runiform() — Uniform and nonuniform pseudorandom variates

Description

runiform(r, c) returns an \( r \times c \) real matrix containing uniformly distributed random variates over \((0,1)\). runiform() is the same function as Stata’s runiform() function.

runiform(r, c, a, b) returns an \( ir \times jc \) real matrix containing uniformly distributed random variates over \((a,b)\). The matrices \( a \) and \( b \) must be \( r\)-conformable, where \( i = \max(\text{rows}\,(a),\text{rows}\,(b)) \) and \( j = \max(\text{cols}\,(a),\text{cols}\,(b)) \).

runiformint(r, c, a, b) returns an \( ir \times jc \) real matrix containing uniformly distributed random integer variates over \([a,b]\). The matrices \( a \) and \( b \) must be \( r\)-conformable, where \( i = \max(\text{rows}\,(a),\text{rows}\,(b)) \) and \( j = \max(\text{cols}\,(a),\text{cols}\,(b)) \).

rseed() returns the current random-variate seed in an encrypted string form. rseed(newseed) sets the seed: an integer can be specified. rseed(newseed) has the same effect as Stata’s set seed command; see [R] set seed.

rngstate() returns the current state of the random-number generator. rngstate() returns the same thing as Stata’s c(rngstate); see [R] set seed and [P] creturn. rngstate(newstate) sets the state of the random-number generator using a string previously obtained from rngstate(). rngstate(newstate) has the same effect as Stata’s set rngstate newstate; see [R] set seed.

rbeta(r, c, a, b) returns an \( ir \timesjc \) real matrix containing beta random variates. The real-valued matrices \( a \) and \( b \) contain the beta shape parameters. The matrices \( a \) and \( b \) must be \( r\)-conformable, where \( i = \max(\text{rows}\,(a),\text{rows}\,(b)) \) and \( j = \max(\text{cols}\,(a),\text{cols}\,(b)) \).

rbinomial(r, c, n, p) returns an \( ir \timesjc \) real matrix containing binomial random variates. The real-valued matrices \( n \) and \( p \) contain the number of trials and the probability parameters, respectively. The matrices \( n \) and \( p \) must be \( r\)-conformable, where \( i = \max(\text{rows}\,(n),\text{rows}\,(p)) \) and \( j = \max(\text{cols}\,(n),\text{cols}\,(p)) \).

rcauchy(r, c, a, b) returns an \( ir \timesjc \) real matrix containing Cauchy random variates. The real-valued matrices \( a \) and \( b \) contain the Cauchy location and scale parameters, respectively. The matrices \( a \) and \( b \) must be \( r\)-conformable, where \( i = \max(\text{rows}\,(a),\text{rows}\,(b)) \) and \( j = \max(\text{cols}\,(a),\text{cols}\,(b)) \).

rchisquare(r, c, df) returns an \( ir \timesjc \) real matrix containing chi-squared random variates. The real-valued matrix \( df \) contains the degrees of freedom parameters, where \( i = \text{rows}(df) \) and \( j = \text{cols}(df) \).

rdiscrete(r, c, p) returns an \( r \times c \) real matrix containing random variates from the discrete distribution specified by the probabilities in the vector \( p \) of length \( k \). The range of the discrete variates is \( 1, 2, \ldots, k \), where \( 2 \leq k \leq 10000 \). The alias method of Walker (1977) is used to sample from the discrete distribution.
When the elements of the matrix \( i \) before the \( \text{rnbinomial}(s) \) must be r-conformable, where the real-valued matrix \( p \) contains the Poisson mean parameters, where \( \text{max}(\text{rows}(m),\text{rows}(a)) = \text{max} \). The real-valued matrices \( a \) and \( b \) contain the mean and standard deviation parameters, respectively. The matrices \( m \) and \( a \) must be r-conformable, where \( \text{rnbinomial}(N,K,n) \) returns an \( i \times j \) real matrix containing hypergeometric random variates. The integer-valued matrix \( N \) contains the population sizes, the integer-valued matrix \( K \) contains the number of elements in each population that have the attribute of interest, and the integer-valued matrix \( n \) contains the sample size. The matrices \( N, K, \) and \( n \) must be r-conformable, where \( \text{max}(\text{rows}(N),\text{rows}(K),\text{rows}(n)) = \text{max} \). The real-valued matrices \( s \) contains scale parameters, where \( \sqrt{3} \). The matrices \( m \) and \( b \) contain the mean and scale parameters, respectively. The matrices \( m \) and \( b \) must be r-conformable, where \( \text{max}(\text{rows}(m),\text{rows}(b)) = \text{max} \). The real-valued matrix \( s \) contains mean-zero logistic random variates. The real-valued matrix \( s \) contains scale parameters, where \( \text{rows}(s) = \text{cols}(s) \). The real-valued matrices \( m \) and \( b \) contain the mean and scale parameters, respectively. The matrices \( m \) and \( b \) must be r-conformable, where \( \text{max}(\text{rows}(m),\text{rows}(b)) = \text{max} \). When the elements of the matrix \( n \) are integer-valued, \( \text{rnbinomial}(r,c,n,p) \) returns an \( i \times j \) real matrix containing negative binomial random variates. The elements of \( n \) can also be nonintegral but must be positive. The matrices \( n \) and \( p \) must be r-conformable, where \( \text{max}(\text{rows}(n),\text{rows}(p)) = \text{max} \). The real-valued matrices \( m \) and \( s \) contain the mean and standard deviation parameters, respectively. The matrices \( m \) and \( s \) must be r-conformable, where \( \text{max}(\text{rows}(m),\text{rows}(s)) = \text{max} \). The real-valued matrix \( m \) contains the Poisson mean parameters, where \( \text{rows}(m) = \text{cols}(m) \). The real-valued matrix \( df \) contains the degrees-of-freedom parameters, where \( \text{rows}(df) = \text{cols}(df) \). The real-valued matrices \( a \) and \( b \) contain the shape and scale parameters, respectively. The matrices \( a \) and \( b \) must be r-conformable, where \( \text{max}(\text{rows}(a),\text{rows}(b)) = \text{max} \).
rweibull(r, c, a, b, g) returns an $ir \times jc$ real matrix containing Weibull random variates. The real-valued matrices $a$, $b$, and $g$ contain the shape, scale, and location parameters, respectively. The matrices $a$, $b$, and $g$ must be $r$-conformable, where $i = \max(\text{rows}(a), \text{rows}(b), \text{rows}(g))$ and $j = \max(\text{cols}(a), \text{cols}(b), \text{cols}(g))$.

rweibullph(r, c, a, b) returns an $ir \times jc$ real matrix containing Weibull (proportional hazards) random variates. The real-valued matrices $a$ and $b$ contain the shape and scale parameters, respectively. The matrices $a$ and $b$ must be $r$-conformable, where $i = \max(\text{rows}(a), \text{rows}(b))$ and $j = \max(\text{cols}(a), \text{cols}(b))$.

rweibullph(r, c, a, b, g) returns an $ir \times jc$ real matrix containing Weibull (proportional hazards) random variates. The real-valued matrices $a$, $b$, and $g$ contain the shape, scale, and location parameters, respectively. The matrices $a$, $b$, and $g$ must be $r$-conformable, where $i = \max(\text{rows}(a), \text{rows}(b), \text{rows}(g))$ and $j = \max(\text{cols}(a), \text{cols}(b), \text{cols}(g))$.

**Syntax**

```
real matrix runiform(real scalar r, real scalar c)
real matrix runiform(real scalar r, real scalar c, real matrix a, real matrix b)
real matrix runiformint(real scalar r, real scalar c, real matrix a, real matrix b)
string scalar rseed()
void rseed(real scalar newseed)
string scalar rngstate()
void rngstate(string scalar newstate)
real matrix rbeta(real scalar r, real scalar c, real matrix a, real matrix b)
real matrix rbinomial(real scalar r, real scalar c, real matrix n, real matrix p)
real matrix rcauchy(real scalar r, real scalar c, real matrix a, real matrix b)
real matrix rchi2(real scalar r, real scalar c, real matrix df)
real matrix rdiscrete(real scalar r, real scalar c, real colvector p)
real matrix rexponential(real scalar r, real scalar c, real matrix b)
real matrix rgamma(real scalar r, real scalar c, real matrix a, real matrix b)
```
The functions described here generate random variates. The parameter limits for each generator are the same as those documented for Stata's random-number functions, except for rdiscrete(), which has no Stata equivalent.

In the example below, we generate and summarize 1,000 random normal deviates with a mean of 3 and standard deviation of 1.

: rseed(13579)
: x = rnormal(1000, 1, 3, 1)
: meanvariance(x)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.99162691</td>
</tr>
<tr>
<td>2</td>
<td>1.056033182</td>
</tr>
</tbody>
</table>

The next example uses a $1 \times 3$ vector of gamma shape parameters to generate a $1000 \times 3$ matrix of gamma random variates, $X$. 

```stata
real matrix rhypergeometric(real scalar r, real scalar c, real matrix N,
                             real matrix K, real matrix n)

real matrix rigaussian(real scalar r, real scalar c, real matrix m,
                       real matrix a)

real matrix rlaplace(real scalar r, real scalar c, real matrix m,
                      real matrix b)

real matrix rlogistic(real scalar r, real scalar c)
real matrix rlogistic(real scalar r, real scalar c, real matrix s)
real matrix rlogistic(real scalar r, real scalar c, real matrix m, real matrix s)

real matrix rnbinomial(real scalar r, real scalar c, real matrix n, real matrix p)

real matrix rnormal(real scalar r, real scalar c, real matrix m, real matrix s)

real matrix rpoisson(real scalar r, real scalar c, real matrix m)

real matrix rt(real scalar r, real scalar c, real matrix df)

real matrix rweibull(real scalar r, real scalar c, real matrix a, real matrix b)
real matrix rweibull(real scalar r, real scalar c, real matrix a, real matrix b,
                     real matrix g)

real matrix rweibullph(real scalar r, real scalar c, real matrix a, real matrix b)
real matrix rweibullph(real scalar r, real scalar c, real matrix a, real matrix b,
                      real matrix g)
```
: a = (0.5, 1.5, 2.5)
: rseed(13579)
: X = rgamma(1000, 1, a, 1)
: mean(X)

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 0.5022154504 & 1.502187839 & 2.417570905 \\
\end{array}
\]

: diagonal(variance(X))'

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 0.5082196561 & 1.434504411 & 2.512575559 \\
\end{array}
\]

The first column of \( X \) contains gamma variates with shape parameter 0.5, the second column contains gamma variates with shape parameter 1.5, and the third column contains gamma variates with shape parameter 2.5.

Below we generate a \( 4 \times 3 \) matrix of beta variates where we demonstrate the use of two \( r \)-conformable parameter matrices, \( a \) and \( b \).

: a = (0.5, 1.5, 2.5)
: b = (0.5, 0.75, 1.0 \ 1.25, 1.5, 1.75)
: rseed(13579)
: rbeta(2, 1, a, b)

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 0.8389820448 & 0.9707672865 & 0.2122592494 \\
2 & 0.5997013245 & 0.6617211509 & 0.8775212495 \\
3 & 0.9552933701 & 0.1133821372 & 0.8006242906 \\
4 & 0.2279075363 & 0.4298247049 & 0.6683477165 \\
\end{array}
\]

The \( 4 \times 3 \) shape-parameter matrices used to generate these beta variates are given below:

: J(2, 1, J(rows(b), 1, a))

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 0.5 & 1.5 & 2.5 \\
2 & 0.5 & 1.5 & 2.5 \\
3 & 0.5 & 1.5 & 2.5 \\
4 & 0.5 & 1.5 & 2.5 \\
\end{array}
\]

: J(2, 1, b)

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 0.5 & 0.75 & 1 \\
2 & 1.25 & 1.5 & 1.75 \\
3 & 0.5 & 0.75 & 1 \\
4 & 1.25 & 1.5 & 1.75 \\
\end{array}
\]

This example illustrates how to restart a random-number generator from a particular point in its sequence. We begin by setting the seed and drawing some uniform variates.
We save off the current state of the random-number generator, so that we can subsequently return to this point in the sequence.

```
: rseed(12345)
: x = runiform(1,3)
: x
1   2   3
1 .3576297229 .400442617 .689383317
```

Having saved off the state, we draw some more numbers from the sequence.

```
: rngstate = rngstate()
: x = runiform(1,3)
: x
1   2   3
1 .5597355706 .574451294 .2076905269
```

Now we restore the state of the random-number generator to where it was and obtain the same numbers from the sequence.

```
: rngstate(rngstate)
: x = runiform(1,3)
: x
1   2   3
1 .5597355706 .574451294 .2076905269
```
Conformability

runiform(r, c):
  r: 1 × 1
  c: 1 × 1
  result: r × c

runiform(r, c, a, b):
  r: 1 × 1
  c: 1 × 1
  a: 1 × 1 or i × 1 or 1 × j or i × j
  b: 1 × 1 or i × 1 or 1 × j or i × j
  result: r × c or ir × c or r × jc or ir × jc

runifmint(r, c, a, b):
  r: 1 × 1
  c: 1 × 1
  a: 1 × 1 or i × 1 or 1 × j or i × j
  b: 1 × 1 or i × 1 or 1 × j or i × j
  result: r × c or ir × c or r × jc or ir × jc

rseed():
  result: 1 × 1

rseed(newseed):
  newseed: 1 × 1
  result: void

rngstate():
  result: 1 × 1

rngstate(newstate):
  newstate: 1 × 1
  result: void

rbeta(r, c, a, b):
  r: 1 × 1
  c: 1 × 1
  a: 1 × 1 or i × 1 or 1 × j or i × j
  b: 1 × 1 or i × 1 or 1 × j or i × j
  result: r × c or ir × c or r × jc or ir × jc
rbinomial(r, c, n, p):
  r: 1 × 1
  c: 1 × 1
  n: 1 × 1 or i × 1 or l × j or i × j
  p: 1 × 1 or i × 1 or l × j or i × j
result: r × c or i r × c or r × j c or i r × j c
rcauchy(r, c, a, b):
  r: 1 × 1
  c: 1 × 1
  a: 1 × 1 or i × 1 or l × j or i × j
  b: 1 × 1 or i × 1 or l × j or i × j
result: r × c or i r × c or r × j c or i r × j c
rchi2(r, c, df):
  r: 1 × 1
c: 1 × 1
df: i × j
result: i r × j c
rdiscrete(r, c, p):
  r: 1 × 1
c: 1 × 1
  p: k × 1
result: r × c
rexponential(r, c, b):
  r: 1 × 1
c: 1 × 1
  b: 1 × 1 or i × 1 or l × j or i × j
result: r × c or i r × c or r × j c or i r × j c
rgamma(r, c, a, b):
  r: 1 × 1
c: 1 × 1
  a: 1 × 1 or i × 1 or l × j or i × j
  b: 1 × 1 or i × 1 or l × j or i × j
result: r × c or i r × c or r × j c or i r × j c
rhypergeometric(r, c, N, K, n):
  r: 1 × 1
c: 1 × 1
  K: 1 × 1 or i × 1 or l × j or i × j
  N: 1 × 1
  n: 1 × 1 or i × 1 or l × j or i × j
result: r × c or i r × c or r × j c or i r × j c
rigaussian(r, c, m, a):
  r: 1 × 1
c: 1 × 1
  m: 1 × 1 or i × 1 or l × j or i × j
  a: 1 × 1 or i × 1 or l × j or i × j
result: r × c or i r × c or r × j c or i r × j c
runiform$(r, c, m, b)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ m: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ b: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ \text{result: } r \times c \text{ or } ir \times c \text{ or } r \times jc \text{ or } ir \times jc \]

rlaplace$(r, c, m, b)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ m: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ b: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ \text{result: } r \times c \text{ or } ir \times c \text{ or } r \times jc \text{ or } ir \times jc \]

rlogistic$(r, c)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ \text{result: } r \times c \]

rlogistic$(r, c, s)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ s: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ \text{result: } r \times c \text{ or } ir \times c \text{ or } r \times jc \text{ or } ir \times jc \]

rlogistic$(r, c, m, s)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ m: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ s: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ \text{result: } r \times c \text{ or } ir \times c \text{ or } r \times jc \text{ or } ir \times jc \]

rnbinomial$(r, c, n, p)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ n: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ p: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ \text{result: } r \times c \text{ or } ir \times c \text{ or } r \times jc \text{ or } ir \times jc \]

rnormal$(r, c, m, s)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ m: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ s: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ \text{result: } r \times c \text{ or } ir \times c \text{ or } r \times jc \text{ or } ir \times jc \]

rpoisson$(r, c, m)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ m: \ i \times j \]
  \[ \text{result: } ir \times jc \]

rt$(r, c, df)$:
  \[ r: \ 1 \times 1 \]
  \[ c: \ 1 \times 1 \]
  \[ df: \ 1 \times 1 \text{ or } i \times 1 \text{ or } 1 \times j \text{ or } i \times j \]
  \[ \text{result: } r \times c \text{ or } ir \times c \text{ or } r \times jc \text{ or } ir \times jc \]
rweibull(r, c, a, b):
  r: 1 x 1
c: 1 x 1
  a: 1 x 1 or i x 1 or 1 x j or i x j
  b: 1 x 1 or i x 1 or 1 x j or i x j
result: r x c or ir x c or r x jc or ir x jc

rweibull(r, c, a, b, g):
  r: 1 x 1
c: 1 x 1
  a: 1 x 1 or i x 1 or 1 x j or i x j
  b: 1 x 1 or i x 1 or 1 x j or i x j
g: 1 x 1 or i x 1 or 1 x j or i x j
result: r x c or ir x c or r x jc or ir x jc

rweibullph(r, c, a, b):
  r: 1 x 1
c: 1 x 1
  a: 1 x 1 or i x 1 or 1 x j or i x j
  b: 1 x 1 or i x 1 or 1 x j or i x j
result: r x c or ir x c or r x jc or ir x jc

rweibullph(r, c, a, b, g):
  r: 1 x 1
c: 1 x 1
  a: 1 x 1 or i x 1 or 1 x j or i x j
  b: 1 x 1 or i x 1 or 1 x j or i x j
g: 1 x 1 or i x 1 or 1 x j or i x j
result: r x c or ir x c or r x jc or ir x jc

Diagnostics

All random-variate generators abort with an error if r < 0 or c < 0.

rseed(see) aborts with error if a string seed is specified and it is malformed.

rngstate(newstate) aborts with error if the specified newstate is malformed, which almost certainly is the case if newstate was not previously obtained from rngstate().

runiform(r, c, a, b), runiformint(r, c, a, b), rnormal(r, c, m, s), rbeta(r, c, a, b), rbinomial(r, c, n, p), rcauchy(r, c, a, b), rgamma(r, c, a, b), rhypergeometric(r, c, N, K, n), rigaussian(r, c, m, a), rlaplace(r, c, m, b), rlogistic(r, c, m, s), rnbinomial(r, c, n, p), rweibull(r, c, a, b), rweibullph(r, c, a, b), and rweibullph(r, c, a, b, g) abort with an error if the parameter matrices do not conform. See r-conformability in [M-6] Glossary for rules on matrix conformability.

rdiscrete() aborts with error if the probabilities in p are not in [0,1] or do not sum to 1.

References

runiform() — Uniform and nonuniform pseudorandom variates


Also see

[M-4] **Standard** — Functions to create standard matrices

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