

Package ‘rlecuyer’

November 21, 2019

Version 0.3-5

Date 2019-11-19

Title R Interface to RNG with Multiple Streams

Description Provides an interface to the C implementation of the random number generator with multiple independent streams developed by L'Ecuyer et al (2002). The main purpose of this package is to enable the use of this random number generator in parallel R applications.

License GPL (>= 2)

URL <http://www.iro.umontreal.ca/~lecuyer/myftp/papers/streams00.pdf>

NeedsCompilation yes

Author Hana Sevcikova [aut, cre],
Tony Rossini [aut],
Pierre L'Ecuyer [cph] (author of the underlying C code)

Maintainer Hana Sevcikova <hanas@uw.edu>

Repository CRAN

Date/Publication 2019-11-21 22:40:02 UTC

R topics documented:

AdvanceState	2
CreateStream	2
CurrentStream	3
DeleteStream	4
GetState	5
GetStreams	5
IncreasedPrecis	6
ResetStream	6
SetAntithetic	7
SetPackageSeed	8
uniform	8
WriteState	10

Index**11**

AdvanceState	<i>Advance the state of a stream</i>
--------------	--------------------------------------

Description

.lec.AdvanceState advances the state of a stream by n steps (see below).

Usage

.lec.AdvanceState (name, e, c)

Arguments

name	name of the stream.
e, c	if $e > 0$ then $n = 2^e + c$; if $e < 0$ then $n = -2^{-e} + c$; if $e = 0$ then $n = c$.

Details

.lec.AdvanceState is a wrapper function for the C function RngStream_AdvanceState (L'Ecuyer et al, 2002).

Value

None.

References

P. L'Ecuyer, R. Simard, E.J.Chen and W.D.Kelton: An Object-Oriented Random-Number Package With Many Long Streams and Substreams; Operations Research, vol. 50, nr. 6, 2002.

CreateStream	<i>Spawn new streams</i>
--------------	--------------------------

Description

.lec.CreateStream creates new streams of random numbers.

Usage

.lec.CreateStream (names)

Arguments

`names` a character string or a vector of character strings naming the streams to be created. The argument must be provided and the names must be unique within the set of existing streams. If for one `i` a stream of the name `names[i]` already exists, its state is replaced by the state of the new created stream.

Details

`.lec.CreateStream` is a wrapper function for the C function `RngStream_CreateStream` (L'Ecuyer et al, 2002). The state of the created stream returned by the C function is stored in the global object `.lec.Random.seed.table`.

Value

None.

References

P. L'Ecuyer, R. Simard, E.J.Chen and W.D.Kelton: An Object-Oriented Random-Number Package With Many Long Streams and Substreams; Operations Research, vol. 50, nr. 6, 2002.

Examples

```
nstreams <- 10      # number of streams
names <- paste("mystream", 1:nstreams, sep="")
.lec.CreateStream(names)
.lec.WriteStateFull(names)
```

CurrentStream	<i>Set/unset the current stream</i>
---------------	-------------------------------------

Description

`.lec.CurrentStream` sets the current stream for usage with the standard R functions for generating random numbers such as `runif` or `rnorm`. `.lec.CurrentStreamEnd` unsets it.

Usage

```
.lec.CurrentStream (name)
.lec.CurrentStreamEnd (kind.old = c("Marsaglia-Multicarry",
                                   "Kinderman-Ramage"))
```

Arguments

`name` a character string giving the name of the stream.
`kind.old` a length 2 character vector, the old rng kinds (possibly returned by `.lec.CurrentStream`).

Details

`.lec.CurrentStream` sets the `RNGkind` to user-defined. All succeeding calls of R built-in generators will generate random numbers from the stream name, until `.lec.CurrentStreamEnd` is called. `.lec.CurrentStreamEnd` updates the RNG state of the stream name in the table `.lec.Random.seed.table` and sets the `RNGkind` to `kind.old`. These two functions are meant to be always used as a pair. Thus, one can arbitrarily switch generating between different streams.

Value

`.lec.CurrentStream` returns a two-element character vector of the RNG and normal kinds in use before the call. `.lec.CurrentStreamEnd` returns a character string giving the name of the unset current stream.

Examples

```
nstreams <- 10      # number of streams
names <- paste("mystream", 1:nstreams, sep="")
.lec.CreateStream(names)
for (i in 1:nstreams) { # generate 10 RNs from each stream
  .lec.CurrentStream(names[i])
  print(paste("stream no.", i))
  print(runif(10))
  .lec.CurrentStreamEnd()
}
```

DeleteStream

Remove streams

Description

`.lec.DeleteStream` removes streams from the global state table.

Usage

```
.lec.DeleteStream (names)
```

Arguments

names a character string or a vector of character strings naming the streams to be deleted.

Details

All streams given in the argument `names` are removed from the table `.lec.Random.seed.table`.

Value

None.

GetState	<i>Return current state of a stream</i>
----------	---

Description

Returns current state (Cg values) of the stream name.

Usage

```
.lec.GetState (name)
```

Arguments

name a character string giving the name of the stream.

Value

a vector of six integer values that identifies the current state of the stream.

See Also

[SetPackageSeed](#)

GetStreams	<i>Return names of existing streams</i>
------------	---

Description

Returns names of existing streams stored in `.lec.Random.seed.table`.

Usage

```
.lec.GetStreams ()
```

Value

a vector of character strings.

IncreasedPrecis *Switch between 32 and 53 bits of resolution*

Description

Switch between 32 and 53 bits of resolution as described in L'Ecuyer et al (2002).

Usage

```
.lec.IncreasedPrecis (name, incp=FALSE)
```

Arguments

name	name of the stream.
incp	see L'Ecuyer et al (2002).

Details

.lec.IncreasedPrecis is a wrapper function for the C function RngStream_IncreasedPrecis.

Value

None.

References

P. L'Ecuyer, R. Simard, E.J.Chen and W.D.Kelton: An Object-Oriented Random-Number Package With Many Long Streams and Substreams; Operations Research, vol. 50, nr. 6, 2002.

ResetStream *Reset the state of a stream*

Description

Resets the state of a stream to its initial state, beginning of the current substream or beginning of the next substream.

Usage

```
.lec.ResetNextSubstream(name)
.lec.ResetStartStream(name)
.lec.ResetStartSubstream(name)
```

Arguments

name	a character string giving the name of the stream.
------	---

Details

- .lec.ResetNextSubstream reinitializes the stream to the beginning of its next substream.
- .lec.ResetStartStream reinitializes the stream to its initial state.
- .lec.ResetStartSubstream reinitializes the stream to the beginning of its current substream.

Value

None.

See Also

[SetPackageSeed](#)

SetAntithetic	<i>Switch between U and $1-U$ variates</i>
---------------	--

Description

Switches between U and $1-U$ variates.

Usage

```
.lec.SetAntithetic (name, anti=FALSE)
```

Arguments

name	name of the stream.
anti	if anti=TRUE then antithetic variates are generated (i.e. $1-U$), until this function is called again with anti=FALSE.

Value

None.

SetPackageSeed	<i>Set the initial seed of the package or stream</i>
----------------	--

Description

Sets the initial seed of the package or stream.

Usage

```
.lec.SetPackageSeed(seed)
.lec.SetSeed (name, seed)
```

Arguments

name	a character string giving the name of the stream.
seed	a vector of six integers. If it is shorter, the seed is extended to the length of 6 by default values 12345. If it is longer, it is truncated to the length of 6 by eliminating the last elements.

Details

Each state of a stream is given by three integer vectors of length 6: Ig gives the initial state of the stream, Bg gives the starting state of the substream that contains the current state, Cg gives the current state. Function `.lec.SetPackageSeed` sets Cg, Bg and Ig to the value of seed. Function `.lec.SetSeed` sets Ig to seed. L'Ecuyer recommends to use the [ResetStream](#) functions instead of `SetSeed`.

Value

The (possibly modified) seed that has been used.

See Also

[ResetNextSubstream](#)

uniform	<i>Generate random numbers</i>
---------	--------------------------------

Description

```
.lec.uniform generates  $U(0, 1)$  random numbers.
.lec.uniform.int generates random numbers from the discrete uniform distribution over integers.
```


Usage

```
.lec.uniform (name, n = 1)

.lec.uniform.int (name, n = 1, a = 0, b = 10)
```

Arguments

name	name of the stream.
n	number of random numbers to be generated.
a,b	interval from which the integer random numbers should be generated.

Details

`.lec.uniform` and `.lec.uniform.int`, respectively, are wrapper functions for the C functions `RngStream_RandU01` and `RngStream_RandInt`, respectively (L'Ecuyer et al, 2002).

Note: Since the stream is here identified by name, there is no need for using the [CurrentStream](#) pair.

Value

A vector of n random numbers.

References

P. L'Ecuyer, R. Simard, E.J.Chen and W.D.Kelton: An Object-Oriented Random-Number Package With Many Long Streams and Substreams; *Operations Research*, vol. 50, nr. 6, 2002.

See Also

[.lec.CurrentStream](#)

Examples

```
nstreams <- 10      # number of streams
seed<-rep(1,6)
.lec.SetPackageSeed(seed)
names <- paste("mystream",1:nstreams,sep="")
.lec.CreateStream(names)
for (i in 1:nstreams) # generate 10 RNs from each stream
  print(.lec.uniform(names[i],10))
.lec.DeleteStream(names)
```

`WriteState`*Output of the current state of streams*

Description

`.lec.WriteState` writes the current state of given streams (Cg values).

`.lec.WriteStateFull` writes the values of all internal state variables of given streams.

Usage

`.lec.WriteState (names)`

`.lec.WriteStateFull (names)`

Arguments

`names` a character string or a vector of character strings naming the streams.

Value

None

Index

*Topic **distribution**

- AdvanceState, 2
- CreateStream, 2
- CurrentStream, 3
- DeleteStream, 4
- GetState, 5
- GetStreams, 5
- IncreasedPrecis, 6
- ResetStream, 6
- SetAntithetic, 7
- SetPackageSeed, 8
- uniform, 8
- WriteState, 10
- .lec.AdvanceState (AdvanceState), 2
- .lec.CreateStream (CreateStream), 2
- .lec.CurrentStream, 9
- .lec.CurrentStream (CurrentStream), 3
- .lec.CurrentStreamEnd (CurrentStream), 3
- .lec.DeleteStream (DeleteStream), 4
- .lec.GetState (GetState), 5
- .lec.GetStreams (GetStreams), 5
- .lec.IncreasedPrecis (IncreasedPrecis), 6
- .lec.ResetNextSubstream (ResetStream), 6
- .lec.ResetStartStream (ResetStream), 6
- .lec.ResetStartSubstream (ResetStream), 6
- .lec.SetAntithetic (SetAntithetic), 7
- .lec.SetPackageSeed (SetPackageSeed), 8
- .lec.SetSeed (SetPackageSeed), 8
- .lec.WriteState (WriteState), 10
- .lec.WriteStateFull (WriteState), 10
- .lec.uniform (uniform), 8

- AdvanceState, 2

- CreateStream, 2
- CurrentStream, 3, 9

- DeleteStream, 4

- GetState, 5
- GetStreams, 5

- IncreasedPrecis, 6

- ResetNextSubstream, 8
- ResetStream, 6, 8
- rnorm, 3
- runif, 3

- SetAntithetic, 7
- SetPackageSeed, 5, 7, 8

- uniform, 8

- WriteState, 10