Package ‘mlsjunkgen’

May 2, 2021

Title Use the MLS Junk Generator Algorithm to Generate a Stream of Pseudo-Random Numbers

Version 0.1.2

Description Generate a stream of pseudo-random numbers generated using the MLS Junk Generator algorithm. Functions exist to generate single pseudo-random numbers as well as a vector, data frame, or matrix of pseudo-random numbers.


BugReports https://github.com/scumdogsteev/mlsjunkgen/issues

Depends R (>= 3.1.3)

License MIT + file LICENSE

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Steve Myles [aut, cre]

Maintainer Steve Myles <steve@mylesandmyles.info>

Repository CRAN

Date/Publication 2021-05-02 00:40:02 UTC

R topics documented:

  junkgen ......................................................... 2
  mlsjunkgen ..................................................... 2
  mlsjunkgend .................................................... 3
  mlsjunkgenm ..................................................... 4
  mlsjunkgenv ..................................................... 5

Index 7
Generate a single pseudo-random number using the MLS Junk Generator algorithm

Description

Based on user input seeds, this function generates a pseudo-random number. This is called by the mlsjunkgen package’s other functions to generate a pseudo-random number stream.

Usage

junkgen(w, x, y, z)

Arguments

w the first seed required by the MLS Junk Generator algorithm
x the first seed required by the MLS Junk Generator algorithm
y the first seed required by the MLS Junk Generator algorithm
z the first seed required by the MLS Junk Generator algorithm

Value

A numeric vector containing a single pseudo-random number

Examples

# Generate a pseudo-random number with user-specified seeds

w <- 1
x <- 2
y <- 3
z <- 4

junkgen(w = w, x = x, y = y, z = z) # returns "[1] 0.9551644"

mlsjunkgen: Use the MLS Junk Generator Algorithm to Generate a Stream of Pseudo-Random Numbers

Description

mlsjunkgen: Use the MLS Junk Generator Algorithm to Generate a Stream of Pseudo-Random Numbers
mlsjunkgen functions

- `junkgen`: generate a single pseudo-random number; called by the other functions
- `mlsjunkgenv`: generate a vector stream of pseudo-random numbers
- `mlsjunkgend`: generate a data frame of pseudo-random numbers
- `mlsjunkgenm`: generate a matrix of pseudo-random numbers

---

**mlsjunkgend**

*Generate a data frame of pseudo-random numbers using the MLS Junk Generator algorithm*

**Description**

Based on user input seeds, this function generates a data frame of n pseudo-random numbers and names the column containing these as "RN" for "random numbers." This is achieved by calling `junkgen`.

**Usage**

`mlsjunkgend(n = 1, w, x, y, z, round = 5)`

**Arguments**

- `n` the number of pseudo-random numbers to generate; defaults to 1
- `w` the first seed required by the MLS Junk Generator algorithm
- `x` the first seed required by the MLS Junk Generator algorithm
- `y` the first seed required by the MLS Junk Generator algorithm
- `z` the first seed required by the MLS Junk Generator algorithm
- `round` the number of decimal places to which to round the pseudo-random numbers; default = 5

**Value**

A numeric vector containing a single pseudo-random number

**Examples**

```r
# Generate a pseudo-random number data frame with 10 observations from user-specified seeds
w <- 1
x <- 2
y <- 3
z <- 4
mlsjunkgend(n = 10, w = w, x = x, y = y, z = z) # returns a data frame of 10 observations
```
# Specifying different values for n and round
mlsjunkgend(n = 5, w = w, x = x, y = y, z = z, round = 2)
# returns a data frame identical to the above example but with only 5 observations
# rounded to 2 decimal places

# using the default value of n (1) is identical to assigning the rounded result of
# junkgen to a data frame of 1 observation
round(junkgen(w = w, x = x, y = y, z = z), 5) # returns "[1] 0.95516"
mlsjunkgend(w = w, x = x, y = y, z = z)
# returns the following:
#   RN
#   1 0.95516

---

**mlsjunkgenm**  
*Generate a matrix of pseudo-random numbers using the MLS Junk Generator algorithm*

---

**Description**

Based on user input seeds, this function generates a vector of n pseudo-random numbers by calling *mlsjunkgenv* which in turn calls *junkgen*.

**Usage**

```r
mlsjunkgenm(nrow = 1, ncol = 1, w, x, y, z, round = 5)
```

**Arguments**

- `nrow`: the number of rows for the matrix; defaults to 1
- `ncol`: the number of columns for the matrix; defaults to 1
- `w`: the first seed required by the MLS Junk Generator algorithm
- `x`: the first seed required by the MLS Junk Generator algorithm
- `y`: the first seed required by the MLS Junk Generator algorithm
- `z`: the first seed required by the MLS Junk Generator algorithm
- `round`: the number of decimal places to which to round the pseudo-random numbers; default = 5

**Value**

A numeric vector containing a single pseudo-random number
Examples

# Generate a 4x4 matrix of pseudo-random numbers with user-specified seeds

```r
w <- 1
x <- 2
y <- 3
z <- 4

mlsjunkgenm(nrow = 4, ncol = 4, w = w, x = x, y = y, z = z) # returns a 4x4 matrix
```

# the sixteen values in the above matrix are equivalent to the following call
# to mlsjunkgenv

```r
mlsjunkgenv(n = 16, w = w, x = x, y = y, z = z)
```

# matrices need not be square
# this returns a 3x2 matrix of pseudo-random numbers with 2 decimal places

```r
mlsjunkgenm(nrow = 3, ncol = 2, w = w, x = x, y = y, z = z, round = 2)
```

# using the default value of n (1) generates a 1x1 matrix the value of which
# is identical to running junkgen and rounding the result to 5 decimal places

```r
round(junkgen(w = w, x = x, y = y, z = z), 5) # returns "[1] 0.95516"
mlsjunkgenv(w = w, x = x, y = y, z = z) # returns a 1x1 matrix with single element = "0.95516"
```

---

### Description

Based on user input seeds, this function generates a vector of \( n \) pseudo-random numbers by calling `junkgen`.

### Usage

```r
mlsjunkgenv(n = 1, w, x, y, z, round = 5)
```

### Arguments

- **\( n \)**: the number of pseudo-random numbers to generate; defaults to 1
- **\( w \)**: the first seed required by the MLS Junk Generator algorithm
- **\( x \)**: the first seed required by the MLS Junk Generator algorithm
- **\( y \)**: the first seed required by the MLS Junk Generator algorithm
- **\( z \)**: the first seed required by the MLS Junk Generator algorithm
- **\( \text{round} \)**: the number of decimal places to which to round the pseudo-random numbers; default = 5
Value

A numeric vector containing a single pseudo-random number

Examples

# Generate a pseudo-random number stream of length 5 with user-specified seeds

w <- 1
x <- 2
y <- 3
z <- 4

# the following call returns "[1] 0.95516 0.66908 0.21235 0.34488 0.11995"
mlsjunkgenv(n = 5, w = w, x = x, y = y, z = z)

# Specifying different values for n and round

mlsjunkgenv(n = 3, w = w, x = x, y = y, z = z, round = 2) # returns "[1] 0.96 0.67 0.21"

# using the default value of n (1) is identical to running junkgen and rounding
# the result to 5 decimal places

round(junkgen(w = w, x = x, y = y, z = z),5) # returns "[1] 0.95516"
mlsjunkgenv(w = w, x = x, y = y, z = z) # returns "[1] 0.95516"
Index

junkgen, 2, 3
mlsjunkgen, 2
mlsjunkgen-package (mlsjunkgen), 2
mlsjunkgend, 3, 3
mlsjunkgenm, 3, 4
mlsjunkgenv, 3, 5