RStudio PACKAGE ‘rnvutils’

Version 1.0

Title Random Number and Variates Utilities

Description Provide utilities for generating random numbers, testing uniformity and independence of a vector of random numbers, and generating random variates from various specific distributions

Depends R (>= 3.0.2)

URL https://tinyurl.com/rnvutils
(Contains *.tar.gz* file for the package, *.zip* file for all files related to the package, *.pdf* file for the package specifications, and *.R* for testing all rnvutils functions)

Authors Tricia Janylle Sta. Maria [aut]
Emogene Tejada [aut]
John Robert Torres [aut,cre]
Carl Jacob Villar [aut]

Install from Package Archive File (.tar.gz)

Date Publication 14 December 2020

HOW TO INSTALL

USING THE TOOLBAR

1. Make sure that you have downloaded the Package Archive File of rnvutils (**rnvutils_1.0.tar.gz**).
2. In the menu toolbar, click **Tools** > **Install packages**.
3. In the new window, select **Install From Package Archive File (.tar.gz)** and browse the file **rnvutils_1.0.tar.gz** for the **Package Archive**.
4. Press **Enter** or click **Install**.
5. After the installation, find **rnvutils** in the list of **Packages** at the default lower-right pane of RStudio.
6. You may click on the **rnvutils** package to see all details.
7. To load rnvutils, mark its checkbox or type **library('rnvutils')** in the console.

USING THE CONSOLE

1. Make sure that you have downloaded the Package Archive File of rnvutils (**rnvutils_1.0.tar.gz**).
2. Set your working directory using the **setwd()** function. The directory should be the path where the **rnvutils_1.0.tar.gz** is located. This can also be done manually in the **File Browser** tab at the default lower-right pane of RStudio.
3. Once the directory is properly set, type **install.packages("rnvutils_1.0.tar.gz",repos=NULL)** in the console.
4. After the installation, find **rnvutils** in the list of **Packages** at the lower-right pane of RStudio.
5. You may click on the **rnvutils** package to see all details.
6. To load **rnvutils**, mark its checkbox or type **library('rnvutils')** in the console.

Note: After successfully installing and loading rnvutils, you can view its DESCRIPTION file and HELP PAGES for all its functions in the Packages tab. You just need to find rnvutils in the list of Packages. If these do not appear, it means you are unsuccessful in installing and loading rnvutils. Moreover, all functions of this package can be tested using the rnvutils_TEST.R.

RNVUTILS FUNCTIONS

The ‘rnvutils’ package is a collection of R functions that provide utilities for random numbers and random variates. These functions can be classified into four main topics namely congruential random number generators, test of uniformity of random numbers, test of independence of random numbers, and random variates generation.

Specified below are the descriptions, inputs, and outputs of each function in the rnvutils package. Please note that all errors in the arguments are caught by RStudio’s default error detection. Smart and responsible usage of each function is shared with the end-users.

---

autocorr.test(randnums,start,lag,alpha) Test the independence of selected random numbers using Autocorrelation Test

Description Performs test of hypothesis on the independence of selected random numbers using Autocorrelation Test

randnums A vector of random numbers

start The index of the random number in the vector where the selection starts

lag The lag or space between the random numbers being tested

alpha The level of significance for the test of hypothesis

Return A test of hypothesis including computed statistic, tabular value, p-value, and conclusion

---

freq.test(randnums,nc,alpha) Test the uniformity of random numbers using Frequency Test

Description Performs test of hypothesis on the uniformity of random numbers using Frequency Test

randnums A vector of random numbers (each random number is from 0 to 1, inclusive) nc The number of intervals from 0 to 1 to be used

alpha The level of significance for the test of hypothesis

Return A test of hypothesis including computed statistic, tabular value, p-value, and conclusion

----

inv.cg(a1,a2,k,w0,n) Generate random numbers using the theory of inverse congruence

Description Generate uniform random numbers from 0 to 1 inclusive using Inverse Congruential Generators (ICGs)

a1 An integer

a2 An integer

k An integer

w0 The number or a seed to start

n The number of uniform random numbers to be generated

Return A vector of uniform random numbers

----

lin.cg(a1,a2,k,w0,n) Generate random numbers using the theory of linear congruence

Description Generate uniform random numbers from 0 to 1 inclusive using Linear Congruential Generators (LCGs)

a1 An integer

a2 An integer

k An integer

w0 The number or a seed to start

n The number of uniform random numbers to be generated

Return A vector of uniform random numbers

----

mult.cg(a1,k,w0,n) Generate random numbers using the theory of multiplicative congruence

Description Generate uniform random numbers from 0 to 1 inclusive using Multiplicative Congruential Generators (MCGs)

a1 An integer

k An integer

w0 The number or a seed to start

n The number of uniform random numbers to be generated

Return A vector of uniform random numbers

----

poker.test.2d(randnums,alpha) Test the independence of random numbers (two digits on decimal) using Poker Test

Description Performs test of hypothesis on the independence of two-digit decimal numbers using Poker Test

randnums A vector of two-digit decimal numbers

alpha The level of significance for the test of hypothesis

Return A test of hypothesis including computed statistic, tabular value, p-value, and conclusion

----

poker.test.3d(randnums,alpha) Test the independence of random numbers (three digits on decimal) using Poker Test

Description Performs test of hypothesis on the independence of three-digit decimal numbers using Poker Test

randnums A vector of three-digit decimal numbers

alpha The level of significance for the test of hypothesis

Return A test of hypothesis including computed statistic, tabular value, p-value, and conclusion

----

poker.test.4d(randnums,alpha) Test the independence of random numbers (four digits on decimal) using Poker Test

Description Performs test of hypothesis on the independence of four-digit decimal numbers using Poker Test

randnums A vector of four-digit decimal numbers

alpha The level of significance for the test of hypothesis

Return A test of hypothesis including computed statistic, tabular value, p-value, and conclusion

---

runs.down(vec) Determine the number of runs down

Description Count the number of runs down in a sequence of numbers to be used in the Runs Test

vec A vector of numbers

Return The number of runs

---

runs.test(randnums, alpha) Test the independence of random numbers using Runs Test

Description Performs test of hypothesis on the independence of selected random numbers using Runs Test

randnums A vector of random numbers

alpha The level of significance for the test of hypothesis

Return A test of hypothesis including computed statistic, tabular value, p-value, and conclusion

---

runs.up(vec) Determine the number of runs up

Description Count the number of runs up in a sequence of numbers to be used in the Runs Test

vec A vector of numbers

Return The number of runs up

---

rv.bern(rvs,p) Produce random variates from the Bernoulli Distribution

Description Use Uniform random numbers to generate random variates from a Bernoulli Distribution with parameter p

rvs The number of random variates to be generated

p The probability parameter of a Bernoulli Distribution

Return A vector of random variates from the Bernoulli Distribution

---

rv.beta(rvs,alpha, beta) Produce random variates from the Beta Distribution

Description Use Uniform random numbers to generate random variates from a Beta Distribution with parameters alpha and beta

rvs The number of random variates to be generated

alpha The first shape parameter of a Beta Distribution

beta The second shape parameter of a Beta Distribution

Return A vector of random variates from the Bernoulli Distribution

---

rv.bin(rvs,n,p) Produce random variates from the Binomial Distribution

Description Use Uniform random numbers to generate random variates from a Binomial Distribution with parameters n and p

rvs The number of random variates to be generated

n The number of trials parameter of a Binomial Distribution

p The probability parameter of a Binomial Distribution

Return A vector of random variates from the Binomial Distribution

---

rv.exp(rvs,lambda) Produce random variates from the Exponential Distribution

Description Use Uniform random numbers to generate random variates from an Exponential Distribution with parameter lambda

rvs The number of random variates to be generated

lambda The rate parameter of an Exponential Distribution

Return A vector of random variates from the Exponential Distribution

---

rv.geom(rvs,p) Produce random variates from the Geometric Distribution

Description Use Uniform random numbers to generate random variates from a Geometric Distribution with parameter p

rvs The number of random variates to be generated

p The probability parameter of a Geometric Distribution

Return A vector of random variates from the Geometric Distribution

---

rv.norm.BM(rvps) Produce random variates from the Standard Normal Distribution using the Box-Muller Method

Description Use Uniform random numbers to generate random variates from a Standard Normal Distribution using the Box-Muller Method. The Box-Muller Method generates two random variates per random number.

rvps The number of pairs of random variates to be generated

Return A vector of random variates from the Standard Normal Distribution

---

rv.norm.CLT(rvs,rc) Produce random variates from the Standard Normal Distribution using the Central Limit Theorem

Description Use Uniform random numbers to generate random variates from a Standard Normal Distribution using the Central Limit Theorem

rvs The number of random variates to be generated

rc The number of random numbers to be used in generating a random variate from a Standard Normal Distribution using the Central Limit Theorem

Return A vector of random variates from the Standard Normal Distribution

---

rv.pois(rvs,lambda) Produce random variates from the Poisson Distribution

Description Use Uniform random numbers to generate random variates from a Poisson Distribution with parameter lambda

rvs The number of random variates to be generated

lambda The rate parameter of a Poisson Distribution

Return A vector of random variates from the Poisson Distribution

---

rv.unif(rvs,lower, upper) Produce random variates from the Continuous Uniform Distribution

Description Use Uniform random numbers to generate random variates from a Continuous Uniform Distribution with parameters lower and upper (bounds)

rvs The number of random variates to be generated

lower The lower bound of a Continuous Uniform Distribution

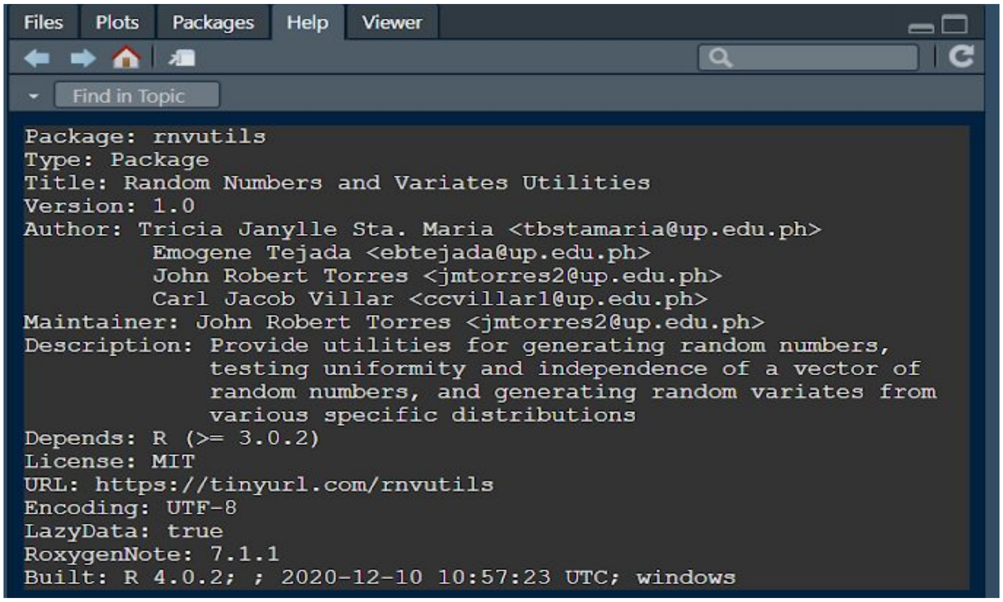
upper The upper bound of a Continuous Uniform Distribution

Return A vector of random variates from the Continuous Uniform Distribution

---

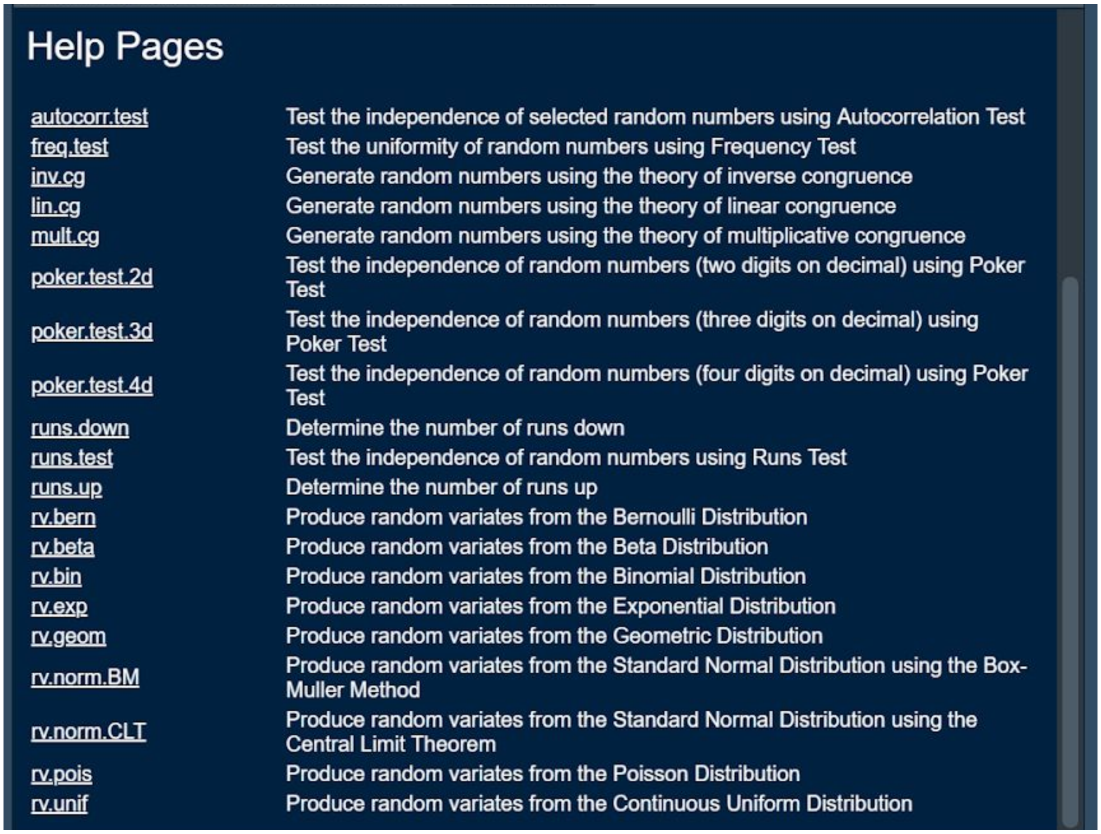
APPENDIX

DESCRIPTION



```
Package: rnvutils
Type: Package
Title: Random Numbers and Variates Utilities
Version: 1.0
Author: Tricia Janylle Sta. Maria <tbstamaria@up.edu.ph>
         Emogene Tejada <ebtejada@up.edu.ph>
         John Robert Torres <jmtorres2@up.edu.ph>
         Carl Jacob Villar <ccvillar1@up.edu.ph>
Maintainer: John Robert Torres <jmtorres2@up.edu.ph>
Description: Provide utilities for generating random numbers,
             testing uniformity and independence of a vector of
             random numbers, and generating random variates from
             various specific distributions
Depends: R (>= 3.0.2)
License: MIT
URL: https://tinyurl.com/rnvutils
Encoding: UTF-8
LazyData: true
RoxygenNote: 7.1.1
Built: R 4.0.2; ; 2020-12-10 10:57:23 UTC; windows
```

HELP PAGES



Help Pages

autocorr.test	Test the independence of selected random numbers using Autocorrelation Test
freq.test	Test the uniformity of random numbers using Frequency Test
inv.cg	Generate random numbers using the theory of inverse congruence
lin.cg	Generate random numbers using the theory of linear congruence
mult.cg	Generate random numbers using the theory of multiplicative congruence
poker.test.2d	Test the independence of random numbers (two digits on decimal) using Poker Test
poker.test.3d	Test the independence of random numbers (three digits on decimal) using Poker Test
poker.test.4d	Test the independence of random numbers (four digits on decimal) using Poker Test
runs.down	Determine the number of runs down
runs.test	Test the independence of random numbers using Runs Test
runs.up	Determine the number of runs up
rv.bern	Produce random variates from the Bernoulli Distribution
rv.beta	Produce random variates from the Beta Distribution
rv.bin	Produce random variates from the Binomial Distribution
rv.exp	Produce random variates from the Exponential Distribution
rv.geom	Produce random variates from the Geometric Distribution
rv.norm.BM	Produce random variates from the Standard Normal Distribution using the Box-Muller Method
rv.norm.CLT	Produce random variates from the Standard Normal Distribution using the Central Limit Theorem
rv.pois	Produce random variates from the Poisson Distribution
rv.unif	Produce random variates from the Continuous Uniform Distribution