Package ‘covsim’

May 12, 2020

Type Package
Title VITA and IG Simulation for Given Covariance and Marginals
Version 0.1.0
Description User specifies population covariance matrix. Marginal information may be fully
specified, for which the package implements the VITA (VIne-To-Anything) algorithm.
Alternatively, marginal skewness and kurtosis may be specified, for which the package
License GPL (>= 2)
Depends R (>= 3.1.0)
Imports rvinecopulib (>= 0.5.1.1.0), lavaan (>= 0.6-5), nleqslv,
PearsonDS, MASS, stats, Rcpp, gsl
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
NeedsCompilation no
Author Njål Foldnes [aut, cre],
Steffen Grønneberg [aut]
Maintainer Njål Foldnes <njal.foldnes@gmail.com>
Repository CRAN
Date/Publication 2020-05-12 10:10:06 UTC

R topics documented:

rIG ......................................................... 2
vita ...................................................... 3

Index 5
Simulation of non-normal data

Description

Using the IG method to simulate non-normal data

Usage

\[
\text{rIG}(N, \text{sigma.target}, \text{skewness}, \text{excesskurtosis}, \text{reps} = 1, \text{typeA} = \text{"symm"})
\]

Arguments

- \(N\): Number of observations to simulate.
- \(\text{sigma.target}\): Target population covariance matrix
- \(\text{skewness}\): Target skewness
- \(\text{excesskurtosis}\): Target excess kurtosis
- \(\text{reps}\): Number of simulated samples
- \(\text{typeA}\): Symmetrical (default) or triangular \(A\) matrix

Value

A list of simulated samples

Author(s)

Njål Foldnes (<njal.foldnes@gmail.com>)

References


Examples

```
set.seed(1234)
model <-'
  # measurement model
  ind60 =~ x1 + x2 + x3
  dem60 =~ y1 + y2 + y3 + y4
  dem65 =~ y5 + y6 + y7 + y8
  # regressions
  dem60 ~ ind60
  dem65 ~ ind60 + dem60
  # residual correlations
  y1 ~~ y5
```
y2 ≈ y4 + y6
y3 ≈ y7
y4 ≈ y8
y6 ≈ y8

fit <- lavaan::sem(model, data=lavaan::PoliticalDemocracy)
population.sigma <- lavaan::lavInspect(fit, "sigma.hat")
population.skew <- c(0, 0, 0, 1, 1, 1, 2, 2)
population.excesskurt <- c(1, 1, 1, 3, 3, 3, 7, 7)
my.samples <- rIG(N=10^3, sigma=population.sigma,
    skewness=population.skew,
    excesskurt=population.excesskurt,
    reps=5)

vita

Calibrate a regular vine

Description

vita implements the VITA (VIne-To-Anything) algorithm. Covariance matrix and margins are specified, and vita calibrates the pair-copulas in each node of the tree to match the target covariance.

Usage

vita(
margins,
sigma.target,
vc = NULL,
family_set = c("clayton", "gauss", "joe", "gumbel", "frank"),
Nmax = 10^6,
umrootpoints = 10,
conflevel = 0.995,
numpoints = 4,
verbose = TRUE,
cores = parallel::detectCores()
)

Arguments

margins A list where each element corresponds to a margin. Each margin element is a list containing the distribution family ("distr") and additional parameters. Must be a distribution available in the stats package.

sigma.target The target covariance matrix that is to be matched. The diagonal elements must contain the variances of marginal distributions.

vc A vine dist object as specified by the rvinecopulib package. This object specifies the vine that is to be calibrated. If not provided, a D-vine is assumed.
family_set  A vector of one-parameter pair-copula families that is to be calibrated at each node in the vine. Possible entries are "gauss", "clayton", "joe", "gumbel" and "frank". Calibration of pair-copula families is attempted in the order provided.

Nmax  The sample size used for calibration. Reduce for faster calibration, at the cost of precision.

numrootpoints  The number of estimated roots at the initial calibration stage, which determines a search interval where Nmax samples are drawn.

conflevel  Confidence level for determining search interval.

numpoints  The number of samples drawn with size Nmax, to determine the root within search interval. To increase precision increase this number. To calibrate faster (but less precisely), may be reduced to a number no lower than 2.

verbose  If TRUE, outputs details of calibration of each bicopula.

cores  Number of cores to use. If larger than 1, computations are done in parallel. May be determined with parallel:detectCores()

Value

If a feasible solution was found, a vine to be used for simulation.

References


Examples

```r
set.seed(1)# define a target covariance. 3 dimensions.
sigma.target <- cov(MASS::mvrnorm(10, mu=rep(0,3), Sigma=diag(1, 3)))
#normal margins that match the covariances:
marginsnorm <- lapply(sqrt(diag(sigma.target)),function(X) list(distr="norm", sd=X) )
#calibrate with a default D-vine, with rather low precision (default Nmax is 10^6)
# if cores=1 is removed, all cores are used, with a speed gain
calibrated.vine <- vita(marginsnorm, sigma.target =sigma.target, Nmax=10^5, cores=1)
#check
#round(cov(rvinecopulib::rvine(10^5, calibrated.vine))-sigma.target, 3)

#margins are normal but dependence structure is not
#pairs(rvinecopulib::rvine(500, calibrated.vine))
```
Index

rIG, 2
vita, 3