Package ‘covsim’

October 12, 2022

Type  Package
Title  VITA, IG and PLSIM Simulation for Given Covariance and Marginals
Version  1.0.0
Description  Random sampling from distributions with user-specified population covariance matrix. Marginal information may be fully specified, for which the package implements the VITA (VIne-To-Anything) algorithm Grønneberg and Foldnes (2017) <doi:10.1007/s11336-017-9569-6>. See also Grønneberg, Foldnes and Marcoulides (2022) <doi:10.18637/jss.v102.i03>. Alternatively, marginal skewness and kurtosis may be specified, for which the package implements the IG (independent generator) and PLSIM (piecewise linear) algorithms, see Foldnes and Olsson (2016) <doi:10.1080/00273171.2015.1133274> and Foldnes and Grønneberg (2021) <doi:10.1080/10705511.2021.1949323>, respectively.

License  GPL (>= 2)
Depends  R (>= 3.5.0)
Imports  rvinecopulib (>= 0.5.1.1.0), lavaan (>= 0.6-5), nleqslv, PearsonDS, MASS, stats, Rcpp, gsl, tmvtnorm, Matrix
Encoding  UTF-8
RoxygenNote  7.1.2
Suggests  rmarkdown, knitr, R.rsp, GGally, psych
VignetteBuilder  knitr
NeedsCompilation  no
Author  Njaal Foldnes [aut, cre], Steffen Grønneberg [aut]
Maintainer  Njaal Foldnes <njal.foldnes@gmail.com>
Repository  CRAN
Date/Publication  2022-04-29 22:40:09 UTC

R topics documented:

rIG .......................................................... 2
rPLSIM ...................................................... 3
vita .......................................................... 4
Description

Using the IG method to simulate non-normal data

Usage

```r
rIG(N, sigma.target, skewness, excesskurtosis, reps = 1, typeA = c("symm", "triang"))
```

Arguments

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

Value

A list of simulated samples.

Author(s)

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

References

rPLSIM

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, 1, 3, 3, 3, 7, 7, 7)
my.samples <- rIG(N=10^3, sigma=population.sigma,
                   skewness=population.skew,
                   excesskurt=population.excesskurt,
                   reps=5)

rPLSIM Simulation of non-normal data

Description

Using the piecewise linear PLSIM method to simulate non-normal data

Usage

rPLSIM(
  N,
  sigma.target,
  skewness,
  excesskurtosis,
  reps = 1,
  numsegments = 4,
  gammalist = NULL,
  monot = FALSE,
  verbose = TRUE
)
**Arguments**

- **N**
  - Number of observations to simulate.
- **sigma.target**
  - Target population covariance matrix
- **skewness**
  - Target skewness
- **excesskurtosis**
  - Target excess kurtosis
- **reps**
  - Number of simulated samples
- **numsegments**
  - The number of line segments in each marginal
- **gammalist**
  - A list of breakpoints in each margin
- **monot**
  - True if piecewise linear functions are forced to be monotonous. The copula will then be normal.
- **verbose**
  - If true, progress details of the procedure are printed

**Value**

A list with two elements. First element: the list of simulated samples. Second element: The fitted piecewise linear functions and the intermediate correlations matrix.

**Author(s)**

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

**References**


**Examples**

```r
set.seed(1)
sigma.target <- cov(MASS::mvnorm(5, rep(0,3), diag(3)))
res <- covsim::rPLSIM(10^5, sigma.target, skewness=rep(1,3), excesskurtosis=rep(4,3))
my.sample <- res[[1]][[1]]
```

---

**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,
  numrootpoints = 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

ing non-normal data for structural equation models using copulas. Journal of Statistical Software.
doi:10.18637/jss.v102.i03
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(X=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
rPLSIM, 3
vita, 4