

# Package ‘sbgcop’

July 23, 2025

**Title** Semiparametric Bayesian Gaussian Copula Estimation and Imputation

**Version** 1.0

**Date** 2025-06-18

**Maintainer** Peter Hoff <peter.hoff@duke.edu>

**Description** Estimation and inference for parameters in a Gaussian copula model, treating the univariate marginal distributions as nuisance parameters as described in Hoff (2007) <doi:10.1214/07-AOAS107>. This package also provides a semiparametric imputation procedure for missing multivariate data.

**License** GPL (>= 2)

**URL** <https://pdhoff.github.io/>

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Peter Hoff [aut, cre]

**Repository** CRAN

**Date/Publication** 2025-06-19 15:40:02 UTC

## Contents

sbgcop-package . . . . .	2
ldmvnorm . . . . .	3
plotci.sA . . . . .	4
qM.sM . . . . .	4
rwish . . . . .	5
sbgcop.mcmc . . . . .	6
sR.sC . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

sbgcop-package	<i>Semiparametric Bayesian Gaussian Copula Estimation and Imputation</i>
----------------	--

---

## Description

Estimation and inference for parameters in a Gaussian copula model, treating univariate marginal distributions as nuisance parameters as described in Hoff (2007) <doi:10.1214/07-AOAS107>. This package also provides a semiparametric imputation procedure for missing multivariate data.

## Details

Package: sbgcop  
Type: Package  
Version: 0.980  
Date: 2018-05-25  
License: GPL Version 2 or later

This function produces MCMC samples from the posterior distribution of a correlation matrix, using a scaled inverse-Wishart prior distribution and an extended rank likelihood. It also provides imputation for missing values in a multivariate dataset.

## Author(s)

Peter Hoff <peter.hoff@duke.edu>

## References

Hoff (2007) “Extending the rank likelihood for semiparametric copula estimation”

## Examples

```
fit<-sbgcop.mcmc(swiss)
summary(fit)
plot(fit)
```

---

ldmvnorm	<i>Log Multivariate Normal Density</i>
----------	--

---

**Description**

Computes the log of the multivariate normal density

**Usage**

```
ldmvnorm(Y, S)
```

**Arguments**

Y	an n x p matrix
S	a p x p positive definite matrix

**Details**

This function computes the log density of the data matrix Y under the model that the rows are independent samples from a mean-zero multivariate normal distribution with covariance matrix S.

**Value**

A real number.

**Author(s)**

Peter Hoff

**Examples**

```
Y<-matrix(rnorm(9*7),9,7)
ldmvnorm(Y,diag(7))
```

---

 plotci.sA

*Plot Confidence Bands for Association Parameters*


---

**Description**

Plots 95

**Usage**

```
plotci.sA(sA, ylabs = colnames(sA[, , 1]), mgp = c(1.75, 0.75, 0))
```

**Arguments**

sA	a p x p x nsamp array
ylabs	a p x 1 vector of names for plotting labels
mgp	margin parameters

**Author(s)**

Peter Hoff

**Examples**

```
fit<-sbgcop.mcmc(swiss)
plotci.sA(fit$C.psamp)
```

---

 qM.sM

*Matrix Quantiles*


---

**Description**

Computes quantiles along the third dimension of a 3-d array.

**Usage**

```
qM.sM(sM, quantiles = c(0.025, 0.5, 0.975))
```

**Arguments**

sM	an m x n x s array
quantiles	quantiles to be computed

**Value**

an array of dimension  $m \times n \times l$ , where  $l$  is the length of quantiles

**Author(s)**

Peter Hoff

---

rwish

*Sample from the Wishart Distribution*

---

**Description**

Generate a random sample from the Wishart distribution.

**Usage**

```
rwish(S0, nu)
```

**Arguments**

$S0$	a positive definite matrix
$nu$	a positive integer

**Details**

Return the sum of  $nu$  i.i.d. rank-one matrices generated as  $z\%*t(z)$ , where  $z$  is a sample from a multivariate normal distribution with covariance  $S0$ . The resulting random variable has mean  $nu*S0$ .

**Value**

a positive definite matrix.

**Author(s)**

Peter Hoff

sbgcop.mcmc

*Semiparametric Bayesian Gaussian copula estimation and imputation***Description**

sbgcop.mcmc is used to semiparametrically estimate the parameters of a Gaussian copula. It can be used for posterior inference on the copula parameters, and for imputation of missing values in a matrix of ordinal and/or continuous values.

**Usage**

```
sbgcop.mcmc(Y, S0 = diag(dim(Y)[2]), n0 = dim(Y)[2] + 2, nsamp = 100,
  odens = max(1, round(nsamp/1000)), impute = any(is.na(Y)),
  plugin.threshold = 100, plugin.marginal = (apply(Y, 2, function(x) {
  length(unique(x)) }) > plugin.threshold), seed = 1, verb = TRUE)
```

**Arguments**

Y	an n x p matrix. Missing values are allowed.
S0	a p x p positive definite matrix
n0	a positive integer
nsamp	number of iterations of the Markov chain.
odens	output density: number of iterations between saved samples.
impute	save posterior predictive values of missing data(TRUE/FALSE)?
plugin.threshold	if the number of unique values of a variable exceeds this integer, then plug-in the empirical distribution as the marginal.
plugin.marginal	a logical of length p. Gives finer control over which margins to use the empirical distribution for.
seed	an integer for the random seed
verb	print progress of MCMC(TRUE/FALSE)?

**Details**

This function produces MCMC samples from the posterior distribution of a correlation matrix, using a scaled inverse-Wishart prior distribution and an extended rank likelihood. It also provides imputation for missing values in a multivariate dataset.

**Value**

An object of class psgc containing the following components:

C.psamp	an array of size p x p x nsamp/odens, consisting of posterior samples of the correlation matrix.
---------	--

Y.pmean	the original datamatrix with imputed values replacing missing data
Y.impute	an array of size $n \times p \times \text{nsamp}/\text{odens}$ , consisting of copies of the original data matrix, with posterior samples of missing values included.
LPC	the log-probability of the latent variables at each saved sample. Used for diagnostic purposes.

**Author(s)**

Peter Hoff

**References**<http://www.stat.washington.edu/hoff/>**Examples**

```
fit<-sbgcop.mcmc(swiss)
summary(fit)
plot(fit)
```

sR.sC

*Compute Regression Parameters***Description**

Compute an array of regression parameters from an array of correlation parameters.

**Usage**

```
sR.sC(sC)
```

**Arguments**

sC a  $p \times p \times \text{nsamp}$  array of, made up of nsamp correlation matrices.

**Details**

For each of the nsamp correlation matrices C, a matrix of regression parameters is computed via  $R[j, -j] \leftarrow C[j, -j] \%*\% \text{solve}(C[-j, -j])$

**Value**

a  $p \times p \times \text{nsamp}$  array of regression parameters.

**Author(s)**

Peter Hoff

**Examples**

```
fit<-sbgcop.mcmc(swiss)
```

```
plotci.sA(sR.sC(fit$C.psamp))
```



# Index

- \* **array**
  - plotci.sA, 4
  - qM.sM, 4
  - sR.sC, 7
- \* **datagen**
  - rwish, 5
- \* **distribution**
  - ldmvnorm, 3
  - rwish, 5
- \* **fit<-sbgcop.mcmc(swiss)**
  - qM.sM, 4
- \* **models**
  - sbgcop.mcmc, 6
- \* **multivariate**
  - ldmvnorm, 3
  - qM.sM, 4
  - rwish, 5
  - sbgcop-package, 2
  - sbgcop.mcmc, 6
  - sR.sC, 7
- \* **qC<-qM.sM(fit\$C.psamp)**
  - qM.sM, 4
- \* **qR<-qM.sM(sR.sC(fit\$C.psamp))**
  - qM.sM, 4
- \* **regression**
  - sR.sC, 7

ldmvnorm, 3

plot.psgc (sbgcop.mcmc), 6

plotci.sA, 4

print.sum.psgc (sbgcop.mcmc), 6

qM.sM, 4

rwish, 5

sbgcop (sbgcop-package), 2

sbgcop-package, 2

sbgcop.mcmc, 6

sR.sC, 7

summary.psgc (sbgcop.mcmc), 6