Package ‘copulaSFM’

May 8, 2026

Type Package

Title Copula-Based Stochastic Frontier Models

Version 0.2.0

Description Provides estimation procedures for copula-based stochastic frontier models for cross-sectional data. The package implements maximum likelihood estimation of stochastic frontier models allowing flexible dependence structures between inefficiency and noise terms through various copula families (e.g., Gaussian and Student-t). It enables estimation of technical efficiency scores, log-likelihood values, and information criteria (AIC and BIC). The implemented framework builds upon stochastic frontier analysis introduced by Aigner, Lovell and Schmidt (1977) <doi:10.1016/0304-4076(77)90052-5> and the copula theory described in Joe (2014, ISBN:9781466583221). Empirical applications of copula-based stochastic frontier models can be found in Wiboonpongse et al. (2015) <doi:10.1016/j.ijar.2015.06.001> and Maneejuk et al. (2017, ISBN:9783319562176).

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats, graphics, truncnorm, VineCopula, MASS

NeedsCompilation no

Author Woraphon Yamaka [aut, cre],
Paravee Maneejuk [aut],
Nuttaphong Kaewtathip [aut]

Maintainer Woraphon Yamaka <woraphon.econ@gmail.com>

Repository CRAN

Date/Publication 2026-03-05 04:10:02 UTC

Contents

---

copSFM *Copula based Stochastic frontier Model*

Description

In the standard stochastic frontier model, the two-sided error term V and the one-sided technical inefficiency error term W are assumed to be independent. In this paper, we relax this assumption by modeling the dependence between V and W using copulas.

Usage

```
copSFM(Y, X, family,
                     RHO, LB, UB,
                     RHO2 = NULL, LB2 = NULL, UB2 = NULL,
                     nSim = 50,
                     seed = NULL,
                     maxit = 10000)
```

Arguments

Y	vector of dependent variable
X	matrix of independent variable
family	Copula function eg. Gaussain=1, Student-t=2 (see, Vinecopula package)
RHO	Initial value of the first copula parameter (passed to `par` in `VineCopula`). Note that this parameter is not always a correlation; its interpretation depends on the chosen copula family.
LB	The lower bound of the copula parameter
UB	The upper bound of the copula parameter
RHO2	Optional initial value of the second copula parameter (passed to `par2` in `VineCopula`). This is required for copula families that have two parameters (e.g. Student-t where `RHO2` represents degrees of freedom, and BB families where `RHO2` is a shape parameter). If `NULL`, a one-parameter copula is assumed.
LB2	Optional lower bound for the second copula parameter `RHO2`. Must be supplied when `RHO2` is not `NULL`.
UB2	Optional upper bound for the second copula parameter `RHO2`. Must be supplied when `RHO2` is not `NULL`.
nSim	Number of Monte Carlo draws used to approximate the likelihood integral for each observation. Larger values improve accuracy but increase computation time.
seed	Optional integer seed for reproducibility of the simulation draws used in the likelihood. If `NULL`, no seed is set.
maxit	Maximum number of iterations for the optimizer (`stats::optim` with `"L-BFGS-B"`).

Details

The model follows the stochastic frontier decomposition

$$Y_i = x_i\beta + v_i - u_i,$$

where v_i is a two-sided noise term and $u_i \geq 0$ is the inefficiency term. Dependence between u_i and v_i is introduced through a bivariate copula density $c(\cdot, \cdot)$ from the `VineCopula` package. The log-likelihood is evaluated by simulating draws of u_i from the truncated ALD and approximating the likelihood integral by Monte Carlo averaging (`nSim` draws per observation).

When a two-parameter copula family is used, `RHO2` must be provided along with bounds `LB2` and `UB2`.

The following copula families are supported, together with their parameter bounds:

```
1  = Gaussian copula (par: (LB = -0.99, UB = 0.99))
2  = Student t copula (par: (LB = -0.99, UB = 0.99); par2: (LB = 0, UB = Inf))
3  = Clayton copula (par: (LB = 0.1, UB = Inf))
4  = Gumbel copula (par: [LB = 0.99, UB = Inf))
5  = Frank copula (par: (LB = -Inf, UB =  0) U (LB = 0, UB = Inf))
6  = Joe copula (par: (LB = 0.99, UB = Inf))
7  = BB1 (Clayton-Gumbel) copula (par: (LB = 0, UB = Inf); par2: [LB = 0.99, UB = Inf))
8  = BB6 copula (par: (LB = 0.99, UB = Inf); par2: (LB = 0, UB = Inf))
9  = BB7 (Joe-Clayton) copula (par: [LB = 0.99, UB = Inf); par2: (LB = 0, UB = Inf))
10 = BB8 copula (par: (LB = 0, UB = 1); par2: (LB = 0, UB =  Inf))
13 = Survival Clayton (180 degrees rotation; par: (LB = 0, UB = Inf))
14 = Survival Gumbel (180 degrees rotation; par: [LB = 0.99, UB = Inf))
16 = Survival Joe (180 degrees rotation; par: (LB = 0.99, UB = Inf))
17 = Survival BB1 (par: (LB = 0, UB = Inf); par2: [LB = 0.99, UB = Inf))
18 = Survival BB6 (par: (LB = 0.99, UB = Inf); par2: (0, UB = Inf))
19 = Survival BB7 (par: [LB = 0.99, UB = Inf); par2: (0, UB = Inf))
20 = Survival BB8 (par: (LB = 0, 0.99); par2: (LB = 0, UB = Inf))
23 = Rotated Clayton (90 degrees; par: (LB = -Inf, UB = 0))
24 = Rotated Gumbel (90 degrees; par: (LB = -Inf, UB = -0.99])
26 = Rotated Joe (90 degrees; par: (LB = -Inf, UB = -0.99))
27 = Rotated BB1 (90 degrees; par: (LB = -Inf, UB = 0); par2: [LB = 0.99, UB = Inf))
28 = Rotated BB6 (90 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
29 = Rotated BB7 (90 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
30 = Rotated BB8 (90 degrees; par: (LB = -0.99, UB = 0); par2: (LB = 0, UB = Inf))
33 = Rotated Clayton (270 degrees; par: (LB = -Inf, UB = 0))
34 = Rotated Gumbel (270 degrees; par: (LB = -Inf, UB = -0.99])
36 = Rotated Joe (270 degrees; par: (LB = -Inf, UB = -0.99))
37 = Rotated BB1 (270 degrees; par: (LB = -Inf, UB = 0); par2: [LB = 0.99, UB = Inf))
38 = Rotated BB6 (270 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
39 = Rotated BB7 (270 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
40 = Rotated BB8 (270 degrees; par: (LB = -0.99, UB = 0); par2: (LB = 0, UB = Inf))
```

See the **VineCopula** package (Nagler et al., 2025) for additional details on copula parameterization.

Value

`result`	The result contain the estimated parameters, standard errors, t-stat, and p-value
`AIC`	Akaiki Information Criteria
`BIC`	Bayesian Information Criteria
`Loglikelihood`	Maximum Log-likelihood function
`convergence`	Convergence code returned by `stats::optim`. A value of 0 indicates successful convergence.

Author(s)

Woraphon Yamaka, Paravee Maneejuk, and Nuttaphong Kaewtathip

References

Wiboonpongse, A., Liu, J., Sriboonchitta, S., & Denoeux, T.(2015). Modeling dependence between error components of the stochastic frontier model using copula: application to intercrop coffee production in Northern Thailand. International Journal of Approximate Reasoning, 65, 34-44.

Maneejuk, P., Yamaka, W., & Sriboonchitta, S.(2017). Analysis of global competitiveness using copula-based stochastic frontier kink model. In Robustness in Econometrics (pp. 543-559). Springer, Cham.

Nagler, T., Schepsmeier, U., Stoeber, J., Brechmann, E. C., Graeler, B., Erhardt, T., ... & Killiches, M. (2025, July). Package 'VineCopula'.

Examples

```
## --------------------------------------------------
## Example: Simulation and Model Estimation
## --------------------------------------------------

# Generate simulated stochastic frontier data
data=sfa.simu(nob=50, alpha=c(1,2,0.5),sigV=1,sigU=0.5,family=1,rho=0.5)

## --------------------------------------------------
## Example 1: One-parameter copula (Gaussian)
## --------------------------------------------------

# Specify the copula family and corresponding parameter bounds.
# For example:
# family = 1  -> Gaussian copula   (parameter range: -0.99, 0.99)
# family = 2  -> Student-t copula  (parameter range: -0.99, 0.99)
# family = 3  -> Clayton copula    (parameter range:  0.1,  Inf)

model=copSFM(Y=data$Y,X=data$X,family=1,RHO=0.5,LB=-0.99,UB=0.99)

## --------------------------------------------------
## Example 2: Two-parameter copula (Student-t)
## --------------------------------------------------

# For two-parameter copulas (e.g., Student-t, BB families),
```

```
# the second parameter (RHO2) must be supplied along with
# its corresponding lower and upper bounds.
# For the Student-t copula, RHO2 typically represents
# the degrees of freedom parameter.

data=sfa.simu(nob=50, alpha=c(1,2,0.5),sigV=1,sigU=0.5,family=2,rho=0.3,rho2=4)

model_t <- copSFM(
  Y = data$Y,
  X = data$X,
  family = 2,
  RHO = 0.3,
  LB = -0.99,
  UB = 0.99,
  RHO2 = 45,
  LB2 = 1,
  UB2 = Inf,
  nSim = 50,
  seed = NULL
)
model_t
```

`sfa.simu` *Simulate Data for Stochastic Frontier Analysis*

Description

generates simulated data from a copula-based stochastic frontier model (SFM). The function creates dependent error components using a specified copula family and then constructs output data according to the stochastic frontier specification.

Usage

```
sfa.simu(nob, alpha, sigV, sigU, family, rho, rho2,seed = NULL )
```

Arguments

`nob`	Number of observations to be simulated.
`alpha`	Vector of frontier coefficients including the intercept and slope parameters.
`sigV`	Standard deviation of noise term V.
`sigU`	Standard deviation of inefficiency term U.
`family`	Integer code specifying the copula family used to model the dependence between the noise term and the inefficiency term.
`rho`	Primary dependence parameter of the copula.

rho2 Optional second copula parameter (passed to par2 in VineCopula). This must be provided for copula families that require a second parameter (e.g., degrees of freedom for the Student-t copula, or a shape parameter for BB copulas).

seed Optional integer seed for reproducibility. If NULL, no seed is set.

Details

The simulated data follow the stochastic frontier model

$$y_i = x_i'\alpha + V_i - U_i$$

where

$V_i \sim N(0, \sigma_V^2)$ is a symmetric noise term and $U_i \sim |N(0, \sigma_U^2)|$ is a non-negative inefficiency term.

The dependence between V_i and U_i is modeled using a copula. Let (U_1, U_2) be a pair of uniform random variables generated from a copula $C_\rho(u_1, u_2)$ with dependence parameter ρ. The noise and inefficiency terms are obtained through the transformations

$$V_i = \Phi^{-1}(U_1)\sigma_V$$

$$U_i = F_{TN}^{-1}(U_2; 0, \sigma_U^2)$$

where $\Phi^{-1}(\cdot)$ is the inverse standard normal distribution function and $F_{TN}^{-1}(\cdot)$ is the inverse cumulative distribution function of the truncated normal distribution with support $[0, \infty)$.

The following copula families are supported, together with their parameter bounds:

```
1  = Gaussian copula (par: (LB = -0.99, UB = 0.99))
2  = Student t copula (par: (LB = -0.99, UB = 0.99); par2: (LB = 0, UB = Inf))
3  = Clayton copula (par: (LB = 0.1, UB = Inf))
4  = Gumbel copula (par: [LB = 0.99, UB = Inf))
5  = Frank copula (par: (LB = -Inf, UB =  0) U (LB = 0, UB = Inf))
6  = Joe copula (par: (LB = 0.99, UB = Inf))
7  = BB1 (Clayton-Gumbel) copula (par: (LB = 0, UB = Inf); par2: [LB = 0.99, UB = Inf))
8  = BB6 copula (par: (LB = 0.99, UB = Inf); par2: (LB = 0, UB = Inf))
9  = BB7 (Joe-Clayton) copula (par: [LB = 0.99, UB = Inf); par2: (LB = 0, UB = Inf))
10 = BB8 copula (par: (LB = 0, UB = 1); par2: (LB = 0, UB =  Inf))
13 = Survival Clayton (180 degrees rotation; par: (LB = 0, UB = Inf))
14 = Survival Gumbel (180 degrees rotation; par: [LB = 0.99, UB = Inf))
16 = Survival Joe (180 degrees rotation; par: (LB = 0.99, UB = Inf))
17 = Survival BB1 (par: (LB = 0, UB = Inf); par2: [LB = 0.99, UB = Inf))
18 = Survival BB6 (par: (LB = 0.99, UB = Inf); par2: (0, UB = Inf))
19 = Survival BB7 (par: [LB = 0.99, UB = Inf); par2: (0, UB = Inf))
20 = Survival BB8 (par: (LB = 0, 0.99); par2: (LB = 0, UB = Inf))
23 = Rotated Clayton (90 degrees; par: (LB = -Inf, UB = 0))
24 = Rotated Gumbel (90 degrees; par: (LB = -Inf, UB = -0.99])
26 = Rotated Joe (90 degrees; par: (LB = -Inf, UB = -0.99))
27 = Rotated BB1 (90 degrees; par: (LB = -Inf, UB = 0); par2: [LB = 0.99, UB = Inf))
```

```
28 = Rotated BB6 (90 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
29 = Rotated BB7 (90 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
30 = Rotated BB8 (90 degrees; par: (LB = -0.99, UB = 0); par2: (LB = 0, UB = Inf))
33 = Rotated Clayton (270 degrees; par: (LB = -Inf, UB = 0))
34 = Rotated Gumbel (270 degrees; par: (LB = -Inf, UB = -0.99])
36 = Rotated Joe (270 degrees; par: (LB = -Inf, UB = -0.99))
37 = Rotated BB1 (270 degrees; par: (LB = -Inf, UB = 0); par2: [LB = 0.99, UB = Inf))
38 = Rotated BB6 (270 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
39 = Rotated BB7 (270 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
40 = Rotated BB8 (270 degrees; par: (LB = -0.99, UB = 0); par2: (LB = 0, UB = Inf))
```

See the **VineCopula** package (Nagler et al., 2025) for additional details on copula parameterization.

Value

A list containing simulated output and inputs.

Examples

```
# Example: Simulate data from a copula-based stochastic frontier model
# Set a seed for reproducibility
set.seed(1)
sim <- sfa.simu(nob = 20, alpha = c(1, 0.5, -0.2), sigV = 1, sigU = 1, family = 1, rho = 0.2)
```

TE1 *Technical efficiency measure.*

Description

Computing and plotting the technical efficiency.

Usage

```
TE1(theta, Y, X, family,
                 nSim = 200,
                 rho2 = NULL,
                 seed = NULL,
                 plot = FALSE)
```

Arguments

theta	Numeric vector of estimated model parameters. The expected ordering is: regression coefficients (including intercept), σ_v, σ_u, copula parameter rho, and optionally rho2 if a two-parameter copula is used.
Y	Numeric vector of dependent variable observations.
X	Numeric matrix (or object coercible to a matrix) of independent variables (without intercept column).

`family`	Integer specifying the copula family (see `VineCopula::BiCopPDF`). Some families require a second parameter `rho2`.
`nSim`	Number of Monte Carlo draws used to approximate the conditional expectation. Larger values reduce simulation noise but increase computation time.
`rho2`	Optional second copula parameter. If `NULL` and `theta` contains an additional element beyond `rho`, the function will use that element as `rho2`.
`seed`	Optional integer seed for reproducibility in simulation-based computation. If `NULL`, no seed is set.
`plot`	Logical. If `TRUE`, produces a plot of sorted technical efficiency values.

Details

Technical efficiency is computed as a simulated conditional expectation:

$$TE_i = E[\exp(-U_i) \mid w_i],$$

where $w_i = Y_i - x_i^{\beta}$ and the weights are constructed using the asymmetric Laplace density for the noise term and the copula density capturing dependence between the inefficiency component and the noise component.

The inefficiency term U is generated from a truncated asymmetric Laplace distribution on $(0, \infty)$.

If `plot = TRUE`, the function produces a line plot of sorted technical efficiency values.

The following copula families are supported, together with their parameter bounds:

```
1  = Gaussian copula (par: (LB = -0.99, UB = 0.99))
2  = Student t copula (par: (LB = -0.99, UB = 0.99); par2: (LB = 0, UB = Inf))
3  = Clayton copula (par: (LB = 0.1, UB = Inf))
4  = Gumbel copula (par: [LB = 0.99, UB = Inf))
5  = Frank copula (par: (LB = -Inf, UB =  0) U (LB = 0, UB = Inf))
6  = Joe copula (par: (LB = 0.99, UB = Inf))
7  = BB1 (Clayton-Gumbel) copula (par: (LB = 0, UB = Inf); par2: [LB = 0.99, UB = Inf))
8  = BB6 copula (par: (LB = 0.99, UB = Inf); par2: (LB = 0, UB = Inf))
9  = BB7 (Joe-Clayton) copula (par: [LB = 0.99, UB = Inf); par2: (LB = 0, UB = Inf))
10 = BB8 copula (par: (LB = 0, UB = 1); par2: (LB = 0, UB =  Inf))
13 = Survival Clayton (180 degrees rotation; par: (LB = 0, UB = Inf))
14 = Survival Gumbel (180 degrees rotation; par: [LB = 0.99, UB = Inf))
16 = Survival Joe (180 degrees rotation; par: (LB = 0.99, UB = Inf))
17 = Survival BB1 (par: (LB = 0, UB = Inf); par2: [LB = 0.99, UB = Inf))
18 = Survival BB6 (par: (LB = 0.99, UB = Inf); par2: (0, UB = Inf))
19 = Survival BB7 (par: [LB = 0.99, UB = Inf); par2: (0, UB = Inf))
20 = Survival BB8 (par: (LB = 0, 0.99); par2: (LB = 0, UB = Inf))
23 = Rotated Clayton (90 degrees; par: (LB = -Inf, UB = 0))
24 = Rotated Gumbel (90 degrees; par: (LB = -Inf, UB = -0.99])
26 = Rotated Joe (90 degrees; par: (LB = -Inf, UB = -0.99))
27 = Rotated BB1 (90 degrees; par: (LB = -Inf, UB = 0); par2: [LB = 0.99, UB = Inf))
28 = Rotated BB6 (90 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
29 = Rotated BB7 (90 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
30 = Rotated BB8 (90 degrees; par: (LB = -0.99, UB = 0); par2: (LB = 0, UB = Inf))
```

```
33 = Rotated Clayton (270 degrees; par: (LB = -Inf, UB = 0))
34 = Rotated Gumbel (270 degrees; par: (LB = -Inf, UB = -0.99])
36 = Rotated Joe (270 degrees; par: (LB = -Inf, UB = -0.99))
37 = Rotated BB1 (270 degrees; par: (LB = -Inf, UB = 0); par2: [LB = 0.99, UB = Inf))
38 = Rotated BB6 (270 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
39 = Rotated BB7 (270 degrees; par: (LB = -Inf, UB = -0.99]; par2: (LB = 0, UB = Inf))
40 = Rotated BB8 (270 degrees; par: (LB = -0.99, UB = 0); par2: (LB = 0, UB = Inf))
```

See the **VineCopula** package (Nagler et al., 2025) for additional details on copula parameterization.

Value

A numeric vector of technical efficiency values (one per observation).

Author(s)

Woraphon Yamaka, Paravee Maneejuk, and Nuttaphong Kaewtathip

References

Wiboonpongse, A., Liu, J., Sriboonchitta, S., & Denoeux, T. (2015). Modeling dependence between error components of the stochastic frontier model using copula: application to intercrop coffee production in Northern Thailand. International Journal of Approximate Reasoning, 65, 34-44.

Nagler, T., Schepsmeier, U., Stoeber, J., Brechmann, E. C., Graeler, B., Erhardt, T., ... & Killiches, M. (2025, July). Package 'VineCopula'.

Examples

```
## ------------------------------------------------
## Example: Simulation and Model Estimation
## ------------------------------------------------
data=sfa.simu(nob=50, alpha=c(1,2,0.5),sigV=1,sigU=0.5,family=1,rho=0.5)

# Estimate the copula-based stochastic frontier model
model=copSFM(Y=data$Y,X=data$X,family=1,RHO=0.5,LB=-0.99,UB=0.99, nSim = 50, seed = 123)

## ------------------------------------------------
## Example: Computing Technical Efficiency
## ------------------------------------------------

# Compute technical efficiency estimates
te1=TE1(model$result[,1],Y=data$Y,X=data$X,family=1, nSim = 50, seed = 123, plot = TRUE)
```

Index