Package ‘CommonMean.Copula’

January 4, 2022

Type Package

Title Common Mean Vector under Copula Models

Version 1.0.4

Date 2022-01-04

Author Jia-Han Shih

Maintainer Jia-Han Shih <tommy355097@gmail.com>

Description Estimate bivariate common mean vector under copula models with known correlation. In the current version, available copulas are the Clayton, Gumbel, Frank, Farlie-Gumbel-Morgenstern (FGM), and normal copulas. See Shih et al. (2019) <doi:10.1080/02331888.2019.1581782> and Shih et al. (2021) <under review> for details under the FGM and general copulas, respectively.

Depends pracma, mvtnorm

License GPL-2

Encoding UTF-8

RoxygenNote 7.1.2

Repository CRAN

NeedsCompilation no

Date/Publication 2022-01-04 11:50:07 UTC

R topics documented:

CommonMean.Copula-package *Common Mean Vector under Copula Models*

Description

Estimate bivariate common mean vector under copula models with known correlation. A maximum likelihood estimation procedure is employed. In the current version, available copulas are the Clayton, Gumbel, Frank, Farlie-Gumbel-Morgenstern (FGM), and normal copulas. See Shih et al. (2019) and Shih et al. (2021) for details under the FGM and general copulas, respectively.

Details

The method implemented in this package can be used for bivariate meta-analyses. See Shih et al. (2019) and Shih et al. (2021) for the example of bivariate entrance exam data analysis.

Author(s)

Jia-Han Shih

Maintainer: Jia-Han Shih <tommy355097@gmail.com>

References

Shih J-H, Konno Y, Chang Y-T, Emura T (2019) Estimation of a common mean vector in bivariate meta-analysis under the FGM copula, Statistics 53(3): 673-95.

Shih J-H, Konno Y, Emura T (2021-) Copula-based estimation methods for a common mean vector for bivariate meta-analyses, under review.

CommonMean.Copula *Estimate bivariate common mean vector under copula models*

Description

Estimate the common mean vector under copula models with known correlation. A maximum likelihood estimation procedure is employed. See Shih et al. (2019) and Shih et al. (2021) for details under the Farlie-Gumbel-Morgenstern (FGM) and general copulas, respectively.

Usage

```
CommonMean.Copula(Y1, Y2, Sigma1, Sigma2, rho, copula = "Clayton")
```

Arguments

- `Y1` Outcome 1
- `Y2` Outcome 2
- `Sigma1` Standard deviation of outcome 1.
- `Sigma2` Standard deviation of outcome 2.
- `rho` Correlation coefficient between outcomes.
- `copula` The copula to be used with possible options "Clayton", "Gumbel", "Frank", "FGM", and "normal".

Details

We apply "optim" routine to maximize the log-likelihood function. In addition, boundary corrected correlations will be used (Shih et al., 2019).

Value

- `Outcome 1` Outcome 1.
- `Outcome 2` Outcome 2.
- `Correlation` Correlation coefficient between outcomes.
- `Sample size` Sample size.
- `Copula` Selected copula.
- `Copula parameter` Copula parameter.
- `Corrected correlation` Boundary corrected correlations.
- `CommonMean 1` Estimation results of outcome 1.
- `CommonMean 2` Estimation results of outcome 2.
- `V` Covariance matrix of the common mean vector estimate.
- `Log-likelihood values` Fitted log-likelihood values.

Note

When rho is 1 or -1, there are some computational issues since the copula parameter may correspond to infinite or negative infinite under some copulas. For the Clayton copula, if rho > 0.95, it will be approximated by 0.95. For the Frank copula, if rho > 0.95 or rho < -0.95, it will be approximated by 0.95 or -0.95, respectively.

References

Shih J-H, Konno Y, Chang Y-T, Emura T (2019) Estimation of a common mean vector in bivariate meta-analysis under the FGM copula, Statistics 53(3): 673-95.

Shih J-H, Konno Y, Emura T (2021-) Copula-based estimation methods for a common mean vector for bivariate meta-analyses, under review.

Examples

```
library(CommonMean.Copula)
Y1 = c(35,25,30,50,60) # outcome 1
Y2 = c(30,30,50,65,40) # outcome 2
Sigma1 = c(1.3,1.4,1.5,2.0,1.8) # SE of outcome 1
Sigma2 = c(1.7,1.9,2.5,2.2,1.8) # SE of outcome 2
rho = c(0.4,0.7,0.6,0.7,0.6) # correlation between two outcomes
CommonMean.Copula(Y1,Y2,Sigma1,Sigma2,rho) # input
```

Index