

# Package ‘copulareg’

July 22, 2025

**Type** Package  
**Title** Copula Regression  
**Version** 0.1.0  
**Description** Fits multivariate models in an R-vine pair copula construction framework, in such a way that the conditional copula can be easily evaluated. In addition, the package implements functionality to compute or approximate the conditional expectation via the conditional copula.  
**License** MIT + file LICENCE  
**Imports** rvinecopulib (>= 0.5.4.1.0)  
**Encoding** UTF-8  
**LazyData** true  
**RoxygenNote** 7.1.0  
**NeedsCompilation** no  
**Author** Simon Boge Brant [aut, cre, cph],  
Ingrid Hobæk Haff [aut]  
**Maintainer** Simon Boge Brant <simonbb@math.uio.no>  
**Repository** CRAN  
**Date/Publication** 2021-02-22 20:30:05 UTC

## Contents

|                             |                   |
|-----------------------------|-------------------|
| copulareg . . . . .         | <a href="#">2</a> |
| plot.copulareg . . . . .    | <a href="#">3</a> |
| predict.copulareg . . . . . | <a href="#">4</a> |
| <b>Index</b>                | <a href="#">5</a> |

---

copulareg

*copulareg*


---

## Description

This function fits joint distributions with an R-vine pair copula structure, that is constructed in a specific way so that the conditional density and distribution of the variable  $y$  can be computed explicitly.

## Usage

```
copulareg.default(
  y,
  x,
  var_type_y,
  var_type_x,
  family_set = c("gaussian", "clayton", "gumbel"),
  extra_x = NULL,
  extra_y = NULL
)
```

## Arguments

|                         |                                                                                                                                                                                                                      |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>y</code>          | A vector of $n$ observations of the (univariate) outcome variable $y$                                                                                                                                                |
| <code>x</code>          | A ( $n \times p$ ) matrix of $n$ observations of $p$ covariates                                                                                                                                                      |
| <code>var_type_y</code> | A character that has to be specified as "d" or "c" to indicate whether $y$ is discrete or continuous, respectively.                                                                                                  |
| <code>var_type_x</code> | A vector of $p$ characters that have to take the value "c" or "d" to indicate whether each margin of the covariates is discrete or continuous.                                                                       |
| <code>family_set</code> | A vector of strings that specifies the set of pair-copula families that the fitting algorithm chooses from. For an overview of which values that can be specified, see the documentation for <a href="#">bicop</a> . |
| <code>extra_x</code>    | Optional extra values of $x$ to use for estimating the margins of the covariates.                                                                                                                                    |
| <code>extra_y</code>    | Optional extra values of $y$ to use to estimate the margin of $y$ .                                                                                                                                                  |

## Value

An object of the class 'copulareg', which contains an object of the class 'vinecop', see `rvinecop-ulib::vinecop`, an environment that stores the transformed variables at each level of the pair copula for the training data, the marginal distributions of the covariates, the marginal distribution of the response, and the  $y$ -training values.

**Examples**

```
# Compile some test data
data('ChickWeight')
set.seed(10)
tr <- sample(c(TRUE, FALSE), nrow(ChickWeight), TRUE, c(0.7, 0.3))
y_tr <- ChickWeight$weight[tr]
y_te <- ChickWeight$weight[!tr]
x_tr <- apply(ChickWeight[tr, -1], 2, as.numeric)
x_te <- apply(ChickWeight[!tr, -1], 2, as.numeric)
var_type_x <- apply(x_tr, 2,
  function(x) if(length(unique(x)) < 10) "d" else "c")

# Fit model to training data
md <- copulareg::copulareg(y_tr, x_tr, "c", var_type_x)

# Predict for a new data matrix
pred <- predict(md, new_x = x_te)

# Plot residuals for test data against covariates
plot(data.frame(residual = y_te - pred, x_te))

# Plot residuals against fitted
plot(md)

# Plot prediction error against predicted values
plot(md, new_x=x_te, new_y=y_te)
```

---

|                |                |
|----------------|----------------|
| plot.copulareg | <i>predict</i> |
|----------------|----------------|

---

**Description**

Plot the residuals against the fitted values for a copulareg object, or predicted values against the prediction error

**Usage**

```
## S3 method for class 'copulareg'
plot(x, new_x = NULL, new_y = NULL, ...)
```

**Arguments**

|       |                                                                                                                                                               |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x     | Model fit as returned by copulareg                                                                                                                            |
| new_x | optional matrix of covariate values to compute the predicted values of the outcome for. If not specified, the fitted values for the training sample are used. |
| new_y | Optional vector if the plot should show predicted values and prediction error.                                                                                |
| ...   | additional parameters to plot.                                                                                                                                |

**Value**

No return value, called to produce plot.

---

|                   |                |
|-------------------|----------------|
| predict.copulareg | <i>predict</i> |
|-------------------|----------------|

---

**Description**

Computes predictions based on a fitted copulareg model.

**Usage**

```
## S3 method for class 'copulareg'
predict(object, new_x = NULL, eps = 0.01, cont_method = "Localmedian", ...)
```

**Arguments**

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object      | Model fit as returned by copulareg                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| new_x       | optional matrix of covariate values to compute the predicted values of the outcome for. If not specified, the predicted values for the training sample is returned.                                                                                                                                                                                                                                                                                                                                                                         |
| eps         | Interval between each interpolation point when integrating to evaluate the predicted value of y, in the case where y is continuous. If y is discrete this parameter is ignored.                                                                                                                                                                                                                                                                                                                                                             |
| cont_method | Specifies the method used to compute the expected values. Can be specified as 'Localmedian' or 'Trapezoidalsurv'. The first method divides the range of the observed values of y into subintervals according to the argument 'eps', where the sub-integral is approximated as the measure of the interval weighted by the local median on the interval. The second method computes the integral by integrating the survival function using the trapezoidal rule, by transforming the outcome into a positive variable by adding a constant. |
| ...         | unused.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

**Value**

A vector of predicted y-values for each row of new\_x, or for the training data if new\_x is not supplied.

# Index

`bicop`, [2](#)

`copulareg`, [2](#)

`plot.copulareg`, [3](#)

`predict.copulareg`, [4](#)