

Package ‘vinereg’

August 31, 2026

Type Package

Title D-Vine Quantile Regression

Version 0.13.0

Maintainer Thomas Nagler <mail@tnagler.com>

Description Implements D-vine quantile regression models with parametric or nonparametric pair-copulas. See Kraus and Czado (2017) <[doi:10.1016/j.csda.2016.12.009](https://doi.org/10.1016/j.csda.2016.12.009)> and Schallhorn et al. (2017) <[doi:10.48550/arXiv.1705.08310](https://doi.org/10.48550/arXiv.1705.08310)>.

License GPL-3

SystemRequirements C++17

Imports rvinecopulib (>= 0.7.1.1.0), kde1d (>= 1.1.0), Rcpp, assertthat

LinkingTo rvinecopulib, RcppEigen, Rcpp, BH, wdm, RcppThread, kde1d

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, ggplot2, AppliedPredictiveModeling, quantreg, tidyr, dplyr, purrr, scales, mgcv, testthat, covr

VignetteBuilder knitr

URL <https://tnagler.github.io/vinereg/>

BugReports <https://github.com/tnagler/vinereg/issues>

Encoding UTF-8

NeedsCompilation yes

Author Thomas Nagler [aut, cre],
Dani Kraus [ctb]

Repository CRAN

Date/Publication 2026-08-31 21:50:02 UTC

Contents

vinereg-package	2
cpl	3
cpdf	3
cpit	4
plot_effects	5
predict.vinereg	6
vinereg	7
vinereg-methods	9

Index	11
--------------	-----------

vinereg-package	<i>vinereg: D-vine regression models</i>
-----------------	--

Description

Fit D-vine copula regression models for conditional mean and quantile prediction with continuous or discrete variables. The package provides automatic covariate selection, fixed-order models, conditional distribution diagnostics, and marginal fitted-quantile plots.

Author(s)

Maintainer: Thomas Nagler <mail@tnagler.com>

Authors:

- Thomas Nagler <mail@tnagler.com>

Other contributors:

- Dani Kraus [contributor]

References

Kraus, D. and Czado, C. (2017). D-vine copula based quantile regression. *Computational Statistics & Data Analysis*, 110, 1–18. doi:10.1016/j.csda.2016.12.009

Schallhorn, N., Kraus, D., Nagler, T., and Czado, C. (2017). D-vine quantile regression with discrete variables. doi:10.48550/arXiv.1705.08310

See Also

Useful links:

- <https://tnagler.github.io/vinereg/>
- Report bugs at <https://github.com/tnagler/vinereg/issues>

c1l	<i>Conditional log-likelihood</i>
-----	-----------------------------------

Description

Calculates the conditional log-likelihood of the response given the covariates.

Usage

```
c1l(object, newdata, cores = 1)
```

Arguments

object	an object of class <code>vinereg</code> .
newdata	a data frame containing the response and covariates from the original formula, with matching classes and factor levels.
cores	integer; the number of cores to use for computations.

Value

The scalar conditional log-likelihood evaluated on `newdata`.

Examples

```
# simulate data
x <- matrix(rnorm(100), 50, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(50, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)

c1l(fit, dat)
fit$stats$c1l
```

cpdf	<i>Conditional density or probability mass</i>
------	--

Description

Calculates the conditional density of a continuous response or conditional probability mass of a discrete response given the covariates.

Usage

```
cpdf(object, newdata, cores = 1)
```

Arguments

object	an object of class <code>vinereg</code> .
newdata	a data frame containing the response and covariates from the original formula, with matching classes and factor levels.
cores	integer; the number of cores to use for computations.

Value

A numeric vector containing one conditional density or probability mass per row of `newdata`.

Examples

```
# simulate data
x <- matrix(rnorm(100), 50, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(50, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)

cpdf(fit, dat)
```

cpit	<i>Conditional probability integral transform</i>
------	---

Description

Evaluates the fitted conditional distribution of the response given the covariates. For a discrete response, this is the conditional CDF at the observed category, not a randomized probability integral transform.

Usage

```
cpit(object, newdata, cores = 1)
```

Arguments

object	an object of class <code>vinereg</code> .
newdata	a data frame containing the response and covariates from the original formula, with matching classes and factor levels.
cores	integer; the number of cores to use for computations.

Value

A numeric vector containing one conditional CDF value per row of `newdata`.

Examples

```
# simulate data
x <- matrix(rnorm(100), 50, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(50, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)

hist(cpit(fit, dat)) # should be approximately uniform
```

plot_effects

*Plot marginal effects of a D-vine regression model***Description**

The points show fitted conditional quantiles against each requested variable. For variable X_k , the smooth curve estimates $E[\hat{Q}_\alpha(Y \mid X) \mid X_k = x]$. It therefore averages over the conditional distribution of the other variables. A curve for an unselected variable can vary when that variable is associated with selected predictors. The curve is descriptive and is not a partial-dependence or causal effect.

Usage

```
plot_effects(object, alpha = c(0.1, 0.5, 0.9), vars = object$order)
```

Arguments

object	a vinereg object
alpha	vector of quantile levels.
vars	vector of expanded variable names to display.

Value

A `ggplot2::ggplot()` object.

Examples

```
# simulate data
x <- matrix(rnorm(100), 50, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(50, 2, 0.5)))

# fit vine regression model
fit <- vinereg(y ~ ., dat)
plot_effects(fit)
```

predict.vinereg	<i>Predict conditional mean and quantiles from a D-vine regression model</i>
-----------------	--

Description

Predict conditional mean and quantiles from a D-vine regression model

Usage

```
## S3 method for class 'vinereg'
predict(object, newdata, alpha = 0.5, cores = 1, ...)

## S3 method for class 'vinereg'
fitted(object, alpha = 0.5, ...)
```

Arguments

object	an object of class vinereg.
newdata	a data frame containing the covariates from the original formula, with matching classes and factor levels. If omitted, the model frame used for fitting is used.
alpha	vector of quantile levels; NA predicts the mean based on an average of the 1:10 / 11-quantiles.
cores	integer; the number of cores to use for computations.
...	unused.

Value

A data frame with one row per observation and one column per value of alpha. Columns are named by their quantile level; the conditional mean column is named mean.

See Also

[vinereg](#)

Examples

```
# simulate data
x <- matrix(rnorm(100), 50, 2)
y <- x %*% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(50, 2, 0.5)))

## fixed variable order (no selection)
(fit <- vinereg(y ~ ., dat, order = c("x.2", "x.1", "z")))

# model predictions
mu_hat <- predict(fit, newdata = dat, alpha = NA) # mean
```

```

med_hat <- predict(fit, newdata = dat, alpha = 0.5) # median

# observed vs predicted
plot(cbind(y, mu_hat))

```

vinereg

D-vine regression models

Description

Sequential estimation of a regression D-vine for the purpose of quantile prediction as described in Kraus and Czado (2017).

Usage

```

vinereg(
  formula,
  data,
  family_set = "parametric",
  selcrit = "aic",
  order = NA,
  par_1d = list(),
  weights = numeric(),
  cores = 1,
  ...,
  uscale = FALSE
)

```

Arguments

formula	a two-sided formula containing untransformed variable names. Compute transformations and interactions in data before fitting.
data	data frame (or object coercible by <code>as.data.frame()</code>) containing the variables in the model.
family_set	see family_set argument of <code>rvinecopulib::bicop()</code> .
selcrit	selection criterion based on conditional log-likelihood. "aic" (default) and "bic" penalize model complexity; "loglik" imposes no penalty.
order	the order of covariates in the D-vine, provided as vector of variable names. An unordered factor name expands to all of its dummy variables in level order. Expanded dummy names remain accepted. The order is selected automatically if order = NA (default).
par_1d	list of options passed to <code>kde1d::kde1d()</code> , must be one value for each margin, e.g. <code>list(xmin = c(0, 0, NaN))</code> if the response and first covariate have non-negative support.

<code>weights</code>	optional numeric vector of nonnegative observation weights. Supply one value per row of data or per row of the model frame after missing values have been omitted. Missing weights are omitted as well.
<code>cores</code>	integer; the number of cores to use for computations.
<code>...</code>	further arguments passed to <code>rvinecopulib::bicop()</code> .
<code>uscale</code>	if TRUE, vinereg assumes that marginal distributions have been taken care of in a preliminary step.

Details

If discrete variables are declared as `ordered()` or `factor()`, they are handled as described in Panagiotelis et al. (2012).

Value

An object of class `vinereg`. It is a list containing the elements

formula the formula used for the fit.

selcrit criterion used for variable selection.

model_frame the data used to fit the regression model.

margins list of marginal models fitted by `kde1d::kde1d()`.

vine an `rvinecopulib::vinecop_dist()` object containing the fitted D-vine.

stats fit statistics such as conditional log-likelihood/AIC/BIC and p-values for each variable's contribution.

order order of the covariates chosen by the variable selection algorithm.

selected_vars indices of selected variables.

factor_map mapping from original variables to expanded variables.

Use `predict.vinereg()` to predict conditional quantiles. `summary.vinereg()` shows the contribution of each selected variable with the associated p-value derived from a likelihood ratio test.

References

Kraus and Czado (2017), D-vine copula based quantile regression, Computational Statistics and Data Analysis, 110, 1-18

Panagiotelis, A., Czado, C., & Joe, H. (2012). Pair copula constructions for multivariate discrete data. Journal of the American Statistical Association, 107(499), 1063-1072.

See Also

`predict.vinereg`

Examples

```
# simulate data
x <- matrix(rnorm(100), 50, 2)
y <- x %%% c(1, -2)
dat <- data.frame(y = y, x = x, z = as.factor(rbinom(50, 2, 0.5)))

# fit vine regression model
(fit <- vinereg(y ~ ., dat))

# inspect model
summary(fit)
plot_effects(fit)

# model predictions
mu_hat <- predict(fit, newdata = dat, alpha = NA) # mean
med_hat <- predict(fit, newdata = dat, alpha = 0.5) # median

# observed vs predicted
plot(cbind(y, mu_hat))

## fixed variable order (no selection)
(fit <- vinereg(y ~ ., dat, order = c("x.2", "x.1", "z")))
```

vinereg-methods

Standard methods for D-vine regression models

Description

These methods extract the fitted model's call information, data, effective degrees of freedom, conditional log-likelihood, and variable-wise summary.

Usage

```
## S3 method for class 'vinereg'
print(x, ...)

## S3 method for class 'vinereg'
summary(object, ...)

## S3 method for class 'vinereg'
logLik(object, ...)

## S3 method for class 'vinereg'
nobs(object, use.fallback = TRUE, ...)

## S3 method for class 'vinereg'
formula(x, ...)
```

```
## S3 method for class 'vinereg'  
model.frame(formula, ...)
```

Arguments

x, object, formula	
	a vinereg object.
...	unused.
use.fallback	unused; included for compatibility with <code>nobs()</code> .

Value

`print()` returns x invisibly. `summary()` returns a data frame with one row for the response and each selected predictor. `logLik()` returns an object of class `logLik`; its `df` attribute contains the effective degrees of freedom. `nobs()` returns the number of observations used for fitting. `formula()` and `model.frame()` return the model formula and frame.

Index

`as.data.frame()`, 7

`c11`, 3

`cpdf`, 3

`cpit`, 4

`fitted.vinereg(predict.vinereg)`, 6

`formula.vinereg(vinereg-methods)`, 9

`ggplot2::ggplot()`, 5

`kde1d::kde1d()`, 7, 8

`logLik.vinereg(vinereg-methods)`, 9

`model.frame.vinereg(vinereg-methods)`, 9

`nobs()`, 10

`nobs.vinereg(vinereg-methods)`, 9

`plot_effects`, 5

`predict.vinereg`, 6, 8

`predict.vinereg()`, 8

`print.vinereg(vinereg-methods)`, 9

`rvinecopulib::bicop()`, 7, 8

`rvinecopulib::vinecop_dist()`, 8

`summary.vinereg(vinereg-methods)`, 9

`vinereg`, 6, 7

`vinereg-methods`, 9

`vinereg-package`, 2