

# Package ‘copulaStan’

February 20, 2026

**Title** Fit Bivariate Gaussian and Clayton Copula Models Using Stan

**Version** 0.5.0

**Description** Provides Stan wrapper functions to fit bivariate Gaussian, Clayton, and Joe copula models with flexible marginal distributions. Models are pre-compiled using 'CmdStan' via the 'instantiate' package for fast sampling.

**Depends** R (>= 4.1.0)

**Imports** posterior, cli, rlang

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Suggests** cmdstanr, knitr, rmarkdown, testthat (>= 3.0.0), pkgdown,  
copula

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**URL** <https://benlug.github.io/copulaStan/>

**Additional\_repositories** <https://stan-dev.r-universe.dev>

**NeedsCompilation** no

**Author** Benedikt Lugauer [aut, cre]

**Maintainer** Benedikt Lugauer <benedikt.lugauer@pm.me>

## Contents

coef.copula_fit . . . . .	2
copula_fit . . . . .	2
fit_bivariate_copula . . . . .	3
get_stan_model . . . . .	4
print.copula_fit . . . . .	5
summary.copula_fit . . . . .	5

## Index

6

`coef.copula_fit`      *Extract point estimates from a copula\_fit*

### Description

Extract point estimates from a `copula_fit`

### Usage

```
## S3 method for class 'copula_fit'
coef(object, ...)
```

### Arguments

<code>object</code>	A <code>copula_fit</code> object.
<code>...</code>	Additional arguments (unused).

### Value

A named numeric vector of posterior means.

`copula_fit`      *Create a copula\_fit object*

### Description

Create a `copula_fit` object

### Usage

```
copula_fit(fit, copula, marginals, data_dim)
```

### Arguments

<code>fit</code>	CmdStanMCMC fit object.
<code>copula</code>	Copula type string.
<code>marginals</code>	Character vector of marginal distribution names.
<code>data_dim</code>	Dimensions of the input data.

### Value

A `copula_fit` S3 object.

---

**fit\_bivariate\_copula** *Fit Bivariate Copula Model*

---

**Description**

Fits a bivariate copula model to data with marginal parameter estimation. Supports Gaussian, Clayton, and Joe copulas with normal, lognormal, exponential, or beta marginal distributions. Uses CmdStan for Bayesian inference via the `cmdstanr` package.

**Usage**

```
fit_bivariate_copula(
  U,
  copula,
  marginals,
  iter = 1000,
  chains = 4,
  warmup = 1000,
  thin = 1,
  seed = NULL,
  adapt_delta = 0.8,
  max_treedepth = 10,
  parallel_chains = 1,
  refresh = 500
)
```

**Arguments**

<code>U</code>	A numeric matrix with exactly two columns containing the observed data. Each column corresponds to one variable.
<code>copula</code>	Character string specifying the copula type. One of "gaussian", "clayton", or "joe".
<code>marginals</code>	A character vector of length 2 specifying the marginal distributions. Each element must be one of "normal", "lognormal", "exponential", or "beta".
<code>iter</code>	Number of sampling iterations per chain (after warmup). Default is 1000.
<code>chains</code>	Number of MCMC chains. Default is 4.
<code>warmup</code>	Number of warmup iterations per chain. Default is 1000.
<code>thin</code>	Thinning rate. Default is 1.
<code>seed</code>	Random seed for reproducibility. Default is <code>NULL</code> .
<code>adapt_delta</code>	Target acceptance rate for NUTS. Default is 0.8.
<code>max_treedepth</code>	Maximum tree depth for NUTS. Default is 10.
<code>parallel_chains</code>	Number of chains to run in parallel. Default is 1.
<code>refresh</code>	How often to print progress (in iterations). Set to 0 for silent. Default is 500.

**Value**

A `copula_fit` object (S3 class) containing:

`fit` The CmdStanMCMC fit object from `cmdstanr`.  
`copula` The copula type used.  
`marginals` The marginal distributions used.  
`data_dim` Dimensions of the input data (rows, columns).

**Examples**

```
## Not run:
library(copula)
library(copulaStan)

set.seed(2024)
n <- 1000
cop <- normalCopula(param = 0.5, dim = 2)
margins <- c("norm", "lnorm")
params <- list(list(mean = 0.8, sd = 2), list(meanlog = 0, sdlog = 0.8))
mvdc_copula <- mvdc(cop, margins = margins, paramMargins = params)
data <- rMvdc(n, mvdc_copula)

fit <- fit_bivariate_copula(data,
  copula = "gaussian",
  marginals = c("normal", "lognormal"),
  seed = 2024
)
print(fit)
summary(fit)

## End(Not run)
```

`get_stan_model`

*Get the compiled Stan model (compiles on first use, caches for reuse)*

**Description**

Get the compiled Stan model (compiles on first use, caches for reuse)

**Usage**

```
get_stan_model()
```

---

```
print.copula_fit      Print a copula_fit object
```

---

**Description**

Print a copula\_fit object

**Usage**

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

**Arguments**

x	A copula_fit object.
...	Additional arguments (unused).

**Value**

Invisibly returns x.

---

```
summary.copula_fit      Summarize a copula_fit object
```

---

**Description**

Summarize a copula\_fit object

**Usage**

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

**Arguments**

object	A copula_fit object.
...	Additional arguments (unused).

**Value**

A tibble of parameter summaries from `posterior::summarise_draws()`.

# Index

```
* internal
    copula_fit, 2
    get_stan_model, 4

coef.copula_fit, 2
copula_fit, 2

fit_bivariate_copula, 3

get_stan_model, 4

print.copula_fit, 5

summary.copula_fit, 5
```