

# Package ‘dda’

November 14, 2025

**Type** Package

**Title** Direction Dependence Analysis

**Version** 0.1.1

**License** MIT + file LICENSE

**Maintainer** Wolfgang Wiedermann <wiedermannw@missouri.edu>

## Description

A collection of tests to analyze the causal direction of dependence in linear models (Wiedermann, W., & von Eye, A., 2025, ISBN: 9781009381390). The package includes functions to perform Direction Dependence Analysis for variable distributions, residual distributions, and independence properties of predictors and residuals in competing causal models. In addition, the package contains functions to test the causal direction of dependence in conditional models (i.e., models with interaction terms) For more information see <<https://www.ddaproject.com>>.

**Imports** dHSIC, energy, foreach, graphics, methods, stats

**Suggests** boot, doParallel, devtools, interactions, iterators, lmtest, moments, spelling, testthat (>= 3.0.0), waldo

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**URL** <https://github.com/wwiedermann/dda>

**NeedsCompilation** no

**Author** Wolfgang Wiedermann [aut, cre],  
Megan Hirni [aut]

**Repository** CRAN

**Date/Publication** 2025-11-14 21:00:02 UTC

**BugReports** <https://github.com/wwiedermann/dda/issues>

## Contents

cdda.indep . . . . .	2
cdda vardist . . . . .	5

dda.indep . . . . .	8
dda.resdist . . . . .	10
dda vardist . . . . .	11

<b>Index</b>	<b>14</b>
--------------	-----------

---

cdda.indep	<i>Conditional Direction Dependence Analysis: Independence Properties</i>
------------	---

---

## Description

cdda.indep evaluates asymmetries of predictor-error independence of competing conditional models ( $y \sim x * m$  vs.  $x \sim y * m$  with  $m$  being a continuous or categorical moderator).

print returns the output of standard linear model coefficients for competing target and alternative models.

plot returns graphs for results of cdda.indep.

summary returns test statistics from the cdda.indep class object.

## Usage

```
cdda.indep(
  formula = NULL,
  pred = NULL,
  mod = NULL,
  modval = NULL,
  data = list(),
  hetero = FALSE,
  diff = FALSE,
  nlfun = NULL,
  hsic.method = "gamma",
  B = 200,
  boot.type = "perc",
  conf.level = 0.95,
  parallelize = FALSE,
  cores = 1,
  ...
)

## S3 method for class 'cdda.indep'
print(x, ...)

## S3 method for class 'cdda.indep'
plot(x = NULL, stat = NULL, ylim = NULL, ...)

## S3 method for class 'cdda.indep'
summary(
```

```

    object,
    nlfun = FALSE,
    hetero = FALSE,
    hsic = TRUE,
    hsic.diff = FALSE,
    dcor = TRUE,
    dcor.diff = FALSE,
    mi.diff = FALSE,
    ...
  )

```

### Arguments

formula	Symbolic formula of the model to be tested or an lm object.
pred	A character indicating the variable name of the predictor which serves as the outcome in the alternative model.
mod	A character indicating the variable name of the moderator.
modval	Characters or a numeric sequence specifying the moderator values used in post-hoc probing. Possible characters include c("mean", "median", "JN").modval = "mean" tests the interaction effect at the moderator values $M - 1SD$ , $M$ , and $M + 1SD$ ; modval = "median" uses $Q1$ , $Md$ , and $Q3$ . The Johnson-Neyman approach is applied when modval = "JN" with conditional effects being evaluated at the boundary values of the significance regions. When a numeric sequence is specified, the pick-a-point approach is used for the selected numeric values.
data	A required data frame containing the variables in the model.
hetero	A logical value indicating whether separate homoscedasticity tests should be returned when using summary, default is FALSE.
diff	A logical value indicating whether differences in HSIC, dCor, and MI values should be computed. Bootstrap confidence intervals are computed using B bootstrap samples.
nlfun	Determines handling of non-linear correlation tests depending on the function used: <ul style="list-style-type: none"> <li>• <b>For cdda.indep:</b> Either a numeric value or a function of .Primitive type used for non-linear correlation tests. When numeric, the value is used in a power transformation.</li> <li>• <b>For summary:</b> A logical value indicating whether non-linear correlation tests should be returned in the output. Default is FALSE.</li> </ul>
hsic.method	A character indicating the inference method for the Hilbert-Schmidt Independence Criterion. Must be one of the four specifications c("gamma", "eigenvalue", "boot", "permutation").hsic.method = "gamma" is the default.
B	Number of permutations for separate dCor tests and number of resamples when hsic.method = c("boot", "permutation") or diff = TRUE.
boot.type	A character indicating the type of bootstrap confidence intervals. Must be one of the two specifications c("perc", "bca"). boot.type = "perc" is the default.
conf.level	Confidence level for bootstrap confidence intervals.

parallelize	A logical value indicating whether bootstrapping is performed on multiple cores. Only used if <code>diff = TRUE</code> .
cores	A numeric value indicating the number of cores. Only used if <code>parallelize = TRUE</code> .
...	Additional arguments to be passed to the function.
x	An object of class <code>cdda.indep</code> when using <code>print</code> or <code>plot</code> .
stat	A character indicating the CDDA statistic to be plotted with the options <code>c("hsic.diff", "dcor.diff", "mi.diff")</code> .
ylim	A numeric vector of length 2 indicating the y-axis limits if <code>NULL</code> , the function will set the limits automatically.
object	An object of class <code>cdda.indep</code> when using <code>summary</code> .
hsic	A logical value indicating whether separate HSIC tests should be returned when using <code>summary</code> , default is <code>TRUE</code> .
hsic.diff	A logical value indicating whether HSIC difference statistics should be returned when using <code>summary</code> , default is <code>FALSE</code> .
dcor	A logical value indicating whether separate Distance Correlation (dCor) tests should be returned when using <code>summary</code> , default is <code>TRUE</code> .
dcor.diff	A logical value indicating whether dCor difference statistics should be returned when using <code>summary</code> , default is <code>FALSE</code> .
mi.diff	A logical value indicating whether Mutual Information (MI) difference statistics should be returned when using <code>summary</code> , default is <code>FALSE</code> .

### Value

An object of class `dda.indep` containing the results of independence tests of Conditional Direction Dependence Analysis.

An object of class `cdda.indep` with competing model coefficients.

### References

Wiedermann, W., & von Eye, A. (2025). *Direction Dependence Analysis: Foundations and Statistical Methods*. Cambridge, UK: Cambridge University Press.

### See Also

[dda.indep](#) for an unconditional version.

### Examples

```
set.seed(321)
n <- 700

## --- generate moderator

z <- sort(rnorm(n))
z1 <- z[z <= 0]
```

```

z2 <- z[z > 0]

## --- x -> y when z <= 0

x1 <- rchisq(length(z1), df = 4) - 4
e1 <- rchisq(length(z1), df = 3) - 3
y1 <- 0.5 * x1 + e1

## --- y -> x when m z > 0

y2 <- rchisq(length(z2), df = 4) - 4
e2 <- rchisq(length(z2), df = 3) - 3
x2 <- 0.25 * y2 + e2

y <- c(y1, y2); x <- c(x1, x2)

d <- data.frame(x, y, z)

m <- lm(y ~ x * z, data = d)

result <- cdda.indep(m,
  pred = "x",
  mod = "z",
  modval = c(-1, 1),
  data = d,
  hetero = TRUE,
  diff = TRUE,
  parallelize = TRUE,
  cores = 2,
  nlfun = 2,
  B = 2)
# Note: Only 2 bootstrap samples are created here to lower computation time

print(result)

plot(result, stat = "dcor.diff")

summary(result, hetero = FALSE)

```

**Description**

cdda.vardist evaluates variable distributions of competing conditional models ( $y \sim x * m$  vs.  $x \sim y * m$  with  $m$  being a continuous or categorical moderator).

print returns the output of standard linear model coefficients for competing target and alternative models.

plot returns graphs for results of cdda.vardist.

summary returns test statistics from the cdda.vardist class object.

## Usage

```
cdda.vardist(
  formula,
  pred = NULL,
  mod = NULL,
  data = list(),
  modval = NULL,
  B = 200,
  boot.type = "perc",
  conf.level = 0.95
)

## S3 method for class 'cdda.vardist'
print(x, ...)

## S3 method for class 'cdda.vardist'
plot(x, stat = NULL, ylim = NULL, ...)

## S3 method for class 'cdda.vardist'
summary(object, skew = TRUE, coskew = FALSE, kurt = TRUE, cokurt = FALSE, ...)
```

## Arguments

formula	Symbolic formula of the model to be tested or a lm object
pred	A character indicating the variable name of the predictor which serves as the outcome in the alternative model.
mod	A character indicating the variable name of the moderator.
data	A required data frame containing the variables in the model.
modval	Characters or a numeric sequence specifying the moderator values used in post-hoc probing. Possible characters include c("mean", "median", "JN"). modval = "mean" tests the interaction effect at the moderator values $M - 1SD$ , $M$ , and $M + 1SD$ ; modval = "median" uses $Q1$ , $Md$ , and $Q3$ . The Johnson-Neyman approach is applied when modval = "JN" with conditional effects being evaluated at the boundary values of the significance regions. When a numeric sequence is specified, the pick-a-point approach is used for the selected numeric values.
B	Number of bootstrap samples.
boot.type	A character indicating the type of bootstrap confidence intervals. Must be one of the two values c("perc", "bca"). boot.type = "bca" is the default.
conf.level	Confidence level for bootstrap confidence intervals.
x	An object of class cdda.vardist when using print or plot.

...	Additional arguments to be passed to the function.
stat	A character indicating the statistic to be plotted, default is "rhs", with options c("coskew", "cokurt", "rhs", "rcc", "rtanh").
ylim	A numeric vector of length 2 indicating the y-axis limits. If NULL, the function will set the limits automatically.
object	An object of class cdda.vardist when using summary.
skew	A logical value indicating whether skewness differences and separate D'Agostino skewness tests should be returned when using summary, default is TRUE.
coskew	A logical value indicating whether co-skewness differences should be returned when using summary, default is FALSE.
kurt	A logical value indicating whether excess kurtosis differences and Anscombe-Glynn kurtosis tests should be returned when using summary, default is TRUE.
cokurt	A logical value indicating whether co-kurtosis differences should be returned when using summary, default is FALSE.

### Value

An object of class cdda.vardist containing the results of conditional direction dependence tests of variable distributions.

### References

Wiedermann, W., & von Eye, A. (2025). *Direction Dependence Analysis: Foundations and Statistical Methods*. Cambridge, UK: Cambridge University Press.

### See Also

[dda.vardist](#) for an unconditional version.

### Examples

```
set.seed(321)
n <- 700

## --- generate moderator

z <- sort(rnorm(n))
z1 <- z[z <= 0]
z2 <- z[z > 0]

## --- x -> y when z <= 0

x1 <- rchisq(length(z1), df = 4) - 4
e1 <- rchisq(length(z1), df = 3) - 3
y1 <- 0.5 * x1 + e1

## --- y -> x when m z > 0

y2 <- rchisq(length(z2), df = 4) - 4
```

```
e2 <- rchisq(length(z2), df = 3) - 3
x2 <- 0.25 * y2 + e2

y <- c(y1, y2); x <- c(x1, x2)

d <- data.frame(x, y, z)

m <- lm(y ~ x * z, data = d)

result <- cdda vardist(m, pred = "x", mod = "z", B = 50,
                      modval = c(-1, 1), data = d)

print(result)

plot(result, stat = "rtanh", ylim = c(-0.05, 0.05))

summary(result, skew = FALSE, kurt = FALSE, coskew = TRUE)
```

---

dda.indep

*Direction Dependence Analysis: Independence Properties*

---

## Description

dda.indep evaluates asymmetries of predictor-error independence of causally competing models ( $y \sim x$  vs.  $x \sim y$ ).

print returns DDA test statistics associated with dda.indep objects.

## Usage

```
dda.indep(
  formula,
  pred = NULL,
  data = list(),
  nlfun = NULL,
  hetero = FALSE,
  hsic.method = "gamma",
  diff = FALSE,
  B = 200,
  boot.type = "perc",
  conf.level = 0.95,
  parallelize = FALSE,
  cores = 1
)

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



**Arguments**

formula	Symbolic formula of the model to be tested or a <code>lm</code> object.
pred	A character indicating the variable name of the predictor which serves as the outcome in the alternative model.
data	An optional data frame containing the variables in the model (by default variables are taken from the environment which <code>dda.indep</code> is called from).
nlfun	Either a numeric value or a function of <code>.Primitive</code> type used for non-linear correlation tests. When <code>nlfun</code> is numeric the value is used in a power transformation.
hetero	A logical value indicating whether separate homoscedasticity tests (i.e., standard and robust Breusch-Pagan tests) should be computed.
hsic.method	A character indicating the inference method for the Hilbert-Schmidt Independence Criterion (HSIC). Must be one of the four specifications <code>c("gamma", "eigenvalue", "bootstrap", "permutation")</code> . <code>hsic.method = "gamma"</code> is the default.
diff	A logical value indicating whether differences in HSIC, Distance Correlation (dCor), and MI values should be computed. Bootstrap confidence intervals are computed using <code>B</code> bootstrap samples.
B	Number of permutations for separate dCor tests and number of resamples if <code>hsic.method = c("bootstrap", "permutation")</code> or <code>diff = TRUE</code> .
boot.type	A vector of character strings representing the type of bootstrap confidence intervals. Must be one of the two specifications <code>c("perc", "bca")</code> . <code>boot.type = "perc"</code> is the default.
conf.level	Confidence level for bootstrap confidence intervals.
parallelize	A logical value indicating whether bootstrapping is performed on multiple cores. Only used if <code>diff = TRUE</code> .
cores	A numeric value indicating the number of cores. Only used if <code>parallelize = TRUE</code> .
x	An object of class <code>dda.indep</code> when using <code>print</code> .
...	Additional arguments to be passed to the function.

**Value**

An object of class `dda.indep` containing the results of independence tests of Direction Dependence Analysis.

**References**

Wiedermann, W., & von Eye, A. (2025). *Direction Dependence Analysis: Foundations and Statistical Methods*. Cambridge, UK: Cambridge University Press.

**See Also**

[cdda.indep](#) for a conditional version.

**Examples**

```

set.seed(123)
n <- 500
x <- rchisq(n, df = 4) - 4
e <- rchisq(n, df = 3) - 3
y <- 0.5 * x + e
d <- data.frame(x, y)

result <- dda.indep(y ~ x, pred = "x", data = d, parallelize = TRUE, cores = 2,
  nlfun = 2, B = 50, hetero = TRUE, diff = TRUE)

print(result)

```

---

dda.resdist

*Direction Dependence Analysis: Residual Distributions*


---

**Description**

dda.resdist evaluates patterns of asymmetry of error distributions of causally competing models ( $y \sim x$  vs.  $x \sim y$ ).

print returns DDA test statistics associated with dda.resdist objects.

**Usage**

```

dda.resdist(
  formula,
  pred = NULL,
  data = list(),
  B = 200,
  boot.type = "perc",
  prob.trans = FALSE,
  conf.level = 0.95
)

## S3 method for class 'dda.resdist'
print(x, ...)

```

**Arguments**

formula	Symbolic formula of the target model to be tested or a lm object.
pred	Variable name of the predictor which serves as the outcome in the alternative model.
data	An optional data frame containing the variables in the model (by default variables are taken from the environment which dda.resdist is called from).
B	Number of bootstrap samples.

<code>boot.type</code>	A vector of character strings representing the type of bootstrap confidence intervals required. Must be one of the two values <code>c("perc", "bca")</code> ; <code>boot.type = "perc"</code> is the default.
<code>prob.trans</code>	A logical value indicating whether a probability integral transformation should be performed prior computation of skewness and kurtosis difference tests.
<code>conf.level</code>	Confidence level for bootstrap confidence intervals.
<code>x</code>	An object of class <code>dda.resdist</code> when using <code>print</code> .
<code>...</code>	Additional arguments to be passed to the method.

### Value

An object of class `ddaresdist` containing the results of direction dependence tests of error distributions.

### References

Wiedermann, W., & von Eye, A. (2025). *Direction Dependence Analysis: Foundations and Statistical Methods*. Cambridge, UK: Cambridge University Press.

### Examples

```
set.seed(123)
n <- 500
x <- rchisq(n, df = 4) - 4
e <- rchisq(n, df = 3) - 3
y <- 0.5 * x + e
d <- data.frame(x, y)

result <- dda.resdist(y ~ x, pred = "x", data = d,
  B = 50, conf.level = 0.90, prob.trans = TRUE)

print(result)
```

### Description

`dda.vardist` evaluates patterns of asymmetry of variable distributions for causally competing models ( $y \sim x$  vs.  $x \sim y$ ).

`print` returns DDA test statistics associated with `dda.vardist` objects.

**Usage**

```
dda.vardist(  
  formula,  
  pred = NULL,  
  data = list(),  
  B = 200,  
  boot.type = "perc",  
  conf.level = 0.95  
)  
  
## S3 method for class 'dda.vardist'  
print(x, ...)
```

**Arguments**

formula	Symbolic formula of the model to be tested or a <code>lm</code> object.
pred	Variable name of the predictor which serves as the outcome in the alternative model.
data	An optional data frame containing the variables in the model (by default variables are taken from the environment which <code>dda.vardist</code> is called from).
B	Number of bootstrap samples.
boot.type	A character indicating the type of bootstrap confidence intervals. Must be one of the two specifications <code>c("perc", "bca")</code> . <code>boot.type = "perc"</code> is the default.
conf.level	Confidence level for bootstrap confidence intervals.
x	An object of class <code>dda.vardist</code> when using <code>print</code> .
...	Additional arguments to be passed to the function.

**Value**

An object of class `dda.vardist` containing the results of direction dependence tests of variable distributions.

An object of class `dda.vardist`.

**References**

Wiedermann, W., & von Eye, A. (2025). *Direction Dependence Analysis: Foundations and Statistical Methods*. Cambridge, UK: Cambridge University Press.

**See Also**

[cdda.vardist](#) for a conditional version.

**Examples**

```
set.seed(123)
n <- 500

x <- rchisq(n, df = 4) - 4
e <- rchisq(n, df = 3) - 3
y <- 0.5 * x + e
d <- data.frame(x, y)

result <- dda.vardist(y ~ x, pred = "x", data = d, B = 50)

print(result)
```

# Index

`cdda.indep`, [2](#), [9](#)

`cdda.vardist`, [5](#), [12](#)

`dda.indep`, [4](#), [8](#)

`dda.resdist`, [10](#)

`dda.vardist`, [7](#), [11](#)

`plot.cdda.indep (cdda.indep)`, [2](#)

`plot.cdda.vardist (cdda.vardist)`, [5](#)

`print.cdda.indep (cdda.indep)`, [2](#)

`print.cdda.vardist (cdda.vardist)`, [5](#)

`print.dda.indep (dda.indep)`, [8](#)

`print.dda.resdist (dda.resdist)`, [10](#)

`print.dda.vardist (dda.vardist)`, [11](#)

`summary.cdda.indep (cdda.indep)`, [2](#)

`summary.cdda.vardist (cdda.vardist)`, [5](#)