

# Package ‘Rfuzzydid’

June 22, 2026

**Type** Package

**Title** Fuzzy Difference in Differences

**Description** Differences in differences is a methodology to measure the impact of a treatment by comparing a control group, which remains untreated at two different time periods, with a treatment group, which receives the treatment at the later date. In many cases, this approach is used in situations where the intervention doesn't fully apply the treatment to the entire treatment group but rather increases the treatment rate. In response to such fuzzy scenarios, de Chaisemartin and D'Haultfoeuille (2018) <[doi:10.1093/restud/rdx049](https://doi.org/10.1093/restud/rdx049)> introduce estimands capable of identifying local average and quantile treatment effects under various assumptions. This R package computes the estimators they are proposing.

**Version** 1.1.2

**Author** Kevin Michael Frick [aut, cre]

**Maintainer** Kevin Michael Frick <[kmfrick@proton.me](mailto:kmfrick@proton.me)>

**License** AGPL-3

**Depends** R (>= 3.6.0)

**Encoding** UTF-8

**Imports** generics, knitr, rmarkdown

**Suggests** broom, foreign, haven, modelsummary, pkgload, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**URL** <https://kmfrick.github.io/Rfuzzydid/>,  
<https://github.com/kmfrick/Rfuzzydid>

**BugReports** <https://github.com/kmfrick/Rfuzzydid/issues>

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2026-06-22 14:50:02 UTC

## Contents

coef.fuzzydid . . . . .	2
confint.fuzzydid . . . . .	3
formula.fuzzydid . . . . .	3
fuzzydid . . . . .	4
glance.fuzzydid . . . . .	7
nobs.fuzzydid . . . . .	8
plot.fuzzydid . . . . .	9
print.fuzzydid . . . . .	9
summary.fuzzydid . . . . .	10
tidy.fuzzydid . . . . .	11
vcov.fuzzydid . . . . .	12
<b>Index</b>	<b>14</b>

---

coef.fuzzydid	<i>coef.fuzzydid</i>
---------------	----------------------

---

### Description

Extract fuzzydid point estimates.

### Usage

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

### Arguments

object	A fuzzydid object.
...	Unused.

### Value

A named numeric vector of LATE and LQTE point estimates.

### Examples

```
df <- expand.grid(i = seq_len(20), g = 0:1, t = 0:1)
df$d <- as.integer(df$i <= c(4, 8, 6, 16)[1 + df$t + 2 * df$g])
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$i / 7)

fit <- fuzzydid(df, y ~ d, group = "g", time = "t", did = TRUE, nose = TRUE)
coef(fit)
```

---

confint.fuzzydid	<i>confint.fuzzydid</i>
------------------	-------------------------

---

**Description**

Extract stored percentile bootstrap confidence intervals.

**Usage**

```
## S3 method for class 'fuzzydid'
confint(object, parm, level = 0.95, ...)
```

**Arguments**

object	A fuzzydid object.
parm	Optional parameter subset.
level	Confidence level. Only 0.95 is currently stored.
...	Unused.

**Value**

A matrix with lower and upper confidence limits.

**Examples**

```
df <- expand.grid(i = seq_len(20), g = 0:1, t = 0:1)
df$d <- as.integer(df$i <= c(4, 8, 6, 16)[1 + df$t + 2 * df$g])
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$i / 7)

fit <- fuzzydid(df, y ~ d, group = "g", time = "t", did = TRUE, nose = TRUE)
confint(fit)
```

---

formula.fuzzydid	<i>formula.fuzzydid</i>
------------------	-------------------------

---

**Description**

Extract the formula used to fit a fuzzydid object.

**Usage**

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

**Arguments**

x                    A fuzzydid object.  
 ...                  Unused.

**Value**

The original formula.

**Examples**

```
df <- expand.grid(i = seq_len(20), g = 0:1, t = 0:1)
df$d <- as.integer(df$i <= c(4, 8, 6, 16)[1 + df$t + 2 * df$g])
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$i / 7)

fit <- fuzzydid(df, y ~ d, group = "g", time = "t", did = TRUE, nose = TRUE)
formula(fit)
```

---

fuzzydid	<i>fuzzydid</i>
----------	-----------------

---

**Description**

Formula-first interface for fuzzy difference-in-differences estimators. Estimation is fully native in R. The object returned by `fuzzydid()` is an estimand summary rather than a predictive regression model: it stores local average and local quantile treatment-effect estimates, bootstrap uncertainty summaries, design cell counts, and metadata needed by extractor methods.

**Usage**

```
fuzzydid(
  data,
  formula,
  group,
  time,
  group_forward = NULL,
  did = FALSE,
  tc = FALSE,
  cic = FALSE,
  lqte = FALSE,
  newcateg = NULL,
  numerator = FALSE,
  partial = FALSE,
  nose = FALSE,
  cluster = NULL,
  breps = 50,
  eqtest = FALSE,
  modelx = NULL,
```

```

sieves = FALSE,
sieveorder = NULL,
tagobs = FALSE,
backend = c("auto", "native"),
seed = NULL,
treatment = NULL
)

```

## Arguments

<code>data</code>	A <code>data.frame</code> .
<code>formula</code>	Formula of the form $y \sim d + \text{covariates}$ .
<code>group</code>	Name of the group variable (backward group for multi-period).
<code>time</code>	Name of the time variable.
<code>group_forward</code>	Optional name of the forward group variable for multi-period designs.
<code>did</code>	Logical; compute the Wald-DID estimator.
<code>tc</code>	Logical; compute the Wald-TC estimator.
<code>cic</code>	Logical; compute the Wald-CIC estimator.
<code>lqte</code>	Logical; compute local quantile treatment effects.
<code>newcateg</code>	Optional numeric vector of upper bounds used to recategorize treatment values for TC/CIC.
<code>numerator</code>	Logical; return estimator numerators for DID/TC/CIC.
<code>partial</code>	Logical; request TC partial-identification bounds.
<code>nose</code>	Logical; skip bootstrap standard errors and confidence intervals.
<code>cluster</code>	Optional name of cluster variable for one-way clustered bootstrap resampling.
<code>breps</code>	Integer number of bootstrap replications.
<code>eqtest</code>	Logical; compute equality tests across requested LATE estimands.
<code>modelx</code>	Optional native covariate-adjusted methods ( <code>ols</code> , <code>logit</code> , <code>probit</code> ). Two entries are required for binary treatments and three for ordered multi-valued treatments.
<code>sieves</code>	Logical; use sieve expansion for continuous covariates.
<code>sieveorder</code>	Optional sieve order control for <code>sieves = TRUE</code> . <code>NULL</code> (default) selects order by deterministic 5-fold CV. A scalar value applies to both outcome and treatment sieve bases. A length-2 vector is accepted for backward compatibility and interpreted as <code>(outcome_order, treatment_order)</code> .
<code>tagobs</code>	Logical; return logical mask of observations used.
<code>backend</code>	One of "auto" or "native".
<code>seed</code>	Optional integer seed used for bootstrap resampling when <code>nose = FALSE</code> . If <code>NULL</code> (default), bootstrap draws use the current RNG state. Supply a value to make bootstrap standard errors, confidence intervals, and diagnostics reproducible.
<code>treatment</code>	Optional treatment variable name for multi-term formulas. If <code>NULL</code> , treatment is inferred from formula RHS when unambiguous.

## Details

fuzzydid() uses complete cases across the outcome, treatment, group, time, optional forward-group, covariate, and cluster variables. Missing NA and NaN values are dropped; non-finite numeric values such as Inf and -Inf are rejected. The outcome and treatment must be numeric vectors. Group and time identifiers must be numeric vectors; with one group variable, group values must be in {0, 1, NA}. Covariates may be numeric, factor, character, or logical vectors. Numeric covariates enter as continuous predictors; factor, character, and logical covariates enter as qualitative predictors expanded to indicator columns. When sieves = TRUE, continuous covariates are expanded to polynomial sieve terms.

Standard errors and confidence intervals are percentile bootstrap summaries. Use seed to make bootstrap draws reproducible. If tagobs = TRUE, the returned object includes a logical vector identifying the input rows retained after complete-case filtering.

## Value

An object of class "fuzzydid". This is a list whose late component is a data frame of requested LATE-type estimators with columns estimator, estimate, std.error, conf.low, and conf.high. eqtest is either NULL or an analogous data frame of pairwise equality contrasts, and lqte is either NULL or a data frame with columns quantile, estimate, std.error, conf.low, and conf.high for local quantile treatment effects. Additional components include matrices, a named list of Stata-style result matrices; tagobs, an optional logical mask of retained observations; sample-size diagnostics n, n11, n10, n01, and n00; bootstrap diagnostics n\_reps, n\_misreps, and share\_failures; and metadata such as backend, call, and options. The estimate tables report point estimates and, unless nose = TRUE, bootstrap standard errors and percentile confidence limits.

## Examples

```
make_example_cell <- function(g, t, ones, n_cell = 20L) {
  data.frame(
    g = rep.int(g, n_cell),
    t = rep.int(t, n_cell),
    d = c(rep.int(1L, ones), rep.int(0L, n_cell - ones))
  )
}

df <- rbind(
  make_example_cell(0L, 0L, 4L),
  make_example_cell(0L, 1L, 8L),
  make_example_cell(1L, 0L, 6L),
  make_example_cell(1L, 1L, 16L)
)
df$id <- seq_len(nrow(df))
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$id / 7)
example_data <- df

fit <- fuzzydid(
  data = example_data,
  formula = y ~ d,
  treatment = NULL,
  group = "g",
```

```

time = "t",
group_forward = NULL,
did = TRUE,
tc = TRUE,
cic = TRUE,
lqte = TRUE,
newcateg = c(0, 1),
cluster = NULL,
modelx = NULL,
sieveorder = NULL,
seed = NULL,
nose = TRUE
)

fit$late

```

glance.fuzzydid

*glance.fuzzydid***Description**

One-row summary for fuzzydid objects.

**Usage**

```

## S3 method for class 'fuzzydid'
glance(x, ...)

```

**Arguments**

x	A fuzzydid object.
...	Unused.

**Value**

A one-row data frame with class "data.frame" summarizing the fitted object. backend reports the computation path and Num.Obs. reports the estimation sample size. N.11, N.10, N.01, and N.00 give the four design cell counts. N.reps, N.misreps, and Share.failures describe bootstrap replication totals and failure rates, or are NA when nose = TRUE.

**Examples**

```

make_example_cell <- function(g, t, ones, n_cell = 20L) {
  data.frame(
    g = rep.int(g, n_cell),
    t = rep.int(t, n_cell),
    d = c(rep.int(1L, ones), rep.int(0L, n_cell - ones))
  )
}

```

```

df <- rbind(
  make_example_cell(0L, 0L, 4L),
  make_example_cell(0L, 1L, 8L),
  make_example_cell(1L, 0L, 6L),
  make_example_cell(1L, 1L, 16L)
)
df$id <- seq_len(nrow(df))
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$id / 7)

fit <- fuzzydid(
  data = df[, c("y", "g", "t", "d")],
  formula = y ~ d,
  group = "g",
  time = "t",
  did = TRUE,
  nose = TRUE
)

generics::glance(fit)

```

---

nobs.fuzzydid

*nobs.fuzzydid*


---

## Description

Extract the estimation sample size.

## Usage

```
## S3 method for class 'fuzzydid'
nobs(object, ...)
```

## Arguments

object	A fuzzydid object.
...	Unused.

## Value

Integer number of observations used for estimation.

## Examples

```

df <- expand.grid(i = seq_len(20), g = 0:1, t = 0:1)
df$d <- as.integer(df$i <= c(4, 8, 6, 16)[1 + df$t + 2 * df$g])
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$i / 7)

fit <- fuzzydid(df, y ~ d, group = "g", time = "t", did = TRUE, nose = TRUE)
nobs(fit)

```

---

```
plot.fuzzydid      plot.fuzzydid
```

---

**Description**

Plot fuzzydid point estimates and stored confidence intervals as a base R dot-and-whisker plot.

**Usage**

```
## S3 method for class 'fuzzydid'
plot(x, ...)
```

**Arguments**

```
x          A fuzzydid object.
...        Unused.
```

**Value**

The input x, returned invisibly.

**Examples**

```
df <- expand.grid(i = seq_len(20), g = 0:1, t = 0:1)
df$d <- as.integer(df$i <= c(4, 8, 6, 16)[1 + df$t + 2 * df$g])
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$i / 7)

fit <- fuzzydid(df, y ~ d, group = "g", time = "t", did = TRUE,
               breps = 5, seed = 1)

plot(fit)
```

---

```
print.fuzzydid     print.fuzzydid
```

---

**Description**

Print a compact fuzzydid object header and estimator table.

**Usage**

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

**Arguments**

```
x          A fuzzydid object.
...        Unused.
```

**Value**

The input x, returned invisibly.

**Examples**

```
df <- expand.grid(i = seq_len(20), g = 0:1, t = 0:1)
df$d <- as.integer(df$i <= c(4, 8, 6, 16)[1 + df$t + 2 * df$g])
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$i / 7)

fit <- fuzzydid(df, y ~ d, group = "g", time = "t", did = TRUE, nose = TRUE)
print(fit)
```

---

summary.fuzzydid	<i>summary.fuzzydid</i>
------------------	-------------------------

---

**Description**

Print a compact summary table for fuzzydid results.

**Usage**

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

**Arguments**

object	A fuzzydid object.
...	Unused.

**Value**

The input object, returned invisibly with class "fuzzydid", after printing the available estimator tables. This method is called for its side effect of displaying the late, eqtest, and lqte components in a compact tabular form.

**Examples**

```
make_example_cell <- function(g, t, ones, n_cell = 20L) {
  data.frame(
    g = rep.int(g, n_cell),
    t = rep.int(t, n_cell),
    d = c(rep.int(1L, ones), rep.int(0L, n_cell - ones))
  )
}

df <- rbind(
  make_example_cell(0L, 0L, 4L),
  make_example_cell(0L, 1L, 8L),
```

```

  make_example_cell(1L, 0L, 6L),
  make_example_cell(1L, 1L, 16L)
)
df$id <- seq_len(nrow(df))
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$id / 7)

fit <- fuzzydid(
  data = df[, c("y", "g", "t", "d")],
  formula = y ~ d,
  group = "g",
  time = "t",
  did = TRUE,
  nose = TRUE
)

summary(fit)

```

---

tidy.fuzzydid

*tidy.fuzzydid*


---

## Description

Tidy extractor for fuzzydid objects.

## Usage

```

## S3 method for class 'fuzzydid'
tidy(x, ...)

```

## Arguments

x	A fuzzydid object.
...	Unused.

## Value

A data frame with class "data.frame" and one row per available estimate or contrast. component identifies whether the row comes from the LATE table, equality-test table, or LQTE table. model and term label the estimator or contrast. estimate is the point estimate, while std.error, conf.low, and conf.high contain bootstrap uncertainty summaries when available and NA otherwise. If no estimates are available, an empty data frame with the same columns is returned.

## Examples

```

make_example_cell <- function(g, t, ones, n_cell = 20L) {
  data.frame(
    g = rep.int(g, n_cell),
    t = rep.int(t, n_cell),
    d = c(rep.int(1L, ones), rep.int(0L, n_cell - ones))
  )
}

```

```

  )
}

df <- rbind(
  make_example_cell(0L, 0L, 4L),
  make_example_cell(0L, 1L, 8L),
  make_example_cell(1L, 0L, 6L),
  make_example_cell(1L, 1L, 16L)
)
df$id <- seq_len(nrow(df))
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$id / 7)

fit <- fuzzydid(
  data = df[, c("y", "g", "t", "d")],
  formula = y ~ d,
  group = "g",
  time = "t",
  did = TRUE,
  nose = TRUE
)

generics::tidy(fit)

```

---

vcov.fuzzydid

*vcov.fuzzydid*


---

## Description

Extract the bootstrap covariance matrix for stored estimates.

## Usage

```
## S3 method for class 'fuzzydid'
vcov(object, ...)
```

## Arguments

object	A fuzzydid object.
...	Unused.

## Value

A covariance matrix for `coef(object)`.

**Examples**

```
df <- expand.grid(i = seq_len(20), g = 0:1, t = 0:1)
df$d <- as.integer(df$i <= c(4, 8, 6, 16)[1 + df$t + 2 * df$g])
df$y <- 1 + 0.5 * df$g + 0.4 * df$t + 2 * df$d + sin(df$i / 7)

fit <- fuzzydid(df, y ~ d, group = "g", time = "t", did = TRUE,
               breps = 5, seed = 1)

vcov(fit)
```

# Index

`coef.fuzzydid`, [2](#)  
`confint.fuzzydid`, [3](#)  
  
`formula.fuzzydid`, [3](#)  
`fuzzydid`, [4](#)  
  
`glance.fuzzydid`, [7](#)  
  
`nobs.fuzzydid`, [8](#)  
  
`plot.fuzzydid`, [9](#)  
`print.fuzzydid`, [9](#)  
  
`summary.fuzzydid`, [10](#)  
  
`tidy.fuzzydid`, [11](#)  
  
`vcov.fuzzydid`, [12](#)