

Package ‘causaljudgment’

October 6, 2026

Title Computational Models of Causal Judgment

Version 0.1.0

Date 2026-09-08

Description

Provides computational implementations of models of causal judgment, including the Counterfactual Effect Size (CES) model of Quillien and Lucas (2023) <doi:10.1037/rev0000428> and the Necessity-Sufficiency (NS) model of Icard, Komin-sky and Knobe (2017) <doi:10.1016/j.cognition.2017.01.010>.

The package represents causal structures as binary Structural Causal Models and analytically computes causal judgments from counterfactual probability distributions.

License MIT + file LICENSE

Encoding UTF-8

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

Imports dplyr

VignetteBuilder knitr

BugReports <https://github.com/tadeguillien/causaljudgment/issues>

URL <https://tadeguillien.github.io/causaljudgment/>

NeedsCompilation no

Author Tadeq Quillien [aut, cre]

Maintainer Tadeq Quillien <tadeg.quillien@gmail.com>

Repository CRAN

Date/Publication 2026-10-06 07:50:02 UTC

Contents

ces	2
compute_counterfactual_value	3

compute_judgment	3
compute_necessity	4
compute_probabilities	5
compute_sufficiency	6
create_structural_function	6
make_function_list	7
ns	7
verif	8
Index	9

ces	<i>ces(): compute a judgment with the CES model.</i>
-----	--

Description

This function computes a CES judgment on the basis of a probability distribution over counterfactual worlds.

Usage

```
ces(var1, var2, aw_values, d, p_col = "p")
```

Arguments

- var1 Character string giving the candidate cause variable.
- var2 Character string giving the outcome variable.
- aw_values A named list specifying the values of variables in the actual world.
- d The dataframe containing the probability distribution over counterfactual worlds.
- p_col Indicates which column contains the joint probability.

Value

A numeric causal judgment.

compute_counterfactual_value

compute_counterfactual_value(): compute the value of Y conditioned on a counterfactual intervention on X, starting from a given world.

Description

This function computes the value of Y conditioned on a counterfactual intervention on X, starting from a given world. The function is used for computing both necessity and sufficiency.

Usage

```
compute_counterfactual_value(
  intervention_var,
  intervention_value,
  target_var,
  causal_model,
  aw_values
)
```

Arguments

intervention_var	Character string: the variable we intervene upon
intervention_value	Numeric: the post-intervention value of the variable we intervene upon
target_var	Character string: the target variable. We want to compute the value it has after we've intervened on the intervention variable
causal_model	A named list specifying the causal model
aw_values	A named list with the values of the variables in the actual world

Value

A numeric specifying the counterfactual value of the target variable.

compute_judgment

compute_judgment(): compute a causal judgment.

Description

The general causal judgment function. It is essentially a wrapper over the ces() and ns() functions.

Usage

```
compute_judgment(var, outcome, causal_model, actual_world, model, s = 0)
```

Arguments

<code>var</code>	Character string giving the candidate cause variable.
<code>outcome</code>	Character string giving the outcome variable.
<code>causal_model</code>	A named list specifying the causal model.
<code>actual_world</code>	A named list specifying the values of variables in the actual world. The names must match those in <code>causal_model</code> .
<code>model</code>	Character string specifying the causal judgment model. Currently, "ces" and "ns" are supported.
<code>s</code>	Numeric or Named List parameter(s) controlling the adjustment of exogenous variable probabilities toward their actual-world values. Defaults to 0. Usually this is a scalar that applies to all variables in the model, but one can also use a named list that specifies a separate parameter for each variable.

Value

A numeric causal judgment.

Examples

```
# Define a causal model
causal_model <- list(e = "a & b", a = .1, b = .9)

# Define the actual world
actual_world <- list(e = 1, a = 1, b = 1)

# Compute the CES judgment for A causing E
compute_judgment(
  var = "a",
  outcome = "e",
  causal_model = causal_model,
  actual_world = actual_world,
  model = "ces",
  s = .7
)
```

<code>compute_necessity</code>	<i>compute_necessity(): compute whether X was necessary for Y in the actual world.</i>
--------------------------------	--

Description

We check whether intervening on X in the actual world flips the value of Y.

Usage

```
compute_necessity(x_var, y_var, causal_model, actual_world)
```

Arguments

x_var	Character string giving the candidate cause variable.
y_var	Character string giving the outcome variable.
causal_model	A named list of structural functions defining the causal model.
actual_world	A named list giving the values of variables in the actual world.

Value

A logical value indicating whether the candidate cause is necessary for the outcome in the actual world.

compute_probabilities *compute_probability(): compute the probability distribution over counterfactual worlds induced by the causal model and the state of the actual world.*

Description

We compute the distribution using the factorization of the causal model, by computing the marginal probability of exogenous variables and the conditional probabilities of the endogenous variables. Then we take the product of these probabilities to compute the joint distribution.

Usage

```
compute_probabilities(structural_functions, actual_world, s = 0)
```

Arguments

structural_functions	A named list of functions defining the structural equations and probability distributions of the variables in the causal model.
actual_world	A named list giving the values of variables in the actual world.
s	Numeric or Named List parameter(s) controlling the adjustment of exogenous variable probabilities toward their actual-world values. Defaults to 0. Usually this is a scalar that applies to all variables in the model, but one can also use a named list that specifies a separate parameter for each variable.

Value

A data frame containing one row for each possible world, probability columns for each variable, and a column p giving the probability of each world.

compute_sufficiency	<i>compute_sufficiency()</i> : computes the sufficiency of a candidate cause for an outcome.
---------------------	--

Description

This function computes the sufficiency of a candidate cause for an outcome.

Usage

```
compute_sufficiency(var, outcome, actual_world, causal_model, d)
```

Arguments

var	Character string giving the candidate cause variable.
outcome	Character string giving the outcome variable.
actual_world	A named list giving the values of variables in the actual world.
causal_model	A named list of structural functions defining the causal model.
d	A data frame containing the joint probability distribution over possible worlds.

Value

A numeric value representing the sufficiency of the candidate cause for the outcome.

create_structural_function	<i>Parse a structural function (or exogenous probability)</i>
----------------------------	---

Description

Parses a string representing a structural equation or probability and converts it into an R function. Variable names appearing in the equation become arguments to the resulting function. Numeric inputs are interpreted as exogenous probabilities.

Usage

```
create_structural_function(equation_string)
```

Arguments

equation_string	A string representing a structural equation, or a numeric value representing an exogenous probability.
-----------------	--

Details

For example, "a & b" is converted into a function equivalent to `function(a, b) a & b`.

Value

An R function implementing the structural equation or, for an exogenous variable, a function returning its probability.

make_function_list	<i>Create structural functions for a causal model</i>
--------------------	---

Description

Converts a named causal model into a named list of R functions. Each variable in the causal model is converted using `create_structural_function()`.

Usage

```
make_function_list(vars)
```

Arguments

vars	A named list specifying the causal model. Elements representing structural equations should be strings, while exogenous variables are specified by their probabilities.
------	---

Value

A named list of functions corresponding to the variables in the causal model.

ns	<i>ns(): compute a judgment with the NS model.</i>
----	--

Description

This function computes a NS judgment on the basis of a probability distribution over counterfactual worlds

Usage

```
ns(var, outcome, actual_world, d, causal_model)
```

Arguments

var	Character string giving the candidate cause variable.
outcome	Character string giving the outcome variable.
actual_world	A named list specifying the values of variables in the actual world.
d	The dataframe containing the probability distribution over counterfactual worlds.
causal_model	A named list specifying the causal model.

Value

A numeric causal judgment.

verif	<i>verif()</i> : <i>verify consistency of an endogenous variable</i>
-------	--

Description

Checks whether the value of an endogenous variable is consistent with the values of its parent variables under its structural function.

Usage

```
verif(outcome, args, fun)
```

Arguments

outcome	Numeric value of the endogenous variable.
args	Values of the parent variables, supplied as a list or list-like object suitable for passing to fun.
fun	A structural function defining the value of the endogenous variable as a function of its parents.

Value

A logical value indicating whether the structural function produces the specified value of outcome.

Index

ces, [2](#)
compute_counterfactual_value, [3](#)
compute_judgment, [3](#)
compute_necessity, [4](#)
compute_probabilities, [5](#)
compute_sufficiency, [6](#)
create_structural_function, [6](#)
create_structural_function(), [7](#)

make_function_list, [7](#)

ns, [7](#)

verif, [8](#)