

# Package: causens (via r-universe)

June 1, 2026

**Type** Package

**Title** Perform Causal Sensitivity Analyses Using Various Statistical Methods

**Version** 0.0.3

**Description** While data from randomized experiments remain the gold standard for causal inference, estimation of causal estimands from observational data is possible through various confounding adjustment methods. However, the challenge of unmeasured confounding remains a concern in causal inference, where failure to account for unmeasured confounders can lead to biased estimates of causal estimands. Sensitivity analysis within the framework of causal inference can help adjust for possible unmeasured confounding. In `causens`, three main methods are implemented: adjustment via sensitivity functions (Brumback, Hernán, Haneuse, and Robins (2004) [doi:10.1002/sim.1657](https://doi.org/10.1002/sim.1657)) and Li, Shen, Wu, and Li (2011) [doi:10.1093/aje/kwr096](https://doi.org/10.1093/aje/kwr096)), Bayesian parametric modelling and Monte Carlo approaches (McCandless, Lawrence C and Gustafson, Paul (2017) [doi:10.1002/sim.7298](https://doi.org/10.1002/sim.7298)).

**License** MIT + file LICENSE

**URL** <https://kuan-liu-lab.github.io/causens/>,  
<https://github.com/Kuan-Liu-Lab/causens>

**Suggests** devtools, knitr, pkgdown, PSweight, rjags, rmarkdown, roxygen2, testthat (>= 3.0.0), usethis, waldo

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**BugReports** <https://github.com/Kuan-Liu-Lab/causens/issues>

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Repository** <https://kuan-liu-lab.r-universe.dev>

**Date/Publication** 2025-06-03 21:54:12 UTC

**RemoteUrl** <https://github.com/kuan-liu-lab/causens>

**RemoteRef** HEAD

**RemoteSha** a5c37d7fc5d8ac1e5dc8be9c14a41f49a55ae5f2

## Contents

bayesian_causens . . . . .	2
causens_monte_carlo . . . . .	3
causens_sf . . . . .	4
create_jags_model . . . . .	4
gData_U_binary_Y_binary . . . . .	5
gData_U_binary_Y_cont . . . . .	6
gData_U_cont_Y_binary . . . . .	7
gData_U_cont_Y_cont . . . . .	7
plot_causens . . . . .	8
process_model_formula . . . . .	9
sf . . . . .	10
simulate_data . . . . .	10
summary.bayesian_causens . . . . .	11
summary.causens_sf . . . . .	12
summary.monte_carlo_causens . . . . .	12

**Index** **14**

---

bayesian_causens	<i>Bayesian parametric sensitivity analysis for causal inference</i>
------------------	--

---

## Description

This function runs a Bayesian sensitivity analysis for causal inference using JAGS or Stan as a backend. For now, only JAGS is supported.

## Usage

```
bayesian_causens(
  trt_model,
  outcome_model,
  U_model,
  data,
  beta_uy = ~dunif(-2, 2),
  alpha_uz = ~dunif(-2, 2),
  backend = "jags",
  output_trace = FALSE,
  ...
)
```

**Arguments**

<code>trt_model</code>	The treatment model object as a formula.
<code>outcome_model</code>	The outcome model object as a formula.
<code>U_model</code>	The unmeasured confounder model object as a formula.
<code>data</code>	A data frame containing the exposure, outcome, and confounder variables.
<code>beta_uy</code>	Prior distribution for the effect of the missing confounder U on the outcome Y.
<code>alpha_uz</code>	Prior distribution for the effect of the missing confounder U on the treatment assignment mechanism Z.
<code>backend</code>	The backend to use for the sensitivity analysis. Currently only "jags" is supported.
<code>output_trace</code>	Whether to output the full trace of the MCMC sampler.
<code>...</code>	Additional arguments to be passed to the backend.

**Value**

A list of posterior samples for the causal effect of the exposure variable on the outcome, as well as the confounder-adjusted causal effect.

---

`causens_monte_carlo` *Monte Carlo sensitivity analysis for causal effects*

---

**Description**

This function performs a Monte Carlo sensitivity analysis for causal effects.

**Usage**

```
causens_monte_carlo(outcome, exposure, confounders, data, ...)
```

**Arguments**

<code>outcome</code>	The name of the outcome variable in the data frame.
<code>exposure</code>	The name of the exposure variable in the data frame.
<code>confounders</code>	The name of the confounders in the data frame.
<code>data</code>	A data frame containing the exposure, outcome, and confounder variables.
<code>...</code>	Additional arguments to be passed to the function.

**Value**

The estimated causal effect.

---

causens\_sf

*Bayesian Estimation of ATE Subject to Unmeasured Confounding*


---

### Description

This function provides an estimate of the Average Treatment Effect (ATE) using Bayesian modelling.

### Usage

```
causens_sf(
  trt_model,
  outcome,
  data,
  bootstrap = FALSE,
  B = 1000,
  seed = 123,
  ...
)
```

### Arguments

trt_model	A model formula specifying the treatment model.
outcome	The name of the outcome variable.
data	A data frame containing the exposure, outcome, and confounder variables.
bootstrap	A logical indicating whether to perform bootstrap estimation of the 95% confidence interval.
B	If the bootstrap argument is TRUE, the number of bootstrap samples to generate.
seed	An integer to set the random seed for reproducibility.
...	Additional arguments to be passed to the sensitivity function.

### Value

A point estimate of the corrected ATE.

---

create\_jags\_model

*Create an JAGS model for Bayesian sensitivity analysis*


---

### Description

Creates a JAGS model available as a string, or .txt file, where priors are initialized to be uninformative by default.

**Usage**

```
create_jags_model(binary_outcome, beta_uy, alpha_uz)
```

**Arguments**

binary\_outcome Boolean indicating whether the outcome is binary.

beta\_uy Prior distribution for the effect of the missing confounder U on the outcome Y.

alpha\_uz Prior distribution for the effect of the missing confounder U on the treatment assignment mechanism Z.

No inputs are given to this function (for now) since data-related information is provided in `jags.model()` during model initialization.

---

```
gData_U_binary_Y_binary
```

*Generate data with a binary unmeasured confounder and binary outcome*

---

**Description**

Generate data with a binary unmeasured confounder and binary outcome

**Usage**

```
gData_U_binary_Y_binary(
  ymodel = "linear",
  N = 500,
  alpha_uz = 0.2,
  beta_uy = 0.5,
  treatment_effects = 1,
  seed = 123
)
```

**Arguments**

ymodel A string indicating the functional form of the outcome model.

N The number of observations to be generated.

alpha\_uz Unmeasured confounder coefficient in the propensity score model.

beta\_uy Unmeasured confounder coefficient in the outcome model.

treatment\_effects The treatment effect.

seed The seed for the random number generator.

**Value**

A data frame with the simulated dataset with U binary, Y binary.

---

gData\_U\_binary\_Y\_cont *Generate data with a binary unmeasured confounder and continuous outcome*

---

### Description

Generate data with a binary unmeasured confounder and continuous outcome

### Usage

```
gData_U_binary_Y_cont(  
  ymodel = "linear",  
  N = 500,  
  alpha_uz = 0.2,  
  beta_uy = 0.5,  
  treatment_effects = 1,  
  seed = 123  
)
```

### Arguments

ymodel	A string indicating the functional form of the outcome model.
N	The number of observations to be generated.
alpha_uz	Unmeasured confounder coefficient in the propensity score model.
beta_uy	Unmeasured confounder coefficient in the outcome model.
treatment_effects	The treatment effect.
seed	The seed for the random number generator.

### Value

A data frame with the simulated dataset with U binary, Y continuous.

### Examples

```
fulldata <- gData_U_binary_Y_cont(  
  ymodel = "linear",  
  N = 500,  
  alpha_uz = 0.2,  
  beta_uy = 0.5,  
  treatment_effects = 1,  
  seed = 123  
)  
table(fulldata$Z)
```

---

`gData_U_cont_Y_binary` *Generate data with a continuous unmeasured confounder and a binary outcome*

---

### Description

Generate data with a continuous unmeasured confounder and a binary outcome

### Usage

```
gData_U_cont_Y_binary(  
  ymodel = "linear",  
  N = 500,  
  alpha_uz = 0.2,  
  beta_uy = 0.5,  
  treatment_effects = 1,  
  seed = 123  
)
```

### Arguments

<code>ymodel</code>	A string indicating the functional form of the outcome model.
<code>N</code>	The number of observations to be generated.
<code>alpha_uz</code>	Unmeasured confounder coefficient in the propensity score model.
<code>beta_uy</code>	Unmeasured confounder coefficient in the outcome model.
<code>treatment_effects</code>	The treatment effect.
<code>seed</code>	The seed for the random number generator.

### Value

A data frame with the simulated dataset with U continuous, Y binary.

---

`gData_U_cont_Y_cont` *Generate data with a continuous unmeasured confounder and continuous outcome*

---

### Description

Generate data with a continuous unmeasured confounder and continuous outcome

**Usage**

```
gData_U_cont_Y_cont(
  ymodel = "linear",
  N = 500,
  alpha_uz = 0.2,
  beta_uy = 0.5,
  treatment_effects = 1,
  seed = 123
)
```

**Arguments**

ymodel	A string indicating the functional form of the outcome model.
N	The number of observations to be generated.
alpha_uz	Unmeasured confounder coefficient in the propensity score model.
beta_uy	Unmeasured confounder coefficient in the outcome model.
treatment_effects	The treatment effect.
seed	The seed for the random number generator.

**Value**

A data frame with the simulated dataset with U continuous, Y continuous.

---

plot_causens	<i>Plot ATE with respect to sensitivity function value when it is constant, i.e. <math>c(1, e) = c1</math> and <math>c(0, e) = c0</math>.</i>
--------------	---

---

**Description**

This function plots 1) the ATE as a function of the sensitivity function value when it is constant and 2) its associate 95 interval obtained via bootstrapping, if desired. The latter process can be take a few seconds to minutes.

**Usage**

```
plot_causens(
  trt_model,
  data,
  outcome,
  c1_upper = 0.5,
  c1_lower = 0,
  r = 1,
  by = 0.01
)
```

**Arguments**

trt_model	The treatment model object as a formula or fitted glm.
data	A data frame containing the variables of interest.
outcome	The name of the outcome variable.
c1_upper	The upper bound for the sensitivity function value.
c1_lower	The lower bound for the sensitivity function value.
r	The ratio between c1 and c0.
by	The increment for the sensitivity function value.

**Value**

A plot of the ATE as a function of c1 values.

---

process\_model\_formula *Process model formula*

---

**Description**

This helper function takes in a model in its ‘formula’ format and returns a list comprising of various information, including the fitted model as an R object.

**Usage**

```
process_model_formula(model, data)
```

**Arguments**

model	The model object as a formula or fitted glm.
data	A data frame used for fitting the model.

**Value**

A list containing the fitted model, model formula, response variable name, and confounder names.

---

sf	<i>Calculate sensitivity of treatment effect estimate to unmeasured confounding</i>
----	---

---

### Description

This function calculates the sensitivity of a treatment effect estimate to unmeasured confounding, as described in Rosenbaum (2002). The sensitivity is defined as the maximum strength of association between the unmeasured confounder and the treatment assignment that would be needed to explain away the observed treatment effect estimate. This function assumes that the treatment assignment is binary and that the outcome is continuous.

### Usage

```
sf(z, e, form = "constant", c1 = 0, c0 = 0, s1 = 0, s0 = 0)
```

### Arguments

z	Treatment assignment (binary: 0 or 1)
e	Propensity score value (numeric)
form	Form of the sensitivity function (character: "constant" or "linear")
c1	Value of the sensitivity function when $z = 1$ (numeric)
c0	Value of the sensitivity function when $z = 0$ (numeric)
s1	Slope of the sensitivity function when $z = 1$ (numeric)
s0	Slope of the sensitivity function when $z = 0$ (numeric)

### Value

Sensitivity of treatment effect estimate to unmeasured confounding (numeric)  
sf(z, e, form)

---

simulate_data	<i>Generate data with unmeasured confounder</i>
---------------	---

---

### Description

Generate data with unmeasured confounder

**Usage**

```
simulate_data(
  ymodel = "linear",
  N = 500,
  u_type = "binary",
  y_type = "continuous",
  seed = 123,
  alpha_uz = 0.2,
  beta_uy = 0.5,
  treatment_effects = 1,
  informative_u = FALSE
)
```

**Arguments**

ymodel	A string indicating the functional form of the outcome model.
N	The number of observations to be generated.
u_type	A string indicating the type of the unmeasured confounder: "binary" or "continuous".
y_type	A string indicating the type of the outcome: "binary" or "continuous".
seed	The seed for the random number generator.
alpha_uz	Unmeasured confounder coefficient in the propensity score model.
beta_uy	Unmeasured confounder coefficient in the outcome model.
treatment_effects	The treatment effect.
informative_u	A boolean indicating whether the unmeasured confounder is driven by covariates.

**Value**

A data frame with the simulated dataset.

---

```
summary.bayesian_causens
```

*Summarize the results of a causal sensitivity analysis via Bayesian modelling of an unmeasured confounder.*

---

**Description**

Summarize the results of a causal sensitivity analysis via Bayesian modelling of an unmeasured confounder.

**Usage**

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

**Arguments**

object            An object of class bayesian\_causens.  
 ...              Additional arguments to be passed to summary.

**Value**

A summary of results from a Bayesian causal sensitivity analysis.

---

summary.causens\_sf    *Summarize the results of a causal sensitivity analysis via sensitivity function.*

---

**Description**

Summarize the results of a causal sensitivity analysis via sensitivity function.

**Usage**

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

**Arguments**

object            An object of class causens\_sf.  
 ...              Additional arguments to be passed to summary.

**Value**

A summary of the results of the causal sensitivity analysis.

---

summary.monte\_carlo\_causens  
                           *Summarize the results of a causal sensitivity analysis via the Monte Carlo method.*

---

**Description**

Summarize the results of a causal sensitivity analysis via the Monte Carlo method.

**Usage**

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

**Arguments**

object	An object of class <code>causens_monte_carlo</code> .
...	Additional arguments to be passed to <code>summary</code> .

**Value**

A summary of results from a Monte Carlo causal sensitivity analysis.

# Index

bayesian\_causens, 2

causens\_monte\_carlo, 3  
causens\_sf, 4  
create\_jags\_model, 4

gData\_U\_binary\_Y\_binary, 5  
gData\_U\_binary\_Y\_cont, 6  
gData\_U\_cont\_Y\_binary, 7  
gData\_U\_cont\_Y\_cont, 7

plot\_causens, 8  
process\_model\_formula, 9

sf, 10  
simulate\_data, 10  
summary.bayesian\_causens, 11  
summary.causens\_sf, 12  
summary.monte\_carlo\_causens, 12