Package: causalMed (via r-universe)

November 7, 2026

Title Parametric G-Formula for Longitudinal Mediation and Total Effects

Version 0.1.3

Description Causal mediation analysis for longitudinal data with time-varying exposures, mediators and confounders, covering both interventional (<doi:10.1002/sim.7426>) and natural (<doi:10.1515/jci-2016-0006>) direct and indirect effects. Estimation uses the parametric g-formula (<doi:10.1016/0270-0255(86)90088-6>), which is also exposed directly so that counterfactual means or risks can be estimated under static, dynamic or time-varying hypothetical interventions.

Depends R (>= 3.5.1)

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports data.table, nnet, future, future.apply, progressr

Suggests gfoRmula, Hmisc, knitr, rmarkdown, survival, testthat

URL https://github.com/adayim/causalMed, https://adayim.github.io/causalMed/

BugReports https://github.com/adayim/causalMed/issues

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

Repository https://adayim.r-universe.dev

Date/Publication 2026-10-13 12:21:39 UTC

RemoteUrl https://github.com/adayim/causalmed

RemoteRef HEAD

RemoteSha 1b7a9112e313ec60f9cf69d93a112ee06e80ad04

Contents

dyn_int 2
gformula 3
gvhd 6
mediation 7
nonsurvivaldata 13
print.gformula 15
recodes 16
spec_model 17
summary.gformula 19
survivaldata 19
yamamurodata 21

Index **23**

`dyn_int` *Define a dynamic intervention rule*

Description

Captures an R expression for a dynamic (rule-based) exposure intervention. The expression is evaluated at each time step inside the simulated `data.table`, so any column in the current Monte Carlo dataset can be referenced by name (including the exposure itself, which holds its natural-course draw at the time of evaluation).

Usage

```
dyn_int(expr)
```

Arguments

`expr`	An R expression that returns a numeric (or logical coerced to numeric) vector with one value per row. Column names from the current simulated dataset are in scope (e.g. `L1`, `A`).

Value

An object of class `"causalMed_dynint"`.

Examples

```
# Treat only if the natural-course value of A exceeds 0:
dyn_int(as.numeric(A > 0))

# Treat if L1 > 0 or L2 equals 1:
dyn_int(as.numeric(L1 > 0 | L2 == 1))
```

`gformula` *Parametric g-formula for time-varying interventions*

Description

Estimates the mean outcome, or the risk for a survival outcome, that would be seen if everyone followed each of several exposure strategies, and the contrasts between them (the total effect), using the parametric g-formula (Robins 1986). The models in `models` are fitted to the observed data and then used to simulate each subject forward in time under every intervention.

Usage

```
gformula(
  data,
  id_var,
  base_vars,
  exposure,
  time_var,
  models,
  intervention = NULL,
  ref_int = 0,
  init_recode = NULL,
  in_recode = NULL,
  out_recode = NULL,
  return_fitted = FALSE,
  mc_sample = NULL,
  return_data = FALSE,
  R = 500,
  quiet = FALSE,
  seed = 12345
)
```

Arguments

`data`	A `data.frame` in long format.
`id_var`	Character. Name of the subject identifier.
`base_vars`	Character vector of time-fixed baseline covariates (may be empty), with no missing values. Only these and `id_var` are carried into the simulated cohort; every other variable a model uses, including lags, must be created by the recode hooks.
`exposure`	Character. Name of the exposure to intervene on.
`time_var`	Character. Name of the numeric time variable. Each distinct value is one simulated step, in ascending order.
`models`	List of `spec_model` objects, in the order the variables are generated.

intervention	Named list of interventions. Each element is NULL (the natural course: exposure drawn from its fitted model), a 0/1 value or vector with one value per distinct time point (a static intervention), or a dyn_int rule, e.g. list(natural = NULL, treat_if_high = dyn_int(as.numeric(L1 > 0))). NULL (the default) runs the natural course only.
ref_int	Reference for the contrasts: 0 or "natural" (default) for the natural course, or the position or name of an element of intervention. See Details.
init_recode	recodes applied at the first time step, before the models are evaluated, e.g. to set lags to their baseline value.
in_recode	recodes applied at the start of each later time step, before the models are evaluated, e.g. to update lags.
out_recode	recodes applied at the end of each time step, the first included, e.g. to advance cumulative counts or carry an absorbing state forward.
return_fitted	Logical. Return the full fitted model objects (default FALSE: calls and coefficients only).
mc_sample	Number of subjects in the simulated Monte Carlo cohort. The default, NULL, uses 50 times the number of subjects in data and reports the value unless quiet = TRUE. A larger value reduces Monte Carlo error, which falls as 1/sqrt(mc_sample); it does not change the estimand.
return_data	Logical. Return the simulated data (default FALSE; can be large).
R	Number of bootstrap replicates (default 500); R = 1 skips the bootstrap. Replicates run through future.apply::future_lapply: sequentially unless a parallel plan is set with future::plan(), e.g. plan(multisession).
quiet	Logical. Suppress progress messages (default FALSE).
seed	Integer random seed (default 12345) for the Monte Carlo simulation and the bootstrap replicates; the global RNG state is restored on exit. NULL disables seeding, so repeated calls differ.

Details

Data. Long format: one row per subject per time point. For a survival outcome the data must be in risk-set form, with every row after the event and after loss to follow-up removed. A "censor" model may be declared; it is simulated in the natural course but does not change any reported risk, each of which is the risk under eliminated loss to follow-up.

Models. models is a list of spec_model objects in the order the variables are generated within a time point. The "outcome" or "survival" model gives the predicted outcome (Pred_Y) under each intervention: the fitted mean or risk of an end-of-follow-up outcome, or the cumulative risk built from the fitted hazards.

Interventions. intervention is a named list, for example list(natural = NULL, always = 1, never = 0). ref_int = 0 or "natural" (the default) compares against the natural course: the NULL element if there is one, otherwise a natural element is added. Because the natural course draws the exposure from its fitted model, models then needs an "exposure" model. An integer position or a name selects one of your interventions instead, and no natural course is added.

Rank-deficient models. Terms that cannot be estimated (a collinear term, or one that is constant in the rows used for fitting, such as time in an outcome recorded only at the last time point) are

dropped from the simulation, as predict would drop them, and named in the warning summary printed on exit.

The results rest on the correct temporal ordering and specification of the models and on the usual g-formula assumptions (consistency, positivity, no unmeasured confounding), which the package cannot check.

Value

An object of class "gformula", printed by print.gformula, with components:

- effect_size: mean outcome (risk, for survival) under each intervention (Intervention, Est).
- estimate: with two or more interventions, the risk difference and risk ratio of each against ref_int (Intervention, Risk_type, Estimate).
- With R > 1, both tables gain Sd and percentile (perct_lcl, perct_ucl) and normal-approximation (norm_lcl, norm_ucl) limits, and boot_estimates holds the per-replicate estimates ($interventions, $contrasts).
- sim_data: with return_data = TRUE, the simulated data as an end-of-follow-up snapshot: one row per Monte Carlo subject per intervention, each variable at its last simulated time step, with the accumulated Pred_Y. Earlier time steps are not kept.
- fitted_models: the fitted models, as full objects when return_fitted = TRUE, otherwise their calls and coefficients.
- data_summary: numbers of subjects, rows and time points in data.
- observed: a nonparametric benchmark printed beside the estimates: the observed mean outcome at the last time point, or the product-limit cumulative incidence for a survival outcome.
- call, all.args: the matched call and the evaluated arguments.

References

Robins, J. M. (1986). A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7(9–12), 1393–1512. doi:10.1016/02700255(86)900886

Keil, A. P., Edwards, J. K., Richardson, D. B., Naimi, A. I., & Cole, S. R. (2014). The parametric g-formula for time-to-event data: intuition and a worked example. *Epidemiology*, 25(6), 889–897. doi:10.1097/EDE.0000000000000160

See Also

spec_model, recodes, dyn_int, mediation for direct and indirect effects, and vignette("causalMed-03-gformula").

Examples

```
data(nonsurvivaldata)

# Models in the order the variables are generated: A -> L1 -> L2 -> Y
models <- list(
  spec_model(A ~ V + lag1_A + lag1_L1 + lag1_L2 + time,
```

```
            var_type = "binary", mod_type = "exposure"),
  spec_model(L1 ~ V + A + lag1_L1 + time,
            var_type = "normal", mod_type = "covariate"),
  spec_model(L2 ~ V + A + lag1_L2 + time,
            var_type = "binary", mod_type = "covariate"),
  spec_model(Y_bin ~ V + A + L1 + L2,
            var_type = "binary", mod_type = "outcome")
)

fit <- gformula(
  data = nonsurvivaldata, id_var = "id", time_var = "time",
  base_vars = "V", exposure = "A", models = models,
  intervention = list(natural = NULL, always = 1, never = 0),
  init_recode = recodes(lag1_A = 0, lag1_L1 = 0, lag1_L2 = 0),
  in_recode   = recodes(lag1_A = A, lag1_L1 = L1, lag1_L2 = L2),
  mc_sample = 2000,
  R = 1,          # use R > 1 (e.g. 500) for bootstrap confidence intervals
  quiet = TRUE
)
fit
```

---

gvhd *Bone marrow transplant data (graft-versus-host disease)*

Description

Person-day data for the 137 leukemia patients who received bone marrow transplants under a radiation-free regimen at four medical centers, as used in the parametric g-formula illustration of Keil et al. (2014). Each subject contributes one row per day of follow-up (day 1 up to death, loss to follow-up, or day 1825), giving 108,714 rows. The dataset reproduces the person-day file created in Appendix 1 of that paper: 80 deaths and 39 losses to follow-up.

Usage

```
gvhd
```

Format

A data frame with 108,714 rows and 21 variables:

id unique subject identifier (1-137)

age baseline age in years

male male indicator (1 = male, 0 = female)

cmv baseline cytomegalovirus immune status (1 = positive, 0 = negative)

all leukemia type (1 = acute lymphocytic, 0 = acute myeloid)

wait waiting time from leukemia diagnosis to transplantation, in months (waiting days / 30.5)

day days since bone marrow transplant (1-1825)

d indicator of death at the end of day `day` (1 = yes, 0 = no)

gvhd indicator of graft-versus-host disease having occurred by day `day` (1 = yes, 0 = no); the exposure

daysgvhd cumulative number of days with GvHD

daysnogvhd cumulative number of days without GvHD

gvhdm1 one-day lag of `gvhd`

relapse indicator of leukemia relapse having occurred by day `day` (1 = yes, 0 = no)

relapsem1 one-day lag of `relapse`

daysrelapse cumulative number of days since relapse

daysnorelapse cumulative number of days relapse-free

platnorm indicator of platelet recovery having occurred by day `day` (1 = platelets returned to normal levels, 0 = not yet)

platnormm1 one-day lag of `platnorm`

daysplatnorm cumulative number of days with normal platelet levels

daysnoplatnorm cumulative number of days without normal platelet levels

censlost indicator of censoring due to loss to follow-up at day `day` (1 = yes, 0 = no)

Details

GvHD (`gvhd`) is the exposure, death (`d`) is the outcome, and relapse (`relapse`) and platelet recovery (`platnorm`) are time-varying confounders. The three time-varying states are *absorbing* (once 1, they remain 1); their `daysX`/`daysnoX` counters and one-day lags (`Xm1`) summarize the history entering each model.

A worked total-effect analysis on this dataset (counterfactual mortality risk had GvHD never occurred versus the natural course), mirroring the model specifications in Appendix 2 of Keil et al. (2014), is walked through in `vignette("causalMed-03-gformula")`.

References

Keil, A. P., Edwards, J. K., Richardson, D. B., Naimi, A. I., & Cole, S. R. (2014). The parametric g-formula for time-to-event data: intuition and a worked example. *Epidemiology*, 25(6), 889-897. doi:10.1097/EDE.0000000000000160

`mediation` *Mediation analysis with the parametric g-formula*

Description

Decomposes the effect of a time-varying binary exposure into a direct effect and an indirect effect through one or more time-varying mediators, using the parametric mediational g-formula. Two estimands are available: interventional effects (`mediation_type = "I"`, the default; Lin et al. 2017, VanderWeele & Tchetgen Tchetgen 2017) and natural effects (`"N"`; Zheng & van der Laan 2017).

Usage

```
mediation(
  data,
  id_var,
  base_vars,
  exposure,
  outcome,
  time_var,
  models,
  init_recode = NULL,
  in_recode = NULL,
  out_recode = NULL,
  mc_sample = NULL,
  mediation_type = c("I", "N"),
  exposure_regime = 1,
  reference_regime = 0,
  n_vw = 2L,
  estimator = c("gcomp", "tmle"),
  tmle_weight_trunc = 0.995,
  return_fitted = FALSE,
  return_data = FALSE,
  R = 500,
  quiet = FALSE,
  seed = 12345L
)
```

Arguments

`data`	A `data.frame` in long format.
`id_var`	Character. Name of the subject identifier.
`base_vars`	Character vector of time-fixed baseline covariates (may be empty), with no missing values. Only these and `id_var` are carried into the simulated cohort; every other variable a model uses, including lags, must be created by the recode hooks.
`exposure`	Character. Name of the exposure to intervene on.
`outcome`	Character. Name of the outcome variable; must be the response of the `"outcome"` or `"survival"` model.
`time_var`	Character. Name of the numeric time variable. Each distinct value is one simulated step, in ascending order.
`models`	List of `spec_model` objects, in the order the variables are generated.
`init_recode`	`recodes` applied at the first time step, before the models are evaluated, e.g. to set lags to their baseline value.
`in_recode`	`recodes` applied at the start of each later time step, before the models are evaluated, e.g. to update lags.
`out_recode`	`recodes` applied at the end of each time step, the first included, e.g. to advance cumulative counts or carry an absorbing state forward.

`mc_sample` Number of subjects in the simulated Monte Carlo cohort. The default, `NULL`, uses 50 times the number of subjects in `data` and reports the value unless `quiet = TRUE`. A larger value reduces Monte Carlo error, which falls as `1/sqrt(mc_sample)`; it does not change the estimand.

`mediation_type` `"I"` (default) for interventional effects or `"N"` for natural effects.

Zheng & van der Laan (2017, Lemma 1) identify the `"N"` effects, which draw the mediator from its conditional distribution under the other regime given each subject's own history, under sequential randomization and positivity. Reading them as *individual-level* natural effects additionally requires a cross-world independence assumption that is not expected to hold when a confounder of the mediator-outcome relationship is affected by prior exposure (Avin, Shpitser & Pearl 2005; VanderWeele & Tchetgen Tchetgen 2017), the setting for which VanderWeele & Tchetgen Tchetgen (2017) propose the interventional effects of `"I"`. With `"N"`, a warning (repeated by `print()`) names any covariate whose model includes the exposure. The check reads formulas only: it does not establish that such a covariate confounds the mediator-outcome relationship, and its silence does not establish that none does.

Interventional indirect effects do not, absent stronger assumptions, satisfy the sharp null criterion of Miles (2023), which the natural indirect effect does: a non-zero interventional indirect effect does not by itself show that the mediator transmits the effect for any individual.

`exposure_regime`

Exposure regime $a(1{:}T)$ whose effect is decomposed: 0/1 values, either one value for every time point or one per distinct value of `time_var`, in sorted order. Default `1` (always exposed).

`reference_regime`

Reference regime $a^*(1{:}T)$, in the same form. Default `0` (never exposed). Must differ from `exposure_regime`.

`n_vw` Integer. Number of mediator permutations averaged for each pool-drawing intervention under `"I"` (`Phi00`, `Phi10`, `Phi11`, `Phi1_k`; not `nat0`/`nat1`), within every bootstrap replicate too. Default `2L`, as in the SAS `mGFORMULA` macro; `1L` is faster with more Monte Carlo noise. Ignored under `"N"`.

`estimator` `"gcomp"` (default): the parametric g-formula plug-in with bootstrap intervals. `"tmle"`: the targeted minimum loss-based estimator of Zheng & van der Laan (2017, Section 4.3), with Wald intervals from the efficient influence curve (`R` and `mc_sample` are ignored).

`"tmle"` requires `mediation_type = "N"`, the default regimes, a single mediator, a binary outcome or survival indicator, and an exposure model. It accepts only lag-style recodes: `in_recode` entries that copy one column (an exposure lag must copy the exposure itself), `init_recode` entries that are a constant or a column name, no `out_recode`, no `subset`, no `var_type = "custom"`, and no model `recode` that reads the exposure or its lags. Other inputs raise an error. Its targeted regressions are additive main-effects working models in the variables your formulas name (transformations and interactions such as `poly()` or `A:M` are not carried over), so the multiple robustness Zheng & van der Laan establish for correctly specified nuisance models is not claimed for this implementation.

Where few subjects follow a regime, the affected fluctuation steps are skipped and weights truncated, with a warning.

`tmle_weight_trunc`
: Quantile in (0, 1] at which the TMLE clever-covariate weights are truncated. Default `0.995`; `1` disables truncation. Ignored unless `estimator = "tmle"`.

`return_fitted`
: Logical. Return the full fitted model objects (default `FALSE`: calls and coefficients only).

`return_data`
: Logical. Return the simulated data (default `FALSE`; can be large).

`R`
: Number of bootstrap replicates (default `500`); `R = 1` skips the bootstrap. Replicates run through `future.apply::future_lapply`: sequentially unless a parallel plan is set with `future::plan()`, e.g. `plan(multisession)`.

`quiet`
: Logical. Suppress progress messages (default `FALSE`).

`seed`
: Integer random seed (default `12345L`) for the Monte Carlo simulation and the bootstrap replicates; the global RNG state is restored on exit. `NULL` disables seeding, so repeated calls differ. Give concurrent analyses different seeds.

Details

Data. Long format: one row per subject per time point. For a survival outcome, remove every row after the event and after loss to follow-up. The exposure must be coded 0/1; the methods implemented here are defined for binary exposures only.

Models. `models` is a list of `spec_model` objects in the order the variables are generated within a time point, for example exposure, confounders, mediator, outcome (**A(t) -> L(t) -> M(t) -> Y(t)**), or **A -> M -> L -> Y** when the confounders respond to the mediator. At least one `"mediator"` model is required. Several mediators, listed in temporal order, are supported under `"I"` only (Yamamuro et al. 2021). A warning is given if the exposure is listed after the mediator or the mediator after the outcome.

Exposure regimes. The effects contrast two *static* regimes, a (`exposure_regime`) and a^* (`reference_regime`). The defaults, always versus never exposed, are the example Lin et al. (2017, *Stat Med*, Section 2.2) and Zheng & van der Laan (2017, Section 2.2) give when defining the effects for arbitrary regimes. Element k of a regime is the exposure at the k-th distinct value of `time_var`: for times {1, 3, 4, 6} a regime has four elements. A regime in which the exposure follows its fitted model outside a window is a dynamic regime, a different estimand, and is not available. Which pair answers a given question is the analyst's judgement; only $a = a^*$ is rejected.

How the mediator is set under `"I"`. The model is first simulated once under each regime with the mediator following its fitted model, and every simulated subject's mediator trajectory $M(1{:}T)$ is stored in a pool. Each decomposition intervention (`Phi00`, `Phi10`, `Phi11`, and `Phi1_k` with several mediators) then gives each subject the whole trajectory of a randomly permuted pool member, each mediator permuted independently, and averages `n_vw` such permutations. The permutation runs over the whole simulated cohort, so the draw is marginal over the baseline covariates as well as the confounder history: the whole-population draw G_a of VanderWeele & Tchetgen Tchetgen (2017), as in the algorithm of Lin et al. (2017, *Epidemiology*). Yamamuro et al. (2021, Eq. 2) and Lin et al. (2017, *Stat Med*, Eq. 4) are written for the draw conditional on baseline covariates, but the algorithms of both papers, and the SAS macros distributed with them, permute the whole cohort. The two forms agree when the counterfactual mediator distribution does not depend on the baseline

covariates, or when the outcome mean is additively separable in them and the mediator; otherwise they are different functionals.

How the mediator is set under `"N"`. The mediator model is re-evaluated on each subject's own simulated history with the exposure history set to the other regime (Zheng & van der Laan 2017, Eq. 5). Only the exposure and its first-order lags (`in_recode` entries that copy the exposure, e.g. `recodes(lag1_A = A)`) are switched, so a mediator model whose formula or `subset` reads any other exposure-derived column (a longer lag, a cumulative count, an `out_recode` copy, a column created by a model's own `recode`) is rejected with an error.

Total effect and residual. The total effect is the natural plug-in contrast `nat1 - nat0`, from runs in which the exposure follows each regime and the mediator follows its fitted model. Under `"I"` the direct and indirect effects sum to the interventional overall effect `Phi11 - Phi00` rather than to the total effect, and the difference is reported as the decomposition residual. Under `"N"` they sum to the total effect and no residual is reported.

Survival outcomes and censoring. The algorithm is that of Lin et al. (2017, *Stat Med*, Section 4): models are fitted among those at risk, no simulated subject is removed, and the cumulative risk $1 - \prod_t (1 - \hat{p}_t)$ is accumulated from the predicted hazards. A `"censor"` model sets the censoring indicator to zero under every intervention, so every risk is the risk under eliminated loss to follow-up; with `estimator = "gcomp"` it does not change any reported risk. It marks the analysis as a survival setting and is used by `estimator = "tmle"`.

Warnings. Model-fitting warnings (e.g. non-convergence) are collected and printed once, with a repeat count, when the function returns.

Value

An object of class `"gformula"`, printed by `print.gformula`, with components:

- `effect_size`: mean outcome (risk, for survival) under each intervention (`Intervention`, `Est`). Under `"I"`: `Phi00` $= E[Y_{0,G_0}]$, `Phi10` $= E[Y_{1,G_0}]$, `Phi11` $= E[Y_{1,G_1}]$ (mediator from a permuted pool), and `Phi1_k` (k = 1, ..., N-1) with N mediators; under `"N"`, `Phi00` and `Phi11` use each subject's own mediator. Always `nat0` $= E[Y_0]$ and `nat1` $= E[Y_1]$. With non-default regimes, 0 and 1 in these labels stand for a^* and a.
- `estimate`: the decomposition, on the risk-difference (`RD`) and risk-ratio (`RR`) scales. Rows: `"Indirect effect"` (`Phi11 - Phi10`; one row per mediator, `"Indirect effect (<mediator>)"`, with several), `"Direct effect"` (`Phi10 - Phi00`), `"Total effect"` (`nat1 - nat0`), `"TE - (Direct + Indirect)"` (`"I"` only), and `"Mediation Proportion"`: the summed indirect effects as a percentage of the interventional overall effect `Phi11 - Phi00`, the quantity Lin et al. (2017, *Stat Med*, Table 2) report. It is not a share of the total effect, from which it differs by the residual. No separate risk-ratio proportion is given: $RR_{IDE}(\prod_k RR_{IIE_k} - 1)/(RR_{OE} - 1)$ simplifies to the same number. `RR` is `NA` for the residual and proportion rows.
- With `R > 1`, both tables gain `Sd` and percentile (`perct_lcl`, `perct_ucl`) and normal-approximation (`norm_lcl`, `norm_ucl`) limits, with an `_RR` suffix on the risk-ratio scale; `boot_estimates` holds the per-replicate estimates (`$interventions`, `$effects`).
- `sim_data`: with `return_data = TRUE`, the simulated data as an end-of-follow-up snapshot: one row per Monte Carlo subject per intervention, each variable at its last simulated time step, with the accumulated `Pred_Y`. With `n_vw > 1` only the last permutation is kept while `effect_size$Est` averages all of them, so `mean(Pred_Y)` reproduces `Est` for the pool-drawing interventions only when `n_vw = 1` (always for `nat0`/`nat1`).

- `fitted_models`: the fitted models, as full objects when `return_fitted = TRUE`, otherwise their calls and coefficients.
- `data_summary`: numbers of subjects, rows and time points in `data`, and `regime_support`, the number of observed subjects whose exposure matches each regime at every time they were observed (`n_following`) and among those observed at every time point (`n_complete`). A subject with a missing exposure counts for neither regime. The estimation does not use these counts.
- `observed`: a nonparametric benchmark printed beside the estimates: the observed mean outcome at the last time point, or the product-limit cumulative incidence for a survival outcome.
- `intermediate_confounders`: under `"N"`, covariates whose model includes the exposure (see `mediation_type`).
- `tmle_diag`: with `estimator = "tmle"`, the subject-level efficient influence curve (`eic`), its column means (`eic_mean`) and the number of subjects (`n`).
- `call`, `all.args`: the matched call and the evaluated arguments.

References

Avin, C., Shpitser, I., & Pearl, J. (2005). Identifiability of path-specific effects. *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 357-363.

Lin, S. H., Young, J. G., Logan, R., & VanderWeele, T. J. (2017). Mediation analysis for a survival outcome with time-varying exposures, mediators, and confounders. *Statistics in Medicine*, 36(26), 4153–4166. doi:10.1002/sim.7426

Lin, S. H., Young, J., Logan, R., Tchetgen Tchetgen, E. J., & VanderWeele, T. J. (2017). Parametric mediational g-formula approach to mediation analysis with time-varying exposures, mediators, and confounders. *Epidemiology*, 28(2), 266–274. doi:10.1097/EDE.0000000000000609

Miles, C. H. (2023). On the causal interpretation of randomised interventional indirect effects. *Journal of the Royal Statistical Society: Series B*, 85(4), 1154–1172. doi:10.1093/jrsssb/qkad066

VanderWeele, T. J., & Tchetgen Tchetgen, E. J. (2017). Mediation analysis with time varying exposures and mediators. *Journal of the Royal Statistical Society: Series B*, 79(3), 917–938. doi:10.1111/rssb.12194

Yamamuro, S., Shinozaki, T., Iimuro, S., & Matsuyama, Y. (2021). Mediational g-formula for time-varying treatment and repeated-measured multiple mediators: Application to atorvastatin's effect on cardiovascular disease via cholesterol lowering and anti-inflammatory actions in elderly type 2 diabetics. *Statistical Methods in Medical Research*, 30(8), 1782–1799. doi:10.1177/09622802211025988

Zheng, W., & van der Laan, M. (2017). Longitudinal mediation analysis with time-varying mediators and exposures, with application to survival outcomes. *Journal of Causal Inference*, 5(2). doi:10.1515/jci20160006

See Also

`spec_model`, `recodes`, `gformula` for total effects, and `vignette("causalMed-02-mediation")`.

Examples

```
data(nonsurvivaldata)
```

```
# Models in the order the variables are generated: A -> L1 -> L2 -> M -> Y
models <- list(
  spec_model(A ~ V + lag1_A + lag1_L1 + lag1_L2 + time,
             var_type = "binary", mod_type = "exposure"),
  spec_model(L1 ~ V + A + lag1_L1 + time,
             var_type = "normal", mod_type = "covariate"),
  spec_model(L2 ~ V + A + lag1_L2 + time,
             var_type = "binary", mod_type = "covariate"),
  spec_model(M ~ V + A + L1 + L2 + lag1_M + time,
             var_type = "normal", mod_type = "mediator"),
  spec_model(Y_bin ~ V + A + M + L1 + L2,
             var_type = "binary", mod_type = "outcome")
)

fit <- mediation(
  data = nonsurvivaldata, id_var = "id", time_var = "time",
  base_vars = "V", exposure = "A", outcome = "Y_bin", models = models,
  init_recode = recodes(lag1_A = 0, lag1_L1 = 0, lag1_L2 = 0, lag1_M = 0),
  in_recode   = recodes(lag1_A = A, lag1_L1 = L1, lag1_L2 = L2, lag1_M = M),
  mc_sample = 2000,
  R = 1,          # use R > 1 (e.g. 500) for bootstrap confidence intervals
  quiet = TRUE
)
fit
```

`nonsurvivaldata` *Example Dataset for a Non-Survival Outcome*

Description

A simulated dataset with time-varying and baseline variables for 3000 subjects over 5 time points, including exposure, mediator, confounders, and an end-of-follow-up outcome. Every subject contributes all 5 rows (there is no attrition), which makes this the simpler of the two simulated examples.

Usage

```
nonsurvivaldata
```

Format

A data frame with 15,000 rows (3000 subjects x 5 time points) and 13 variables:

id Unique subject identifier (1-3000).

time Time variable (0 to 4).

V Time-fixed baseline covariate.

L1 Time-varying confounder 1 (continuous).

L2 Time-varying confounder 2 (binary).

A Time-varying binary exposure.

M Time-varying mediator (continuous).

Y_bin Binary end-of-follow-up outcome. Recorded **only at** `time == 4`; `NA` at `time` 0-3.

Y_cont Continuous end-of-follow-up outcome, on the same schedule as `Y_bin`: recorded **only at** `time == 4`, `NA` otherwise.

lag1_A Exposure at the previous time point; `NA` at `time == 0`.

lag1_L1 Previous value of `L1`; `NA` at `time == 0`.

lag1_L2 Previous value of `L2`; `NA` at `time == 0`.

lag1_M Previous value of `M`; `NA` at `time == 0`.

Details

The simulated longitudinal data-generating structure can be summarized as:

$A_t \leftarrow V, L1_{t-1}, L2_{t-1}, A_{t-1}, t; \quad L1_t \leftarrow V, A_t, L1_{t-1}, t; \quad L2_t \leftarrow V, A_t, L2_{t-1}, t; \quad M_t \leftarrow V, A_t, L1_t, L2_t, M_{t-1}, t; \quad Y$

The exposure, mediator and confounders evolve at every time point, but the outcome is realised once, at the end of follow-up.

The generating parameters are, with $V \sim N(0,1)$ and all lagged terms set to zero at $t = 0$:

```
logit P(A_t = 1) =  1.25 + 0.50 V + 0.25 L1_{t-1} + 0.25 L2_{t-1}
                         + 0.20 A_{t-1} + 0.01 t
E(L1_t)          =  0.02 + 0.50 V + 0.18 A_t + 0.35 L1_{t-1} + 0.02 t
                    (Gaussian, SD 0.70)
logit P(L2_t = 1)= -0.10 + 0.40 V + 0.18 A_t + 0.35 L2_{t-1} + 0.02 t
E(M_t)           =  0.05 + 0.50 V + 0.45 A_t + 0.12 L1_t + 0.10 L2_t
                         + 0.30 M_{t-1} + 0.02 t   (Gaussian, SD 0.50)
logit(pi)        = -2.60 + 0.50 V + 0.51 A_4 + 0.45 M_4
                         + 0.25 L1_4 - 0.25 L2_4 + 0.30 A_4 M_4
```

Both outcomes share that same π: `Y_bin` is $\mathrm{Bernoulli}(\pi)$ and `Y_cont` is $\mathrm{Beta}(50\pi,\, 50(1-\pi))$. `Y_cont` therefore lies strictly in $(0, 1)$; fitting it with `var_type = "normal"` is a working approximation, not the generating distribution.

The confounders $L1$ and $L2$ depend on current exposure, so exposure-affected confounding of the mediator-outcome relationship is present – the setting the interventional effects are designed for.

Large-sample true values (risk differences, computed under the same permuted-pool construction the package uses for `mediation_type = "I"`, at 200,000 subjects):

Interventional direct effect (IDE)	0.075
Interventional indirect effect (IIE)	0.071
Total effect (TE)	0.155
Decomposition residual (TE − overall)	0.009

The two pathways are deliberately of comparable size, and the residual is deliberately non-zero, so that the decomposition and its non-additivity are both visible in a worked example.

Two consequences for model specification:

- Because the outcome exists at a single time point, `time` is constant among the rows used to fit the outcome model and must **not** appear in its formula: the term is not estimable and would be dropped (with a warning) from the simulation.
- The `lag1_*` columns are `NA` at `time == 0`, so the rows at `time == 0` are dropped when a model that reads them is fitted. In the simulation, as for any data, a lag must be given its first value by `init_recode`, e.g. `init_recode = recodes(lag1_A = 0, lag1_L1 = 0)`. (The lag columns of `survivaldata` are 0-filled at baseline instead, so no rows are dropped there; the simulation still needs `init_recode`.)

Source

Simulated from the parametric process given under Details.

See Also

`survivaldata` for the survival-outcome counterpart.

`print.gformula` *Print the results of gformula() or mediation()*

Description

Prints the estimated mean outcome (or risk) under each intervention, the contrasts or mediation decomposition, a legend of the interventions, the analysis setup, and the observed nonparametric benchmark.

Usage

```
## S3 method for class 'gformula'
print(x, models = FALSE, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	Object of class `"gformula"`.
`models`	Logical. If `TRUE`, print fitted model details (call and coefficients, or full summary when `return_fitted = TRUE`). Default `FALSE`.
`digits`	Integer. Number of decimal places used when rounding numeric output. Default `max(3, getOption("digits") - 3)`.
`...`	Not used.

Value

`x`, invisibly. The function is called for its printed output.

recodes *Recode variables inside the simulation*

Description

Captures named expressions, unevaluated, to be evaluated inside the simulated data. They are used for the `init_recode`, `in_recode` and `out_recode` hooks of `gformula` and `mediation`, and for the `recode` argument of `spec_model`. Expressions are evaluated in order, so a later one can use the result of an earlier one.

The simulated cohort starts with only the subject identifier and the baseline covariates, so every lag, counter or other derived variable a model uses must be created here, even if it already exists in the data:

- `init_recode`: at the first time step, before the models are evaluated (initial values).
- `in_recode`: at the start of every later time step, before the models are evaluated (lags, functions of time).
- `out_recode`: at the end of every time step, the first included, after the models are evaluated (cumulative counts, carrying an absorbing state forward). A variable it updates needs a starting value from `init_recode`.

Usage

```
recodes(...)
```

Arguments

`...`	Named expressions, e.g. `lag1_A = A` or `daysq = day^2`.

Value

A list of unevaluated expressions of class `"causalMed_recodes"`.

Examples

```
# One-period lags: 0 at the first time step, then the previous value
init <- recodes(lag1_A = 0, lag1_L = 0)
upd  <- recodes(lag1_A = A, lag1_L = L)

# A function of the time variable, e.g. for in_recode
recodes(day_sq = day^2)

# A running count of exposed steps: start at 0 (init_recode), add the
# current step's exposure at the end of each step (out_recode)
init <- recodes(cum_A = 0)
upd  <- recodes(cum_A = cum_A + A)
```

spec_model *Specify a model for one time-varying variable*

Description

Describes the model for one time-varying variable: its formula, its role in the data-generating process (`mod_type`) and how its values are simulated (`var_type`). Nothing is fitted here; `gformula` and `mediation` fit the models, in list order, and use them to simulate counterfactual trajectories.

Usage

```
spec_model(
  formula,
  subset = NULL,
  recode = NULL,
  var_type = c("normal", "binary", "categorical", "custom"),
 mod_type = c("covariate", "exposure", "mediator", "outcome", "censor", "survival"),
  custom_fit = NULL,
  custom_sim = NULL,
  truncate = TRUE,
  ...
)
```

Arguments

`formula`	Model formula, e.g. `L ~ A + lag1_L + time`. Every variable must exist in the analysis data, or be created by a recode hook.
`subset`	Optional unquoted logical expression, e.g. `platnormm1 == 0`. The model is fitted on the rows where it holds and, at each simulated time step, draws only those rows; other rows keep their current value. This is how an absorbing state is modelled.
`recode`	Optional `recodes` applied before this model is fitted and before its response is simulated.
`var_type`	How values are simulated: `"normal"` (default) Gaussian draws around the fitted linear model, with the residual standard deviation. `"binary"` Bernoulli draws from a logistic model. `"categorical"` Draws from a multinomial logistic model (`multinom`), returned in the variable's type in the data (a factor keeps its levels, numeric codes stay numeric); the **Hmisc** package must be installed. `"custom"` A user-supplied fitting function (`custom_fit`) and/or simulation function (`custom_sim`). `"censor"` and `"survival"` models must be `"binary"`; an `"outcome"` model must be `"binary"` or `"normal"`.

`mod_type` Role of the variable: `"covariate"` (default), `"exposure"`, `"mediator"`, `"outcome"` (end-of-follow-up outcome), `"survival"` (discrete-time event indicator) or `"censor"` (loss to follow-up). Under every intervention the censoring indicator is set to zero, so risks are the risks under eliminated loss to follow-up, and with `estimator = "gcomp"` a censoring model does not change them. It is simulated in the natural course of `gformula` and used by `estimator = "tmle"` in `mediation`.

`custom_fit` Fitting function for `var_type = "custom"` (ignored, with a warning, otherwise); default `glm`. Its *name* is recorded and evaluated later, so it must be reachable from the global environment or a package namespace: a function defined inside another function or `local()` will not be found. A namespace-qualified name (e.g. `truncreg::truncreg`) also works on parallel bootstrap workers. Without `custom_sim`, the fitted object must have a `terms` component and a `coef` method.

`custom_sim` Simulation function `function(fit, newdata)` returning one simulated value per row of `newdata`. It replaces the draw implied by `var_type`, so it should return a draw from the model's distribution rather than a fitted mean. If it is omitted with `var_type = "custom"`, values are drawn from a normal distribution around the linear predictor; that is the fitted mean only under an identity link, so a fit with another link is rejected. For `"outcome"` and `"survival"` models the reported risk is always computed from the fitted model, not from `custom_sim`.

`truncate` Logical (default `TRUE`). Clip simulated numeric values (`"normal"` draws and numeric `custom_sim` output) to the range of the response observed in the data. This matches the default `sim_trunc = TRUE` of **gfoRmula**. `FALSE` draws from the untruncated distribution. No effect on `"binary"` or `"categorical"` variables.

`...` Further arguments to the fitting function (`glm`, `multinom` or `custom_fit`), e.g. `family`.

Details

Choosing a model and a simulation rule that suit each variable is the analyst's responsibility; the package checks only what it can verify mechanically.

Value

An object of class `"causalMed_gmodel"`: the unevaluated fitting call plus `subset`, `recode`, `var_type`, `mod_type`, `custom_sim` and `truncate`.

See Also

`gformula`, `mediation`, `recodes`

Examples

```
# A binary covariate modelled only among those not yet in the state
spec_model(platnorm ~ all + cmv + male + age + gvhdm1 + daysgvhd + wait,
           var_type = "binary", mod_type = "covariate",
```

```
            subset = platnormm1 == 0)

# A count covariate: Poisson fit with a matching simulation function
sim_poisson <- function(fit, newdata) {
  rpois(nrow(newdata), predict(fit, newdata = newdata, type = "response"))
}
spec_model(daysgvhd ~ all + cmv + male + age + wait,
           var_type = "custom", mod_type = "covariate",
           custom_sim = sim_poisson, family = poisson(link = "log"))
```

`summary.gformula` *Summarise the results of gformula() or mediation()*

Description

Prints the same results as `print.gformula`, followed by the coefficient table of every fitted model.

Usage

```
## S3 method for class 'gformula'
summary(object, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	Object of class `"gformula"`.
`digits`	Integer. Number of decimal places used when rounding numeric output. Default `max(3, getOption("digits") - 3)`.
`...`	Not used.

Value

`object`, invisibly. The function is called for its printed output.

`survivaldata` *Example Dataset for a Survival Outcome*

Description

A simulated dataset with time-varying and baseline variables for 3000 subjects with a discrete-time survival outcome, suitable for use with the g-formula and mediation functions under survival settings.

Usage

```
survivaldata
```

Format

A data frame with 10,223 rows and 11 variables, in long at-risk format (one row per subject per time point while still at risk; 3000 subjects, 1822 events, 283 censored, no missing values):

id Unique subject identifier (1-3000).

time Time variable (integer, 0 to 4).

V Time-fixed baseline covariate.

L Time-varying confounder (continuous).

A Time-varying binary exposure.

M Time-varying binary mediator.

Y Event indicator at this time point (1 = event, 0 = still at risk).

C Loss-to-follow-up indicator at this time point (1 = censored). Censoring is drawn only among subjects who did not have the event at that time, so `Y` and `C` are never both 1.

lag1_A Exposure at the previous time point; 0 at `time == 0`.

lag1_M Previous value of `M`; 0 at `time == 0`.

lag1_L Previous value of `L`; 0 at `time == 0`.

Details

The data-generating structure can be summarized as:

$$A_t \leftarrow V, L_{t-1}, A_{t-1}, t; \quad L_t \leftarrow V, A_t, L_{t-1}, t; \quad M_t \leftarrow V, A_t, L_t, M_{t-1}, t; \quad Y_t \leftarrow V, A_t, M_t, L_t, A_t{*}M_t, t.$$

The generating parameters are, with $V \sim N(0,1)$ and all lagged terms set to zero at $t = 0$:

```
logit P(A_t = 1) = -0.40 + 0.60 V + 0.40 L_{t-1} + 0.58 A_{t-1} + 0.05 t
E(L_t)           =  0.50 + 0.60 V + 0.30 A_t + 0.20 L_{t-1} + 0.05 t
                    (Gaussian, SD 0.10)
logit P(M_t = 1) = -0.95 + 0.50 V + 1.50 A_t + 0.40 L_t
                         + 0.20 M_{t-1} + 0.05 t
logit P(Y_t = 1) = -3.40 - 0.50 V + 0.25 A_t + 1.70 M_t + 0.30 L_t
                         + 0.50 A_t M_t
logit P(C_t = 1) = -3.80 + 0.30 V + 0.30 L_t + 0.10 t
```

The hazard carries no time trend. The confounder L depends on current exposure, so exposure-affected confounding of the mediator-outcome relationship is present.

Large-sample true values (cumulative risk by the fifth period, computed under the same permuted-pool construction the package uses for `mediation_type = "I"`, at 200,000 subjects):

Risk under no exposure	0.39
Risk under exposure	0.79
Interventional direct effect (IDE)	0.194
Interventional indirect effect (IIE)	0.191
Total effect (TE)	0.400

Decomposition residual (TE − overall) 0.015

Follow-up stops at the event or at loss to follow-up, whichever comes first, so no rows are retained for a subject after either. Of the 3000 subjects, 1822 have the event, 283 are censored and the remaining 895 reach the end of the fifth period.

Censoring depends on V and on the current confounder L, so it is informative marginally but conditionally independent of the counterfactual event process given that observed history. The target estimand, the risk with censoring eliminated, is therefore unchanged by it, and the true values above are the same as they would be without it. With `estimator = "gcomp"` a censoring model is not required; a `mod_type = "censor"` model can be supplied to simulate censoring in the natural course of `gformula()`, or for `estimator = "tmle"`.

Source

Simulated from the parametric process given under Details.

See Also

`nonsurvivaldata` for the end-of-follow-up counterpart.

`yamamurodata` *Yamamuro et al. (2021) Multi-Mediator Simulation Dataset*

Description

A single dataset simulated from the data-generating process of the Yamamuro et al. (2021) multi-mediator simulation study: a time-varying binary treatment, a time-varying confounder, two sequential continuous mediators, and a discrete-time survival outcome over three visits. The large-sample true interventional effects for this process are documented below, so estimates obtained with `mediation` can be compared against known values.

Usage

```
yamamurodata
```

Format

A data frame with 29482 rows and 15 variables (long at-risk format: one row per subject per visit until the event):

id Unique subject identifier.

time Visit index (0, 1, 2).

V Time-fixed binary baseline covariate.

A Time-varying binary treatment.

L Time-varying continuous confounder.

M1 First mediator (continuous), affected by `A` and `L`.

M2 Second mediator (continuous), affected by `A`, `L`, and `M1`.

Y Event indicator (1 = event at this visit).

lag1_A, lag1_L, lag1_M1, lag1_M2 Previous-visit values (0 at baseline).

L0base, M10base, M20base Baseline (visit 0) values of `L`, `M1`, `M2`, carried as time-fixed columns so the Monte Carlo simulation can be seeded via `init_recode`.

Details

The within-visit ordering is $\mathbf{A} \to \mathbf{L} \to \mathbf{M1} \to \mathbf{M2} \to \mathbf{Y}$; the data-generating equations are those of the SAS `%simdata` macro in the supplementary material of Yamamuro et al. (2021).

The published study design generates 1000 replicate datasets of $n = 1000$ subjects and averages the estimates. A single replicate of that size is dominated by sampling error (between-replicate SD of the total effect is about 1.7 percentage points), so the dataset shipped here is one draw of $n = 10000$ subjects (seed 2468) from the identical process, making single-dataset estimates informative.

Large-sample true values (risk differences in percentage points, computed from the data-generating process at $n = 10^7$; Monte Carlo SE in parentheses):

Total effect (TE = $E[Y_1] - E[Y_0]$)	-6.36 (0.011)
Interventional direct effect (IDE)	-3.20 (0.013)
Interventional indirect effect via M1	-2.29 (0.011)
Interventional indirect effect via M2	-0.97 (0.009)
Decomposition residual (TE $-$ overall)	0.10 (0.016)

The residual arises because the total effect lets the mediators follow their fitted models, while the direct and indirect effects draw them from a permuted pool; see Details of `mediation`.

Source

Simulated from the data-generating process of Yamamuro, S., Shinozaki, T., Iimuro, S., & Matsuyama, Y. (2021). Mediational g-formula for time-varying treatment and repeated-measured multiple mediators. *Statistical Methods in Medical Research*, 30(8), 1782-1799. doi:10.1177/09622802211025988

Index

∗ **datasets**
gvhd, 6
nonsurvivaldata, 13
survivaldata, 19
yamamurodata, 21

coef, *18*

dyn_int, 2, *4, 5*

gformula, 3, *12*, *16–18*
glm, *18*
gvhd, 6

mediation, *5*, 7, *16–18*, *21, 22*
multinom, *17*

nonsurvivaldata, 13, *21*

predict, *5*
print.gformula, *5*, *11*, 15, *19*

recodes, *4, 5*, *8*, *12*, 16, *17, 18*

spec_model, *3–5*, *8*, *10*, *12*, *16*, 17
summary.gformula, 19
survivaldata, *15*, 19

yamamurodata, 21