Package ‘lmtp’

July 18, 2020

Title  Non-Parametric Causal Effects of Feasible Interventions Based on Modified Treatment Policies

Version  0.0.5

Description  Non-parametric estimators for casual effects based on longitudinal modified treatment policies as described in Diaz, Williams, and Hoffman (<arXiv:2006.01366>), traditional point treatment, and traditional longitudinal effects. Continuous, binary, and categorical treatments are allowed as well are censored outcomes. The treatment mechanism is estimated via a density ratio classification procedure irrespective of treatment variable type. For both continuous and binary outcomes, additive treatment effects can be calculated and relative risks and odds ratios may be calculated for binary outcomes. Estimation is enhanced using the Super Learner from 'sl3' available for download from GitHub using 'remotes::install_github("tlverse/sl3@devel")'.

Depends  R (>= 2.10)

License  MIT + file LICENSE

Encoding  UTF-8

LazyData  true

RoxygenNote  7.1.0

Imports  slider, stats, nnls, cli, utils, R6, generics, origami, future (>= 1.17.0), progressr

Suggests  testthat (>= 2.1.0), covr, rmarkdown, knitr, ranger, twang

Enhances  sl3 (>= 1.3.7)

NeedsCompilation  no

Author  Nicholas Williams [aut, cre, cph] (<https://orcid.org/0000-0002-1378-4831>), Iván Díaz [aut, cph] (<https://orcid.org/0000-0001-9056-2047>)

Maintainer  Nicholas Williams <niw4001@med.cornell.edu>

Repository  CRAN

Date/Publication  2020-07-18 09:10:02 UTC
**R topics documented:**

- `create_node_list` ................................................................. 2
- `event_locf` ................................................................. 3
- `lmtp_contrast` ................................................................. 4
- `lmtp_ipw` ................................................................. 5
- `lmtp_sdr` ................................................................. 10
- `lmtp_sub` ................................................................. 15
- `lmtp_tmle` ................................................................. 20
- `sim_cens` ................................................................. 25
- `sim_point_surv` ................................................................. 26
- `sim_t4` ................................................................. 27
- `static_binary_off` ................................................................. 27
- `static_binary_on` ................................................................. 28
- `tidy.lmtp` ................................................................. 29

**Index**

<table>
<thead>
<tr>
<th>create_node_list</th>
<th>Create a node list specification</th>
</tr>
</thead>
</table>

**Description**

Creates a node list specification that is used by the provided estimators. `create_node_list()` is not explicitly called by the analyst, rather it is provided so the analyst can confirm how estimators will use variables before actually performing the estimation procedure.

**Usage**

```r
create_node_list(trt, tau, time_vary = NULL, baseline = NULL, k = Inf)
```

**Arguments**

- `trt`  
  A vector of column names of treatment variables.

- `tau`  
  The number of time points of observation, excluding the final outcome.

- `time_vary`  
  A list of length `tau` with the column names for new `time_vary` to be introduced at each time point. The list should be ordered following the time ordering of the model.

- `baseline`  
  An optional vector of columns names for baseline covariates to be included for adjustment at every timepoint.

- `k`  
  An integer specifying how previous time points should be used for estimation at the given time point. Default is `Inf`, all time points.

**Value**

A list the same length of the `time_vary` parameter with the variables to be used for estimation at that given time point.
**event_locf**

### Examples

```r
a <- c("A_1", "A_2", "A_3", "A_4")
bs <- c("W_1", "W_2")
time_vary <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))

# assuming no Markov property
create_node_list(a, 4, time_vary, bs, k = Inf)

# assuming a Markov property
create_node_list(a, 4, time_vary, bs, k = 0)
```

---

#### Description

A helper function to prepare survival data for use with LMTP estimators by imputing outcome nodes using last outcome carried forward when an observation experiences the event before the end-of-follow-up.

#### Usage

```r
event_locf(data, outcomes)
```

#### Arguments

- **data**  
The dataset to modify.
- **outcomes**  
A vector of outcome nodes ordered by time.

#### Value

A modified dataset with future outcome nodes set to 1 if an observation experienced an event at any previous time point.

#### Examples

```r
event_locf(sim_point_surv, c(paste0("Y.", 0:6)))
```
lmtp_contrast  

Perform Contrasts of LMTP Fits

Description

Estimates contrasts of multiple LMTP fits compared to either a known reference value or a reference LMTP fit.

Usage

```
lmtp_contrast(..., ref, type = c("additive", "rr", "or"))
```

Arguments

- `...`: One or more objects of class `lmtp`.
- `ref`: A reference value or another `lmtp` fit to compare all other fits against.
- `type`: The contrasts of interest. Options are "additive" (the default), "rr", and "or".

Value

A list of class `lmtp_contrast` containing the following components:

- `type`: The type of contrast performed.
- `null`: The null hypothesis.
- `vals`: A dataframe containing the contrasts estimates, standard errors, and confidence intervals.
- `eifs`: Un-centered estimated influence functions for contrasts estimated.

Examples

```r
library(lmtp)
head(sim_cens)
a <- c("A1", "A2")
nodes <- list(c("L1"), c("L2"))
cens <- c("C1", "C2")
y <- "y"
# mean population outcome
psi_null <- lmtp_tmle(sim_cens, a, y, time_vary = nodes,
                      cens = cens, shift = NULL, folds = 2)

# treatment rule, everyone is increased by 0.5
d <- function(data, x) data[[x]] + 0.5
psi_rule1 <- lmtp_tmle(sim_cens, a, y, time_vary = nodes,
                      cens = cens, shift = d, folds = 2)

# treatment rule, everyone is decreased by 0.5
```
\texttt{lmtp_ipw}

\begin{verbatim}
d <- function(data, x) data[[x]] - 0.5
psi_rule2 <- lmtp_tmle(sim_cens, a, y, time_vary = nodes,
    cens = cens, shift = d, folds = 2)

# Example 1.1
# Additive effect of rule 1 compared to a known constant
lmtp_contrast(psi_rule1, ref = 0.9)

# Example 1.2
# Additive effect of rule 1 compared to the population mean outcome
lmtp_contrast(psi_rule1, ref = psi_null)

# Example 1.3
# Additive effects of rule 1 and 2 compared to the population mean outcome
lmtp_contrast(psi_rule1, psi_rule2, ref = psi_null)

# Example 1.4
# Relative risk of rule 1 compared to observed exposure
lmtp_contrast(psi_rule1, ref = psi_null, type = "rr")

# Example 1.5
# Odds of rule 1 compared to observed exposure
lmtp_contrast(psi_rule1, ref = psi_null, type = "or")
\end{verbatim}

---

\textbf{Description}

Inverse probability of treatment weighting estimator for the effects of traditional causal effects and modified treatment policies for both point treatment and longitudinal data with binary, continuous, or time-to-event outcomes. Supports binary, categorical, and continuous exposures.

\textbf{Usage}

\begin{verbatim}
lmtp_ipw(
    data, trt, outcome, baseline = NULL, time_vary = NULL, cens = NULL, k = Inf, id = NULL, shift, learners = NULL, folds = 10, bound = 1e-05
)
\end{verbatim}
Arguments

data A data frame in wide format containing all necessary variables for the estimation problem.

trt A vector containing the column names of treatment variables ordered by time.

outcome The column name of the outcome variable. In the case of time-to-event analysis, a vector containing the column names of intermediate outcome variables and the final outcome variable ordered by time. Only numeric values are allowed. If the outcome type is binary, data should be coded as 0 and 1.

baseline An optional vector of column names of baseline covariates to be included for adjustment at every timepoint.

time_vary A list the same length as the number of time points of observation with the column names for new time-varying covariates introduced at each time point. The list should be ordered following the time ordering of the model.

cens An optional vector of column names of censoring indicators the same length as the number of time points of observation. If missingness in the outcome is present or if time-to-event outcome, must be provided.

k An integer specifying how previous time points should be used for estimation at the given time point. Default is Inf, all time points.

id An optional column name containing cluster level identifiers.

shift A two argument function that specifies how treatment variables should be shifted. See examples for how to specify shift functions for continuous, binary, and categorical exposures.

learners An optional sl3 learner stack for estimation of the exposure mechanism. If not specified, will default to using a generalized linear model.

folds The number of folds to be used for cross-fitting. Minimum allowable number is two folds.

bound Determines that maximum and minimum values (scaled) predictions will be bounded by. The default is 1e-5, bounding predictions by 1e-5 and 0.9999.

Value

A list of class lmtp containing the following components:

estimator The estimator used, in this case "IPW".

theta The estimated population LMTP effect.

standard_error NA

low NA

high NA

shift The shift function specifying the treatment policy of interest.

density_ratios An n x Tau matrix of the estimated density ratios.

weights_r A list the same length as folds, containing the Super Learner ensemble weights at each time-point for each fold for the propensity.
Examples

library(lmtp)

# Example 1.1
# Point treatment, continuous exposure, continuous outcome, no loss-to-follow-up
# Interested in the effect of a population wide decrease in A of 5 units
# The true value under this intervention is about 519.
set.seed(56)
n <- 1000
W <- rnorm(n, 10, 5)
A <- 23 + 5*W + rnorm(n)
Y <- 7.2*A + 3*W + rnorm(n)
ex1_dat <- data.frame(W, A, Y)
d <- function(data, x) data[[x]] - 5
psi1.1 <- lmtp_ipw(ex1_dat, "A", "Y", baseline = "W", shift = d, folds = 2)
psi1.1

# Example 1.2
# Point treatment, continuous exposure, continuous outcome, no loss-to-follow-up
# Interested in the effect of a modified treatment policy where A is decreased by 15
# units only among observations whose observed A was above 80.
# The true value under this intervention is about 513.
d <- function(data, x) (data[[x]] > 80)*(data[[x]] - 15) + (data[[x]] <= 80)*data[[x]]
psi1.2 <- lmtp_ipw(ex1_dat, "A", "Y", baseline = "W", shift = d, folds = 2)
psi1.2

# Example 2.1
# Longitudinal setting, time-varying continuous exposure bounded by 0,
# time-varying covariates, and a binary outcome with no loss-to-follow-up.
# Interested in the effect of a treatment policy where exposure decreases by
# one unit at every time point if an observations observed exposure is greater
# than or equal to 2. The true value under this intervention is about 0.305.
head(sim_t4)
# specifying treatment variables
a <- c("A_1", "A_2", "A_3", "A_4")
# specifying time varying covariates
tv <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))
# treatment policy function to be applied at all time points
d <- function(data, trt) {
a <- data[[trt]]
(a - 1) * (a - 1 >= 1) + a * (a - 1 < 1)
}
progressr::with_progress(
  psi2.1 <- lmtp_ipw(sim_t4, a, "Y", time_vary = tv, shift = d, folds = 2)
))
psi2.1

# Example 2.2
# Example 2.1 assumed that the outcome (as well as the treatment variables)
# were directly affected by all other nodes in the past. In certain situations,
# domain specific knowledge may suggest otherwise leading to a Markov processes.
# This can be controlled using the k argument.
progressr::with_progress(
    psi2.2 <- lmtp_ipw(sim_t4, a, "Y", time_vary = tv, shift = d,
                        k = 0, folds = 2)
)
psi2.2

# Example 2.3
# Using the same data as examples 2.1 and 2.2.
# Now estimating the effect of a dynamic modified treatment policy.
a <- c("A_1", "A_2", "A_3", "A_4")
time_varying <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))
# our same shift function
shift <- function(data, trt) {
    (data[[trt]] - 1) * (data[[trt]] - 1 >= 1) + data[[trt]] * (data[[trt]] - 1 < 1)
}
# creating a dynamic mtp that applies the shift function
# but also depends on history and the current time
dynamic_mtp <- function(data, trt) {
    if (trt == "A_1") {
        # if its the first time point, follow the same mtp as before
        shift(data, trt)
    } else {
        # otherwise check if the time varying covariate equals 1
        ifelse(data[[sub("A", "L", trt)]] == 1,
                shift(data, trt), # if yes continue with the policy
                data[[trt]]) # otherwise do nothing
    }
}
psi2.3 <- lmtp_ipw(sim_t4, a, "Y", time_vary = time_varying,
                   k = 0, shift = dynamic_mtp, folds = 2)
psi2.3

# Example 2.4
# Using the same data as examples 2.1, 2.2, and 2.3, but now treating the exposure
# as an ordered categorical variable. To account for the exposure being a
# factor we just need to modify the shift function (and the original data)
# so as to respect this.
for (i in a) {
    sim_t4[[i]] <- factor(sim_t4[[i]], levels = 0:5, ordered = TRUE)
}
d <- function(data, trt) {
    out <- list()
    a <- data[[trt]]
    for (i in 1:length(a)) {
        if (as.character(a[i]) %in% c("0", "1")) {
            out[[i]] <- as.character(a[i])
        } else {
            out[[i]] <- as.numeric(as.character(a[i])) - 1
        }
    }
    factor(unlist(out), levels = 0:5, ordered = TRUE)
}
# Example 3.1
# Longitudinal setting, time-varying binary treatment, time-varying covariates
# and baseline covariates with no loss-to-follow-up. Interested in a traditional
# causal effect where treatment is set to 1 at all time points for all observations.
data("iptwExWide", package = "twang")
a <- paste0("tx", 1:3)
baseline <- c("gender", "age")
tv <- list(c("use0"), c("use1"), c("use2"))
progressr::with_progress({
  psi3.1 <-
  lmtp_ipw(iptwExWide, a, "outcome", baseline = baseline, time_vary = tv,
            shift = static_binary_on, folds = 2)
})

psi3.1

# Example 4.1
# Longitudinal setting, time-varying continuous treatment, time-varying covariates,
# binary outcome with right censoring. Interested in the mean population outcome under
# the observed exposures in a hypothetical population with no loss-to-follow-up.
head(sim_cens)
a <- c("A1", "A2")
tv <- list(c("L1"), c("L2"))
cens <- c("C1", "C2")
y <- "Y"
psi4.1 <- lmtp_ipw(sim_cens, a, y, time_vary = tv, cens = cens,
                     shift = NULL, folds = 2)

psi4.1

# Example 4.2
# Using the same data as example 4.1, but now interested in the causal effect of a
# treatment policy where exposure increased by 0.5 units at all time points. The
# true value under this intervention is about 0.88.
d <- function(data, x) data[[x]] + 0.5
psi4.2 <- lmtp_ipw(sim_cens, a, y, time_vary = tv,
                     cens = cens, shift = d, folds = 2)

psi4.2

# Example 5.1
# Time-to-event analysis with a binary time-invariant exposure. Interested in
# the effect of treatment being given to all observations on the cumulative
# incidence of our time-to-event outcome.
a <- "trt"
# for a survival problem, the outcome argument now takes a vector of outcomes
# if an observation experiences the event prior to the end of follow-up, all future
# outcome nodes should be set to 1 (i.e., last observation carried forward).
y <- paste0("Y.", 1:6)
cens <- paste0("C.", 0:5)
baseline <- c("W1", "W2")
progressr::with_progress(
  psi5.1 <- lmtp_ipw(sim_point_surv, a, y, baseline, cens = cens,
                     shift = static_binary_on, folds = 2)
)
psi5.1

lmtp_sdr  

LMTP Sequential Doubly Robust Estimator

Description

Sequentially doubly robust estimator for the effects of traditional causal effects and modified treatment policies for both point treatment and longitudinal data with binary, continuous, or time-to-event outcomes. Supports binary, categorical, and continuous exposures.

Usage

lmtp_sdr(
  data,    
  trt,     
  outcome, 
  baseline = NULL, 
  time_vary = NULL, 
  cens = NULL, 
  shift,   
  k = Inf,  
  outcome_type = c("binomial", "continuous"), 
  id = NULL, 
  bounds = NULL, 
  learners_outcome = NULL, 
  learners_trt = NULL, 
  folds = 10, 
  bound = 1e-05 
)

Arguments

data A data frame in wide format containing all necessary variables for the estimation problem.

trt A vector containing the column names of treatment variables ordered by time.

outcome The column name of the outcome variable. In the case of time-to-event analysis, a vector containing the columns names of intermediate outcome variables and the final outcome variable ordered by time. Only numeric values are allowed. If the outcome type is binary, data should be coded as 0 and 1.

baseline An optional vector of columns names of baseline covariates to be included for adjustment at every timepoint.
time_vary A list the same length as the number of time points of observation with the column names for new time-varying covariates introduced at each time point. The list should be ordered following the time ordering of the model.
cens An optional vector of column names of censoring indicators the same length as the number of time points of observation. If missingness in the outcome is present or if time-to-event outcome, must be provided.
shift A two argument function that specifies how treatment variables should be shifted. See examples for how to specify shift functions for continuous, binary, and categorical exposures.
k An integer specifying how previous time points should be used for estimation at the given time point. Default is \( \text{Inf} \), all time points.
outcome_type Outcome variable type (i.e., continuous, binomial).
id An optional column name containing cluster level identifiers.
bounds An optional vector of the bounds for continuous outcomes. If \( \text{NULL} \), the bounds will be taken as the minimum and maximum of the observed data. Should be left as \( \text{NULL} \) if the outcome type is binary.
learners_outcome An optional \texttt{sl3} learner stack for estimation of the outcome regression. If not specified, will default to using a generalized linear model.
learners_trt An optional \texttt{sl3} learner stack for estimation of the exposure mechanism. If not specified, will default to using a generalized linear model.
folds The number of folds to be used for cross-fitting. Minimum allowable number is two folds.
bound Determines that maximum and minimum values (scaled) predictions will be bounded by. The default is 1e-5, bounding predictions by 1e-5 and 0.9999.

Value

A list of class \texttt{lmtp} containing the following components:

- \texttt{estimator} The estimator used, in this case “SDR”.
- \texttt{theta} The estimated population LMTP effect.
- \texttt{standard_error} The estimated standard error of the LMTP effect.
- \texttt{low} Lower bound of the 95% confidence interval of the LMTP effect.
- \texttt{high} Upper bound of the 95% confidence interval of the LMTP effect.
- \texttt{eif} The estimated, un-centered, influence function of the estimate.
- \texttt{shift} The shift function specifying the treatment policy of interest.
- \texttt{outcome_reg} An n x Tau + 1 matrix of outcome regression predictions. The mean of the first column is used for calculating theta.
- \texttt{density_ratios} An n x Tau matrix of the estimated density ratios.
- \texttt{weights_m} A list the same length as \texttt{folds}, containing the Super Learner ensemble weights at each time-point for each fold for the outcome regression.
- \texttt{weights_r} A list the same length as \texttt{folds}, containing the Super Learner ensemble weights at each time-point for each fold for the propensity.
- \texttt{outcome_type} The outcome variable type.
Examples

```r
library(lmtp)
# Example 1.1
# Point treatment, continuous exposure, continuous outcome, no loss-to-follow-up
# Interested in the effect of a population wide decrease in A of 5 units
# The true value under this intervention is about 519.
set.seed(56)
n <- 1000
W <- rnorm(n, 10, 5)
A <- 23 + 5*W + rnorm(n)
Y <- 7.2*A + 3*W + rnorm(n)
ex1_dat <- data.frame(W, A, Y)
d <- function(data, x) data[[x]] - 5
psi1.1 <- lmtp_sdr(ex1_dat, "A", "Y", "W", shift = d,
                   outcome_type = "continuous", folds = 2)
psi1.1

# Example 1.2
# Point treatment, continuous exposure, continuous outcome, no loss-to-follow-up
# Interested in the effect of a modified treatment policy where A is decreased by 15
# units only among observations whose observed A was above 80.
# The true value under this intervention is about 513.
d <- function(data, x) (data[[x]] > 80)*(data[[x]] - 15) + (data[[x]] <= 80)*data[[x]]
psi1.2 <- lmtp_sdr(ex1_dat, "A", "Y", "W", shift = d,
                   outcome_type = "continuous", folds = 2)
psi1.2

# Example 2.1
# Longitudinal setting, time-varying continuous exposure bounded by 0,
# time-varying covariates, and a binary outcome with no loss-to-follow-up.
# Interested in the effect of a treatment policy where exposure decreases by
# one unit at every time point if an observations observed exposure is greater
# than or equal to 2. The true value under this intervention is about 0.305.
head(sim_t4)
# specifying treatment variables
a <- c("A_1", "A_2", "A_3", "A_4")
# specifying time varying covariates
tv <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))
# treatment policy function to be applied at all time points
d <- function(data, trt) {
  a <- data[[trt]]
  (a - 1) * (a - 1 >= 1) + a * (a - 1 < 1)
}
progressr::with_progress({
  psi2.1 <- lmtp_sdr(sim_t4, a, "Y", time_vary = tv, shift = d, folds = 2)
})
psi2.1

# Example 2.2
# Example 2.1 assumed that the outcome (as well as the treatment variables)
# were directly affected by all other nodes in the past. In certain situations,
lmtp_sdr

# domain specific knowledge may suggest otherwise leading to a Markov processes.
# This can be controlled using the k argument.
progressr::with_progress(
  psi2.2 <- lmtp_sdr(sim_t4, a, "Y", time_vary = tv, shift = d,
                     k = 0, folds = 2)
})
psi2.2

# Example 2.3
# Using the same data as examples 2.1 and 2.2.
# Now estimating the effect of a dynamic modified treatment policy.
a <- c("A_1", "A_2", "A_3", "A_4")
time_varying <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))
# our same shift function
shift <- function(data, trt) {
  (data[[trt]] - 1) * (data[[trt]] - 1 >= 1) + data[[trt]] * (data[[trt]] - 1 < 1)
}
# creating a dynamic mtp that applies the shift function
# but also depends on history and the current time
dynamic_mtp <- function(data, trt) {
  if (trt == "A_1") {
    # if its the first time point, follow the same mtp as before
    shift(data, trt)
  } else {
    # otherwise check if the time varying covariate equals 1
    ifelse(data[sub("A", "L", trt)] == 1,
            shift(data, trt), # if yes continue with the policy
            data[[trt]])) # otherwise do nothing
  }
} psi2.3 <- lmtp_sdr(sim_t4, a, "Y", time_vary = time_varying,
                     k = 0, shift = dynamic_mtp, folds = 2)
psi2.3

# Example 2.4
# Using the same data as examples 2.1, 2.2, and 2.3, but now treating the exposure
# as an ordered categorical variable. To account for the exposure being a
# factor we just need to modify the shift function (and the original data)
# so as to respect this.
for (i in a) {
  sim_t4[[i]] <- factor(sim_t4[[i]], levels = 0:5, ordered = TRUE)
}
d <- function(data, trt) {
  out <- list()
  a <- data[[trt]]
  for (i in 1:length(a)) {
    if (as.character(a[i]) %in% c("0", "1")) {
      out[[i]] <- as.character(a[i])
    } else {
      out[[i]] <- as.numeric(as.character(a[i])) - 1
    }
  }
  factor(unlist(out), levels = 0:5, ordered = TRUE)
progressr::with_progress(
    psi2.4 <- lntp_sdr(sim_t4, a, "Y", time_vary = tv, shift = d, k = 0, folds = 2)
))
psi2.4

# Example 3.1
# Longitudinal setting, time-varying binary treatment, time-varying covariates
# and baseline covariates with no loss-to-follow-up. Interested in a traditional
# causal effect where treatment is set to 1 at all time points for all observations.
data("iptwExWide", package = "twang")
a <- paste0("tx", 1:3)
baseline <- c("gender", "age")
tv <- list(c("use0"), c("use1"), c("use2"))
progressr::with_progress(
    psi3.1 <-
    lntp_sdr(iptwExWide, a, "outcome", baseline = baseline, time_vary = tv,
             shift = static_binary_on, outcome_type = "continuous",
             folds = 2)
))
psi3.1

# Example 4.1
# Longitudinal setting, time-varying continuous treatment, time-varying covariates,
# binary outcome with right censoring. Interested in the mean population outcome under
# the observed exposures in a hypothetical population with no loss-to-follow-up.
head(sim_cens)
a <- c("A1", "A2")
tv <- list(c("L1"), c("L2"))
cens <- c("C1", "C2")
y <- "Y"
psi4.1 <- lntp_sdr(sim_cens, a, y, time_vary = tv, cens = cens,
                    shift = NULL, folds = 2)
psi4.1

# Example 4.2
# Using the same data as example 4.1, but now interested in the causal effect of a
# treatment policy where exposure increased by 0.5 units at all time points. The
# true value under this intervention is about 0.88.
d <- function(data, x) data[[x]] + 0.5
psi4.2 <- lntp_sdr(sim_cens, a, y, time_vary = tv,
                    cens = cens, shift = d, folds = 2)
psi4.2

# Example 5.1
# Time-to-event analysis with a binary time-invariant exposure. Interested in
# the effect of treatment being given to all observations on the cumulative
# incidence of our time-to-event outcome.
a <- "trt"
# for a survival problem, the outcome argument now takes a vector of outcomes
# if an observation experiences the event prior to the end of follow-up, all future
# outcome nodes should be set to 1 (i.e., last observation carried forward).
y <- paste0("Y.", 1:6)
lmtp_sub

```r
cens <- paste0("C.", 0:5)
baseline <- c("W1", "W2")
progressr::with_progress(
  psi5.1 <- lmtp_sdr(sim_point_surv, a, y, baseline, cens = cens,
                      shift = static_binary_on, folds = 2)
)
psi5.1
```

---

**lmtp_sub**

**LMTP Substitution Estimator**

**Description**

G-computation estimator for the effects of traditional causal effects and modified treatment policies for both point treatment and longitudinal data with binary, continuous, or time-to-event outcomes. Supports binary, categorical, and continuous exposures.

**Usage**

```r
lmtp_sub(
  data,
  trt,
  outcome,
  baseline = NULL,
  time_vary = NULL,
  cens = NULL,
  shift,
  k = Inf,
  outcome_type = c("binomial", "continuous"),
  id = NULL,
  bounds = NULL,
  learners = NULL,
  folds = 10,
  bound = 1e-05
)
```

**Arguments**

- **data**: A data frame in wide format containing all necessary variables for the estimation problem.
- **trt**: A vector containing the column names of treatment variables ordered by time.
- **outcome**: The column name of the outcome variable. In the case of time-to-event analysis, a vector containing the columns names of intermediate outcome variables and the final outcome variable ordered by time. Only numeric values are allowed. If the outcome type is binary, data should be coded as 0 and 1.
baseline  An optional vector of columns names of baseline covariates to be included for adjustment at every timepoint.

time_vary  A list the same length as the number of time points of observation with the column names for new time-varying covariates introduced at each time point. The list should be ordered following the time ordering of the model.

cens  An optional vector of column names of censoring indicators the same length as the number of time points of observation. If missingness in the outcome is present or if time-to-event outcome, must be provided.

shift  A two argument function that specifies how treatment variables should be shifted. See examples for how to specify shift functions for continuous, binary, and categorical exposures.

k  An integer specifying how previous time points should be used for estimation at the given time point. Default is Inf, all time points.

outcome_type  Outcome variable type (i.e., continuous, binomial).

id  An optional column name containing cluster level identifiers.

bounds  An optional vector of the bounds for continuous outcomes. If NULL, the bounds will be taken as the minimum and maximum of the observed data. Should be left as NULL if the outcome type is binary.

learners  An optional sl3 learner stack for estimation of the outcome regression. If not specified, will default to using a generalized linear model.

folds  The number of folds to be used for cross-fitting. Minimum allowable number is two folds.

bound  Determines that maximum and minimum values (scaled) predictions will be bounded by. The default is 1e-5, bounding predictions by 1e-5 and 0.9999.

Value

A list of class mlmp containing the following components:

estimator  The estimator used, in this case "substitution".

theta  The estimated population LMTP effect.

standard_error  NA

low  NA

high  NA

shift  The shift function specifying the treatment policy of interest.

outcome_reg  An n x Tau + 1 matrix of outcome regression predictions. The mean of the first column is used for calculating theta.

weights_m  A list the same length as folds, containing the Super Learner ensemble weights at each time-point for each fold for the outcome regression.

outcome_type  The outcome variable type.
Examples

library(lmtp)

# Example 1.1
# Point treatment, continuous exposure, continuous outcome, no loss-to-follow-up
# Interested in the effect of a population wide decrease in A of 5 units
# The true value under this intervention is about 519.
set.seed(56)
n <- 1000
W <- rnorm(n, 10, 5)
A <- 23 + 5*W + rnorm(n)
Y <- 7.2*A + 3*W + rnorm(n)
ex1_dat <- data.frame(W, A, Y)
d <- function(data, x) data[[x]] - 5
psi1.1 <- lmtp_sub(ex1_dat, "A", "Y", "W", shift = d,
                   outcome_type = "continuous", folds = 2)
psi1.1

# Example 1.2
# Point treatment, continuous exposure, continuous outcome, no loss-to-follow-up
# Interested in the effect of a modified treatment policy where A is decreased by 15
# units only among observations whose observed A was above 80.
# The true value under this intervention is about 513.
d <- function(data, x) (data[[x]] > 80)*(data[[x]] - 15) + (data[[x]] <= 80)*data[[x]]
psi1.2 <- lmtp_sub(ex1_dat, "A", "Y", "W", shift = d,
                   outcome_type = "continuous", folds = 2)
psi1.2

# Example 2.1
# Longitudinal setting, time-varying continuous exposure bounded by 0,
# time-varying covariates, and a binary outcome with no loss-to-follow-up.
# Interested in the effect of a treatment policy where exposure decreases by
# one unit at every time point if an observations observed exposure is greater
# than or equal to 2. The true value under this intervention is about 0.305.
head(sim_t4)
# specifying treatment variables
a <- c("A_1", "A_2", "A_3", "A_4")
# specifying time varying covariates
tv <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))
# treatment policy function to be applied at all time points
d <- function(data, trt) {
a <- data[[trt]]
  (a - 1) * (a - 1 >= 1) + a * (a - 1 < 1)
}
progressr::with_progress({
  psi2.1 <- lmtp_sub(sim_t4, a, "Y", time_vary = tv, shift = d, folds = 2)
})
psi2.1

# Example 2.2
# Example 2.1 assumed that the outcome (as well as the treatment variables)
# were directly affected by all other nodes in the past. In certain situations,
# domain specific knowledge may suggest otherwise leading to a Markov processes.
# This can be controlled using the k argument.

```r
progressr::with_progress(
  psi2.2 <- lmtp_sub(sim_t4, a, "Y", time_vary = tv, shift = d,
                      k = 0, folds = 2)
)
psi2.2
```

**Example 2.3**

# Using the same data as examples 2.1 and 2.2.
# Now estimating the effect of a dynamic modified treatment policy.

```r
a <- c("A_1", "A_2", "A_3", "A_4")
time_varying <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))
# our same shift function
shift <- function(data, trt) {
  (data[[trt]] - 1) * (data[[trt]] - 1 >= 1) + data[[trt]] * (data[[trt]] - 1 < 1)
}
# creating a dynamic mtp that applies the shift function
# but also depends on history and the current time
dynamic_mtp <- function(data, trt) {
  if (trt == "A_1") {
    # if its the first time point, follow the same mtp as before
    shift(data, trt)
  } else {
    # otherwise check if the time varying covariate equals 1
    ifelse(data[["A", "L", trt]] == 1,
            shift(data, trt), # if yes continue with the policy
            data[[trt]]) # otherwise do nothing
  }
}
psi2.3 <- lmtp_sub(sim_t4, a, "Y", time_vary = time_varying,
                   k = 0, shift = dynamic_mtp, folds = 2)
psi2.3
```

**Example 2.4**

# Using the same data as examples 2.1, 2.2, and 2.3, but now treating the exposure
# as an ordered categorical variable. To account for the exposure being a
# factor we just need to modify the shift function (and the original data)
# so as to respect this.

```r
for (i in a) {
  sim_t4[[i]] <- factor(sim_t4[[i]], levels = 0:5, ordered = TRUE)
}
d <- function(data, trt) {
  out <- list()
  a <- data[[trt]]
  for (i in 1:length(a)) {
    if (as.character(a[i]) %in% c("0", "1")) {
      out[[i]] <- as.character(a[i])
    } else {
      out[[i]] <- as.numeric(as.character(a[i])) - 1
    }
  }
}
```
```r
factor(unlist(out), levels = 0:5, ordered = TRUE)
}
progressr::with_progress(
  psi2.4 <- lmtp_sub(sim_t4, a, "Y", time_vary = tv, shift = d, k = 0, folds = 2)
))
psi2.4

# Example 3.1
# Longitudinal setting, time-varying binary treatment, time-varying covariates
# and baseline covariates with no loss-to-follow-up. Interested in a traditional
# causal effect where treatment is set to 1 at all time points for all observations.
data("iptwExWide", package = "twang")
a <- paste0("tx", 1:3)
baseline <- c("gender", "age")
tv <- list(c("use0"), c("use1"), c("use2"))
progressr::with_progress(
  psi3.1 <-
    lmtp_sub(iptwExWide, a, "outcome", baseline = baseline, time_vary = tv,
             shift = static_binary_on, outcome_type = "continuous",
             folds = 2)
))
psi3.1

# Example 4.1
# Longitudinal setting, time-varying continuous treatment, time-varying covariates,
# binary outcome with right censoring. Interested in the mean population outcome under
# the observed exposures in a hypothetical population with no loss-to-follow-up.
head(sim_cens)
a <- c("A1", "A2")
tv <- list(c("L1"), c("L2"))
cens <- c("C1", "C2")
y <- "Y"
psi4.1 <- lmtp_sub(sim_cens, a, y, time_vary = tv,
                   cens = cens, shift = NULL, folds = 2)
psi4.1

# Example 4.2
# Using the same data as example 4.1, but now interested in the causal effect of a
# treatment policy where exposure increased by 0.5 units at all time points. The
# true value under this intervention is about 0.88.
d <- function(data, x) data[[x]] + 0.5
psi4.2 <- lmtp_sub(sim_cens, a, y, time_vary = tv, cens = cens, shift = d, folds = 2)
psi4.2

# Example 5.1
# Time-to-event analysis with a binary time-invariant exposure. Interested in
# the effect of treatment being given to all observations on the cumulative
# incidence of our time-to-event outcome.
a <- "trt"
# for a survival problem, the outcome argument now takes a vector of outcomes
# if an observation experiences the event prior to the end of follow-up, all future
# outcome nodes should be set to 1 (i.e., last observation carried forward).
y <- paste0("Y.", 1:6)
```
cens <- paste0("C.", 0:5)
baseline <- c("W1", "W2")
progressr::with_progress({
  psi5.1 <- lmtp_sub(sim_point_surv, a, y, baseline, cens = cens,
                     shift = static_binary_on, folds = 2)
})
psi5.1

lmtp_tmle

**LMTP** Targeted Maximum Likelihood Estimator

**Description**

Targeted maximum likelihood estimator for the effects of traditional causal effects and modified treatment policies for both point treatment and longitudinal data with binary, continuous, or time-to-event outcomes. Supports binary, categorical, and continuous exposures.

**Usage**

```r
lmtp_tmle(  
data,  
trt,  
outcome,  
baseline = NULL,  
time_vary = NULL,  
cens = NULL,  
shift,  
k = Inf,  
outcome_type = c("binomial", "continuous"),  
id = NULL,  
bounds = NULL,  
learners_outcome = NULL,  
learners_trt = NULL,  
folds = 10,  
bound = 1e-05)
```

**Arguments**

- **data**  
  A data frame in wide format containing all necessary variables for the estimation problem.
- **trt**  
  A vector containing the column names of treatment variables ordered by time.
- **outcome**  
  The column name of the outcome variable. In the case of time-to-event analysis, a vector containing the column names of intermediate outcome variables and the final outcome variable ordered by time. Only numeric values are allowed. If the outcome type is binary, data should be coded as 0 and 1.
baseline  An optional vector of columns names of baseline covariates to be included for adjustment at every timepoint.

time_vary  A list the same length as the number of time points of observation with the column names for new time-varying covariates introduced at each time point. The list should be ordered following the time ordering of the model.

cens  An optional vector of column names of censoring indicators the same length as the number of time points of observation. If missingness in the outcome is present or if time-to-event outcome, must be provided.

shift  A two argument function that specifies how treatment variables should be shifted. See examples for how to specify shift functions for continuous, binary, and categorical exposures.

k  An integer specifying how previous time points should be used for estimation at the given time point. Default is Inf, all time points.

outcome_type  Outcome variable type (i.e., continuous, binomial).

id  An optional column name containing cluster level identifiers.

bounds  An optional vector of the bounds for continuous outcomes. If NULL, the bounds will be taken as the minimum and maximum of the observed data. Should be left as NULL if the outcome type is binary.

learners_outcome  An optional s13 learner stack for estimation of the outcome regression. If not specified, will default to using a generalized linear model.

learners_trt  An optional s13 learner stack for estimation of the exposure mechanism. If not specified, will default to using a generalized linear model.

folds  The number of folds to be used for cross-fitting. Minimum allowable number is two folds.

bound  Determines that maximum and minimum values (scaled) predictions will be bounded by. The default is 1e-5, bounding predictions by 1e-5 and 0.9999.

Value

A list of class lmtp containing the following components:

estimator  The estimator used, in this case "TMLE".

theta  The estimated population LMTP effect.

standard_error  The estimated standard error of the LMTP effect.

low  Lower bound of the 95% confidence interval of the LMTP effect.

high  Upper bound of the 95% confidence interval of the LMTP effect.

eif  The estimated, un-centered, influence function of the estimate.

shift  The shift function specifying the treatment policy of interest.

outcome_reg  An n x Tau + 1 matrix of outcome regression predictions. The mean of the first column is used for calculating theta.

density_ratios  An n x Tau matrix of the estimated density ratios.
weights_m  A list the same length as folds, containing the Super Learner ensemble weights at each time-point for each fold for the outcome regression.

weights_r  A list the same length as folds, containing the Super Learner ensemble weights at each time-point for each fold for the propensity.

outcome_type  The outcome variable type.

Examples

library(lmtp)

# Example 1.1
# Point treatment, continuous exposure, continuous outcome, no loss-to-follow-up
# Interested in the effect of a population wide decrease in A of 5 units
# The true value under this intervention is about 519.
set.seed(56)
n <- 1000
W <- rnorm(n, 10, 5)
A <- 23 + 5*W + rnorm(n)
Y <- 7.2*A + 3*W + rnorm(n)
ex1_dat <- data.frame(W, A, Y)
d <- function(data, x) data[[x]] - 5
psi1.1 <- lmtp_tmle(ex1_dat, "A", "Y", "W", shift = d,
                   outcome_type = "continuous", folds = 2)
psi1.1

# Example 1.2
# Point treatment, continuous exposure, continuous outcome, no loss-to-follow-up
# Interested in the effect of a modified treatment policy where A is decreased by 15 units only among observations whose observed A was above 80.
# The true value under this intervention is about 513.
d <- function(data, x) (data[[x]] > 80)*(data[[x]] - 15) + (data[[x]] <= 80)*data[[x]]
psi1.2 <- lmtp_tmle(ex1_dat, "A", "Y", "W", shift = d,
                   outcome_type = "continuous", folds = 2)
psi1.2

# Example 2.1
# Longitudinal setting, time-varying continuous exposure bounded by 0,
# time-varying covariates, and a binary outcome with no loss-to-follow-up.
# Interested in the effect of a treatment policy where exposure decreases by one unit at every time point if an observation's observed exposure is greater than or equal to 2. The true value under this intervention is about 0.305.
head(sim_t4)
# specifying treatment variables
a <- c("A_1", "A_2", "A_3", "A_4")
# specifying time varying covariates
tv <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4"))
# treatment policy function to be applied at all time points
d <- function(data, trt) {
a <- data[[trt]]
(a - 1) * (a - 1 >= 1) + a * (a - 1 < 1)
}


# Example 2.2
# Example 2.1 assumed that the outcome (as well as the treatment variables) 
# were directly affected by all other nodes in the past. In certain situations, 
# domain specific knowledge may suggest otherwise leading to a Markov processes. 
# This can be controlled using the k argument.

```r
progressr::with_progress(
    psi2.2 <- lmtp_tmle(sim_t4, a, "Y", time_vary = tv, shift = d, 
                       k = 0, folds = 2)
))
psi2.2
```

# Example 2.3
# Using the same data as examples 2.1 and 2.2.
# Now estimating the effect of a dynamic modified treatment policy.

```r
a <- c("A_1", "A_2", "A_3", "A_4") 
time_varying <- list(c("L_1"), c("L_2"), c("L_3"), c("L_4")) 
shift <- function(data, trt) {

    (data[[trt]] - 1) * (data[[trt]] - 1 >= 1) + data[[trt]] * (data[[trt]] - 1 < 1)
}

dynamic_mtp <- function(data, trt) {

    if (trt == "A_1") {
        shift(data, trt) 
    } else {

        ifelse(data[[sub("A", "L", trt)]] == 1, 
            shift(data, trt), 
            # if yes continue with the policy 
            data[[trt]]) 
    }

}

psi2.3 <- lmtp_tmle(sim_t4, a, "Y", time_vary = time_varying, 
                    k = 0, shift = dynamic_mtp, folds = 2)
psi2.3
```

# Example 2.4
# Using the same data as examples 2.1, 2.2, and 2.3, but now treating the exposure 
# as an ordered categorical variable. To account for the exposure being a 
# factor we just need to modify the shift function (and the original data) 
# so as to respect this.

```r
for (i in a) {

    sim_t4[[i]] <- factor(sim_t4[[i]], levels = 0:5, ordered = TRUE)
}
d <- function(data, trt) {

    out <- list() 
    a <- data[[trt]]
    ```
for (i in 1:length(a)) {
    if (as.character(a[i]) %in% c("0", "1")) {
        out[i] <- as.character(a[i])
    } else {
        out[i] <- as.numeric(as.character(a[i])) - 1
    }
}
factor(unlist(out), levels = 0:5, ordered = TRUE)
}

progressr::with_progress(
    psi2.4 <- lmtp_tmle(sim_t4, a, "Y", time_vary = tv, shift = d, k = 0, folds = 2)
))
psi2.4

# Example 3.1
# Longitudinal setting, time-varying binary treatment, time-varying covariates
# and baseline covariates with no loss-to-follow-up. Interested in a traditional
# causal effect where treatment is set to 1 at all time points for all observations.
data("iptwExWide", package = "twang")
a <- paste0("tx", 1:3)
baseline <- c("gender", "age")
tv <- list(c("use0"), c("use1"), c("use2"))
progressr::with_progress(
    psi3.1 <-
        lmtp_tmle(iptwExWide, a, "outcome", baseline = baseline, time_vary = tv,
            shift = static_binary_on, outcome_type = "continuous",
            folds = 2)
))
psi3.1

# Example 4.1
# Longitudinal setting, time-varying continuous treatment, time-varying covariates,
# binary outcome with right censoring. Interested in the mean population outcome under
# the observed exposures in a hypothetical population with no loss-to-follow-up.
head(sim_cens)
a <- c("A1", "A2")
tv <- list(c("L1"), c("L2"))
cens <- c("C1", "C2")
y <- "Y"
psi4.1 <- lmtp_tmle(sim_cens, a, y, time_vary = tv, cens = cens, shift = NULL, folds = 2)
psi4.1

# Example 4.2
# Using the same data as example 4.1, but now interested in the causal effect of a
# treatment policy where exposure increased by 0.5 units at all time points. The
# true value under this intervention is about 0.88.
d <- function(data, x) data[[x]] + 0.5
psi4.2 <- lmtp_tmle(sim_cens, a, y, time_vary = tv,
            cens = cens, shift = d, folds = 2)
psi4.2

# Example 5.1
# Time-to-event analysis with a binary time-invariant exposure. Interested in
# the effect of treatment being given to all observations on the cumulative incidence of our time-to-event outcome.
a <- "trt"
# for a survival problem, the outcome argument now takes a vector of outcomes
# if an observation experiences the event prior to the end of follow-up, all future outcome nodes should be set to 1 (i.e., last observation carried forward).
y <- paste0("Y.", 1:6)
cens <- paste0("C.", 0:5)
baseline <- c("W1", "W2")
progressr::with_progress({
  psi5.1 <- lmtp_tmle(sim_point_surv, a, y, baseline, cens = cens,
                     shift = static_binary_on, folds = 2)
})
psi5.1

---

**sim_cens**  
*Simulated Longitudinal Data With Censoring*

**Description**

A dataset with a binary outcome, two time varying treatment nodes, two time varying covariates, and two censoring indicators.

**Usage**

`sim_cens`

**Format**

A data frame with 1000 rows and 10 variables:

**L1** Time varying covariate time 1  
**A1** Treatment node at time 1, effected by L_1  
**C1** Censoring indicator that the observation is observed after time 1  
**L2** Time varying covariate at time 2, effected by L_1 and A_1  
**A2** Treatment node at time 2, effected by L_2 and A_1  
**C2** Censoring indicator that the observation is observed after time 2  
**Y** Binary outcome at time 3, effected by L_2 and A_2
Simulated Point-treatment Survival Data

Description

A dataset with a time-to-event outcome, two baseline nodes, a binary point treatment, six past-time outcome nodes, and six censoring indicators.

Usage

`sim_point_surv`

Format

A data frame with 2000 rows and 16 variables:

- **W1** Binary baseline variable.
- **W2** Categorical baseline variable.
- **trt** Binary treatment variable.
- **Y.0** Outcome node at time 0.
- **C.0** Censoring indicator that the observation is observed future time points.
- **Y.1** Outcome node at time 1.
- **C.1** Censoring indicator that the observation is observed future time points.
- **Y.2** Outcome node at time 2.
- **C.2** Censoring indicator that the observation is observed future time points.
- **Y.3** Outcome node at time 3.
- **C.3** Censoring indicator that the observation is observed future time points.
- **Y.4** Outcome node at time 4.
- **C.4** Censoring indicator that the observation is observed future time points.
- **Y.5** Outcome node at time 5.
- **C.5** Censoring indicator that the observation is observed future time points.
- **Y.6** Final outcome node.
**Simulated Longitudinal Data**

**Description**
A dataset with a binary outcome, four time varying treatment nodes, and four time varying covariates.

**Usage**
sim_t4

**Format**
A data frame with 5000 rows and 10 variables:

- **ID**: observation ID
- **L_1**: Time varying covariate time 1
- **A_1**: Treatment node at time 1, effected by L_1
- **L_2**: Time varying covariate time 1, effected by L_1 and A_1
- **A_2**: Treatment node at time 2, effected by L_2 and A_1
- **L_3**: Time varying covariate time 1, effected by L_2 and A_2
- **A_3**: Treatment node at time 3, effected by L_3 and A_2
- **L_4**: Time varying covariate time 1, effected by L_3 and A_3
- **A_4**: Treatment node at time 3, effected by L_4 and A_3
- **Y**: Binary outcome at time 5, effected by L_4 and A_4

**Turn All Treatment Nodes Off**

**Description**
A pre-packaged shift function for use with provided estimators when the exposure is binary. Used to estimate the population intervention effect when all treatment variables are set to 0.

**Usage**
static_binary_off(data, trt)

**Arguments**

- **data**: A data frame containing the treatment variables.
- **trt**: The name of the current treatment variable.
Value

A dataframe with all treatment nodes set to 0.

See Also

lmtp_tmle(), lmtp_sdr(), lmtp_sub(), lmtp_ipw()

Examples

data("iptwExWide", package = "twang")
a <- paste0("tx", 1:3)
baseline <- c("gender", "age")
tv <- list(c("use0"), c("use1"), c("use2"))
lmtp_sdr(iptwExWide, a, "outcome", baseline = baseline, time_vary = tv,
         shift = static_binary_off, outcome_type = "continuous", folds = 2)

Description

A pre-packaged shift function for use with provided estimators when the exposure is binary. Used to estimate the population intervention effect when all treatment variables are set to 1.

Usage

static_binary_on(data, trt)

Arguments

data A dataframe containing the treatment variables.

trt The name of the current treatment variable.

Value

A dataframe with all treatment nodes set to 1.

See Also

lmtp_tmle(), lmtp_sdr(), lmtp_sub(), lmtp_ipw()
Examples

data("iptwExWide", package = "twang")
a <- paste0("tx", 1:3)
baseline <- c("gender", "age")
tv <- list(c("use0"), c("use1"), c("use2"))
lmtp_sdr(iptwExWide, a, "outcome", baseline = baseline, time_vary = tv,
        shift = static_binary_on, outcome_type = "continuous", folds = 2)

tidy.lmtp       Tidy a(n) lmtp object

Description

Tidy a(n) lmtp object

Usage

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

Arguments

x    A lmtp object produced by a call to \code{lmtp::lmtp_tmle()}, \code{lmtp::lmtp_sdr()}, \code{lmtp::lmtp_sub()}, or \code{lmtp::lmtp_ipw}.

...   Unused, included for generic consistency only.

Examples

a <- c("A1", "A2")
nodes <- list(c("L1"), c("L2"))
cens  <- c("C1", "C2")
y <- "y"
fit   <- lmtp_tmle(sim_cens, a, y, time_vary = nodes, cens = cens, shift = NULL, folds = 2)
tidy(fit)
Index

* datasets
  sim_cens, 25  
  sim_point_surv, 26  
  sim_t4, 27

create_node_list, 2

event_locf, 3

lmtp::lmtp_ipw(), 29  
lmtp::lmtp_sdr(), 29  
lmtp::lmtp_sub(), 29  
lmtp::lmtp_tmle(), 29  
lmtp_contrast, 4  
lmtp_ipw, 5  
lmtp_ipw(), 28  
lmtp_sdr, 10  
lmtp_sdr(), 28  
lmtp_sub, 15  
lmtp_sub(), 28  
lmtp_tmle, 20  
lmtp_tmle(), 28

sim_cens, 25  
sim_point_surv, 26  
sim_t4, 27  
static_binary_off, 27  
static_binary_on, 28

tidy.lmtp, 29