Package: tters (via r-universe)

July 2, 2026

Title Sequential Target Trial Emulation Data Expansion (Rust + Polars Backend)

Version 0.1.1

Description Fast, verified data-expansion stage for sequential target trial emulation, backed by a Rust and Polars engine via the 'extendr' crate. Reproduces, bit-for-bit, the expansion output of the 'TrialEmulation' R package. The heavy lifting lives in the 'tte-expand' Rust core crate; this package is a thin binding layer.

License Apache License (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.2)

Imports bit64, methods, utils

Suggests arrow, data.table, testthat (>= 3.0.0), TrialEmulation (>= 0.0.4.11)

Config/testthat/edition 3

Config/rextendr/version 0.5.0

SystemRequirements Cargo (Rust's package manager), rustc >= 1.95.0

URL https://github.com/oldschoolcool2/rust-tte

BugReports https://github.com/oldschoolcool2/rust-tte/issues

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libclang-dev

Repository https://oldschoolcool2.r-universe.dev

Date/Publication 2026-07-02 21:59:54 UTC

RemoteUrl https://github.com/oldschoolcool2/rust-tte

RemoteRef main

RemoteSha 3edab88dd16d61b1fee6b948e8ad513bdb48d7d1

RemoteSubdir bindings/tters

Contents

`expand_df` *Expand an in-memory cohort* `data.frame` *into the sequential target-trial layout and return the result as a* `data.frame` *— the frame-in/frame-out analogue of* `expand_parquet()`*, with no intermediate Parquet.*

Description

The cohort arrives as an R `data.frame` (a `list` of equal-length columns); columns are marshalled dtype-exactly into a Polars frame (R `integer` -> `Int32`, `double` -> `Float64`, `bit64::integer64` -> `Int64`), expanded by the verified core, and the six structural columns are marshalled back to an R `data.frame`. A 64-bit integer column (an `integer64`, e.g. a large `id`) round-trips exactly via a pure-safe bit reinterpret (no precision loss).

Usage

```
expand_df(
  cohort,
  id_col,
  period_col,
  treatment_col,
  eligible_col,
  outcome_col,
  first_period,
  last_period,
```

```
  estimand
)
```

Arguments

`cohort`	An R `data.frame` of long person-time rows.
`id_col`, `period_col`, `treatment_col`	Column names in `cohort`.
`eligible_col`, `outcome_col`	Eligibility / outcome column names.
`first_period`, `last_period`	Inclusive integer bounds on `trial_period`.
`estimand`	`"ITT"` or `"PP"`. Case-insensitive.

Value

A `data.frame` with the six structural columns (an `integer64` input column is returned as `integer64`). Errors in the core engine surface as R errors.

---

expand_parquet *Expand a prepared person-time Parquet dataset into the sequential target-trial layout and write the result to* `output_path`.

Description

This is a thin FFI shim. All dtype-exact, deterministic Polars work lives in the `tte_expand` core crate (which is `#![forbid(unsafe_code)]`). The binding crate cannot forbid unsafe because the extendr macros emit the FFI registrar. Every `tte_expand::ExpandError` is mapped to an R error condition.

Usage

```
expand_parquet(
  input_path,
  output_path,
  id_col,
  period_col,
  treatment_col,
  eligible_col,
  outcome_col,
  first_period,
  last_period,
  estimand
)
```

Arguments

input_path	Path to the input Parquet file.
output_path	Path where the expanded Parquet is written.
id_col, period_col, treatment_col	Column names in the input.
eligible_col, outcome_col	Eligibility / outcome column names (TrialEmulation defaults are "eligible" / "outcome").
first_period, last_period	Inclusive integer bounds on trial_period.
estimand	"ITT" (no artificial censoring) or "PP" (per-protocol, censor each trial at the first treatment deviation). Case-insensitive.

Value

NULL, invisibly; the expansion is written to output_path. Errors in the core engine surface as R errors.

Examples

```
## Not run:
expand_parquet(
  "input.parquet", "expanded.parquet",
  "id", "period", "treatment", "eligible", "outcome",
  0L, .Machine$integer.max, "ITT"
)

## End(Not run)
```

---

expand_trial *Expand a target-trial person-time dataset (ergonomic wrapper)*

Description

User-facing wrapper around the extendr-generated expand_parquet() that validates inputs and uses sensible defaults. The heavy lifting happens in the Rust core crate.

Usage

```
expand_trial(
  input_path,
  output_path,
  id_col = "id",
  period_col = "period",
  treatment_col = "treatment",
  eligible_col = "eligible",
```

```
  outcome_col = "outcome",
  first_period = 0L,
  last_period = .Machine$integer.max,
  estimand = "ITT"
)
```

Arguments

`input_path`	Path to an existing input Parquet file.
`output_path`	Path to write the expanded Parquet file.
`id_col`, `period_col`, `treatment_col`, `eligible_col`, `outcome_col`	Column names. Defaults match the TrialEmulation conventions.
`first_period`, `last_period`	Inclusive integer period bounds.
`estimand`	`"ITT"` (intention-to-treat, no artificial censoring) or `"PP"` (per-protocol, censor each trial at the first treatment deviation).

Value

`output_path`, invisibly.

Examples

```
## Not run:
expand_trial("input.parquet", "expanded.parquet", estimand = "PP")

## End(Not run)
```

`expand_trial_df` *Expand a target-trial cohort data.frame in memory (ergonomic wrapper)*

Description

Frame-in / frame-out analogue of `expand_trial()`: takes an in-memory cohort `data.frame` and returns the expanded trial frame as a `data.frame`, with no intermediate Parquet. Wraps the extendr-generated `expand_df()`.

Usage

```
expand_trial_df(
  cohort,
  id_col = "id",
  period_col = "period",
  treatment_col = "treatment",
  eligible_col = "eligible",
  outcome_col = "outcome",
```

```
  first_period = 0L,
  last_period = .Machine$integer.max,
  estimand = "ITT"
)
```

Arguments

cohort	A data.frame (or tibble / data.table / Arrow Table) of long person-time rows. Coerced with as.data.frame(), which preserves an integer64 column's class.
id_col, period_col, treatment_col, eligible_col, outcome_col	Column names. Defaults match the TrialEmulation conventions.
first_period, last_period	Inclusive integer period bounds.
estimand	"ITT" (intention-to-treat, no artificial censoring) or "PP" (per-protocol, censor each trial at the first treatment deviation).

Details

Column dtypes are preserved exactly: R integer <-> Polars Int32, double <-> Float64, and bit64::integer64 <-> Int64. A 64-bit integer column (e.g. a large id) round-trips exactly as integer64 with no precision loss above 2^53 (a pure-safe bit reinterpret, not a numeric cast).

Value

A data.frame with the six structural columns (id, trial_period, followup_time, assigned_treatment, treatment, outcome); an integer64 input column is returned as integer64.

See Also

expand_trial() for the Parquet-path equivalent.

Examples

```
## Not run:
cohort <- arrow::read_parquet("input.parquet")
expanded <- expand_trial_df(cohort, estimand = "PP")

## End(Not run)
```

---

expand_trial_weighted *Expand a dataset and attach pre-computed inverse-probability weights*

Description

User-facing wrapper around the extendr-generated expand_weighted_parquet(). It expands input_path under estimand, joins the per-(id, period) factor table at factors_path (id, period, weight_factor), and writes the six structural columns plus the cumulative-product weight. Weight *values* are produced upstream in R; the engine reproduces only their deterministic accumulation.

Usage

```
expand_trial_weighted(
  input_path,
  factors_path,
  output_path,
  id_col = "id",
  period_col = "period",
  treatment_col = "treatment",
  eligible_col = "eligible",
  outcome_col = "outcome",
  first_period = 0L,
  last_period = .Machine$integer.max,
  estimand = "PP"
)
```

Arguments

`input_path`	Path to an existing input Parquet file.
`factors_path`	Path to the per-`(id, period)` factor Parquet file.
`output_path`	Path to write the weighted Parquet file.
`id_col`, `period_col`, `treatment_col`, `eligible_col`, `outcome_col`	Column names. Defaults match the TrialEmulation conventions.
`first_period`, `last_period`	Inclusive integer period bounds.
`estimand`	`"ITT"` or `"PP"`; selects the weight model upstream, but the application arithmetic is identical for both.

Value

`output_path`, invisibly.

Examples

```
## Not run:
expand_trial_weighted(
  "input.parquet", "factors.parquet", "weighted.parquet", estimand = "PP"
)

## End(Not run)
```

`expand_trial_weighted_df` *Expand a cohort data.frame and attach pre-computed weights, in memory (wrapper)*

Description

Frame-in / frame-out analogue of `expand_trial_weighted()`: takes an in-memory cohort `data.frame` and a pre-computed factor `data.frame` (`id`, `period`, `weight_factor`), and returns the weighted, expanded frame as a `data.frame`. Wraps the extendr-generated `expand_weighted_df()`. A `bit64::integer64` `id` in either frame round-trips exactly.

Usage

```
expand_trial_weighted_df(
  cohort,
  factors,
  id_col = "id",
  period_col = "period",
  treatment_col = "treatment",
  eligible_col = "eligible",
  outcome_col = "outcome",
  first_period = 0L,
  last_period = .Machine$integer.max,
  estimand = "PP"
)
```

Arguments

`cohort`	A `data.frame` (or tibble / `data.table` / Arrow `Table`) of long person-time rows. Coerced with `as.data.frame()`.
`factors`	A `data.frame` with columns `id`, `period`, `weight_factor`. Coerced with `as.data.frame()`.
`id_col`, `period_col`, `treatment_col`, `eligible_col`, `outcome_col`	Column names. Defaults match the TrialEmulation conventions.
`first_period`, `last_period`	Inclusive integer period bounds.
`estimand`	`"ITT"` or `"PP"`; selects the weight model upstream, but the application arithmetic is identical for both.

Value

A `data.frame` with the six structural columns plus `weight`.

See Also

`expand_trial_weighted()` for the Parquet-path equivalent.

Examples

```
## Not run:
weighted <- expand_trial_weighted_df(cohort, factors, estimand = "PP")

## End(Not run)
```

`expand_trial_weighted_fitted` *Fit IPW weights and expand a cohort into a weighted trial frame (ergonomic wrapper)*

Description

User-facing wrapper around the extendr-generated `expand_weighted_fitted_parquet()`. It takes a raw person-time cohort straight to a weighted, expanded trial frame in one call — fitting the switching and/or IPCW models in Rust (no pre-computed factor table), expanding under `estimand`, and accumulating the fitted factor into the cumulative `weight`. The six structural columns are bit-exact; `weight` matches the Oracle within the staged ~1e-6 tolerance. Robust/sandwich variance and the marginal structural model stay in R.

Usage

```
expand_trial_weighted_fitted(
  input_path,
  output_path,
  id_col = "id",
  period_col = "period",
  treatment_col = "treatment",
  eligible_col = "eligible",
  outcome_col = "outcome",
  first_period = 0L,
  last_period = .Machine$integer.max,
  estimand = "PP",
  switch_numerator = NULL,
  switch_denominator = NULL,
  censor_col = NULL,
  censor_numerator = NULL,
  censor_denominator = NULL,
  pool_censor = "none"
)
```

Arguments

`input_path`	Path to an existing input Parquet cohort (long person-time).
`output_path`	Path to write the weighted, expanded Parquet.
`id_col`, `period_col`, `treatment_col`, `eligible_col`, `outcome_col`	Column names. Defaults match the TrialEmulation conventions.
`first_period`, `last_period`	Inclusive integer period bounds.
`estimand`	`"ITT"` or `"PP"`. Per-protocol runs the artificial-censoring state machine and the switching models; intention-to-treat skips both.

`switch_numerator`, `switch_denominator`
: Character vectors of covariate column names for the switching numerator (stabiliser) / denominator models, or `NULL` (the default) to omit switching weights.

`censor_col`
: Name of the `{0,1}` censoring-indicator column; the modelled response is `1 - censor_col`. `NULL` (the default) omits IPCW weights.

`censor_numerator`, `censor_denominator`
: Character vectors of covariate column names for the IPCW numerator / denominator models.

`pool_censor`
: How the IPCW models are pooled across the previous-treatment strata: `"none"`, `"numerator"`, or `"both"`.

Details

Model presence follows the same rule as `fit_trial_weights()`: a switching model is fitted when either `switch_*` covariate vector is non-`NULL`; an IPCW model is fitted when `censor_col` is non-`NULL`.

Value

`output_path`, invisibly.

See Also

`fit_trial_weights()` to write only the `(id, period, weight_factor)` factor table.

Examples

```
## Not run:
# Per-protocol switch + IPCW censoring, raw cohort to weighted frame in one call:
expand_trial_weighted_fitted(
  "cohort.parquet", "weighted.parquet", estimand = "PP",
  switch_numerator = "x2", switch_denominator = c("x2", "x1"),
  censor_col = "censored",
  censor_numerator = "x2", censor_denominator = c("x2", "x1"),
  pool_censor = "none"
)

## End(Not run)
```

`expand_trial_weighted_fitted_df`
: *Fit IPW weights and expand a cohort data.frame in one call, in memory (wrapper)*

Description

Frame-in / frame-out analogue of `expand_trial_weighted_fitted()`: takes a raw cohort `data.frame` straight to a weighted, expanded `data.frame` in one call — fitting the switching and/or IPCW models in Rust (no pre-computed factor table), expanding under `estimand`, and accumulating the fitted factor into the cumulative `weight`. The six structural columns are bit-exact; `weight` matches the Oracle within the staged ~1e-6 tolerance. Wraps the extendr-generated `expand_weighted_fitted_df()`. A `bit64::integer64` `id` round-trips exactly.

Usage

```
expand_trial_weighted_fitted_df(
  cohort,
  id_col = "id",
  period_col = "period",
  treatment_col = "treatment",
  eligible_col = "eligible",
  outcome_col = "outcome",
  first_period = 0L,
  last_period = .Machine$integer.max,
  estimand = "PP",
  switch_numerator = NULL,
  switch_denominator = NULL,
  censor_col = NULL,
  censor_numerator = NULL,
  censor_denominator = NULL,
  pool_censor = "none"
)
```

Arguments

`cohort`	A `data.frame` (or tibble / `data.table` / Arrow `Table`) of long person-time rows. Coerced with `as.data.frame()`.
`id_col`, `period_col`, `treatment_col`, `eligible_col`, `outcome_col`	Column names. Defaults match the TrialEmulation conventions.
`first_period`, `last_period`	Inclusive integer period bounds.
`estimand`	`"ITT"` or `"PP"`.
`switch_numerator`, `switch_denominator`	Character vectors of covariate column names for the switching numerator / denominator models, or `NULL` to omit switching weights.
`censor_col`	Name of the `{0,1}` censoring-indicator column; the modelled response is `1 - censor_col`. `NULL` omits IPCW weights.
`censor_numerator`, `censor_denominator`	Character vectors of covariate column names for the IPCW numerator / denominator models.
`pool_censor`	How the IPCW models are pooled across the previous-treatment strata: `"none"`, `"numerator"`, or `"both"`.

Details

Model presence follows the same rule as `fit_trial_weights_df()`.

Value

A `data.frame` with the six structural columns plus `weight`.

See Also

`expand_trial_weighted_fitted()` for the Parquet-path equivalent; `fit_trial_weights_df()` to return only the factor table.

Examples

```
## Not run:
weighted <- expand_trial_weighted_fitted_df(
  cohort, estimand = "PP",
  switch_numerator = "x2", switch_denominator = c("x2", "x1"),
  censor_col = "censored",
  censor_numerator = "x2", censor_denominator = c("x2", "x1"),
  pool_censor = "none"
)

## End(Not run)
```

---

expand_trials_tters *Expand a sequence of target trials with the Rust + Polars engine*

Description

A drop-in replacement for `TrialEmulation::expand_trials()` that runs the expensive expansion in Rust (`tters`) instead of R, then stores the result through the `trial_sequence`'s registered `te_datastore`. The produced frame is byte-equivalent to the default path (structural columns bit-exact, `weight` to within machine precision), so the downstream — `load_expanded_data()`, `sample_controls()`, `fit_msm()` — behaves identically.

Usage

```
expand_trials_tters(object, fallback = TRUE, quiet = FALSE)
```

Arguments

`object`	A configured `trial_sequence` (ITT or PP). The AT estimand is not yet supported and falls back to R.
`fallback`	If `TRUE` (default), any failure of the Rust path (including an unsupported estimand or a missing toolchain) falls back to `TrialEmulation::expand_trials()` with a message. If `FALSE`, the error is raised.
`quiet`	If `TRUE`, suppress the fallback message.

Details

Estimation stays entirely in R. Weight *models* are fit by `calculate_weights()`; this function reads that per-period `wt` verbatim and Rust performs only the deterministic expansion and weight accumulation.

Set up the `trial_sequence` exactly as for `TrialEmulation::expand_trials()` (`set_data()` -> optional weight models + `calculate_weights()` -> `set_outcome_model()` -> `set_expansion_options()`), then call this instead of `expand_trials()`. The registered output may be `save_to_tters()` or any other `te_datastore` (e.g. `save_to_datatable()`); the speedup comes from the Rust expansion, not the store.

Value

The updated `trial_sequence`, with its `@expansion@datastore` populated — the same object type `TrialEmulation::expand_trials()` returns.

See Also

`save_to_tters()`; `TrialEmulation::expand_trials()`.

Examples

```
## Not run:
library(TrialEmulation)
data("data_censored")
trial <- trial_sequence("ITT") |>
  set_data(data = data_censored) |>
  set_outcome_model(adjustment_terms = ~x2) |>
  set_expansion_options(output = save_to_tters(), chunk_size = 0)
trial <- expand_trials_tters(trial)
trial <- load_expanded_data(trial, seed = 1234, p_control = 0.5)
trial <- fit_msm(trial)

## End(Not run)
```

---

expand_weighted_df *Expand an in-memory cohort and attach pre-computed inverse-probability weights, returning the weighted frame as a* `data.frame` *— the frame-in/frame-out analogue of* `expand_weighted_parquet()`.

Description

Both the cohort and the per-`(id, period)` factor table (`id, period, weight_factor`) are passed as R `data.frames`; the engine expands under `estimand`, joins the factors, and accumulates the cumulative-product `weight`. A 64-bit integer `id` (`bit64::integer64`) in either frame round-trips exactly.

Usage

```
expand_weighted_df(
  cohort,
  factors,
  id_col,
  period_col,
  treatment_col,
  eligible_col,
  outcome_col,
  first_period,
  last_period,
  estimand
)
```

Arguments

cohort	An R data.frame of long person-time rows.
factors	An R data.frame with columns id, period, weight_factor.
id_col, period_col, treatment_col	Column names in cohort.
eligible_col, outcome_col	Eligibility / outcome column names.
first_period, last_period	Inclusive integer bounds on trial_period.
estimand	"ITT" or "PP". Case-insensitive.

Value

A data.frame with the six structural columns plus weight. Errors in the core engine surface as R errors.

---

expand_weighted_fitted_df

Fit the IPW weights for an in-memory cohort, expand, apply, and return the weighted trial frame as a data.frame *— a raw cohort* data.frame *straight to a weighted, expanded* data.frame *in one call (no pre-computed factor table, no intermediate Parquet). The frame-in/frame-out analogue of* expand_weighted_fitted_parquet(). *A 64-bit integer* id *(*bit64::integer64*) round-trips exactly.*

Description

Fit the IPW weights for an in-memory cohort, expand, apply, and return the weighted trial frame as a data.frame — a raw cohort data.frame straight to a weighted, expanded data.frame in one call (no pre-computed factor table, no intermediate Parquet). The frame-in/frame-out analogue of expand_weighted_fitted_parquet(). A 64-bit integer id (bit64::integer64) round-trips exactly.

Usage

```
expand_weighted_fitted_df(
  cohort,
  id_col,
  period_col,
  treatment_col,
  eligible_col,
  outcome_col,
  first_period,
  last_period,
  estimand,
  use_switch,
  switch_numerator,
  switch_denominator,
  use_censor,
  censor_col,
  censor_numerator,
  censor_denominator,
  pool_censor
)
```

Arguments

`cohort`	An R `data.frame` of long person-time rows.
`id_col`, `period_col`, `treatment_col`	Column names in `cohort`.
`eligible_col`, `outcome_col`	Eligibility / outcome column names.
`first_period`, `last_period`	Inclusive integer bounds on `trial_period`.
`estimand`	`"ITT"` or `"PP"`. Case-insensitive.
`use_switch`	Whether to fit per-protocol switching-weight models.
`switch_numerator`, `switch_denominator`	Covariate columns for the switching numerator/denominator models (ignored when `use_switch` is `FALSE`).
`use_censor`	Whether to fit inverse-probability-of-censoring (IPCW) models.
`censor_col`	Name of the `{0,1}` censoring-indicator column; the response is `1 - censor_col` (ignored when `use_censor` is `FALSE`).
`censor_numerator`, `censor_denominator`	Covariate columns for the IPCW numerator/denominator models (ignored when `use_censor` is `FALSE`).
`pool_censor`	How the IPCW models are pooled across the previous-treatment strata: `"none"`, `"numerator"`, or `"both"`. Case-insensitive.

Value

A `data.frame` with the six structural columns plus `weight`. Errors in the core engine (including weight-fit failures) surface as R errors.

`expand_weighted_fitted_parquet` *Fit the IPW weights in Rust, expand the cohort, apply the weights, and write the weighted trial frame — a raw cohort to a weighted, expanded frame in one call (no pre-computed factor table).*

Description

A thin FFI shim over `tte_expand::expand_weighted_fitted_parquet`: the fully in-Rust analogue of `expand_weighted_parquet()`. It fits the switching and/or IPCW models from the spec (as `fit_weights_parquet()` does), expands under `estimand`, joins and accumulates the fitted factor, and writes the six structural columns plus the cumulative-product `weight`. Structural columns are bit-exact; `weight` matches the Oracle within the staged ~1e-6 tolerance (ADR-2).

Usage

```
expand_weighted_fitted_parquet(
  input_path,
  output_path,
  id_col,
  period_col,
  treatment_col,
  eligible_col,
  outcome_col,
  first_period,
  last_period,
  estimand,
  use_switch,
  switch_numerator,
  switch_denominator,
  use_censor,
  censor_col,
  censor_numerator,
  censor_denominator,
  pool_censor
)
```

Arguments

`input_path`	Path to the input Parquet cohort (long person-time).
`output_path`	Path where the weighted, expanded Parquet is written.

`id_col`, `period_col`, `treatment_col`
: Column names in the input.

`eligible_col`, `outcome_col`
: Eligibility / outcome column names.

`first_period`, `last_period`
: Inclusive integer bounds on `trial_period`.

`estimand`
: `"ITT"` or `"PP"`. Case-insensitive.

`use_switch`
: Whether to fit per-protocol switching-weight models.

`switch_numerator`, `switch_denominator`
: Covariate columns for the switching numerator/denominator models (ignored when `use_switch` is `FALSE`).

`use_censor`
: Whether to fit inverse-probability-of-censoring (IPCW) models.

`censor_col`
: Name of the `{0,1}` censoring-indicator column; the response is `1 - censor_col` (ignored when `use_censor` is `FALSE`).

`censor_numerator`, `censor_denominator`
: Covariate columns for the IPCW numerator/denominator models (ignored when `use_censor` is `FALSE`).

`pool_censor`
: How the IPCW models are pooled across the previous-treatment strata: `"none"`, `"numerator"`, or `"both"`. Case-insensitive.

Value

`NULL`, invisibly; the weighted expansion is written to `output_path`. Errors in the core engine (including weight-fit failures) surface as R errors.

Examples

```
## Not run:
expand_weighted_fitted_parquet(
  "cohort.parquet", "weighted.parquet",
  "id", "period", "treatment", "eligible", "outcome",
  0L, .Machine$integer.max, "PP",
  TRUE, c("x2"), c("x2", "x1"),
  FALSE, "", character(0), character(0), "none"
)

## End(Not run)
```

`expand_weighted_parquet`
: *Expand a person-time Parquet dataset and attach pre-computed inverse-probability weights, writing the weighted frame to* `output_path`.

Description

A thin FFI shim over `tte_expand::expand_weighted_parquet`: it expands the input under `estimand`, joins the per-`(id, period)` factor table at `factors_path` (`id, period, weight_factor`), and writes the six structural columns plus the cumulative-product `weight`. The weight *values* come from R (the `glm` fit); the engine only reproduces their deterministic accumulation.

Usage

```
expand_weighted_parquet(
  input_path,
  factors_path,
  output_path,
  id_col,
  period_col,
  treatment_col,
  eligible_col,
  outcome_col,
  first_period,
  last_period,
  estimand
)
```

Arguments

`input_path`	Path to the input Parquet file.
`factors_path`	Path to the per-`(id, period)` factor Parquet (`id, period, weight_factor`).
`output_path`	Path where the weighted Parquet is written.
`id_col`, `period_col`, `treatment_col`	Column names in the input.
`eligible_col`, `outcome_col`	Eligibility / outcome column names.
`first_period`, `last_period`	Inclusive integer bounds on `trial_period`.
`estimand`	`"ITT"` or `"PP"`; selects the weight *model* upstream, but the application arithmetic (join + cumulative product) is identical for both.

Value

`NULL`, invisibly; the weighted expansion is written to `output_path`. Errors in the core engine surface as R errors.

Examples

```
## Not run:
expand_weighted_parquet(
  "input.parquet", "factors.parquet", "weighted.parquet",
  "id", "period", "treatment", "eligible", "outcome",
  0L, .Machine$integer.max, "PP"
```

```
)

## End(Not run)
```

`fit_trial_weights` *Fit inverse-probability weights for a target-trial cohort (ergonomic wrapper)*

Description

User-facing wrapper around the extendr-generated `fit_weights_parquet()` that *fits* the IPW switching and/or IPCW censoring models in Rust and writes the per-`(id, period)` factor table `(id, period, weight_factor)` — the table `expand_trial_weighted()` consumes. Unlike that pre-computed-factor path, here the weight *models* are fitted in Rust (the `weights-fit` surface): a faithful port of `TrialEmulation`'s design preparation plus a deterministic binomial-logit solver. Robust/sandwich variance and the marginal structural model stay in R.

Usage

```
fit_trial_weights(
  input_path,
  output_path,
  id_col = "id",
  period_col = "period",
  treatment_col = "treatment",
  eligible_col = "eligible",
  outcome_col = "outcome",
  first_period = 0L,
  last_period = .Machine$integer.max,
  estimand = "PP",
  switch_numerator = NULL,
  switch_denominator = NULL,
  censor_col = NULL,
  censor_numerator = NULL,
  censor_denominator = NULL,
  pool_censor = "none"
)
```

Arguments

`input_path`	Path to an existing input Parquet cohort (long person-time).
`output_path`	Path to write the `(id, period, weight_factor)` Parquet.
`id_col, period_col, treatment_col, eligible_col, outcome_col`	Column names. Defaults match the TrialEmulation conventions.
`first_period, last_period`	Inclusive integer period bounds.

`estimand` "ITT" or "PP". Per-protocol runs the artificial-censoring state machine and the switching models; intention-to-treat skips both.

`switch_numerator`, `switch_denominator`
Character vectors of covariate column names for the switching numerator (stabiliser) / denominator models, or `NULL` (the default) to omit switching weights.

`censor_col` Name of the `{0,1}` censoring-indicator column; the modelled response is `1 - censor_col`. `NULL` (the default) omits IPCW weights.

`censor_numerator`, `censor_denominator`
Character vectors of covariate column names for the IPCW numerator / denominator models.

`pool_censor` How the IPCW models are pooled across the previous-treatment strata: "none", "numerator", or "both".

Details

A switching model is fitted when either `switch_numerator` or `switch_denominator` is non-`NULL`; an IPCW censoring model is fitted when `censor_col` is non-`NULL`. Covariates are character vectors of column names; `character(0)` (or `NULL`) yields an intercept-only model.

Value

`output_path`, invisibly.

See Also

`expand_trial_weighted_fitted()` to fit and expand in a single call.

Examples

```
## Not run:
# Per-protocol switching weights (numerator ~ x2, denominator ~ x2 + x1):
fit_trial_weights(
  "cohort.parquet", "factors.parquet", estimand = "PP",
  switch_numerator = "x2", switch_denominator = c("x2", "x1")
)

## End(Not run)
```

---

fit_trial_weights_df *Fit inverse-probability weights for a cohort data.frame, in memory (wrapper)*

Description

Frame-in / frame-out analogue of `fit_trial_weights()`: fits the IPW switching and/or IPCW censoring models in Rust from an in-memory cohort `data.frame` and returns the per-`(id, period)` factor table as a `data.frame` (`id`, `period`, `weight_factor`) — the table `expand_trial_weighted_df()` consumes. Wraps the extendr-generated `fit_weights_df()`. A `bit64::integer64` `id` round-trips exactly (the returned `id` is `integer64`).

Usage

```
fit_trial_weights_df(
  cohort,
  id_col = "id",
  period_col = "period",
  treatment_col = "treatment",
  eligible_col = "eligible",
  outcome_col = "outcome",
  first_period = 0L,
  last_period = .Machine$integer.max,
  estimand = "PP",
  switch_numerator = NULL,
  switch_denominator = NULL,
  censor_col = NULL,
  censor_numerator = NULL,
  censor_denominator = NULL,
  pool_censor = "none"
)
```

Arguments

cohort	A data.frame (or tibble / data.table / Arrow Table) of long person-time rows. Coerced with as.data.frame().
id_col, period_col, treatment_col, eligible_col, outcome_col	Column names. Defaults match the TrialEmulation conventions.
first_period, last_period	Inclusive integer period bounds.
estimand	"ITT" or "PP".
switch_numerator, switch_denominator	Character vectors of covariate column names for the switching numerator / denominator models, or NULL to omit switching weights.
censor_col	Name of the {0,1} censoring-indicator column; the modelled response is 1 - censor_col. NULL omits IPCW weights.
censor_numerator, censor_denominator	Character vectors of covariate column names for the IPCW numerator / denominator models.
pool_censor	How the IPCW models are pooled across the previous-treatment strata: "none", "numerator", or "both".

Details

Model presence follows the same NULL-driven rule as fit_trial_weights(): a switching model is fitted when either switch_* covariate vector is non-NULL; an IPCW model is fitted when censor_col is non-NULL.

Value

A data.frame with columns id, period, weight_factor.

See Also

fit_trial_weights() for the Parquet-path equivalent; expand_trial_weighted_fitted_df() to fit and expand in a single call.

Examples

```
## Not run:
factors <- fit_trial_weights_df(
  cohort, estimand = "PP",
  switch_numerator = "x2", switch_denominator = c("x2", "x1")
)

## End(Not run)
```

---

fit_weights_df — *Fit the inverse-probability weight factor for an in-memory cohort and return the per-*(id, period) *factor table (*id, period, weight_factor*) as a* data.frame *— the frame-in/frame-out analogue of* fit_weights_parquet().

Description

Fit the inverse-probability weight factor for an in-memory cohort and return the per-(id, period) factor table (id, period, weight_factor) as a data.frame — the frame-in/frame-out analogue of fit_weights_parquet().

Usage

```
fit_weights_df(
  cohort,
  id_col,
  period_col,
  treatment_col,
  eligible_col,
  outcome_col,
  first_period,
  last_period,
  estimand,
  use_switch,
  switch_numerator,
  switch_denominator,
  use_censor,
  censor_col,
  censor_numerator,
  censor_denominator,
  pool_censor
)
```

Arguments

cohort	An R data.frame of long person-time rows.
id_col, period_col, treatment_col	Column names in cohort.
eligible_col, outcome_col	Eligibility / outcome column names.
first_period, last_period	Inclusive integer bounds on trial_period.
estimand	"ITT" or "PP". Case-insensitive.
use_switch	Whether to fit per-protocol switching-weight models.
switch_numerator, switch_denominator	Covariate columns for the switching numerator/denominator models (ignored when use_switch is FALSE).
use_censor	Whether to fit inverse-probability-of-censoring (IPCW) models.
censor_col	Name of the {0,1} censoring-indicator column; the response is 1 - censor_col (ignored when use_censor is FALSE).
censor_numerator, censor_denominator	Covariate columns for the IPCW numerator/denominator models (ignored when use_censor is FALSE).
pool_censor	How the IPCW models are pooled across the previous-treatment strata: "none", "numerator", or "both". Case-insensitive.

Value

A data.frame with columns id, period, weight_factor (a 64-bit integer id is returned as bit64::integer64). Errors in the core engine (including weight-fit failures) surface as R errors.

---

fit_weights_parquet *Fit the inverse-probability* **weight factor** *for a Parquet cohort in Rust and write the per-*(id, period) *factor table (*id, period, weight_factor*).*

Description

A thin FFI shim over tte_expand::fit_weights_parquet (the weights-fit surface). Unlike expand_weighted_parquet(), which *applies* a pre-computed factor table, this *fits* the IPW models in Rust: it ports TrialEmulation's data_manipulation + censor_func design preparation and binds a deterministic binomial-logit solver for the switching and/or IPCW censoring models, then forms wt = wt_switch * wtC. The structural design is exact; the fitted factors reproduce R glm within the staged ~1e-6 tolerance (ADR-2), not bit-for-bit. Robust/sandwich variance and the marginal structural model stay in R.

Usage

```
fit_weights_parquet(
  input_path,
  output_path,
  id_col,
  period_col,
  treatment_col,
  eligible_col,
  outcome_col,
  first_period,
  last_period,
  estimand,
  use_switch,
  switch_numerator,
  switch_denominator,
  use_censor,
  censor_col,
  censor_numerator,
  censor_denominator,
  pool_censor
)
```

Arguments

input_path	Path to the input Parquet cohort (long person-time).
output_path	Path where the (id, period, weight_factor) table is written.
id_col, period_col, treatment_col	Column names in the input.
eligible_col, outcome_col	Eligibility / outcome column names.
first_period, last_period	Inclusive integer bounds on trial_period.
estimand	"ITT" or "PP"; per-protocol runs the artificial-censoring state machine and (with switching covariates) the switch models. Case-insensitive.
use_switch	Whether to fit per-protocol switching-weight models.
switch_numerator, switch_denominator	Covariate columns for the switching numerator (stabiliser) and denominator models (ignored when use_switch is FALSE).
use_censor	Whether to fit inverse-probability-of-censoring (IPCW) models.
censor_col	Name of the {0,1} censoring-indicator column; the response is 1 - censor_col (ignored when use_censor is FALSE).
censor_numerator, censor_denominator	Covariate columns for the IPCW numerator/denominator models (ignored when use_censor is FALSE).
pool_censor	How the IPCW models are pooled across the previous-treatment strata: "none", "numerator", or "both". Case-insensitive.

Value

NULL, invisibly; the factor table is written to output_path. Errors in the core engine (including weight-fit failures) surface as R errors.

Examples

```
## Not run:
fit_weights_parquet(
  "cohort.parquet", "factors.parquet",
  "id", "period", "treatment", "eligible", "outcome",
  0L, .Machine$integer.max, "PP",
  TRUE, c("x2"), c("x2", "x1"),
  FALSE, "", character(0), character(0), "none"
)

## End(Not run)
```

---

save_to_tters *Create a* te_datastore_tters *storage backend*

Description

Constructor (the save_to_* convention) for the Rust-backed te_datastore subclass. Like the reference backends it does no work — it returns an empty store to hand to TrialEmulation::set_expansion_options(). The expansion is run later by expand_trials_tters().

Usage

```
save_to_tters()
```

Details

Requires the TrialEmulation (and data.table) package: the returned object is an S4 subclass of TrialEmulation's te_datastore, so the class only exists when TrialEmulation is installed.

Value

A te_datastore_tters object with N = 0L and an empty data slot.

See Also

expand_trials_tters() to populate it with a Rust-fast expansion.

Examples

```
## Not run:
library(TrialEmulation)
trial_sequence("ITT") |>
  set_data(data = data_censored) |>
  set_outcome_model(adjustment_terms = ~x2) |>
  set_expansion_options(output = save_to_tters(), chunk_size = 0)

## End(Not run)
```

Index