

Package: PhysioClinStats (via r-universe)

October 3, 2026

Type Package

Title Clinical Inference Engine for Physiological Signal Studies

Version 0.3.1

Description Statistical inference for rehabilitation studies:
mixed-effects / MMRM longitudinal models, single-case (N-of-1)
designs, estimands with multiple imputation, and per-estimate
uncertainty. Heavy modelling backends are optional (guarded)
Suggests.

Depends R (>= 4.2)

Imports digest, PhysioExperiment (>= 2.0.0), methods, splines, stats

Suggests PhysioCore, knitr, rmarkdown, testthat (>= 3.2.0), mrmr,
emmeans, lme4, SingleCaseES, scan, ggplot2,
SummarizedExperiment, S4Vectors, lmerTest, nlme, flexmix, mice,
rbmi, survival, mediation, WeightIt, ipw, patchwork, rstanarm,
brms

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL <https://github.com/x-biosignal/PhysioClinStats>,
<https://x-biosignal.github.io/PhysioClinStats/>

BugReports <https://github.com/x-biosignal/PhysioExperiment/issues>

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs zlib1g-dev

Repository <https://x-biosignal.r-universe.dev>

Date/Publication 2026-10-03 13:11:47 UTC

RemoteUrl <https://github.com/x-biosignal/PhysioClinStats>

RemoteRef HEAD

RemoteSha 4aabf952bf67c3fa38b3c304e7c19c04dfcf5eea

Contents

analyseEstimand	2
bayesEstimate	4
causalMediation	5
clinStatsBackends	6
conformalInterval	6
conformalPredict	8
coxModel	9
credibleInterval	10
defineEstimand	11
doseResponse	12
ESTIMAND_STRATEGIES	13
estimatedMarginalMeans	13
fitMixedModel	14
fitMMRM	16
latentClassGrowth	17
milestoneHazard	18
multipleImputation	19
pairwiseContrasts	20
peToLong	21
poolEstimates	21
proportionalRecoveryRule	22
recoveryTrajectoryLME	24
requireBackend	25
scedABAB	25
scedCelerationLine	26
scedNAP	27
scedPEM	28
scedPND	29
scedTau	30
scedTauU	31
scedTwoSDBand	32
survivalFit	33
targetTrialEmulate	34
targetTrialProtocol	35
timeToMilestone	36
Index	38

Description

End-to-end analysis of a declared `defineEstimand` estimand: multiply-impute the dropout-missing longitudinal response (in wide form, so the within-subject correlation is respected), fit an MMRM (`fitMMRM`) to each completed data set, and pool the fixed effects with Rubin's rules (`poolEstimates()`). Treatment-policy and hypothetical strategies use the standard MAR imputation; other strategies warn that they need a bespoke imputation model.

Usage

```
analyseEstimand(  
  data,  
  estimand,  
  response,  
  treatment,  
  time,  
  subject,  
  covariates = NULL,  
  m = 20,  
  method = "norm",  
  seed = NULL,  
  covariance = "unstructured"  
)
```

Arguments

<code>data</code>	Long-format data (subject x time rows) with the response, treatment, time, subject, and any (subject-level) covariates.
<code>estimand</code>	An "estimand" from <code>defineEstimand()</code> .
<code>response, treatment, time, subject</code>	Column names.
<code>covariates</code>	Optional subject-level covariate column names.
<code>m</code>	Number of imputations (default 20).
<code>method</code>	mice method (default "norm", appropriate for a continuous longitudinal response).
<code>seed</code>	Optional integer seed.
<code>covariance</code>	MMRM covariance structure (default "unstructured").

Value

An `AnalysisResult` (type "estimand_analysis") carrying the estimand attributes and the pooled fixed-effect table.

References

ICH E9(R1); Rubin 1987; Mallinckrodt 2008 (MMRM).

See Also

`defineEstimand()`, `multipleImputation()`, `poolEstimates()`

bayesEstimate

Bayesian estimate of a mean (conjugate Normal-Normal or Stan)

Description

For a numeric sample, returns the analytic Normal-Normal conjugate posterior for the mean: a normal prior $N(\text{prior_mean}, \text{prior_sd}^2)$ combined with a normal likelihood of known scale σ gives a normal posterior in closed form (a flat prior, $\text{prior_sd} = \text{Inf}$, reproduces the sampling posterior $N(\text{mean}(y), \sigma^2/n)$). For a model formula the estimation is delegated to `rstanarm`/`brms` when installed, and errors informatively otherwise – the analytic numeric-vector path always runs.

Usage

```
bayesEstimate(
  y,
  data = NULL,
  prior_mean = 0,
  prior_sd = Inf,
  sigma = NULL,
  level = 0.95,
  seed = NULL
)
```

Arguments

<code>y</code>	A numeric vector (analytic path) or a model formula (Stan path).
<code>data</code>	Data frame for the formula path.
<code>prior_mean, prior_sd</code>	Normal prior mean and SD (default 0 and Inf, a flat prior).
<code>sigma</code>	Known likelihood SD; NULL uses the sample SD of <code>y</code> .
<code>level</code>	Credible level (default 0.95).
<code>seed</code>	Optional integer random seed recorded in the provenance.

Value

An `AnalysisResult` (from `PhysioCore`) with `type = "bayes"` whose `result` holds `posterior_mean`, `posterior_sd`, `ci_lower`, `ci_upper`, the `level/method`, plus a provenance log with the seed.

References

Gelman, A. et al. (2013). Bayesian Data Analysis, 3rd ed.

See Also

[credibleInterval\(\)](#), [conformalInterval\(\)](#)

Examples

```
set.seed(1)
bayesEstimate(rnorm(30, mean = 5), prior_mean = 0, prior_sd = 10)
```

causalMediation	<i>Model-based causal mediation analysis</i>
-----------------	--

Description

Calls `mediation::mediate()` after checking that the mediator and outcome models describe the same ordered observations. Estimates rely on sequential ignorability, consistency, positivity, absence of a treatment-induced mediator-outcome confounder, and correct mediator and outcome models. An indirect association, even when statistically significant, is not evidence that a biological mechanism has been established.

Usage

```
causalMediation(
  model_m,
  model_y,
  treat,
  mediator,
  control.value = 0,
  treat.value = 1,
  covariates = NULL,
  sims = 1000L,
  boot = FALSE,
  boot.ci.type = "perc",
  conf.level = 0.95,
  robustSE = FALSE,
  sensitivity = TRUE,
  seed = NULL,
  ...
)
```

Arguments

<code>model_m, model_y</code>	Supported fitted mediator and outcome models, or <code>AnalysisResult</code> objects whose <code>result\$fit</code> contains those models.
<code>treat, mediator</code>	Treatment and mediator variable names.
<code>control.value, treat.value</code>	Treatment contrast supplied to the backend.
<code>covariates</code>	Optional named list fixing covariates for a conditional mediation estimand.
<code>sims</code>	Positive integer number of simulations or bootstrap replicates.
<code>boot</code>	Use the nonparametric bootstrap.

<code>boot.ci.type</code>	Bootstrap confidence-interval type.
<code>conf.level</code>	Confidence level in $(0, 1)$.
<code>robustSE</code>	Request heteroskedasticity-consistent uncertainty for supported <code>lm/glm</code> models.
<code>sensitivity</code>	Run <code>mediation::medsens()</code> for an all-linear model pair.
<code>seed</code>	Optional integer seed. The caller's global RNG state is restored.
<code>...</code>	Further arguments passed to <code>mediation::mediate()</code> .

Value

An `AnalysisResult` with a complete ACME/ADE/total/proportion table, the backend object, settings, formulas, assumptions, and sensitivity diagnostics.

<code>clinStatsBackends</code>	<i>Which optional modelling backends are available</i>
--------------------------------	--

Description

The inference functions load heavy modelling packages only when present (guarded via `requireNamespace`). This reports availability so callers can fail early with a helpful message.

Usage

```
clinStatsBackends()
```

Value

A named logical vector for `mrmr`, `emmeans`, `lme4`, and `SingleCaseES`.

Examples

```
clinStatsBackends()
```

<code>conformalInterval</code>	<i>Conformal prediction interval for a fitted model</i>
--------------------------------	---

Description

Distribution-free predictive intervals with finite-sample marginal coverage at least $1 - \alpha$, requiring only exchangeability. "split" conformal (Lei et al. 2018) scores an independent calibration set with the fitted model and takes the conformal quantile of the absolute residuals; "jackknife_plus" (Barber et al. 2021) refits the model leave-one-out over the calibration data and combines the leave-one-out predictions and residuals.

Usage

```
conformalInterval(
  model,
  calib,
  newx,
  alpha = 0.1,
  type = c("split", "jackknife_plus"),
  seed = NULL
)
```

Arguments

model	A fitted model supporting <code>predict(model, newdata=)</code> (and, for "jackknife_plus", <code>update(model, data=)</code>), e.g. an <code>lm</code> .
calib	A data frame with the response and predictors: an independent calibration set for "split", or the full data to refit over for "jackknife_plus".
newx	A data frame of predictor values to predict for.
alpha	Miscoverage level (default 0.1 for 90% intervals).
type	"split" (default) or "jackknife_plus".
seed	Optional integer random seed recorded in the provenance.

Details

The coverage guarantee requires the calibration data to be exchangeable with, and (for "split") independent of, the data the model was trained on. When the model uses a transformed response (e.g. $\log(y) \sim x$) the interval is on that transformed scale.

Value

An `AnalysisResult` (from `PhysioCore`) with `type = "conformal"` whose `result` holds point, lower, upper (one per `newx` row), the `alpha/method/quantile`, and a provenance log capturing the seed.

References

Lei, J. et al. (2018). *JASA* 113(523):1094-1111. Barber, R.F. et al. (2021). *Ann Statist* 49(1):486-507.

See Also

[conformalPredict\(\)](#), [bayesEstimate\(\)](#)

Examples

```
set.seed(1)
train <- data.frame(x = rnorm(60)); train$y <- 2 * train$x + rnorm(60)
calib <- data.frame(x = rnorm(60)); calib$y <- 2 * calib$x + rnorm(60)
fit <- lm(y ~ x, data = train)
```

```
res <- conformalInterval(fit, calib, data.frame(x = c(-1, 0, 1)))
PhysioExperiment::resultValue(res)$lower
```

conformalPredict

Conformal prediction band for a single new patient

Description

Convenience wrapper around split-conformal [conformalInterval\(\)](#) for one new observation, returning the guaranteed-coverage predicted band.

Usage

```
conformalPredict(model, calib, newx, alpha = 0.1, seed = NULL)
```

Arguments

model	A fitted model (see conformalInterval()).
calib	The calibration data frame (response + predictors).
newx	A one-row data frame of predictors for the new patient.
alpha	Miscoverage level (default 0.1).
seed	Optional integer random seed.

Value

An `AnalysisResult` with `type = "conformal"` for the single patient.

See Also

[conformalInterval\(\)](#)

Examples

```
set.seed(1)
d <- data.frame(x = rnorm(80)); d$y <- 1.5 * d$x + rnorm(80)
fit <- lm(y ~ x, data = d[1:40, ])
conformalPredict(fit, d[41:80, ], data.frame(x = 0.5))
```

coxModel

*Cox proportional-hazards model with a PH check***Description**

Fits a Cox proportional-hazards model with `survival::coxph`, tests the proportional-hazards assumption with `survival::cox.zph`, and warns when the global test is significant. Returns hazard ratios with confidence intervals.

Usage

```
coxModel(
  data,
  formula = NULL,
  time = "time",
  event = "event",
  covariates = NULL,
  conf_level = 0.95,
  ph_alpha = 0.05
)
```

Arguments

<code>data</code>	A <code>data.frame</code> .
<code>formula</code>	A model formula whose response is <code>survival::Surv(time, event)</code> , or supply <code>time/event/covariates</code> instead.
<code>time, event, covariates</code>	Alternative to <code>formula</code> : the time and event column names and a character vector of covariate columns.
<code>conf_level</code>	Confidence level (default 0.95).
<code>ph_alpha</code>	Significance level for the global PH warning (default 0.05).

Value

An `AnalysisResult` (type = "cox_model") whose `estimate` is the hazard-ratio vector, with `result$coefficients` (coef, HR, SE, z, p, CI), `result$ph_test` (the `cox.zph` table), `result$ph_violated` and `result$fit`.

References

Cox DR (1972). Regression models and life-tables. JRSS B, 34(2). Grambsch PM, Therneau TM (1994). Proportional hazards tests and diagnostics based on weighted residuals. Biometrika, 81(3).

See Also

[survivalFit\(\)](#), [milestoneHazard\(\)](#)

Examples

```
if (requireNamespace("survival", quietly = TRUE)) {  
  coxModel(survival::lung, time = "time", event = "status",  
           covariates = c("age", "sex", "ph.ecog"))  
}
```

credibleInterval	<i>Credible interval from posterior samples</i>
------------------	---

Description

The highest-density interval (HDI, the narrowest interval carrying level probability) or the equal-tailed quantile interval, from a vector of posterior draws.

Usage

```
credibleInterval(x, level = 0.95, method = c("hdi", "quantile"))
```

Arguments

x	Numeric vector of posterior samples.
level	Credible level (default 0.95).
method	"hdi" (default, narrowest) or "quantile" (equal-tailed).

Value

Numeric c(lower, upper).

See Also

[bayesEstimate\(\)](#)

Examples

```
set.seed(1)  
credibleInterval(rnorm(2000), method = "hdi")
```

defineEstimand	<i>Define an ICH E9(R1) estimand</i>
----------------	--------------------------------------

Description

Constructs a validated estimand from its five ICH E9(R1) attributes. The returned object is a fully-named list (class "estimand") that round-trips through the estimand slot of an [AnalysisResult](#).

Usage

```
defineEstimand(
  treatment,
  population,
  endpoint,
  intercurrent = list(event = NA_character_, strategy = "treatment-policy"),
  summary_measure = "difference in means"
)
```

Arguments

treatment	The treatment condition(s) being compared.
population	The target population.
endpoint	The endpoint / variable of interest.
intercurrent	A named list with event (the intercurrent event) and strategy (one of ESTIMAND_STRATEGIES).
summary_measure	The population-level summary (e.g. "difference in means").

Value

An "estimand" object (a named list of the five attributes plus the strategy's analysis recipe).

References

ICH E9(R1) addendum on estimands and sensitivity analysis (2019).

See Also

[analyseEstimand\(\)](#), [multipleImputation\(\)](#)

Examples

```
defineEstimand(
  treatment = "rehab protocol A vs B", population = "post-stroke",
  endpoint = "6-month gait speed",
  intercurrent = list(event = "treatment discontinuation",
    strategy = "treatment-policy"),
  summary_measure = "difference in means")
```

doseResponse	<i>Therapy dose-response model</i>
--------------	------------------------------------

Description

Fits an outcome (or change from baseline) as a function of a continuous / ordinal therapy dose, with optional confounder adjustment, subject random intercepts and a non-linear dose form. Returns the fitted dose effect, a predicted dose-response curve and the coefficient table.

Usage

```
doseResponse(
  data,
  outcome,
  dose,
  covariates = NULL,
  subject = NULL,
  baseline = NULL,
  form = c("linear", "log", "spline", "emax"),
  df = 3,
  n_grid = 50
)
```

Arguments

data	A data.frame with the columns named below.
outcome	Name of the outcome column.
dose	Name of the therapy-dose column (hours / repetitions / sessions).
covariates	Optional character vector of confounder columns to adjust for (added linearly).
subject	Optional subject-id column; when given, a random intercept per subject is fit via fitMixedModel() (needs lme4).
baseline	Optional baseline column; when given it is added as an adjustment covariate (ANCOVA-style change analysis).
form	Dose form: "linear", "log" (needs positive dose), "spline" (natural cubic spline, df terms) or "emax" ($E_0 + E_{max} \cdot \text{dose} / (ED_{50} + \text{dose})$), fit by stats::nls() ; no subject/covariates).
df	Spline degrees of freedom for form = "spline" (default 3).
n_grid	Number of dose points in the predicted curve (default 50).

Value

A [PhysioExperiment::AnalysisResult](#) of type = "dose_response": estimate is the dose effect (slope, spline coefficients, or $E_0/E_{max}/ED_{50}$); result holds coefficients, the curve (dose, predicted), the fit and the form.

References

Ruberg SJ (1995) Dose response studies. *J Biopharm Stat* 5:1-14.

See Also

`fitMixedModel()`, `proportionalRecoveryRule()`

Examples

```
set.seed(1)
d <- data.frame(dose = rep(c(0, 5, 10, 20), each = 15))
d$change <- 0.4 * d$dose + rnorm(nrow(d), 0, 2)
dr <- doseResponse(d, outcome = "change", dose = "dose", form = "linear")
PhysioExperiment::resultValue(dr)$coefficients
```

ESTIMAND_STRATEGIES	<i>Valid ICH E9(R1) intercurrent-event strategies</i>
---------------------	---

Description

Valid ICH E9(R1) intercurrent-event strategies

Usage

```
ESTIMAND_STRATEGIES
```

Format

An object of class character of length 5.

estimatedMarginalMeans	<i>Estimated marginal means</i>
------------------------	---------------------------------

Description

Computes estimated marginal (least-squares) means from a fitted mixed or MMRM model with `emmeans`, optionally with a follow-up contrast, returning the marginal-mean table (and contrasts) in an `AnalysisResult`.

Usage

```
estimatedMarginalMeans(model, specs, contrasts = NULL, level = 0.95, ...)
```

Arguments

model	An AnalysisResult from <code>fitMixedModel()</code> / <code>fitMMRM()</code> , or a raw model object emmeans understands.
specs	A formula or character spec for the marginal means, e.g. <code>~ treatment time</code> .
contrasts	Optional contrast method passed to <code>emmeans::contrast</code> (e.g. <code>"pairwise"</code> , <code>"trt.vs.ctrl"</code>); NULL (default) returns means only.
level	Confidence level for the intervals (default 0.95).
...	Further arguments forwarded to <code>emmeans::emmeans</code> .

Value

An AnalysisResult (type = "emmeans") with `result$emmeans` (the marginal-mean table) and, when requested, `result$contrasts`.

References

Lenth RV (2024). emmeans: Estimated Marginal Means. R package.

See Also

`pairwiseContrasts()`, `fitMMRM()`

Examples

```
if (requireNamespace("mmrm", quietly = TRUE) &&
    requireNamespace("emmeans", quietly = TRUE)) {
  data(fev_data, package = "mmrm")
  fit <- fitMMRM(fev_data, "FEV1", "ARMCD", "AVISIT", "USUBJID",
    covariates = c("RACE", "SEX"))
  estimatedMarginalMeans(fit, ~ ARMCD | AVISIT)
}
```

fitMixedModel	<i>Fit a linear mixed-effects model</i>
---------------	---

Description

Fits a linear mixed model with `lme4::lmer` and returns tidy fixed-effect and random-effect tables in an AnalysisResult. Denominator degrees of freedom (and the associated t tests) use `lmerTest` when it is installed - either the Satterthwaite or the Kenward-Roger approximation.

Usage

```
fitMixedModel(  
  data,  
  formula,  
  random = NULL,  
  method = c("REML", "ML"),  
  df = c("satterthwaite", "kenward-roger")  
)
```

Arguments

data	A data.frame in long format.
formula	A model formula. It may carry the random-effects terms directly (e.g. $y \sim x + (x \mid \text{subject})$), or supply them via random.
random	Optional one-sided random-effects formula (e.g. $\sim (1 \mid \text{subject})$) appended to formula when the latter has none.
method	"REML" (default) or "ML".
df	Denominator-df method for the fixed-effect tests: "satterthwaite" (default) or "kenward-roger" (both need lmerTest).

Value

An AnalysisResult (type = "mixed_model") whose estimate is the fixed-effect coefficient vector, with result\$fixed (the tidy fixed-effect table), result\$random (variance components), result\$sigma, result\$fit (the fitted model, for [estimatedMarginalMeans\(\)](#) and result\$formula.

References

Bates D, Maechler M, Bolker B, Walker S (2015). Fitting linear mixed-effects models using lme4. Journal of Statistical Software, 67(1). Kuznetsova A, Brockhoff PB, Christensen RHB (2017). lmerTest package. Journal of Statistical Software, 82(13).

See Also

[fitMERM\(\)](#), [estimatedMarginalMeans\(\)](#)

Examples

```
if (requireNamespace("lme4", quietly = TRUE)) {  
  data(sleepstudy, package = "lme4")  
  fitMixedModel(sleepstudy, Reaction ~ Days + (Days | Subject))  
}
```

fitMMRM

*Fit a mixed model for repeated measures (MMRM)***Description**

Fits the Mallinckrodt MMRM for a longitudinal randomised trial: fixed effects for treatment, categorical time and their interaction (plus any covariates), with a within-subject covariance over the repeated time points. Uses the `mmrm` package when available - with Kenward-Roger or Satterthwaite adjusted degrees of freedom - and falls back to `nlme::gl`s otherwise. The fallback reproduces the `mmrm` fixed effects for the unstructured covariance (its primary use) and approximates the homogeneous `cs/ar1/toeplitz` structures; it reports between-within (containment) degrees of freedom and emits a message, since Kenward-Roger / Satterthwaite df need `mmrm`.

Usage

```
fitMMRM(
  data,
  response,
  treatment,
  time,
  subject,
  covariates = NULL,
  covariance = c("unstructured", "ar1", "compound-symmetry", "toeplitz"),
  df = c("kenward-roger", "satterthwaite")
)
```

Arguments

<code>data</code>	A long-format data.frame with one row per subject-time.
<code>response, treatment, time, subject</code>	Column names of the outcome, the treatment factor, the categorical time factor and the subject id.
<code>covariates</code>	Optional character vector of extra fixed-effect covariate columns (e.g. baseline value, stratifiers).
<code>covariance</code>	Within-subject covariance: "unstructured" (default), "ar1", "compound-symmetry" or "toeplitz".
<code>df</code>	Denominator-df method: "kenward-roger" (default) or "satterthwaite" (used by the <code>mmrm</code> path only).

Value

An `AnalysisResult` (type = "mmrm") whose estimate is the fixed-effect coefficient vector, with `result$coefficients` (estimate, SE, df, t, p), `result$backend` ("mmrm" or "gl"), `result$fit` and `result$formula`.

References

Mallinckrodt CH et al. (2008). Recommendations for the primary analysis of continuous endpoints in longitudinal clinical trials. *Drug Information Journal*, 42. Sabanes Bove D et al. (2023). mrmr: Mixed Models for Repeated Measures. R package.

See Also

`fitMixedModel()`, `estimatedMarginalMeans()`

Examples

```
if (requireNamespace("mrmr", quietly = TRUE)) {
  data(fev_data, package = "mrmr")
  fitMRRM(fev_data, "FEV1", "ARMCD", "AVISIT", "USUBJID",
    covariates = c("RACE", "SEX"))
}
```

latentClassGrowth	<i>Latent-class growth analysis of longitudinal trajectories</i>
-------------------	--

Description

Fits a finite mixture of polynomial growth trajectories to panel data (each subject belongs to one latent class), selecting the number of classes by BIC. Returns per-subject class assignments, posterior class probabilities, and the normalised entropy of the classification.

Usage

```
latentClassGrowth(
  data,
  subject,
  time,
  outcome,
  n_classes = 1:3,
  degree = 1,
  seed = NULL
)
```

Arguments

<code>data</code>	A long-format data frame.
<code>subject, time, outcome</code>	Column names for the grouping factor, the time variable, and the response.
<code>n_classes</code>	Integer vector of class counts to compare (default 1:3); the BIC-minimising count is selected.
<code>degree</code>	Polynomial degree of the per-class growth curve (default 1, linear).
<code>seed</code>	Optional RNG seed (flexmix uses random starts).

Value

An `AnalysisResult` (type "latent_class_growth") whose result holds the selected `n_classes`, the bic table, the per-subject assignment, the posterior probabilities, the entropy, and the fitted flexmix model.

References

Nagin 2005 (group-based trajectory modelling); Gruen & Leisch 2008 (flexmix).

See Also

[recoveryTrajectoryLME\(\)](#), [proportionalRecoveryRule\(\)](#)

Examples

```
set.seed(1)
df <- do.call(rbind, lapply(1:40, function(s) {
  fast <- s <= 20; t <- 0:6
  y <- (if (fast) 5 * t else 0.5 * t) + rnorm(7, 0, 1.5)
  data.frame(subject = s, time = t, y = y)
}))
latentClassGrowth(df, "subject", "time", "y", n_classes = 1:3)
```

milestoneHazard

Hazard ratio for attaining a milestone

Description

Fits a Cox model for the time-to-milestone endpoint against a grouping variable and returns the group hazard ratio - the relative rate of attaining the milestone - with its confidence interval.

Usage

```
milestoneHazard(milestone, group, conf_level = 0.95)
```

Arguments

milestone	A time-to-milestone data.frame from timeToMilestone() (columns time, event), joined with the grouping variable.
group	Column name of the grouping variable.
conf_level	Confidence level (default 0.95).

Value

An `AnalysisResult` (type = "milestone_hazard") as [coxModel\(\)](#).

See Also

[timeToMilestone\(\)](#), [coxModel\(\)](#)

multipleImputation	<i>Multiply impute missing data</i>
--------------------	-------------------------------------

Description

Wraps `mice::mice` for missing-at-random multiple imputation (the default), or delegates to **rbmi** for reference-based (jump-to-reference) imputation. Imputations are reproducible under a fixed seed.

Usage

```
multipleImputation(  
  data,  
  method = "pmm",  
  m = 5,  
  predictors = NULL,  
  seed = NULL,  
  reference_based = FALSE,  
  ...  
)
```

Arguments

<code>data</code>	A data frame with missing values (NA).
<code>method</code>	Imputation method passed to <code>mice</code> (default "pmm"); recycled across incomplete columns.
<code>m</code>	Number of imputations (default 5).
<code>predictors</code>	Optional: a character vector of variables to use as predictors for every imputed column, or a full <code>mice</code> <code>predictorMatrix</code> . <code>NULL</code> uses the <code>mice</code> default.
<code>seed</code>	Optional integer seed for reproducibility.
<code>reference_based</code>	Logical; if <code>TRUE</code> , perform reference-based (jump-to-reference) imputation via rbmi (which must be installed).
<code>...</code>	Further arguments passed to <code>mice::mice</code> .

Value

A `mice` "mids" object (MAR path) with `m` and `method` recorded.

References

van Buuren & Groothuis-Oudshoorn 2011 (`mice`); Carpenter et al. 2013 (reference-based MI); **rbmi**.

See Also

[poolEstimates\(\)](#), [analyseEstimand\(\)](#)

Examples

```
if (requireNamespace("mice", quietly = TRUE)) {
  imp <- multipleImputation(mice::nhanes, m = 5, seed = 1)
}
```

pairwiseContrasts	<i>Pairwise contrasts of estimated marginal means</i>
-------------------	---

Description

Convenience wrapper computing all pairwise differences of the estimated marginal means for specs.

Usage

```
pairwiseContrasts(model, specs, adjust = "tukey", level = 0.95, ...)
```

Arguments

model	An AnalysisResult from <code>fitMixedModel()</code> / <code>fitMMRM()</code> , or a raw model object emmeans understands.
specs	A formula or character spec for the marginal means, e.g. <code>~ treatment time</code> .
adjust	Multiplicity adjustment passed to emmeans (default "tukey").
level	Confidence level for the intervals (default 0.95).
...	Further arguments forwarded to <code>emmeans::emmeans</code> .

Value

An AnalysisResult (type = "emmeans_contrasts") with `result$contrasts`.

See Also

`estimatedMarginalMeans()`

peToLong

Reshape a longitudinal PhysioExperiment to long format

Description

Flattens an epoched or longitudinal PhysioExperiment into a long `data.frame(subject, time, channel, value)` suitable for `fitMixedModel()`, `fitMMRM()` and the marginal-means helpers.

Usage

```
peToLong(pe, assay = NULL, value_name = "value", subjects = NULL)
```

Arguments

pe	A PhysioExperiment. A 2D assay is treated as time x channel for a single subject; a 3D assay as time x channel x subject/trial.
assay	Assay name (default the first).
value_name	Name for the value column (default "value").
subjects	Optional character/factor labels for the third dimension (subjects/trials); defaults to the colnames/index.

Value

A long `data.frame` with columns `subject`, `time`, `channel` and the `value` column.

See Also

`fitMixedModel()`, `fitMMRM()`

poolEstimates

Pool estimates across imputations (Rubin's rules)

Description

Combines the coefficient estimates from a list of models fitted to multiply imputed data sets using Rubin's rules with the Barnard-Rubin degrees-of- freedom adjustment. The pooled estimate, total variance, df, confidence interval, and fraction of missing information reproduce `mice::pool`.

Usage

```
poolEstimates(fits, conf_level = 0.95)
```

Arguments

`fits` A list of fitted models (each with `coef()` and `vcov()`), or a list of tidy data frames with columns `term`, `estimate`, and `std.error` (and optionally `df.residual`).

`conf_level` Confidence level for the pooled interval (default 0.95).

Value

An `AnalysisResult` (type "pooled_estimates") whose `result$estimates` is a data frame of pooled estimate, `std.error`, `df`, `statistic`, `p.value`, `conf.low`, `conf.high`, and `fmi` per term.

References

Rubin 1987; Barnard & Rubin 1999. `mice::pool`.

See Also

[multipleImputation\(\)](#), [analyseEstimand\(\)](#)

Examples

```
set.seed(1)
fits <- lapply(1:5, function(i) lm(mpg ~ hp + wt,
  data = mtcars[sample(nrow(mtcars), replace = TRUE), ]))
poolEstimates(fits)
```

`proportionalRecoveryRule`

Evaluate the proportional recovery rule with an artefact guard

Description

Tests whether recovery follows the proportional recovery rule $\Delta \approx p(\text{max} - \text{initial})$ (default $p = 0.7$), classifies fitters vs non-fitters, handles ceiling effects, and runs two diagnostics against the known statistical artefact: a mixture-vs-single-line comparison and a regression-to-the-mean (label-shuffle) control.

Usage

```
proportionalRecoveryRule(
  initial,
  followup,
  max_score,
  expected_prop = 0.7,
  fitter_tol = 0.2,
  ceiling_frac = 0.9,
  n_shuffle = 999,
  seed = NULL
)
```

Arguments

<code>initial</code>	Numeric baseline scores.
<code>followup</code>	Numeric follow-up scores (same length).
<code>max_score</code>	Scalar maximum attainable score (the recovery ceiling).
<code>expected_prop</code>	Expected recovery proportion (default 0.7).
<code>fitter_tol</code>	Half-width around <code>expected_prop</code> for a "fitter" (default 0.2).
<code>ceiling_frac</code>	Subjects with <code>initial</code> $\geq$ <code>ceiling_frac</code> * <code>max_score</code> are treated as ceiling cases (default 0.9) and excluded from the slope fit.
<code>n_shuffle</code>	Label-shuffles for the regression-to-the-mean control (default 999).
<code>seed</code>	Optional RNG seed.

Details

A genuine proportional rule implies the follow-up regresses on the initial score with slope $1 - p$; a pure regression-to-the-mean / coupling artefact instead reproduces the Δ -vs-potential slope when the follow-up scores are shuffled across subjects. When the observed slope is not distinguishable from that shuffled null, `artefact_suspected` is set.

Value

An `AnalysisResult` (type "proportional_recovery") whose `result` holds the fitted slope (+ CI and test vs `expected_prop`), per-subject classification, the `mixture_preferred` flag, and `artefact_suspected`.

References

Prabhakaran 2008; Winters 2015; Hawe, Scott & Dukelow 2019 (proportional-recovery artefact).

See Also

[recoveryTrajectoryLME\(\)](#), [latentClassGrowth\(\)](#)

Examples

```
set.seed(1)
init <- runif(60, 5, 55)
fu <- init + 0.7 * (66 - init) + rnorm(60, 0, 3) # genuine 0.7 rule
proportionalRecoveryRule(init, fu, max_score = 66)
```

recoveryTrajectoryLME *Fit a population recovery-trajectory NLME*

Description

Fits a nonlinear mixed-effects recovery curve to longitudinal panel data with subject-level random effects (partial pooling), and returns the population fixed effects together with per-subject predicted asymptote, rate, and time-to-90%-recovery.

Usage

```
recoveryTrajectoryLME(
  data,
  subject,
  time,
  outcome,
  model = c("exponential", "asymptotic", "logistic")
)
```

Arguments

data	A long-format data frame.
subject, time, outcome	Column names (character) for the grouping factor, the time variable, and the response.
model	"exponential" (default), "asymptotic" (an alias), or "logistic".

Details

Models: "exponential"/"asymptotic" use the self-starting asymptotic curve $y = A + (R_0 - A)e^{-e^{lrc}t}$ (SSasymp); "logistic" uses SSlogis.

Value

An AnalysisResult (type "recovery_trajectory") whose result holds the fixed_effects, random_effects, a per-subject data frame (asymptote, rate, time_to_90), and the fitted nlme object.

References

Pinheiro & Bates 2000 (nlme); Lindstrom & Bates 1990.

See Also

[proportionalRecoveryRule\(\)](#), [latentClassGrowth\(\)](#)

Examples

```
set.seed(1)
df <- do.call(rbind, lapply(1:12, function(s) {
  A <- 60 + rnorm(1, 0, 5); rate <- 0.3 * exp(rnorm(1, 0, 0.2))
  t <- 0:8; data.frame(subject = s, time = t,
    y = A * (1 - exp(-rate * t)) + rnorm(9, 0, 2))
}))
fit <- recoveryTrajectoryLME(df, "subject", "time", "y")
```

requireBackend

*Require an optional backend or stop with guidance***Description**

Require an optional backend or stop with guidance

Usage

```
requireBackend(pkg)
```

Arguments

pkg Backend package name.

Value

Invisibly TRUE if available; otherwise an error.

scedABAB

*Analyse an ABAB (reversal) single-case design***Description**

Orchestrates the SCED analysis of a reversal design: it separates the phases (by default the four phases A1, B1, A2, B2), and for each intervention contrast (B1 vs A1 and B2 vs A2) computes the non-overlap effect sizes, the 2-SD-band decision and per-phase celeration lines.

Usage

```
scedABAB(
  x,
  phases = c("A1", "B1", "A2", "B2"),
  contrasts = list(c("A1", "B1"), c("A2", "B2")),
  improvement = c("increase", "decrease"),
  value = NULL,
  phase = NULL,
```

```

    time = NULL,
    assay = NULL,
    channel = 1L
  )

```

Arguments

<code>x</code>	A <code>PhysioExperiment</code> single-channel series, or a <code>data.frame(time, value, phase)</code> . Observations are ordered by time before analysis, so a time column need not be pre-sorted; without one, row order is used. Ordering matters because the Tau-U baseline-trend correction and the 2-SD consecutive-run rule are sequence-dependent.
<code>phases</code>	Ordered character vector of the phase labels (default <code>c("A1", "B1", "A2", "B2")</code>).
<code>contrasts</code>	A list of length-2 <code>c(baseline, intervention)</code> phase-label pairs (default the two reversal contrasts).
<code>improvement</code>	"increase" (default) or "decrease".
<code>value, phase, time, assay, channel</code>	Column/assay selectors passed to the extractor when <code>x</code> is a <code>data.frame</code> or <code>PhysioExperiment</code> .

Value

An object of class `sced_abab` (a list) with `phase_data`, `contrasts` (per contrast: NAP/Tau/Tau-U/PND/PEM results, the 2-SD decision and both phase celeration lines) and a summary `data.frame`, with `print` and `plot` methods.

See Also

[scedNAP\(\)](#), [scedTwoSDBand\(\)](#), [scedCelerationLine\(\)](#)

Examples

```

df <- data.frame(
  value = c(10, 11, 9, 18, 19, 20, 10, 12, 11, 21, 22, 20),
  phase = rep(c("A1", "B1", "A2", "B2"), each = 3))
res <- scedABAB(df)
res

```

<code>scedCelerationLine</code>	<i>Celeration (trend) line for a single-case phase</i>
---------------------------------	--

Description

Fits a trend line to one phase, either by the White & Haring split-middle method (the standard SCED hand method) or by ordinary least squares.

Usage

```
scedCelerationLine(
  value,
  time = NULL,
  method = c("split_middle", "ols"),
  ratio_period = 7
)
```

Arguments

value	Numeric phase observations, or pass time/value explicitly.
time	Optional numeric session index (default seq_along(value)).
method	"split_middle" (default) or "ols".
ratio_period	Sessions spanned by the reported celeration ratio (default 7, a weekly ratio for daily data).

Value

An AnalysisResult (type = "sced_celeration") whose estimate is the slope (celeration per session), with result\$intercept, result\$fitted, result\$bounce (the ratio of the largest positive to the largest negative residual magnitude) and result\$celeration_ratio (the multiplicative change over ratio_period sessions).

References

White OR, Haring NG (1980). Exceptional Teaching, 2nd ed. Columbus, OH: Merrill.

See Also

[scedTwoSDBand\(\)](#), [scedABAB\(\)](#)

Examples

```
scedCelerationLine(c(4, 6, 5, 8, 7, 10, 9))
```

scedNAP

Non-overlap of all pairs (NAP)

Description

The probability that a randomly chosen intervention observation exceeds a randomly chosen baseline observation, with ties counted as one half (Parker & Vannest 2009) - equivalent to the area under the ROC curve. The confidence interval is the score interval of Newcombe (2006) and the standard error the unbiased estimator, both as in SingleCaseES.

Usage

```
scedNAP(  
  A_data,  
  B_data = NULL,  
  improvement = c("increase", "decrease"),  
  confidence = 0.95  
)
```

Arguments

- A_data, B_data Numeric baseline (A) and intervention (B) observations, or pass a data.frame to the phase orchestrator [scedABAB\(\)](#).
- improvement "increase" (default) if higher scores are better, or "decrease" if lower scores are the therapeutic goal.
- confidence Confidence level for the interval (default 0.95).

Value

An AnalysisResult (type = "sced_nap") whose estimate is the NAP, with an analytic uncertainty interval and result\$se, result\$p_value.

References

Parker RI, Vannest KJ (2009). An improved effect size for single-case research: non-overlap of all pairs. Behavior Therapy, 40(4), 357-367. Newcombe RG (2006). Confidence intervals for an effect size measure based on the Mann-Whitney statistic. Statistics in Medicine, 25.

See Also

```
scedTau\(\), scedTauU\(\)
```

Examples

```
scedNAP(c(20, 20, 26, 25), c(28, 25, 30, 29))
```

scedPEM	<i>Percentage exceeding the median (PEM)</i>
---------	--

Description

The proportion of intervention-phase observations beyond the baseline median, counting ties as one half (Ma 2006).

Usage

```
scedPEM(A_data, B_data = NULL, improvement = c("increase", "decrease"))
```

Arguments

- A_data, B_data Numeric baseline (A) and intervention (B) observations, or pass a data.frame to the phase orchestrator [scedABAB\(\)](#).
- improvement "increase" (default) if higher scores are better, or "decrease" if lower scores are the therapeutic goal.

Value

An AnalysisResult (type = "sced_pem") whose estimate is the PEM.

References

Ma H-H (2006). An alternative method for quantitative synthesis of single-subject research: percentage of data points exceeding the median. Behavior Modification, 30(5), 598-617.

See Also

[scedPND\(\)](#), [scedNAP\(\)](#)

Examples

scedPEM(c(20, 20, 26, 25), c(28, 25, 30, 29))

scedPND	<i>Percentage of non-overlapping data (PND)</i>
---------	---

Description

The proportion of intervention-phase observations that exceed the most extreme baseline observation (Scruggs, Mastropieri & Casto 1987).

Usage

scedPND(A_data, B_data = NULL, improvement = c("increase", "decrease"))

Arguments

- A_data, B_data Numeric baseline (A) and intervention (B) observations, or pass a data.frame to the phase orchestrator [scedABAB\(\)](#).
- improvement "increase" (default) if higher scores are better, or "decrease" if lower scores are the therapeutic goal.

Value

An AnalysisResult (type = "sced_pnd") whose estimate is the PND in [0, 1].

References

Scruggs TE, Mastropieri MA, Casto G (1987). The quantitative synthesis of single-subject research. Remedial and Special Education, 8(2).

See Also

[scedPEM\(\)](#), [scedNAP\(\)](#), [scedTauU\(\)](#)

Examples

```
scedPND(c(20, 20, 26, 25), c(28, 25, 30, 29))
```

scedTau	<i>Tau non-overlap (Tau)</i>
---------	------------------------------

Description

The rank-correlation non-overlap effect size, $\tau = 2 \text{ NAP} - 1$ (Parker et al. 2011), with the SE and interval derived from [scedNAP\(\)](#).

Usage

```
scedTau(
  A_data,
  B_data = NULL,
  improvement = c("increase", "decrease"),
  confidence = 0.95
)
```

Arguments

A_data, B_data	Numeric baseline (A) and intervention (B) observations, or pass a data.frame to the phase orchestrator scedABAB() .
improvement	"increase" (default) if higher scores are better, or "decrease" if lower scores are the therapeutic goal.
confidence	Confidence level for the interval (default 0.95).

Value

An AnalysisResult (type = "sced_tau") whose estimate is Tau in [-1, 1].

See Also

[scedNAP\(\)](#), [scedTauU\(\)](#)

Examples

```
scedTau(c(20, 20, 26, 25), c(28, 25, 30, 29))
```

scedTauU

*Tau-U with baseline-trend correction***Description**

Parker, Vannest, Davis & Sauber's (2011) Tau-U combines the A-vs-B non-overlap with a correction for baseline (phase A) trend. Two reference conventions differ on the denominator and this function exposes both:

Usage

```
scedTauU(
  A_data,
  B_data = NULL,
  improvement = c("increase", "decrease"),
  method = c("parker", "scan"),
  confidence = 0.95
)
```

Arguments

A_data, B_data	Numeric baseline (A) and intervention (B) observations, or pass a data.frame to the phase orchestrator scedABAB() .
improvement	"increase" (default) if higher scores are better, or "decrease" if lower scores are the therapeutic goal.
method	Denominator convention, "parker" (default) or "scan".
confidence	Confidence level for the interval (default 0.95).

Details

method = "parker" (**default**) matches `SingleCaseES::Tau_U`: $(S_{AB} - S_{trendA})/(mn)$, where S_{AB} is the A-vs-B Kendall S and S_{trendA} the within-baseline Kendall S. Can exceed 1 when a baseline trend runs counter to improvement.

method = "scan" matches `scan::tau_u()`'s "A vs. B - Trend A" row: the same S numerator over a tie-corrected pair count D , giving a Kendall tau-b in [-1, 1].

The Z statistic and p-value come from the tie-corrected variance of the Kendall S numerator and are identical under both methods (they depend on S, not the denominator), matching scan's Z and p.

Value

An `AnalysisResult` (type = "sced_tau_u") whose estimate is Tau-U, with `result$S`, `result$se_S`, `result$z`, `result$p_value` and the component S values.

References

Parker RI, Vannest KJ, Davis JL, Sauber SB (2011). Combining nonoverlap and trend for single-case research: Tau-U. *Behavior Therapy*, 42(2), 284-299.

See Also

[scedNAP\(\)](#), [scedTau\(\)](#)

Examples

```
scedTauU(c(20, 20, 26, 25, 22, 23), c(28, 25, 24, 27, 30, 30, 29, 28))
```

scedTwoSDBand

Two-standard-deviation band decision rule

Description

Draws the Nelson (1984) 2-SD band from the baseline (A) phase - the baseline mean plus and minus two baseline standard deviations - and flags the intervention (B) phase for a systematic shift: the rule fires when a run of consecutive or more successive B points falls on the same side beyond the band.

Usage

```
scedTwoSDBand(
  A_data,
  B_data = NULL,
  improvement = c("increase", "decrease"),
  k = 2,
  consecutive = 2L
)
```

Arguments

A_data, B_data	Numeric baseline and intervention observations.
improvement	"increase" (default) or "decrease"; sets which side of the band counts as improvement.
k	Band half-width in baseline SDs (default 2).
consecutive	Number of successive out-of-band points that trigger the flag (default 2, Nelson's rule).

Value

An `AnalysisResult` (type = "sced_2sd") whose estimate is the logical decision, with `result$mean`, `result$sd`, `result$upper`, `result$lower`, `result$outside` (per-B-point side: -1/0/+1) and `result$first_run_at` (index of the first triggering point, or NA).

References

Nelson LS (1984). The Shewhart control chart - tests for special causes. *Journal of Quality Technology*, 16(4), 237-239. Applied to SCED in Gast & Ledford (2014).

See Also

[scedCelerationLine\(\)](#), [scedABAB\(\)](#)

Examples

```
scedTwoSDBand(c(10, 12, 11, 9, 10), c(15, 16, 17, 16, 18))
```

survivalFit

Kaplan-Meier survival fit with attainment-time summary

Description

Fits a Kaplan-Meier estimator with `survival::survfit`, optionally stratified by a grouping variable, and returns the median and quartile attainment times with their confidence intervals.

Usage

```
survivalFit(
  data,
  time = "time",
  event = "event",
  group = NULL,
  conf_level = 0.95
)
```

Arguments

<code>data</code>	A data.frame with the survival columns.
<code>time, event</code>	Column names of the follow-up time and the event indicator (1 = event, 0 = censored).
<code>group</code>	Optional grouping-variable column name for stratified curves.
<code>conf_level</code>	Confidence level (default 0.95).

Value

An object of class `physio_km` (a list) with `fit` (the `survfit`), quantiles (median / Q1 / Q3 attainment times with CIs, per group) and summary table, plus `print` and `plot` methods.

References

Kaplan EL, Meier P (1958). Nonparametric estimation from incomplete observations. JASA, 53(282), 457-481.

See Also

[coxModel\(\)](#), [timeToMilestone\(\)](#)

Examples

```
if (requireNamespace("survival", quietly = TRUE)) {
  survivalFit(survival::lung, "time", "status", group = "sex")
}
```

targetTrialEmulate	<i>Emulate a declared target trial</i>
--------------------	--

Description

Applies eligibility at the declared time zero and estimates either an intention-to-treat contrast or a per-protocol clone-censor-weight contrast. The result remains conditional on consistency, exchangeability, positivity, correct treatment/censoring/outcome models, and a correctly aligned time zero. A Cox hazard ratio is non-collapsible and is not a marginal risk ratio.

Usage

```
targetTrialEmulate(
  data,
  protocol,
  id,
  time,
  treatment,
  outcome,
  event = NULL,
  baseline_covariates = NULL,
  time_varying_covariates = NULL,
  estimand = c("per_protocol", "intention_to_treat"),
  weight_backend = c("internal", "WeightIt", "ipw"),
  stabilized = TRUE,
  numerator_covariates = NULL,
  weight_truncation = c(0.01, 0.99),
  max_weight = Inf,
  conf.level = 0.95
)
```

Arguments

data	Long person-period data.
protocol	A targetTrialProtocol() object.
id, time, treatment, outcome	Column names. For survival outcomes, outcome is the subject-level outcome/censoring time repeated on each long row and event is its repeated status. Without event, outcome is the repeated fixed-horizon binary outcome.
event	Optional binary event-status column.

baseline_covariates, time_varying_covariates	Declared predictors for treatment/adherence models.
estimand	"per_protocol" or "intention_to_treat".
weight_backend	Propensity backend. Optional backends are validation paths and never silent fallbacks.
stabilized	Use time-based numerator probabilities.
numerator_covariates	Additional numerator-model covariates.
weight_truncation	Lower and upper quantiles applied after raw cumulative weights are retained.
max_weight	Additional positive upper cap applied after quantile truncation.
conf.level	Confidence level.

Value

An AnalysisResult containing effects, models, clone rows, both raw and analysis weights, diagnostics, censoring counts, and provenance.

targetTrialProtocol	<i>Declare a target-trial protocol</i>
---------------------	--

Description

Defines the seven protocol components that must be fixed before emulating a target trial. `time_zero` is a separate alignment rule and is never inferred from the observed data. Static strategies are scalar sustained treatment values. Dynamic strategy functions receive the current row as data and the participant's rows through the current time as history; they must not access later rows.

Usage

```
targetTrialProtocol(
  eligibility,
  treatment_strategies,
  assignment,
  time_zero,
  follow_up,
  outcome,
  causal_contrast,
  analysis_plan,
  protocol_id = NULL,
  version = "1.0.0"
)
```

Arguments

eligibility	Eligibility rule. Use a function of baseline data or a one-sided formula for an executable rule.
treatment_strategies	Named list of at least two static scalar or dynamic function strategies.
assignment, outcome, causal_contrast, analysis_plan	Non-empty, serializable protocol declarations.
time_zero	Explicit scalar time value or alignment rule.
follow_up	Non-empty follow-up declaration. A finite positive scalar is used as the fixed horizon by <code>targetTrialEmulate()</code> .
protocol_id	Optional stable identifier. When omitted, one is derived from the protocol declarations.
version	Semantic protocol version.

Value

An object of class `target_trial_protocol`.

Examples

```
protocol <- targetTrialProtocol(
  eligibility = function(data) data$eligible,
  treatment_strategies = list(never = 0, always = 1),
  assignment = "cloning at eligibility",
  time_zero = 0,
  follow_up = 12,
  outcome = "binary recovery by week 12",
  causal_contrast = "always versus never",
  analysis_plan = "clone-censor-weight"
)
protocol
```

timeToMilestone	<i>Derive a time-to-milestone survival endpoint</i>
-----------------	---

Description

From a long-format longitudinal series, finds for each subject the first time its outcome crosses a clinical threshold, producing a right-censored (time, event) pair per subject: event = 1 at the first crossing, or event = 0 (right-censored) at the last observation - or at `censor_at` - for subjects who never attain the milestone.

Usage

```
timeToMilestone(  
  data,  
  id,  
  time,  
  value,  
  threshold,  
  direction = c("increase", "decrease"),  
  censor_at = NULL  
)
```

Arguments

<code>data</code>	A long-format data.frame with one row per subject-visit.
<code>id, time, value</code>	Column names of the subject id, the visit time and the outcome value.
<code>threshold</code>	The clinical milestone value.
<code>direction</code>	"increase" (default; the milestone is reaching a value $\geq$ threshold) or "decrease" (a value $\leq$ threshold).
<code>censor_at</code>	Optional common administrative censoring time for non-attainers; defaults to each subject's last observation time.

Value

A data.frame with columns `id`, `time`, `event` (1 attained / 0 censored) and `attained`, carrying a `Surv` object as the "surv" attribute.

See Also

[survivalFit\(\)](#), [milestoneHazard\(\)](#)

Examples

```
d <- data.frame(  
  id = rep(c("a", "b"), each = 3), time = rep(1:3, 2),  
  value = c(10, 30, 55, 12, 20, 28))  
timeToMilestone(d, "id", "time", "value", threshold = 50)
```

Index

* datasets

ESTIMAND_STRATEGIES, 13

analyseEstimand, 2
analyseEstimand(), 11, 19, 22
AnalysisResult, 11

bayesEstimate, 4
bayesEstimate(), 7, 10

causalMediation, 5
clinStatsBackends, 6
conformalInterval, 6
conformalInterval(), 4, 8
conformalPredict, 8
conformalPredict(), 7
coxModel, 9
coxModel(), 18, 33
credibleInterval, 10
credibleInterval(), 4

defineEstimand, 3, 11
defineEstimand(), 3
doseResponse, 12

ESTIMAND_STRATEGIES, 11, 13
estimatedMarginalMeans, 13
estimatedMarginalMeans(), 15, 17, 20

fitMixedModel, 14
fitMixedModel(), 12–14, 17, 20, 21
fitMMRM, 3, 16
fitMMRM(), 14, 15, 20, 21

latentClassGrowth, 17
latentClassGrowth(), 23, 24

mediation::mediate(), 5, 6
mediation::medsens(), 6
milestoneHazard, 18
milestoneHazard(), 9, 37

multipleImputation, 19
multipleImputation(), 3, 11, 22

pairwiseContrasts, 20
pairwiseContrasts(), 14
peToLong, 21
PhysioExperiment::AnalysisResult, 12
poolEstimates, 21
poolEstimates(), 3, 19
proportionalRecoveryRule, 22
proportionalRecoveryRule(), 13, 18, 24

recoveryTrajectoryLME, 24
recoveryTrajectoryLME(), 18, 23
requireBackend, 25

scedABAB, 25
scedABAB(), 27–31, 33
scedCelerationLine, 26
scedCelerationLine(), 26, 33
scedNAP, 27
scedNAP(), 26, 29, 30, 32
scedPEM, 28
scedPEM(), 30
scedPND, 29
scedPND(), 29
scedTau, 30
scedTau(), 28, 32
scedTauU, 31
scedTauU(), 28, 30
scedTwoSDBand, 32
scedTwoSDBand(), 26, 27
stats::nls(), 12
survivalFit, 33
survivalFit(), 9, 37

targetTrialEmulate, 34
targetTrialEmulate(), 36
targetTrialProtocol, 35
targetTrialProtocol(), 34

`timeToMilestone`, [36](#)
`timeToMilestone()`, [18](#), [33](#)