

Package ‘lambdastar’

October 11, 2026

Type Package

Title Measurement and Linear Hypothesis Models for Lambda Star

Version 0.8.2

Description Estimates intrinsic and captured noncentrality from parallel measurements and evaluates the numerical parsimony functional for an explicitly encoded hypothesis matrix, formula, or compatible linear model. Includes common-case model comparisons, quantized entropy capacity, controlled temperature integration, and explicit singularity diagnostics. Uses manuscript projection estimates by default, with an explicit alternative population estimator. Supports common nuisance adjustment and ordinary case or cluster percentile bootstrap intervals, with diagnostics for undefined estimates and preserved model coding. Provides reusable row-bound design specifications and explicit conditional term blocks, named linear restrictions, explicit predictor-grid contrasts, and fixed or reevaluated basis recipes under a homogeneous isotropic measurement-fluctuation assumption. Supports explicit known-reference mean hypotheses and paired differences from parallel measurement pairs. Encodes fixed person-by-occasion models with parallel indicators, implicit person adjustment and whole-person bootstrap with distinct sampled copies. The underlying method is described in Hammes (2026) [<doi:10.5281/zenodo.22962377>](https://doi.org/10.5281/zenodo.22962377).

License Apache License (== 2.0)

Copyright 2026 Mike Hammes

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, boot

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Mike Hammes [aut, cre, cph]

Maintainer Mike Hammes <mike.hammes@mikehammes.name>

Repository CRAN

Date/Publication 2026-10-11 07:20:02 UTC

Contents

center_columns	2
confint.lambda_measurement	3
design_volume	4
estimate_lambda_partition_items	4
estimate_lambda_star_items	5
eta_app	6
eta_numerical	7
hypothesis_basis	7
lambdastar	8
lambda_bootstrap_control	9
lambda_compare	10
lambda_contrast	12
lambda_control	13
lambda_design	14
lambda_hypothesis	15
lambda_inference	17
lambda_marginal_contrast	18
lambda_measure	20
lambda_model	21
lambda_paired	23
lambda_panel	25
lambda_reference	26
make_eta_numerical_lookup	28
print.lambda_ci	28
project_onto_basis	29
thermodynamic_model_state	30
xi_eta_numerical	31
Index	33

center_columns	<i>Center a numeric matrix</i>
----------------	--------------------------------

Description

Center a numeric matrix

Usage

```
center_columns(x, name = "x")
```

Arguments

- x Numeric matrix, with observations in rows.
- name Name used in diagnostics.

Value

Matrix centered by column, retaining row and column names.

confint.lambda_measurement

Extract Lambda Star percentile intervals

Description

The stored confidence level is used when level is omitted. A different level requires keep_draws=TRUE and reuses the same parameter draws without resampling. The diagnostics attribute includes estimates, valid/undefined draw counts and explicit statuses for missing intervals. Conditional quantiles are available separately in object\$bootstrap\$conditional_quantiles. See [lambda_inference](#).

Usage

```
## S3 method for class 'lambda_measurement'
confint(object, parm, level = 0.95, ...)

## S3 method for class 'lambda_model'
confint(object, parm, level = 0.95, ...)

## S3 method for class 'lambda_comparison'
confint(object, parm, level = 0.95, ...)
```

Arguments

object	A lambda_measurement, lambda_model or lambda_comparison computed with bootstrap=TRUE or lambda_bootstrap_control().
parm	Optional parameter names or integer positions in the CI table.
level	Confidence level; defaults to the level used during estimation.
...	Reserved; unsupported arguments produce an error.

Value

A lambda_ci matrix with lower/upper columns, diagnostics and inference attributes. Suppressed intervals contain NA and an explicit status.

design_volume	<i>Encoded design volume</i>
---------------	------------------------------

Description

Encoded design volume

Usage

```
design_volume(x, logarithm = FALSE, control = lambda_control())
```

Arguments

x	Centered-space hypothesis design (centering is performed internally).
logarithm	Return log volume instead of its exponential.
control	Numerical controls.

Value

Scalar volume, or log volume. Null and one-direction models have $V=1$.

estimate_lambda_partition_items	<i>Estimate the noncentrality partition for an explicit design</i>
---------------------------------	--

Description

Estimate the noncentrality partition for an explicit design

Usage

```
estimate_lambda_partition_items(
  items,
  x,
  scale = c("composite_mean", "item"),
  control = lambda_control(),
  row_ids = NULL,
  nuisance = NULL,
  reference = NULL
)
```

Arguments

items	Complete parallel measurement matrix.
x	Explicit encoded hypothesis matrix on the same rows as items.
scale	Noncentrality denominator, composite_mean or item.
control	Numerical controls.
row_ids	Optional observation IDs.
nuisance	Optional numeric nuisance design shared by measurements and hypotheses. An intercept is implicit only in the ordinary centered case.
reference	NULL retains ordinary centering. A known scalar or numeric vector on the input rows retains departures from that reference, with no implicit nuisance intercept. It must be prespecified, not fitted here.

Value

Measurement state extended with raw and constrained partitions, degrees of freedom, observed energies and the encoded basis. The raw partition preserves additivity; the constrained partition caps captured structure at the nonnegative total. Kappa is NA when total structure is zero.

estimate_lambda_star_items

Estimate intrinsic structure from parallel measurements

Description

Measurements must target the same stable profile with independent, homogeneous replicate errors. Homogeneous isotropic fluctuation refers to measurement errors, not the total response variance within groups. Finite second moments are required; a Gaussian response distribution is not required for these moment/projection calculations. Complete, already scored items are required; neither reverse scoring nor missing-value substitution is inferred.

Usage

```
estimate_lambda_star_items(
  items,
  scale = c("composite_mean", "item"),
  row_ids = NULL,
  nuisance = NULL,
  control = lambda_control(),
  reference = NULL
)
```

Arguments

items	Finite numeric N by K matrix, $K \geq 2$.
scale	Noncentrality denominator: composite mean (default) or single item.
row_ids	Optional unique observation identifiers in input order.
nuisance	Optional numeric nuisance design on the same rows. The intercept is included by default, but not with a fixed reference; redundant nuisance columns count only once.
control	Numerical controls for nuisance rank determination.
reference	NULL retains ordinary centering. A known scalar or numeric vector on the input rows retains departures from that reference, with no implicit nuisance intercept. It must be prespecified, not fitted here.

Value

Measurement state containing temperature, raw and constrained signal and noncentrality, retained composite profile and row identity. At zero temperature, status is singular_temperature and normalized estimates are NA. composite_response always refers to the mean; item scale only changes the noncentrality denominator, not this observed response.

eta_app	<i>Smooth capacity approximation and geometric activation</i>
---------	---

Description

Smooth capacity approximation and geometric activation

Usage

```
eta_app(temperature, resolution)

xi_eta_app(temperature, resolution)

geometry_activation(temperature, resolution)

xi_a(temperature, resolution)
```

Arguments

temperature	Nonnegative temperature.
resolution	Positive T_q; scalar recycling is supported.

Value

Numeric values. eta_app is a labeled approximation, xi_eta_app its primitive, geometry_activation the Hill profile and xi_a its primitive. Zero-temperature values are mathematical boundary values only.

eta_numerical	<i>Quantized maximum-entropy capacity</i>
---------------	---

Description

Numerically solves the Gibbs distribution over nearest-bin quantizer costs $c_j = (\text{abs}(j) - 1/2)^2$ for nonzero j , with $c_0 = 0$. The target minimum second moment is $T/(12 T_q)$. This evaluates the defining supremum; it is not the entropy of a fitted Gaussian distribution. A log-moment equation avoids premature convergence at small positive temperatures.

Usage

```
eta_numerical(  
  temperature,  
  resolution,  
  control = lambda_control(),  
  details = FALSE  
)
```

Arguments

- temperature Nonnegative temperature, scalar or vector.
- resolution Positive T_q , scalar or vector of compatible length.
- control Numerical controls.
- details Return solver diagnostics in a data frame.

Value

Numeric capacity in nats, or diagnostics with one row per coordinate. The value at zero is the mathematical boundary, not an inferential estimate.

hypothesis_basis	<i>Construct a centered hypothesis basis</i>
------------------	--

Description

The returned X retains the encoded design. Q is used only for projection. Rank is tested after column normalization, making the decision insensitive to changes in predictor units. Redundant or constant columns are rejected.

Usage

```
hypothesis_basis(x, control = lambda_control())
```

Arguments

x	Finite numeric design without an intercept. Zero columns specify H0.
control	Numerical controls from lambda_control() .

Value

List containing X, orthonormal Q, rank, log_volume, volume and control. log_volume is retained even when exponentiation underflows.

lambdastar	<i>Lambda Star: measurement, linear hypotheses and percentile intervals</i>
------------	---

Description

A standalone R package for parallel measurements, nuisance-adjusted linear models, the numerical parsimony functional and ordinary case/cluster percentile intervals. Use [lambda_measure\(\)](#), [lambda_model\(\)](#) and [lambda_compare\(\)](#). The manuscript estimator is the default; an alternative population estimator is explicitly selectable. Points and bootstrap draws use the same selected statistic. See [lambda_inference](#). [lambda_design\(\)](#) binds a reusable encoded model to its input rows; [lambda_hypothesis\(\)](#) selects target terms conditional on explicit adjustment terms. [lambda_contrast\(\)](#) encodes named homogeneous coefficient restrictions; [lambda_marginal_contrast\(\)](#) uses explicit predictor settings and weights. Design constructors support fixed or reevaluated bootstrap basis recipes. The manuscript path assumes stable structure plus zero-mean homogeneous, isotropic background fluctuation with finite second moments. Neither Gaussian responses nor equal total group variances are required. The measurement assumption is used, not established by fitting. The optional population estimator has additional assumptions described in [lambda_inference](#). An installed example is available at `system.file("examples", "quickstart.R", package = "lambdastar")`. A guided lesson with plots, known simulated signal and editable experiments is at `system.file("doc", "LEARN_LMBDASTAR.md", package = "lambdastar")`. Source `system.file("examples", "learn_lambdastar.R", package = "lambdastar")` and call `lambdastar_tutorial()` to create a personal worked example. The centered fixed-model learning path and supported-scope table are at `system.file("doc", "MODEL_SPACE_WORKFLOW.md", package = "lambdastar")`. It links reusable designs, conditional effects, coefficient/grid contrasts, basis policies and the distinction between fixed-profile and population targets. Source `system.file("examples", "intervals_and_targets.R", package = "lambdastar")` and call `lambdastar_intervals_example()` for regular and weak-signal percentile examples (B=5000, 95%, seed=1), with actual draws and undefined-value diagnostics. [lambda_reference\(\)](#) retains a known reference origin and mean direction; [lambda_paired\(\)](#) evaluates after-minus-before parallel measurement pairs. The derivation and resolution conventions are in installed `doc/REFERENCE_MEANS.md`. [lambda_panel\(\)](#) encodes fixed person-by-occasion designs with parallel items, explicit person adjustment and whole-person bootstrap with distinct copies. Its derivation and runnable lesson are in installed `doc/PANEL_MODELS.md`. General covariance/mixed models and uncertainty in an estimated reference remain separate extensions. Ordinary centered calls keep their existing meaning. T_q is supplied on the composite-mean response scale. Measurements must be already scored; the package does not perform imputation or validate a scale.

Author(s)

Maintainer: Mike Hammes <mike.hammes@mikehammes.name> [copyright holder]

Authors:

- Mike Hammes <mike.hammes@mikehammes.name> [copyright holder]

lambda_bootstrap_control

Settings for case or cluster percentile bootstrap intervals

Description

Defaults are 5000 draws, 95 percent two-sided percentile intervals and seed 1. Whole persons and predictors are resampled together. Every draw uses the same selected estimator as the original data. Undefined values are retained as NA; draws are never replaced. Any invalid draw suppresses the affected regular CI. Optional quantiles conditional on valid draws are returned in a separate table.

Usage

```
lambda_bootstrap_control(
  B = 5000L,
  conf.level = 0.95,
  ci.sides = "two.sided",
  ci.method = "percentile",
  resampling = "case",
  strata = NULL,
  seed = 1L,
  min_valid = 200L,
  conditional_quantiles = FALSE,
  keep_draws = FALSE,
  keep_indices = FALSE,
  cluster = NULL,
  percentile_interpolation = "boot"
)
```

Arguments

B	Integer number of draws, at least 2.
conf.level	Confidence level strictly between zero and one.
ci.sides	Currently only "two.sided".
ci.method	Currently only "percentile". This controls interval construction, not the point estimator.
resampling	"case" for whole persons or "cluster" for whole clusters, jointly including measurements, predictors and nuisance variables.

strata	NULL or one data column name. Resample within each stratum, retaining its observed number of persons (case) or clusters (cluster). At least two persons per stratum are required for estimation of the fixed-mixture population variance.
seed	Integer from zero to .Machine\$integer.max. The caller's random state and RNG kind are restored, including absence of .Random.seed.
min_valid	Minimum finite draws for quantiles, default 200.
conditional_quantiles	FALSE by default. If TRUE, separately report quantiles conditional on valid draws when some draws are undefined. These are not confidence intervals for the original unconditional target.
keep_draws	Retain the actual parameter estimates for all B draws, including zeros and NA, plus their per-parameter failure reasons.
keep_indices	Retain the B by N matrix of original input row positions. Cluster resampling instead returns a list of variable-length row vectors.
cluster	Column identifying clusters for resampling="cluster". Labels are scoped within strata. Each stratum draws its original number of clusters with replacement, retaining all rows of every selected cluster.
percentile_interpolation	"boot" (default) uses boot::boot.ci's normal-quantile interpolation; "type7" uses stats::quantile(type=7), as in the manuscript. Both compute ordinary equal-tail percentile intervals.

Value

Settings for the bootstrap argument of lambda_measure(), lambda_model() or lambda_compare().

lambda_compare	<i>Compare two models on one common measurement matrix</i>
----------------	--

Description

This wrapper prepares the measurement state and case selection once and calls the same single-model evaluator twice. No separate F_H formula is used. Complete-case selection spans indicators and variables from BOTH models. delta_F is F_Hb - F_Ha; negative values mean the second model has lower F_H. This is a descriptive difference, not a p-value or automatic model selection.

Usage

```
lambda_compare(
  data,
  model_a,
  model_b,
  indicators,
  T_q,
  measurement_scale = c("mean", "item"),
```

```

na.action = c("fail", "complete"),
contrasts_a = NULL,
contrasts_b = NULL,
capacity = c("numerical", "smooth"),
numerical_control = lambda_control(),
eta_lookup = NULL,
response_tol = sqrt(.Machine$double.eps),
bootstrap = FALSE,
estimator = c("paper", "population"),
nuisance = NULL,
nuisance_contrasts = NULL
)

```

Arguments

<code>data</code>	A data frame or matrix, ordinarily with one row per person. A <code>lambda_panel</code> model instead uses one row per person-occasion observation. Row names identify input observations; panel person labels are declared separately. Otherwise input positions provide the row identities.
<code>model_a, model_b</code>	Two explicitly supplied models or codings, as for <code>model</code> .
<code>indicators</code>	Character names of at least two numeric measurement columns. For <code>lambda_measure</code> only, <code>NULL</code> selects all columns.
<code>T_q</code>	Positive measurement resolution variance on the mean-response scale.
<code>measurement_scale</code>	Noncentrality denominator: "mean" (default) or "item". The observed response remains the item mean in either case.
<code>na.action</code>	"fail" or explicit whole-person "complete" selection.
<code>contrasts_a, contrasts_b</code>	Optional contrasts for the respective formulas.
<code>capacity</code>	"numerical" (default) or the explicit "smooth" approximation.
<code>numerical_control</code>	Controls from <code>lambda_control()</code> .
<code>eta_lookup</code>	Optional numerical integration lookup from <code>make_eta_numerical_lookup()</code> .
<code>response_tol</code>	Nonnegative pointwise tolerance for checking response and saved lm design: $\text{abs}(a-b) \leq \text{response_tol} * \max(1, \text{abs}(a), \text{abs}(b))$.
<code>bootstrap</code>	<code>FALSE</code> for point estimates (default), <code>TRUE</code> for 5000-draw percentile intervals, or settings from <code>lambda_bootstrap_control()</code> . See lambda_inference for estimator and resampling definitions.
<code>estimator</code>	"paper" (default) uses the manuscript's projection bath correction. "population" explicitly selects the alternative random-design population correction; it currently supports case resampling and intercept adjustment only.
<code>nuisance</code>	<code>NULL</code> , a one-sided formula, or a numeric nuisance matrix on the input rows. An intercept is included in ordinary centered analyses. Fixed references use only explicit nuisance columns. Measurements and <code>X_H</code> are adjusted jointly and measurement df are N minus the nuisance rank.

nuisance_contrasts

Optional contrasts for a nuisance formula.

Value

A lambda_comparison with model_a, model_b, delta_F, shared measurement, case provenance and effective settings. At T=0 delta_F is NA.

lambda_contrast	<i>Encode a homogeneous linear coefficient hypothesis</i>
-----------------	---

Description

Constructs $L\beta = 0$ in the non-intercept coefficient coding of a full model. Named columns not supplied in L receive zero weights. The full design must be identifiable after common nuisance adjustment, and L must have independent rows. Intercept/nonzero-reference hypotheses require a different reference space and are not part of this constructor.

Usage

```
lambda_contrast(
  data,
  model,
  L,
  contrasts = NULL,
  nuisance = NULL,
  nuisance_contrasts = NULL,
  encoding = c("reevaluate", "fixed")
)
```

Arguments

data	Common input data frame or matrix, with unique row identities.
model	A whole-model lambda_design, formula, compatible unweighted lm, or encoded numeric matrix. A supplied design owns its coding, nuisance and encoding policy; contradictory constructor arguments are rejected.
L	Finite numeric matrix with unique coefficient column names and optional unique row labels, or a named numeric vector for one restriction.
contrasts	Explicit formula contrast list, as in lambda_model().
nuisance	Optional common one-sided nuisance formula or numeric matrix.
nuisance_contrasts	Contrast list for the nuisance formula.
encoding	Bootstrap basis policy. "reevaluate" (default) evaluates sample-dependent transforms in each draw. "fixed" learns the encoding after common case selection and resamples its rows, including nuisance terms. Persons are still resampled; this does not fix the observed X.

Details

The null-model directions become additional nuisance columns. Consequently, lambda_star and kappa describe structure after adjustment for that null model. The actual ordered contrast coding is retained for F_H, not replaced by the numerical orthogonal basis. Geometry and estimability are rebuilt each draw.

Value

A lambda_design for the conditional contrast space. The realized L, coefficient right inverse, null basis and full-model metadata are available in the evaluated result's model component.

Examples

```
d <- data.frame(x=seq_len(20), z=sin(seq_len(20)))
lambda_contrast(d, ~x+z, c(x=1, z=-1))
```

lambda_control	<i>Numerical settings for the Lambda Star core</i>
----------------	--

Description

Numerical settings for the Lambda Star core

Usage

```
lambda_control(
  rank_tol = 1e-10,
  capacity_tol = 1e-10,
  tail_tol = 1e-14,
  integration_rel_tol = 1e-08,
  integration_abs_tol = 0,
  subdivisions = 200L,
  lookup_rel_tol = 1e-05,
  lookup_abs_tol = 0,
  lookup_max_points = 32769L,
  max_support = 1000000L
)
```

Arguments

rank_tol	Relative rank tolerance after column normalization.
capacity_tol	Relative target-moment tolerance for the Gibbs solver.
tail_tol	Relative truncation tolerance for partition and moment tails.
integration_rel_tol	Relative quadrature tolerance.

integration_abs_tol	Absolute quadrature tolerance (zero by default).
subdivisions	Maximum adaptive quadrature subdivisions.
lookup_rel_tol	Relative local trapezoidal-error target for lookups.
lookup_abs_tol	Absolute dimensionless lookup-error target.
lookup_max_points	Maximum lookup mesh size.
max_support	Maximum positive quantizer-bin index.

Value

A validated list of numerical controls. Lookup error estimates are numerical estimates, not rigorous error certificates.

lambda_design	<i>Bind a reusable encoded model specification to input rows</i>
---------------	--

Description

A design keeps the actual coding and its recipe, not just a projection basis. Formula realization is deferred until common case selection in `lambda_model()` or `lambda_compare()`. Explicit matrices are already encoded. No temperature, fitted projection or measurement state is cached in this object.

Usage

```
lambda_design(
  data,
  model,
  contrasts = NULL,
  nuisance = NULL,
  nuisance_contrasts = NULL,
  encoding = c("reevaluate", "fixed")
)

## S3 method for class 'lambda_design'
print(x, ...)
```

Arguments

data	Common input data frame or matrix, with unique row identities.
model	Formula, compatible unweighted lm, or encoded numeric matrix.
contrasts	Explicit formula contrast list, as in <code>lambda_model()</code> .
nuisance	Optional common one-sided nuisance formula or numeric matrix.
nuisance_contrasts	Contrast list for the nuisance formula.

encoding	Bootstrap basis policy. "reevaluate" (default) evaluates sample-dependent transforms in each draw. "fixed" learns the encoding after common case selection and resamples its rows, including nuisance terms. Persons are still resampled; this does not fix the observed X.
x	A lambda_design object.
...	Reserved for S3 compatibility.

Details

Relevant input columns and row identities are bound exactly. New measurements on the same predictors are allowed for one-sided formulas; changed predictors or row order require a new design. Ordinary deterministic base/stats terms and supported namespace-qualified terms are portable; caller-defined functions must be evaluated explicitly into a matrix before construction.

Factor contrast defaults are recorded and realized as numeric contrasts. During bootstrap, explicit matrices and simple formulas are sliced by row; sample-dependent formula transforms are reevaluated by default, as with the existing formula interface. With encoding="fixed", the raw encoding learned after common case selection is retained during resampling. Projections are rebuilt in either case.

Value

A versioned lambda_design specification. X_H is populated for matrix inputs; formula matrices are available as encoded_x on the evaluated result. B stores the nuisance recipe. row_ids and bound_data record the input binding.

See Also

[lambda_hypothesis\(\)](#), [lambda_contrast\(\)](#), [lambda_marginal_contrast\(\)](#). The installed learning path is at `system.file("doc", "MODEL_SPACE_WORKFLOW.md", package="lambdastar")`.

Examples

```
d <- data.frame(g=factor(rep(c("a", "b"), each=10)), x=seq_len(20))
design <- lambda_design(d, ~g+x)
design
```

lambda_hypothesis

Construct a conditional block from full-model terms

Description

Target and adjustment terms are explicit, disjoint selections from the full model's term labels. Columns are taken from that model's actual coding. Unselected terms are reported as omitted, although their variables still participate in full-recipe evaluation and common case selection.

Usage

```
lambda_hypothesis(
  data,
  model,
  target,
  adjust,
  contrasts = NULL,
  nuisance = NULL,
  nuisance_contrasts = NULL,
  encoding = c("reevaluate", "fixed")
)
```

Arguments

<code>data</code>	Common input data frame or matrix, with unique row identities.
<code>model</code>	Formula, compatible unweighted lm, or encoded numeric matrix.
<code>target</code>	Nonempty character vector of exact full-model term labels.
<code>adjust</code>	Character vector of adjustment term labels. Explicitly use <code>character()</code> for no adjustment terms. No automatic complement is inferred.
<code>contrasts</code>	Explicit formula contrast list, as in <code>lambda_model()</code> .
<code>nuisance</code>	Optional common one-sided nuisance formula or numeric matrix.
<code>nuisance_contrasts</code>	Contrast list for the nuisance formula.
<code>encoding</code>	Bootstrap basis policy. "reevaluate" (default) evaluates sample-dependent transforms in each draw. "fixed" learns the encoding after common case selection and resamples its rows, including nuisance terms. Persons are still resampled; this does not fix the observed X.

Details

The resulting `lambda_star` and `kappa` refer to structure remaining after the specified adjustment. Partial kappas with different adjustment spaces are not an additive partition. F_H comparisons require the same nuisance recipe. This is a term-block constructor, not a general coefficient restriction or an automatic Type I/II/III analysis. The intercept is always removed.

Value

A `lambda_design` for the target block with its effective nuisance recipe, target, adjustment and omitted-term metadata.

Examples

```
d <- data.frame(g=factor(rep(c("a","b"), each=10)), x=seq_len(20))
lambda_hypothesis(d, ~g*x, target="g:x", adjust=c("g","x"))
```

lambda_inference

*Estimator selection and ordinary percentile bootstrap***Description**

estimator="paper" is the default. With nuisance design B (including an intercept), let $d=N-\text{rank}(B)$, $p=\text{rank}(M_B X_H)$, $y=M_B Y_{\text{mean}}$ and Q_H its captured energy. On the mean scale, raw estimates are $\lambda_{\text{star}}=\|y\|^2/T-d$, $\lambda_h=Q_H/T-p$ and $\lambda_{\text{perp}}=\text{RSS}/T-(d-p)$. T is the parallel-item disagreement divided by $d*(K-1)*K$. Item scale changes the denominator, retaining the composite-mean bath correction. Estimates constrain total signal to be nonnegative and captured signal to lie between zero and total.

Details

The selected points are in estimates and the top-level lambda/kappa fields. ρ_{star} is $\lambda_{\text{star}}/(\lambda_{\text{star}}+d)$ for the selected estimator. paper retains the manuscript estimates and raw values. inference and settings identify the estimator independently of whether bootstrap is requested.

estimator="population" explicitly selects the alternative random-design estimator. Its targets at retained N are $\lambda_{\text{star}}=(N-1)*\text{Var}(S)/T$, $\lambda_h=(N-1)*\text{Var}(\Pi_H S)/T$ and $\kappa=\text{Var}(\Pi_H S)/\text{Var}(S)$, where Π_H is the linear population projection. Its raw components are $V_S=\text{var}(Y_{\text{mean}})-T_{\text{mean}}$ and $V_H=\text{var}(Y_{\text{mean}})-\text{RSS}/(N-1-p)$. This correction assumes a linear conditional mean and homogeneous residual variance. It is different from the manuscript fixed-design correction when uncaptured stable signal remains. Set $S_{\text{hat}}=\max(0, V_S)$, $H_{\text{hat}}=\min(S_{\text{hat}}, \max(0, V_H))$; estimates multiply these components by $(N-1)/T$. A non-null model requires positive residual df. population stores these components and estimates only when explicitly selected. This option currently supports intercept adjustment and case resampling; additional nuisance or cluster combinations are rejected explicitly.

With population inference and fixed case strata, variance uses observed proportions w_g , sample variances s_g^2 and means m_g : $\sum w_g [1-(1-w_g)/n_g] s_g^2 + \sum w_g (m_g - \sum w_g m_g)^2$. Each such stratum needs two persons. Paper estimation uses projected energies; specifying resampling strata does not change its point estimator.

Ordinary case bootstrap draws N persons with replacement; all measurements, predictors and nuisance variables travel together. Optional strata retain observed case counts. Cluster bootstrap draws the observed number of whole clusters within each stratum; sample size can vary. Cluster labels are scoped within strata. Nuisance effects are refitted on each draw and d is recomputed. Cluster names and strata are sorted for reproducible sampling; case-stratum ordering follows first occurrence. Complete-case selection is shared by all models and precedes sampling. Factor coding is held fixed.

Every draw uses the selected point estimator, including its constraints. Kappa is computed inside each draw. Direct equal-tail percentiles form intervals for T and lambda/kappa. percentile_interpolation="boot" uses boot::boot.ci(type="perc") and normal-quantile interpolation; "type7" uses stats::quantile(type=7), reproducing the manuscript's interpolation. These choices change interpolation only, never the estimator or resampling design. A constant distribution returns identical endpoints. Quantile messages are retained; unavailable quantiles produce NA with quantile_unavailable. Changing confidence level reuses stored draws.

Zero lambda_star is a valid value in its own distribution. With zero estimated total signal, kappa is undefined and stored as NA with reason undefined_kappa. At positive signal, kappa=0 and kappa=1 are valid observations. Original T=0 is singular and skips inference. A resampled T=0 remains a temperature value; the dependent lambda/kappa values in that draw are undefined.

Invalid draws are counted per parameter with reasons; none are replaced. A nonfinite original estimate yields status undefined_estimate. Otherwise, any undefined kappa draw yields undefined_draws; other invalid draws yield failed_draws. The affected regular interval has NA limits. Other parameters retain their own intervals. min_valid applies to quantile calculation. Optional conditional_quantiles=TRUE returns quantiles of finite draws in a separate conditional_quantiles table; these are conditional descriptive quantiles, not regular confidence intervals. n_undefined counts zero-signal kappa draws; n_failed counts all nonfinite parameter values.

Stored draws contain actual parameter estimates including zeros and NA. Optional original row indices and draw_reasons make each draw inspectable. Factor coding is fixed; lost levels and rank loss have explicit diagnostics. Data-dependent formula transforms are rebuilt; formulas must be deterministic. The caller's RNG kind and state are restored. Intervals are pointwise; F_H and delta_F remain point estimates. Interval availability, width and coverage are distinct properties; kappa coverage has no defined target at true zero signal. A runnable regular/weak-signal example is available at `system.file("examples", "intervals_and_targets.R", package="lambdastar")`. Source that file and call `lambdastar_intervals_example()`; its retained synthetic datasets have known population targets, while the estimator receives only the observed measurement and predictor columns. The installed `doc/MODEL_SPACE_WORKFLOW.md` explains why a fixed-profile projection, a finite-N population target and the expectation of the paper statistic differ.

Fixed-reference designs from `lambda_reference()` use the same paper statistic on departures from a known origin, without an implicit intercept. With no nuisance, `d=N` and `~1` retains the mean direction. Reference values are subset with the original persons on every draw. `lambda_paired()` forms matching after-before item differences before estimating T; its `T_q` is supplied on the mean-difference scale. This path does not extend `estimator="population"`. The derivation is installed as `doc/REFERENCE_MEANS.md`.

References

<https://stat.ethz.ch/R-manual/R-devel/library/boot/html/boot.html> <https://stat.ethz.ch/R-manual/R-devel/library/boot/html/boot.ci.html>

See Also

`lambda_bootstrap_control()`, `confint.lambda_measurement()`

lambda_marginal_contrast

Encode contrasts of explicitly specified predictor settings

Description

The grid lists predictor settings; each weights row is a signed contrast over those grid rows and must sum to zero. Supply all averaging weights explicitly (for example -0.5,-0.5,0.5,0.5 for an equally averaged difference). No covariate values, reference population or equal-cell averaging is inferred. The learned polynomial/spline/scale prediction recipe is used to encode the grid in the full model's coordinates, separately in each bootstrap draw. Unsupported sample-dependent transforms require an explicit coefficient L.

Usage

```
lambda_marginal_contrast(
  data,
  model,
  grid,
  weights,
  contrasts = NULL,
  nuisance = NULL,
  nuisance_contrasts = NULL,
  encoding = c("reevaluate", "fixed")
)
```

Arguments

data	Common input data frame or matrix, with unique row identities.
model	A whole-model lambda_design, formula, compatible unweighted lm, or encoded numeric matrix. A supplied design owns its coding, nuisance and encoding policy; contradictory constructor arguments are rejected.
grid	Data frame of at least two complete predictor settings, with unique row names. It does not contain measurement outcomes.
weights	Named numeric vector or matrix whose columns match the grid row names exactly. Row order/labels define the contrast coding.
contrasts	Explicit formula contrast list, as in lambda_model().
nuisance	Optional common one-sided nuisance formula or numeric matrix.
nuisance_contrasts	Contrast list for the nuisance formula.
encoding	Bootstrap basis policy. "reevaluate" (default) evaluates sample-dependent transforms in each draw. "fixed" learns the encoding after common case selection and resamples its rows, including nuisance terms. Persons are still resampled; this does not fix the observed X.

Value

A lambda_design as for lambda_contrast(), with grid and weights in its hypothesis metadata. The realized coefficient L is stored on results.

Examples

```
d <- data.frame(g=factor(rep(c("a","b"),10)), x=seq_len(20))
grid <- data.frame(g=c("a","b"), x=c(10,10), row.names=c("a10","b10"))
lambda_marginal_contrast(d, ~g*x, grid, c(a10=-1,b10=1))
```

lambda_measure

Prepare parallel measurements from a common data frame

Description

Only complete, already scored parallel measurements are supported. Missing cases are rejected by default. Complete-case selection uses whole persons, never a different number of items per person. Point estimates and bootstrap draws use the same explicitly selected estimator.

Usage

```
lambda_measure(
  data,
  indicators = NULL,
  measurement_scale = c("mean", "item"),
  na.action = c("fail", "complete"),
  bootstrap = FALSE,
  estimator = c("paper", "population"),
  nuisance = NULL,
  nuisance_contrasts = NULL,
  numerical_control = lambda_control(),
  reference = NULL
)
```

Arguments

data	A data frame or matrix, ordinarily with one row per person. A <code>lambda_panel</code> model instead uses one row per person-occasion observation. Row names identify input observations; panel person labels are declared separately. Otherwise input positions provide the row identities.
indicators	Character names of at least two numeric measurement columns. For <code>lambda_measure</code> only, <code>NULL</code> selects all columns.
measurement_scale	Noncentrality denominator: "mean" (default) or "item". The observed response remains the item mean in either case.
na.action	"fail" or explicit whole-person "complete" selection.
bootstrap	<code>FALSE</code> for point estimates (default), <code>TRUE</code> for 5000-draw percentile intervals, or settings from <code>lambda_bootstrap_control()</code> . See lambda_inference for estimator and resampling definitions.

estimator	"paper" (default) uses the manuscript's projection bath correction. "population" explicitly selects the alternative random-design population correction; it currently supports case resampling and intercept adjustment only.
nuisance	NULL, a one-sided formula, or a numeric nuisance matrix on the input rows. An intercept is included in ordinary centered analyses. Fixed references use only explicit nuisance columns. Measurements and X_H are adjusted jointly and measurement df are N minus the nuisance rank.
nuisance_contrasts	Optional contrasts for a nuisance formula.
numerical_control	Controls from lambda_control().
reference	For lambda_measure, NULL uses ordinary centering; a known scalar/vector retains departures from that reference without an implicit intercept. For models use a lambda_reference() design.

Value

A lambda_measurement containing the core measurement state plus included/excluded input positions, original row IDs and effective settings. estimates, inference and settings identify the selected estimator. The paper estimates remain in paper; population is present only when selected.

lambda_model	<i>Evaluate a complete linear hypothesis model</i>
--------------	--

Description

Models describe the hypothesis space after adjustment for the common nuisance design, if supplied separately. Intercepts are removed from the encoded design before ordinary centering. Use ~ 1 for the centered null model. Redundant directions (including a full dummy block from ~ 0 + factor) are rejected, not recoded. A [lambda_design\(\)](#), [lambda_hypothesis\(\)](#), [lambda_contrast\(\)](#) or [lambda_marginal_contrast\(\)](#) object carries its own coding and nuisance recipe. Non-NULL call settings must agree with the object. A [lambda_reference\(\)](#) design instead retains its known origin and explicit mean direction: ~1 is the mean model, ~0 its fixed null. No nuisance intercept is then added implicitly. Reference values travel with rows during resampling. A [lambda_panel\(\)](#) design declares person-by-occasion rows with parallel indicators, person-level complete-case selection and whole-person bootstrap. Its default person adjustment uses within-person demeaning; bootstrap copies receive distinct identities. See lambda_panel for the measurement assumptions. Conditional blocks report total structure and kappa after that adjustment. The manuscript path assumes homogeneous isotropic measurement fluctuation, not Gaussian responses or equal total group variances.

Usage

```
lambda_model(
  data,
  model,
```

```

indicators,
T_q,
measurement_scale = c("mean", "item"),
na.action = c("fail", "complete"),
contrasts = NULL,
capacity = c("numerical", "smooth"),
numerical_control = lambda_control(),
eta_lookup = NULL,
response_tol = sqrt(.Machine$double.eps),
bootstrap = FALSE,
estimator = c("paper", "population"),
nuisance = NULL,
nuisance_contrasts = NULL
)

```

Arguments

<code>data</code>	A data frame or matrix, ordinarily with one row per person. A <code>lambda_panel</code> model instead uses one row per person-occasion observation. Row names identify input observations; panel person labels are declared separately. Otherwise input positions provide the row identities.
<code>model</code>	A <code>lambda_design</code> , one- or two-sided formula, suitable <code>lm</code> object, or numeric <code>X_H</code> matrix with the same input rows as <code>data</code> .
<code>indicators</code>	Character names of at least two numeric measurement columns. For <code>lambda_measure</code> only, <code>NULL</code> selects all columns.
<code>T_q</code>	Positive measurement resolution variance on the mean-response scale.
<code>measurement_scale</code>	Noncentrality denominator: "mean" (default) or "item". The observed response remains the item mean in either case.
<code>na.action</code>	"fail" or explicit whole-person "complete" selection.
<code>contrasts</code>	Optional <code>model.matrix</code> contrasts.arg list for a formula. Matrices and <code>lm</code> objects already specify their coding and reject this argument.
<code>capacity</code>	"numerical" (default) or the explicit "smooth" approximation.
<code>numerical_control</code>	Controls from <code>lambda_control()</code> .
<code>eta_lookup</code>	Optional numerical integration lookup from <code>make_eta_numerical_lookup()</code> .
<code>response_tol</code>	Nonnegative pointwise tolerance for checking response and saved <code>lm</code> design: $\text{abs}(a-b) \leq \text{response_tol} * \max(1, \text{abs}(a), \text{abs}(b))$.
<code>bootstrap</code>	<code>FALSE</code> for point estimates (default), <code>TRUE</code> for 5000-draw percentile intervals, or settings from <code>lambda_bootstrap_control()</code> . See lambda_inference for estimator and resampling definitions.
<code>estimator</code>	"paper" (default) uses the manuscript's projection bath correction. "population" explicitly selects the alternative random-design population correction; it currently supports case resampling and intercept adjustment only.

nuisance	NULL, a one-sided formula, or a numeric nuisance matrix on the input rows. An intercept is included in ordinary centered analyses. Fixed references use only explicit nuisance columns. Measurements and X_H are adjusted jointly and measurement df are N minus the nuisance rank.
nuisance_contrasts	Optional contrasts for a nuisance formula.

Details

One-sided formulas use the item mean as response. A two-sided formula or lm must have a numeric response equal to that mean on the selected rows. Formula variables must be columns of data. Transformations are evaluated after common case selection; transformation-created missing or infinite values are errors. Factors retain their levels. The actual model.matrix coding determines volume.

Only unweighted, univariate lm objects with an intercept, saved model frame, no offset, and exactly the selected rows in the same order are accepted. The saved response and model matrix are checked against data. The stored contrasts are retained. Objects fitted on different cases must be refitted explicitly. An explicit matrix supplies encoded directions without an intercept, in data row order; when it has row names they must match data before case selection.

F_H always uses the composite mean and its temperature, including when measurement_scale = "item" changes the noncentrality denominator. T_q must therefore be supplied on the composite-mean response scale. Zero temperature returns singular_temperature and NA estimates/F_H, never a boundary estimate.

Value

A lambda_model with raw/constrained noncentralities, basis, case provenance, model metadata (formula, columns, term assignment, contrasts, factor levels, rank), and parsimony containing RSS, costs, volume and F_H.

Examples

```
set.seed(17)
d <- data.frame(x = rnorm(40), group = factor(rep(c("a", "b"), 20)))
d$i1 <- d$x + rnorm(40)
d$i2 <- d$x + rnorm(40)
lambda_model(d, ~ group + x, c("i1", "i2"), T_q = 1/12)
```

lambda_paired

Evaluate paired mean differences from parallel measurement pairs

Description

Forms after-before within each person and each matching replicate pair, then calls lambda_model with a lambda_reference design. Temperature is estimated from the difference items, retaining homogeneous within-pair error covariance. Different replicate-pair errors must be independent. Persons and both occasions travel together in case or cluster resampling.

Usage

```
lambda_paired(
  data,
  before,
  after,
  T_q,
  value = 0,
  model = ~1,
  nuisance = NULL,
  contrasts = NULL,
  nuisance_contrasts = NULL,
  encoding = c("reevaluate", "fixed"),
  ...
)
```

Arguments

<code>data</code>	Common input data frame or matrix, with unique row identities.
<code>before, after</code>	Character vectors of matching scored measurement columns, in pair order. Each has the same length $K \geq 2$; columns cannot be reused.
<code>T_q</code>	Positive resolution variance explicitly on the composite-mean difference scale. It is not inferred or doubled from occasion resolutions.
<code>value</code>	Known scalar reference, or finite numeric vector in input-row order. If the vector is named, its names must match the row identities.
<code>model</code>	Formula or explicit encoded matrix. Formula intercepts are retained. lm sources are not accepted; declare the reference recipe explicitly.
<code>nuisance</code>	Optional common one-sided nuisance formula or numeric matrix.
<code>contrasts</code>	Explicit formula contrast list, as in <code>lambda_model()</code> .
<code>nuisance_contrasts</code>	Contrast list for the nuisance formula.
<code>encoding</code>	Bootstrap basis policy. "reevaluate" (default) evaluates sample-dependent transforms in each draw. "fixed" learns the encoding after common case selection and resamples its rows, including nuisance terms. Persons are still resampled; this does not fix the observed X.
<code>...</code>	Other <code>lambda_model</code> settings, including <code>bootstrap</code> , <code>na.action</code> , <code>measurement_scale</code> , <code>capacity</code> and <code>numerical_control</code> .

Value

A `lambda_model` plus paired metadata: column pairs, after-minus-before coefficients and the difference matrix in original input-row order. The default `~1` estimates captured mean departure from value. `T=0` is singular.

lambda_panel	<i>Encode fixed models on person-by-occasion observations</i>
--------------	---

Description

Each row is one person-occasion observation and indicators are parallel measurements within that row. Occasions are not treated as parallel repeats of an unchanged response. The measurement must be homogeneous and isotropic across the retained observation space, with independent parallel replicate errors. Person bootstrap does not correct serial measurement error.

Usage

```
lambda_panel(
  data,
  model = NULL,
  person,
  occasion,
  target = NULL,
  adjust = character(),
  person_effects = c("adjust", "none"),
  contrasts = NULL,
  nuisance = NULL,
  nuisance_contrasts = NULL,
  encoding = c("reevaluate", "fixed")
)
```

Arguments

data	Common input data frame or matrix, with unique row identities.
model	Formula or encoded matrix. NULL encodes a categorical occasion effect. For a numeric time trend supply an explicit formula.
person, occasion	Single column names identifying persons and occasions. Each pair must be unique and nonmissing. Missing observation rows are allowed.
target	Optional exact full-model term labels, as in lambda_hypothesis.
adjust	Explicit adjustment terms when target is supplied.
person_effects	"adjust" (default) removes person intercepts, or "none".
contrasts	Explicit formula contrast list, as in lambda_model().
nuisance	Optional common one-sided nuisance formula or numeric matrix.
nuisance_contrasts	Contrast list for the nuisance formula.
encoding	Bootstrap basis policy. "reevaluate" (default) evaluates sample-dependent transforms in each draw. "fixed" learns the encoding after common case selection and resamples its rows, including nuisance terms. Persons are still resampled; this does not fix the observed X.

Details

Person intercepts are adjusted by default, using group demeaning without a dense dummy matrix. Additional nuisance columns are adjusted in that space. Set `person_effects="none"` to encode person effects explicitly in model. The person column is treated as a categorical identity, including numeric IDs. Formula columns involving person are rebuilt for distinct sampled copies; fixed encoding of such formulas is rejected. Other fixed bases keep their declared encoding. Explicit matrices encode row-bound non-person directions.

With `bootstrap=TRUE` in `lambda_model/lambda_compare`, whole persons are sampled with replacement. An explicit bootstrap control must request cluster resampling with `cluster=person`. Strata must be constant within persons. Duplicate sampled persons receive distinct draw-level identities. Unbalanced observation counts are allowed, so the number of observation rows and retained `df` can vary. `na.action="complete"` excludes an entire person if any required row is incomplete. At least two persons, each with at least two observed occasions, are required.

The default paper estimator, numerical `F_H` and percentile rules are retained. This does not extend the optional population estimator. Comparisons require the same observation map, person adjustment and nuisance specification.

Value

A row-bound `lambda_design` with a panel observation map. Evaluation returns panel metadata; retained bootstrap indices also have `person_instances` giving each sampled copy's draw-local ID in matching row order.

Examples

```
d <- data.frame(id=rep(1:12,each=3), time=rep(0:2,12))
lambda_panel(d, person="id", occasion="time")
lambda_panel(d, ~time, person="id", occasion="time")
lambda_panel(d, ~id*time, person="id", occasion="time",
  target="id:time", adjust="time")
```

lambda_reference

Encode a hypothesis relative to a known reference

Description

The mean direction is retained: `~1` is a free mean and `~0` is the fixed null value. This extends the centered manuscript calculation to a prespecified affine origin. Formula intercepts are retained, including in an explicitly supplied nuisance formula; use `~0+x` for nuisance `x` without an intercept. Reference values must be known independently of this analysis.

Usage

```
lambda_reference(
  data,
  model = ~1,
  value = 0,
```

```

    nuisance = NULL,
    contrasts = NULL,
    nuisance_contrasts = NULL,
    encoding = c("reevaluate", "fixed")
  )

```

Arguments

<code>data</code>	Common input data frame or matrix, with unique row identities.
<code>model</code>	Formula or explicit encoded matrix. Formula intercepts are retained. lm sources are not accepted; declare the reference recipe explicitly.
<code>value</code>	Known scalar reference, or finite numeric vector in input-row order. If the vector is named, its names must match the row identities.
<code>nuisance</code>	Optional common one-sided nuisance formula or numeric matrix.
<code>contrasts</code>	Explicit formula contrast list, as in <code>lambda_model()</code> .
<code>nuisance_contrasts</code>	Contrast list for the nuisance formula.
<code>encoding</code>	Bootstrap basis policy. "reevaluate" (default) evaluates sample-dependent transforms in each draw. "fixed" learns the encoding after common case selection and resamples its rows, including nuisance terms. Persons are still resampled; this does not fix the observed X.

Details

All compared models must declare the same reference and nuisance recipe. The ordinary centered API is unchanged. This path uses `estimator="paper"`; it does not extend the population estimator or the centered L $\beta=0$ helpers. See the installed derivation at `system.file("doc", "REFERENCE_MEANS.md", package="lambda")`.

Value

A row-bound `lambda_design` with an explicit fixed reference. After evaluation, `reference`, `mean_departure` and retained `df` describe the origin and signed mean departure after any nuisance adjustment. Lambda values quantify squared departures; `F_H` uses the uncentered adjusted coding.

Examples

```

d <- data.frame(x=seq_len(20))
lambda_reference(d, ~1, value=50)
lambda_reference(d, ~0, value=50)

```

```
make_eta_numerical_lookup
```

Build a reusable capacity integral lookup

Description

An adaptively refined mesh approximates eta by linear segments. Their integrals are accumulated and evaluated quadratically inside each segment. Below min_ratio, direct quadrature is used, preventing extrapolation of the first interval into the singular boundary region. Local trapezoidal errors are estimated by refinement; these estimates are not rigorous bounds.

Usage

```
make_eta_numerical_lookup(
    max_ratio,
    min_ratio = max_ratio * 1e-06,
    grid_points = 65L,
    control = lambda_control()
)
```

Arguments

max_ratio	Largest T/T _q covered (strictly positive).
min_ratio	Smallest cached ratio, default max_ratio times 1e-6.
grid_points	Initial mesh size, integer >= 3.
control	Numerical controls.

Value

An eta_numerical_lookup object with nodes, capacities, cumulative integrals, estimated errors, and its actual numerical controls.

```
print.lambda_ci
```

Inspect Lambda Star results

Description

Inspect Lambda Star results

Usage

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

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

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

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

## S3 method for class 'lambda_measurement'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'lambda_model'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'lambda_comparison'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'lambda_summary'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

Arguments

digits	Number of significant digits for printing.
...	Reserved for S3 compatibility.
object, x	A Lambda Star result or summary.

Value

Summary methods return inspectable summary lists. Print methods return their input invisibly. Full encoded matrices remain in the original results.

project_onto_basis	<i>Project a vector using an orthonormal basis</i>
--------------------	--

Description

Project a vector using an orthonormal basis

Usage

```
project_onto_basis(x, q)
```

Arguments

x	Finite numeric vector.
q	Orthonormal matrix with the same number of rows as x.

Value

Numeric projected vector. No N by N projector is constructed.

```
thermodynamic_model_state
```

Evaluate the parsimony functional for one encoded hypothesis

Description

The response and supplied temperature must describe the same measurement representation. For a mean of parallel items use its composite_mean temperature, not the single-item temperature. The encoded X determines volume; its orthonormal basis is used only for projection.

Usage

```
thermodynamic_model_state(
  response,
  x,
  temperature,
  resolution,
  model = "H",
  control = lambda_control(),
  eta_method = c("numerical", "smooth"),
  eta_lookup = NULL,
  boundary = FALSE,
  nuisance = NULL,
  reference = NULL
)
```

Arguments

response	Finite numeric response vector.
x	Explicit hypothesis matrix on response rows. Omit an intercept for ordinary centered analyses; retain it explicitly to evaluate a reference mean.
temperature	Nonnegative analysis-scale background variance.
resolution	Positive T _q on the same response scale.

model	Descriptive label for this single model.
control	Numerical controls.
eta_method	Numerical capacity (default) or the labeled smooth approximation.
eta_lookup	Optional lookup for numerical integration.
boundary	Explicitly request the mathematical zero-temperature limit.
nuisance	Optional numeric nuisance design on the response rows. Response and hypothesis directions are projected off this common space; an intercept is implicit only for ordinary centered analyses.
reference	NULL uses the centered reference. A known scalar or numeric vector retains departures from that origin without an implicit intercept.

Value

List containing status, model, n, df_h, df_perp, rss, log_volume, volume, temperature, resolution, xi_eta, xi_a, dimension_cost, geometry_cost, free_energy, numerical diagnostics and encoded basis. At T=0 regular estimation returns NA for the thermal terms and F_H. boundary=TRUE instead labels the result boundary_only; it is not inference.

xi_eta_numerical	<i>Integrate numerical capacity over temperature</i>
------------------	--

Description

Integrate numerical capacity over temperature

Usage

```
xi_eta_numerical(
  temperature,
  resolution,
  lookup = NULL,
  control = lambda_control(),
  details = FALSE
)
```

Arguments

temperature	Nonnegative temperature, scalar or vector.
resolution	Positive T _q , scalar or vector.
lookup	Optional reusable lookup; otherwise direct adaptive quadrature.
control	Numerical controls. A reused lookup must meet these settings.
details	Return values, estimated absolute errors and evaluation methods.

Value

Numeric integral values, or a list containing value, abs_error, method and control. Lookup error estimates describe mesh discretization; solver/truncation tolerances are recorded separately in control.

Index

center_columns, 2
confint.lambda_comparison
 (confint.lambda_measurement), 3
confint.lambda_measurement, 3
confint.lambda_measurement(), 18
confint.lambda_model
 (confint.lambda_measurement), 3

design_volume, 4

estimate_lambda_partition_items, 4
estimate_lambda_star_items, 5
eta_app, 6
eta_numerical, 7

geometry_activation(eta_app), 6

hypothesis_basis, 7

lambda_bootstrap_control, 9
lambda_bootstrap_control(), 11, 18, 20, 22

lambda_compare, 10
lambda_compare(), 8
lambda_contrast, 12
lambda_contrast(), 8, 15, 21
lambda_control, 13
lambda_control(), 8
lambda_design, 14
lambda_design(), 8, 21
lambda_hypothesis, 15
lambda_hypothesis(), 8, 15, 21
lambda_inference, 3, 8, 11, 17, 20, 22
lambda_marginal_contrast, 18
lambda_marginal_contrast(), 8, 15, 21
lambda_measure, 20
lambda_measure(), 8
lambda_model, 21
lambda_model(), 8
lambda_paired, 23
lambda_paired(), 8, 18

lambda_panel, 25
lambda_panel(), 8, 21
lambda_reference, 26
lambda_reference(), 8, 18, 21
lambda_results(print.lambda_ci), 28
lambdastar, 8
lambdastar-package (lambdastar), 8

make_eta_numerical_lookup, 28

print.lambda_ci, 28
print.lambda_comparison
 (print.lambda_ci), 28
print.lambda_comparison_summary
 (print.lambda_ci), 28
print.lambda_design(lambda_design), 14
print.lambda_measurement
 (print.lambda_ci), 28
print.lambda_model(print.lambda_ci), 28
print.lambda_summary(print.lambda_ci), 28

project_onto_basis, 29

summary.lambda_comparison
 (print.lambda_ci), 28
summary.lambda_measurement
 (print.lambda_ci), 28
summary.lambda_model(print.lambda_ci), 28

thermodynamic_model_state, 30

xi_a(eta_app), 6
xi_eta_app(eta_app), 6
xi_eta_numerical, 31