Package 'UniLindleyApprox'

August 6, 2026

Type Package

Title Bayesian Point Estimation Using Lindley's Approximation Under Censoring Schemes

Version 0.1.0

Description Performs Bayesian point estimation using Lindley's Approximation (1980) <doi:10.1111/j.2517-6161.1980.tb01102.x> for arbitrary univariate probability distributions under numerous censoring and truncation schemes. Users supply the probability density function (PDF), cumulative distribution function (CDF), survival function, log-prior density, initial parameter vector, support bounds, and observed data; the package automatically computes Bayesian point estimates under various loss functions using Lindley's approximation. Supported schemes include complete data, right censoring, left censoring, interval censoring, random censoring, block random censoring, Type-I censoring, Type-II censoring, progressive Type-II censoring, progressive first failure censoring, joint Type-I censoring, joint Type-II censoring, balanced joint progressive Type-II censoring, hybrid censoring, hybrid Type-I censoring, hybrid Type-II censoring, Type-I hybrid censoring, Type-II progressively hybrid censoring, doubly Type-II censoring, middle censoring, right truncation, and left truncation. The package computes posterior expectations of arbitrary smooth functions, supports multiple loss functions (squared error loss function (SELF), weighted squared error loss function (WSELF), modified quadratic squared error loss function (MQSELF), precautionary loss function (PLF), entropy loss function (ELF), linear-exponential (LINEX), generalized entropy loss function (GELF), Kullback-Leibler loss function (K-Loss), and user-defined), provides model selection criteria (Akaike information criterion (AIC), Bayesian information criterion (BIC), corrected Akaike information criterion (AICc), Hannan-Quinn information criterion (HQIC), consistent Akaike information criterion (CAIC), Kullback information criterion (KIC)), goodness-of-fit statistics (Kolmogorov-Smirnov, Anderson-Darling, Cramer-von Mises, Watson, Chi-square), residual analysis (Cox-Snell, Martingale, Deviance, Pearson, Generalized, Randomized quantile), comprehensive visualization tools, prediction utilities, and simulation functions for benchmarking estimators. Methods are described in Lindley (1980) <doi:10.1111/j.2517-6161.1980.tb01102.x>, Tierney and Kadane (1986) <doi:10.2307/2234555>, Tierney, Kass, and Kadane (1989) <doi:10.2307/2335663>, Nagar, Kumar, and Krishna (2026)

<doi:10.59467/IJASS.2026.22.1>, Goel, Kumar, and Krishna (2026, ``Estimation in power Lindley distributions using balanced joint progressively Type-II censored data''), Wu and Kus (2009) <doi:10.1016/j.csda.2009.03.010>, Goel and Krishna (2026) <doi:10.1007/s13198-026-03208-w>, Balakrishnan and Aggarwala (2000, ISBN:978-1-4612-1334-5), Mondal and Kundu (2020) <doi:10.1080/03610926.2018.1554128>, Ding and Gui (2023) <doi:10.3390/math11092003>, Prajapati, Mitra, and Kundu (2019) <doi:10.1007/s13571-018-0167-0>, Yadav, Jaiswal, and Yadav (2026) <doi:10.1007/s11135-026-02647-8>, Iyer, Jammalamadaka, and Kundu (2008) <doi:10.1016/j.jspi.2007.03.062>, Banerjee and Kundu (2008) <doi:10.1109/TR.2008.916890>, and Kundu and Joarder (2006) <doi:10.1016/j.csda.2005.05.002>.

License GPL-3

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices, utils, methods, numDeriv, MASS

Suggests testthat (>= 3.0.0), ggplot2, future, parallel, knitr, rmarkdown

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Shikhar Tyagi [aut, cre] (ORCID: <https://orcid.org/0000-0003-1606-0844>), Arvind Pandey [aut], Bhupendra Singh [aut], Vrijesh Tripathi [aut]

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-08-06 13:50:13 UTC

Contents

`UniLindleyApprox-package`

UniLindleyApprox: Bayesian Point Estimation Using Lindley's Approximation

---

Description

The `UniLindleyApprox` package provides a generalized framework for Bayesian parameter estimation using Lindley's Approximation (1980) for arbitrary univariate probability distributions under complete, censored, and truncated data.

Main Function

The main function is `lindley_fit` which performs Bayesian estimation using Lindley's approximation.

Censoring Schemes

The package supports 23 censoring and truncation schemes including:

complete Complete (uncensored) data

right Right censoring

left Left censoring

interval Interval censoring
random Random censoring
type1 Type-I censoring
type2 Type-II censoring
progressive_type2 Progressive Type-II censoring
progressive_first_failure Progressive first failure censoring
joint_type1 Joint Type-I censoring
joint_type2 Joint Type-II censoring
bjpt2 Balanced joint progressive Type-II censoring
hybrid Hybrid censoring
hybrid_type1 Hybrid Type-I censoring
hybrid_type2 Hybrid Type-II censoring
type1_hybrid Type-I hybrid censoring
type2_progressively_hybrid Type-II progressively hybrid censoring
doubly_type2 Doubly Type-II censoring
middle Middle censoring
right_truncation Right truncation
left_truncation Left truncation

Loss Functions

The package computes Bayes estimates under various loss functions:

SELF Squared Error Loss Function
WSELF Weighted Squared Error Loss Function
MQSELF Modified Quadratic Squared Error Loss Function
PLF Precautionary Loss Function
ELF Entropy Loss Function
LINEX Linear Exponential Loss Function
GELF General Entropy Loss Function
K-Loss K-Loss Function

Prior Distributions

Supported prior distributions include:

Gamma `log_prior_gamma`
Beta `log_prior_beta`
Normal `log_prior_normal`
Uniform `log_prior_uniform`
Lognormal `log_prior_lognormal`
Inverse Gamma `log_prior_invgamma`
Weibull `log_prior_weibull`
Exponential `log_prior_exponential`
Jeffreys `log_prior_jeffreys`

Model Selection

Model selection criteria include AIC, AICc, BIC, HQIC, CAIC, and KIC.

Goodness-of-Fit

Goodness-of-fit statistics include KS, AD, CvM, Watson, and Chi-square tests.

Residuals

Residual types include Cox-Snell, Martingale, Deviance, Pearson, Generalized, and Randomized quantile residuals.

Visualization

Visualization functions include posterior surface plots, likelihood profiles, QQ plots, PP plots, and residual plots.

Simulation

Simulation utilities are available for all censoring schemes for benchmarking.

Author(s)

Maintainer: Shikhar Tyagi <shikhar1093tyagi@gmail.com> (ORCID)

Authors:

- Arvind Pandey <arvindmzu@gmail.com>
- Bhupendra Singh <bhupendra.rana@gmail.com>
- Vrijesh Tripathi <vrijesh.tripathi@uwi.edu>

References

Lindley, D. V. (1980). Approximate Bayesian methods. Trabajos de Estadistica y de Investigacion Operativa, 31(1), 223-245.

Tierney, L., & Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. Journal of the American Statistical Association, 81(393), 82-86.

Tierney, L., Kass, R. E., & Kadane, J. B. (1989). Fully exponential Laplace approximations to expectations and variances of nonpositive functions. Journal of the American Statistical Association, 84(407), 710-716.

AIC.lindleyfit *AIC Method for lindleyfit Objects*

Description

AIC Method for lindleyfit Objects

Usage

```
## S3 method for class 'lindleyfit'
AIC(object, ...)
```

Arguments

object	A lindleyfit object.
...	Additional arguments (unused).

Value

AIC value.

AICc *AIC Method for lindleyfit Objects Corrected Akaike Information Criterion (AICc)*

Description

Computes the corrected AIC for fitted models.

Usage

```
AICc(object, ...)
```

Arguments

object	Model fit object.
...	Additional arguments.

Value

AICc value.

AICc.lindleyfit *AICc Method for lindleyfit Objects*

Description

AICc Method for lindleyfit Objects

Usage

```
## S3 method for class 'lindleyfit'
AICc(object, ...)
```

Arguments

object	A lindleyfit object.
...	Additional arguments (unused).

Value

AICc value.

autoplot.lindleyfit *Autoplot Method for lindleyfit Objects*

Description

Creates a comprehensive set of diagnostic plots using ggplot2 if available.

Usage

```
autoplot.lindleyfit(object, which = 1:4, ...)
```

Arguments

object	A lindleyfit object.
which	Which plots to display.
...	Additional arguments.

Value

A ggplot object or list of ggplot objects.

bayes_estimate *Compute Bayes Estimate Under Specified Loss Function*

Description

Computes the Bayes estimate that minimizes the expected loss.

Usage

```
bayes_estimate(fit, loss = "SELF", ...)
```

Arguments

fit	A lindleyfit object.
loss	Loss function name or custom function.
...	Additional parameters for loss function.

Value

Bayes estimate under specified loss function.

BIC.lindleyfit *BIC Method for lindleyfit Objects*

Description

BIC Method for lindleyfit Objects

Usage

```
## S3 method for class 'lindleyfit'
BIC(object, ...)
```

Arguments

object	A lindleyfit object.
...	Additional arguments (unused).

Value

BIC value.

CAIC *Consistent Akaike Information Criterion (CAIC)*

Description

Computes the CAIC for fitted models.

Usage

```
CAIC(object, ...)
```

Arguments

object	Model fit object.
...	Additional arguments.

Value

CAIC value.

CAIC.lindleyfit *CAIC Method for lindleyfit Objects*

Description

Consistent AIC.

Usage

```
## S3 method for class 'lindleyfit'
CAIC(object, ...)
```

Arguments

object	A lindleyfit object.
...	Additional arguments (unused).

Value

CAIC value.

coef.lindleyfit *Coefficient Method for lindleyfit Objects*

Description

Extracts the Bayes estimate.

Usage

```
## S3 method for class 'lindleyfit'
coef(object, ...)
```

Arguments

object	A lindleyfit object.
...	Additional arguments (unused).

Value

Parameter estimates.

compare_predictions *Compare Predictions*

Description

Compares predictions from multiple fitted models.

Usage

```
compare_predictions(..., newdata, type = "density")
```

Arguments

...	List of lindleyfit objects.
newdata	New data points for prediction.
type	Type of prediction.

Value

Data frame with predictions from each model.

diagnostics *Extract Diagnostic Information*

Description

Extracts computational and convergence diagnostics from a fitted model.

Usage

```
diagnostics(object, ...)
```

Arguments

object	Model fit object.
...	Additional arguments.

Value

Diagnostic list.

diagnostics.lindleyfit *Diagnostics Method for lindleyfit Objects*

Description

Computes diagnostic information for the fitted model.

Usage

```
## S3 method for class 'lindleyfit'
diagnostics(object, ...)
```

Arguments

object	A lindleyfit object.
...	Additional arguments (unused).

Value

A list of diagnostic information.

fitted.lindleyfit *Fitted Method for lindleyfit Objects*

Description

Computes fitted values (density, CDF, survival) at data points.

Usage

```
## S3 method for class 'lindleyfit'
fitted(object, type = "density", ...)
```

Arguments

object	A lindleyfit object.
type	Type of fitted values: "density", "cdf", "survival", or "hazard".
...	Additional arguments (unused).

Value

Fitted values.

gof_ad *Anderson-Darling Statistic*

Description

Computes the Anderson-Darling goodness-of-fit statistic.

Usage

```
gof_ad(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments (unused).

Value

AD statistic value.

gof_chisq *Chi-Square Goodness-of-Fit Statistic*

Description

Computes the chi-square goodness-of-fit statistic for binned data.

Usage

```
gof_chisq(fit, bins = NULL, ...)
```

Arguments

fit	A lindleyfit object.
bins	Number of bins or custom bin boundaries.
...	Additional arguments.

Value

Chi-square statistic and p-value.

gof_cvm *Cramér-von Mises Statistic*

Description

Computes the Cramér-von Mises goodness-of-fit statistic.

Usage

```
gof_cvm(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments (unused).

Value

CvM statistic value.

gof_ks *Kolmogorov-Smirnov Statistic*

Description

Computes the Kolmogorov-Smirnov goodness-of-fit statistic.

Usage

```
gof_ks(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments (unused).

Value

KS statistic value.

gof_stats *Goodness-of-Fit Statistics*

Description

Computes various goodness-of-fit statistics for the fitted model.

Usage

```
gof_stats(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments.

Value

A list of goodness-of-fit statistics.

gof_watson *Watson Statistic*

Description

Computes the Watson goodness-of-fit statistic (a variant of Cramér-von Mises).

Usage

```
gof_watson(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments (unused).

Value

Watson statistic value.

HQIC *Hannan-Quinn Information Criterion (HQIC)*

Description

Computes the HQIC for fitted models.

Usage

```
HQIC(object, ...)
```

Arguments

object	Model fit object.
...	Additional arguments.

Value

HQIC value.

HQIC.lindleyfit *HQIC Method for lindleyfit Objects*

Description

Hannan-Quinn Information Criterion.

Usage

```
## S3 method for class 'lindleyfit'
HQIC(object, ...)
```

Arguments

object	A lindleyfit object.
...	Additional arguments (unused).

Value

HQIC value.

independent_prior *Create Independent Product Prior*

Description

Creates a log-prior for independent priors on each parameter.

Usage

```
independent_prior(prior_list, ...)
```

Arguments

prior_list	List of prior functions, one for each parameter.
...	Additional arguments passed to each prior.

Value

Combined log-prior function.

KIC *Kullback Information Criterion (KIC)*

Description

Computes the KIC for fitted models.

Usage

```
KIC(object, ...)
```

Arguments

object	Model fit object.
...	Additional arguments.

Value

KIC value.

KIC.lindleyfit *KIC Method for lindleyfit Objects*

Description

Kullback Information Criterion.

Usage

```
## S3 method for class 'lindleyfit'
KIC(object, ...)
```

Arguments

object	A lindleyfit object.
...	Additional arguments (unused).

Value

KIC value.

`lindley.control` *Control Parameters for Lindley Approximation*

Description

Creates a list of control parameters for the Lindley approximation algorithm.

Usage

```
lindley.control(
  method = "BFGS",
  maxit = 1000,
  reltol = 1e-08,
  abstol = 1e-10,
  hessian = TRUE,
  third_deriv = TRUE,
  deriv_method = "numDeriv",
  eps = 1e-06,
  parallel = FALSE,
  ncores = 2,
  verbose = TRUE,
  trace = 0,
  check_prior = TRUE,
  check_support = TRUE,
  check_hessian = TRUE,
  cache_derivatives = TRUE,
  use_analytical = TRUE
)
```

Arguments

`method`	Optimization method. Options: "BFGS", "L-BFGS-B", "Nelder-Mead", "CG", "SANN", "Brent", "Newton-Raphson". Default: "BFGS".
`maxit`	Maximum number of iterations for optimization. Default: 1000.
`reltol`	Relative convergence tolerance. Default: 1e-8.
`abstol`	Absolute convergence tolerance. Default: 1e-10.
`hessian`	Logical; whether to compute the Hessian matrix. Default: TRUE.
`third_deriv`	Logical; whether to compute third-order derivatives. Default: TRUE.
`deriv_method`	Method for numerical differentiation. Options: "numDeriv", "finite". Default: "numDeriv".
`eps`	Step size for finite differences. Default: 1e-6.
`parallel`	Logical; whether to use parallel computation. Default: FALSE.
`ncores`	Number of cores for parallel computation. Default: 2.
`verbose`	Logical; whether to print progress messages. Default: TRUE.

`trace`	Integer; level of tracing information. Default: 0.
`check_prior`	Logical; whether to validate prior density. Default: TRUE.
`check_support`	Logical; whether to validate parameter support. Default: TRUE.
`check_hessian`	Logical; whether to check Hessian positive definiteness. Default: TRUE.
`cache_derivatives`	Logical; whether to cache derivative computations. Default: TRUE.
`use_analytical`	Logical; whether to use analytical derivatives if provided. Default: TRUE.

Value

A list of control parameters for use in `lindley_fit`.

Examples

```
ctrl <- lindley.control(method = "BFGS", maxit = 500, verbose = FALSE)
```

---

lindley_fit *Bayesian Estimation Using Lindley's Approximation*

Description

Main function for performing Bayesian parameter estimation using Lindley's approximation for arbitrary univariate distributions under various censoring schemes.

Usage

```
lindley_fit(
  data,
  pdf,
  cdf,
  survival,
  log_prior,
  theta0,
  scheme = "complete",
  loss = "SELF",
  control = lindley.control(),
  scheme_params = NULL,
  ...
)
```

Arguments

`data`	Observed data. Format depends on censoring scheme.
`pdf`	Probability density function. Must accept (x, theta) where theta is a parameter vector.
`cdf`	Cumulative distribution function. Must accept (x, theta).
`survival`	Survival function (1 - CDF). Must accept (x, theta).
`log_prior`	Log-prior density function. Must accept theta and return log-density.
`theta0`	Initial parameter vector.
`scheme`	Censoring scheme. See details for supported schemes.
`loss`	Loss function for Bayes estimate. Default: "SELF".
`control`	Control parameters from `lindley.control`.
`scheme_params`	Additional parameters specific to censoring scheme.
`...`	Additional arguments passed to prior and loss functions.

Value

An object of class `lindleyfit` containing:

map_estimate Posterior mode (MAP estimate)

bayes_estimate Bayes estimate under specified loss function

posterior_mean Posterior mean approximation

hessian Hessian matrix at posterior mode

gradient Gradient vector at posterior mode

third_deriv Third-order derivative tensor

log_likelihood Log-likelihood at MAP

log_posterior Log-posterior at MAP

convergence Optimization convergence status

iterations Number of optimization iterations

data Input data

scheme Censoring scheme used

loss Loss function used

control Control parameters

elapsed_time Computation time

Supported Censoring Schemes

complete Complete (uncensored) data

right Right censoring

left Left censoring

interval Interval censoring

random Random censoring

type1 Type-I censoring

type2 Type-II censoring

progressive_type2 Progressive Type-II censoring

progressive_first_failure Progressive first failure censoring

joint_type1 Joint Type-I censoring

joint_type2 Joint Type-II censoring

bjpt2 Balanced joint progressive Type-II censoring

hybrid Hybrid censoring

hybrid_type1 Hybrid Type-I censoring

hybrid_type2 Hybrid Type-II censoring

type1_hybrid Type-I hybrid censoring

type2_progressively_hybrid Type-II progressively hybrid censoring

doubly_type2 Doubly Type-II censoring

middle Middle censoring

right_truncation Right truncation

left_truncation Left truncation

Supported Loss Functions

SELF Squared Error Loss Function

WSELF Weighted Squared Error Loss Function

MQSELF Modified Quadratic Squared Error Loss Function

PLF Precautionary Loss Function

ELF Entropy Loss Function

LINEX Linear Exponential Loss Function

GELF General Entropy Loss Function

K-Loss K-Loss Function

Examples

```
# Exponential distribution with complete data
set.seed(123)
x <- rexp(50, rate = 2)

# Define probability functions
dexp_custom <- function(x, theta) dexp(x, rate = theta[1])
pexp_custom <- function(x, theta) pexp(x, rate = theta[1])
sexp_custom <- function(x, theta) 1 - pexp(x, rate = theta[1])

# Define log-prior (Gamma prior for rate)
logprior <- function(theta) {
  dgamma(theta[1], shape = 2, rate = 1, log = TRUE)
```

```
}

# Fit model
fit <- lindley_fit(
  data = x,
  pdf = dexp_custom,
  cdf = pexp_custom,
  survival = sexp_custom,
  log_prior = logprior,
  theta0 = c(1),
  scheme = "complete",
  loss = "SELF"
)

# Print results
print(fit)
```

`logLik.lindleyfit` *Log-Likelihood Method for lindleyfit Objects*

Description

Log-Likelihood Method for lindleyfit Objects

Usage

```
## S3 method for class 'lindleyfit'
logLik(object, ...)
```

Arguments

`object`	A lindleyfit object.
`...`	Additional arguments (unused).

Value

Log-likelihood value.

log_priors *Prior Distributions for Bayesian Estimation*

Description

Provides log-prior density functions for common prior distributions.

Jeffreys prior is proportional to sqrt(det(I(theta))), where I(theta) is the Fisher information matrix. For common distributions, this simplifies to known forms.

Usage

```
log_prior_gamma(theta, shape = 1, rate = 1, ...)

log_prior_beta(theta, shape1 = 1, shape2 = 1, ...)

log_prior_normal(theta, mean = 0, sd = 1, ...)

log_prior_uniform(theta, min = 0, max = 1, ...)

log_prior_lognormal(theta, meanlog = 0, sdlog = 1, ...)

log_prior_invgamma(theta, shape = 1, scale = 1, ...)

log_prior_weibull(theta, shape = 1, scale = 1, ...)

log_prior_exponential(theta, rate = 1, ...)

log_prior_jeffreys(theta, distribution = "exponential", ...)
```

Arguments

theta	Parameter value.
shape	Shape parameter (default: 1).
rate	Rate parameter (default: 1).
...	Additional parameters.
shape1	First shape parameter (default: 1).
shape2	Second shape parameter (default: 1).
mean	Mean parameter (default: 0).
sd	Standard deviation parameter (default: 1).
min	Lower bound (default: 0).
max	Upper bound (default: 1).
meanlog	Mean of log-scale (default: 0).
sdlog	SD of log-scale (default: 1).
scale	Scale parameter (default: 1).
distribution	Distribution name for Jeffreys prior.

Value

Log-prior density value.

Log-prior density.

Log-prior density.

Log-prior density.

Log-prior density.

Log-prior density.

Log-prior density.

Log-prior density.

Log-prior density.

Log-prior density.

`log_prior_mvn` *Create Multivariate Normal Prior*

Description

Create Multivariate Normal Prior

Usage

```
log_prior_mvn(theta, mean, sigma, ...)
```

Arguments

`theta`	Parameter vector.
`mean`	Mean vector.
`sigma`	Covariance matrix.
`...`	Additional parameters (unused).

Value

Log-prior density.

loss_functions *Loss Functions for Bayesian Estimation*

Description

Implements various loss functions for computing Bayes estimates.

Usage

```
loss_self(theta, estimate, ...)

loss_wself(theta, estimate, w = 1, ...)

loss_mqself(theta, estimate, k = 1, ...)

loss_plf(theta, estimate, a = 1, ...)

loss_elf(theta, estimate, ...)

loss_linex(theta, estimate, a = 1, ...)

loss_gelf(theta, estimate, q = 2, ...)

loss_kloss(theta, estimate, ...)
```

Arguments

theta	True parameter value.
estimate	Estimated value.
...	Additional parameters (unused).
w	Weight parameter (default: 1).
k	Shape parameter (default: 1).
a	Shape parameter (default: 1).
q	Shape parameter (default: 2).

Value

Loss value.

Loss value: (theta - estimate)^2

Loss value: w * (theta - estimate)^2

Loss value: (theta - estimate)^2 / (k + theta^2)

Loss value: (estimate/theta)^a - a * log(estimate/theta) - 1

Loss value: estimate/theta - log(estimate/theta) - 1

Loss value: exp(a * (estimate - theta)) - a * (estimate - theta) - 1

Loss value: (estimate/theta)^q / q - (estimate/theta) + 1/q

Loss value: (theta - estimate)^2 / (theta * estimate)

`mae` *Mean Absolute Error*

Description

Computes mean absolute error between observed and fitted quantities.

Usage

```
mae(fit, type = "density", ...)
```

Arguments

`fit`	A lindleyfit object.
`type`	Type of comparison: "density", "cdf", "survival".
`...`	Additional arguments.

Value

MAE value.

`mse` *Mean Squared Error*

Description

Computes mean squared error between observed and fitted quantities.

Usage

```
mse(fit, type = "density", ...)
```

Arguments

`fit`	A lindleyfit object.
`type`	Type of comparison: "density", "cdf", "survival".
`...`	Additional arguments.

Value

MSE value.

plot.lindleyfit *Plot Method for lindleyfit Objects*

Description

Creates diagnostic plots for the fitted model.

Usage

```
## S3 method for class 'lindleyfit'
plot(x, which = 1:4, ask = TRUE, ...)
```

Arguments

x	A lindleyfit object.
which	Which plots to display: 1=density overlay, 2=QQ plot, 3=PP plot, 4=residuals.
ask	Logical; whether to pause between plots.
...	Additional graphical parameters.

Value

Invisibly returns the object.

plots *Visualization Functions for UniLindleyApprox*

Description

Functions for creating diagnostic and exploratory plots.

plot_fitted *Plot Fitted Distribution*

Description

Plots the fitted distribution over the data histogram.

Usage

```
plot_fitted(fit, type = "density", n = 100, ...)
```

Arguments

fit	A lindleyfit object.
type	Type of plot: "density", "cdf", "survival".
n	Number of points for fitted curve.
...	Additional graphical parameters.

Value

Invisibly returns the fitted values.

---

plot_likelihood_surface

Plot Likelihood Surface

Description

Creates a contour or surface plot of the likelihood function.

Usage

```
plot_likelihood_surface(
  fit,
  theta1_range = NULL,
  theta2_range = NULL,
  n = 50,
  type = "contour",
  ...
)
```

Arguments

fit	A lindleyfit object.
theta1_range	Range for first parameter.
theta2_range	Range for second parameter (for 2D plots).
n	Grid resolution.
type	Plot type: "contour" or "surface".
...	Additional graphical parameters.

Value

Invisibly returns the grid of likelihood values.

plot_posterior_surface *Plot Posterior Surface*

Description

Creates a contour or surface plot of the posterior distribution.

Usage

```
plot_posterior_surface(
  fit,
  theta1_range = NULL,
  theta2_range = NULL,
  n = 50,
  type = "contour",
  ...
)
```

Arguments

fit	A lindleyfit object.
theta1_range	Range for first parameter.
theta2_range	Range for second parameter (for 2D plots).
n	Grid resolution.
type	Plot type: "contour" or "surface".
...	Additional graphical parameters.

Value

Invisibly returns the grid of posterior values.

plot_prior_posterior *Plot Prior vs Posterior*

Description

Compares prior and posterior distributions.

Usage

```
plot_prior_posterior(fit, param_range = NULL, n = 100, ...)
```

Arguments

fit	A lindleyfit object.
param_range	Range for parameter.
n	Number of points.
...	Additional graphical parameters.

Value

Invisibly returns the comparison data.

---

plot_profile *Plot Log-Likelihood Profile*

Description

Creates a profile log-likelihood plot for each parameter.

Usage

```
plot_profile(fit, param_range = NULL, n = 50, ...)
```

Arguments

fit	A lindleyfit object.
param_range	Range for each parameter (list or NULL for automatic).
n	Number of points per profile.
...	Additional graphical parameters.

Value

Invisibly returns the profile data.

plot_residuals *Plot Residuals*

Description

Creates various residual plots.

Usage

```
plot_residuals(fit, type = "cox_snell", which = 1:3, ...)
```

Arguments

fit	A lindleyfit object.
type	Type of residuals: "cox_snell", "martingale", "deviance", "pearson".
which	Which plots: 1=index plot, 2=QQ plot, 3=histogram.
...	Additional graphical parameters.

Value

Invisibly returns the residual values.

posterior_expectation *Compute Posterior Expectation of Arbitrary Function*

Description

General function to compute E[g(theta)|data] using Lindley's approximation.

Usage

```
posterior_expectation(fit, g_fun, ...)
```

Arguments

fit	A lindleyfit object.
g_fun	Function of parameters.
...	Additional arguments.

Value

Posterior expectation approximation.

posterior_mean *Compute Posterior Mean Approximation*

Description

Compute Posterior Mean Approximation

Usage

```
posterior_mean(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments.

Value

Posterior mean approximation.

posterior_median *Compute Posterior Median Approximation*

Description

Approximates posterior median using Lindley's approximation. Note: This is an approximation and may not be accurate for all distributions.

Usage

```
posterior_median(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments.

Value

Posterior median approximation.

predict.lindleyfit *Predict Method for lindleyfit Objects*

Description

Makes predictions for new data.

Usage

```
## S3 method for class 'lindleyfit'
predict(object, newdata = NULL, type = "density", ...)
```

Arguments

object	A lindleyfit object.
newdata	New data points.
type	Type of prediction: "density", "cdf", "survival", "hazard", "quantile".
...	Additional arguments.

Value

Predicted values.

prediction *Prediction Functions for UniLindleyApprox*

Description

Functions for making predictions from fitted models.

prediction_summary *Prediction Summary*

Description

Provides a summary of prediction performance.

Usage

```
prediction_summary(fit, test_data, type = "density")
```

Arguments

fit	A lindleyfit object.
test_data	Test data for evaluation.
type	Type of prediction to evaluate.

Value

List with prediction performance metrics.

---

predict_cumhaz *Predict Cumulative Hazard*

Description

Predicts cumulative hazard values at specified time points.

Usage

```
predict_cumhaz(fit, times, ...)
```

Arguments

fit	A lindleyfit object.
times	Vector of time points.
...	Additional arguments.

Value

Vector of cumulative hazard values.

---

predict_failure_prob *Predict Failure Probability*

Description

Predicts the probability of failure by a specified time.

Usage

```
predict_failure_prob(fit, t, ...)
```

Arguments

fit	A lindleyfit object.
t	Time point.
...	Additional arguments.

Value

Failure probability.

---

predict_hazard *Predict Hazard Function*

Description

Predicts hazard function values at specified time points.

Usage

```
predict_hazard(fit, times, ...)
```

Arguments

fit	A lindleyfit object.
times	Vector of time points.
...	Additional arguments.

Value

Vector of hazard values.

---

predict_interval *Prediction Intervals (Based on Quantiles)*

Description

Computes prediction intervals based on distribution quantiles. Note: This is a prediction interval for future observations, not a confidence interval for parameters (which is outside the scope of Lindley's approximation).

Usage

```
predict_interval(fit, level = 0.95, ...)
```

Arguments

fit	A lindleyfit object.
level	Confidence level (default: 0.95).
...	Additional arguments.

Value

Vector with lower and upper bounds.

---

predict_median *Predict Median Time to Failure*

Description

Predicts the median time to failure from the fitted distribution.

Usage

```
predict_median(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments.

Value

Predicted median time to failure.

---

predict_mttf *Predict Mean Time to Failure*

Description

Predicts the mean time to failure from the fitted distribution.

Usage

```
predict_mttf(fit, ...)
```

Arguments

fit	A lindleyfit object.
...	Additional arguments.

Value

Predicted mean time to failure.

predict_new *Predict for New Data*

Description

Makes predictions for new data points.

Usage

```
predict_new(fit, newdata, type = "density", ...)
```

Arguments

fit	A lindleyfit object.
newdata	New data points.
type	Type of prediction: "density", "cdf", "survival", "hazard".
...	Additional arguments.

Value

Predicted values.

predict_quantiles *Predict Quantiles*

Description

Predicts quantiles of the fitted distribution.

Usage

```
predict_quantiles(fit, probs = c(0.25, 0.5, 0.75), ...)
```

Arguments

fit	A lindleyfit object.
probs	Vector of probabilities.
...	Additional arguments.

Value

Vector of predicted quantiles.

`predict_remaining_lifetime` *Predict Remaining Lifetime*

Description

Predicts the remaining lifetime distribution for a surviving unit.

Usage

```
predict_remaining_lifetime(fit, current_age, ...)
```

Arguments

`fit`	A lindleyfit object.
`current_age`	Current age of the unit.
`...`	Additional arguments.

Value

List with mean and median remaining lifetime.

`predict_survival` *Predict Survival Probabilities*

Description

Predicts survival probabilities at specified time points.

Usage

```
predict_survival(fit, times, ...)
```

Arguments

`fit`	A lindleyfit object.
`times`	Vector of time points.
`...`	Additional arguments.

Value

Vector of survival probabilities.

predict_survival_prob *Predict Probability of Survival to Time T*

Description

Predicts the probability of surviving to a specified time.

Usage

```
predict_survival_prob(fit, t, ...)
```

Arguments

fit	A lindleyfit object.
t	Time point.
...	Additional arguments.

Value

Survival probability.

print.lindleyfit *Print Method for lindleyfit Objects*

Description

Print Method for lindleyfit Objects

Usage

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

Arguments

x	A lindleyfit object.
...	Additional arguments (unused).

Value

Invisibly returns the object.

residuals.lindleyfit *Residuals Method for lindleyfit Objects*

Description

Computes residuals for the fitted model.

Usage

```
## S3 method for class 'lindleyfit'
residuals(object, type = "cox_snell", ...)
```

Arguments

object	A lindleyfit object.
type	Type of residuals: "cox_snell", "martingale", "deviance", "pearson", "generalized", "randomized".
...	Additional arguments.

Value

Residual values.

rmse *Root Mean Squared Error*

Description

Computes root mean squared error between observed and fitted quantities.

Usage

```
rmse(fit, type = "density", ...)
```

Arguments

fit	A lindleyfit object.
type	Type of comparison: "density", "cdf", "survival".
...	Additional arguments.

Value

RMSE value.

simulate_complete *Simulate Complete Data*

Description

Generates complete (uncensored) random samples from a specified distribution.

Usage

```
simulate_complete(n, pdf, cdf, theta, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
theta	True parameter values.
...	Additional arguments.

Value

Vector of simulated observations.

simulate_hybrid *Simulate Hybrid Censored Data*

Description

Generates hybrid censored random samples.

Usage

```
simulate_hybrid(n, pdf, cdf, survival, theta, censoring_time, r, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
censoring_time	Censoring time.
r	Number of failures.
...	Additional arguments.

Value

List with observed data and censoring information.

`simulate_interval` *Simulate Interval Censored Data*

Description

Generates interval censored random samples.

Usage

```
simulate_interval(n, pdf, cdf, survival, theta, interval_width, ...)
```

Arguments

`n`	Sample size.
`pdf`	Probability density function.
`cdf`	Cumulative distribution function.
`survival`	Survival function.
`theta`	True parameter values.
`interval_width`	Width of inspection intervals.
`...`	Additional arguments.

Value

Matrix with lower and upper bounds.

`simulate_left` *Simulate Left Censored Data*

Description

Generates left censored random samples.

Usage

```
simulate_left(n, pdf, cdf, survival, theta, censoring_time, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
censoring_time	Censoring time.
...	Additional arguments.

Value

List with observed and censored components.

---

simulate_left_truncation *Simulate Left Truncated Data*

Description

Generates left truncated random samples.

Usage

```
simulate_left_truncation(n, pdf, cdf, survival, theta, truncation_point, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
truncation_point	Truncation point.
...	Additional arguments.

Value

List with observed data and truncation information.

simulate_middle *Simulate Middle Censored Data*

Description

Generates middle censored random samples.

Usage

```
simulate_middle(n, pdf, cdf, survival, theta, left_bound, right_bound, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
left_bound	Left censoring bound.
right_bound	Right censoring bound.
...	Additional arguments.

Value

List with observed data and censoring information.

simulate_progressive *Simulate Progressive Type-II Censored Data*

Description

Generates progressive Type-II censored random samples.

Usage

```
simulate_progressive(n, pdf, cdf, survival, theta, m, removal_scheme, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
m	Number of failures to observe.
removal_scheme	Removal scheme vector (length m).
...	Additional arguments.

Value

List with observed data and censoring information.

---

simulate_random *Simulate Random Censored Data*

Description

Generates random censored random samples.

Usage

```
simulate_random(
  n,
  pdf,
  cdf,
  survival,
  theta,
  censoring_dist = "exp",
  censoring_params = list(rate = 0.5),
  ...
)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
censoring_dist	Distribution for censoring times.
censoring_params	Parameters for censoring distribution.
...	Additional arguments.

Value

List with observed data and censoring information.

`simulate_right` *Simulate Right Censored Data*

Description

Generates right censored random samples.

Usage

```
simulate_right(n, pdf, cdf, survival, theta, censoring_time, ...)
```

Arguments

`n`	Sample size.
`pdf`	Probability density function.
`cdf`	Cumulative distribution function.
`survival`	Survival function.
`theta`	True parameter values.
`censoring_time`	Censoring time.
`...`	Additional arguments.

Value

List with observed and censored components.

`simulate_right_truncation`
Simulate Right Truncated Data

Description

Generates right truncated random samples.

Usage

```
simulate_right_truncation(n, pdf, cdf, survival, theta, truncation_point, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
truncation_point	Truncation point.
...	Additional arguments.

Value

List with observed data and truncation information.

---

simulate_type1 *Simulate Type-I Censored Data*

Description

Generates Type-I censored random samples.

Usage

```
simulate_type1(n, pdf, cdf, survival, theta, censoring_time, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
censoring_time	Censoring time.
...	Additional arguments.

Value

List with observed data and censoring information.

simulate_type2 *Simulate Type-II Censored Data*

Description

Generates Type-II censored random samples.

Usage

```
simulate_type2(n, pdf, cdf, survival, theta, r, ...)
```

Arguments

n	Sample size.
pdf	Probability density function.
cdf	Cumulative distribution function.
survival	Survival function.
theta	True parameter values.
r	Number of failures to observe.
...	Additional arguments.

Value

List with observed order statistics and censoring information.

simulation *Simulation Utilities for Censoring Schemes*

Description

Functions to generate simulated data under various censoring schemes for benchmarking Bayesian estimators.

`simulation_study` *Simulation Study for Bayesian Estimators*

Description

Performs a simulation study to evaluate Bayesian estimators under repeated sampling.

Usage

```
simulation_study(
  nsim,
  n,
  pdf,
  cdf,
  survival,
  log_prior,
  theta_true,
  theta0,
  scheme = "complete",
  scheme_params = NULL,
  loss = "SELF",
  control = lindley.control(verbose = FALSE),
  parallel = FALSE,
  ncores = 2,
  ...
)
```

Arguments

`nsim`	Number of simulations.
`n`	Sample size per simulation.
`pdf`	Probability density function.
`cdf`	Cumulative distribution function.
`survival`	Survival function.
`log_prior`	Log-prior density function.
`theta_true`	True parameter values.
`theta0`	Initial parameter values.
`scheme`	Censoring scheme.
`scheme_params`	Additional parameters for censoring scheme.
`loss`	Loss function.
`control`	Control parameters.
`parallel`	Logical; whether to use parallel computation.
`ncores`	Number of cores for parallel computation.
`...`	Additional arguments.

Value

Data frame with simulation results.

`summary.lindleyfit` *Summary Method for lindleyfit Objects*

Description

Summary Method for lindleyfit Objects

Usage

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

Arguments

`object`	A lindleyfit object.
`...`	Additional arguments (unused).

Value

A list with summary statistics.

`utilities` *Utility Functions for UniLindleyApprox*

Description

Various helper functions used throughout the package.

Index