

Package ‘TKApprox’

August 7, 2026

Title A General Framework for Bayesian Estimation Using the
‘Tierney’-‘Kadane’ Approximation

Version 0.1.0

Description Provides a distribution-independent framework for Bayesian estimation of arbitrary univariate probability models using the ‘Tierney’-‘Kadane’ approximation (‘Tierney’ & ‘Kadane’, 1986 <[doi:10.1080/01621459.1986.10478240](https://doi.org/10.1080/01621459.1986.10478240)>). Users specify the probability distribution, likelihood, prior distributions, and censoring mechanism, while the package automatically constructs the posterior distribution, computes posterior modes and Hessian matrices, approximates posterior expectations under several Bayesian loss functions, and returns Bayesian parameter estimates, posterior covariance matrices, credible intervals, diagnostic plots, and model comparison statistics. Supports complete, right-, left-, interval-, Type-I, Type-II, progressive Type-II, hybrid, and doubly censored data (‘Lawless’, 2003 <ISBN:978-0-471-37215-8>; ‘Meeker’ & ‘Escobar’, 1998 <ISBN:978-0-471-14328-4>; ‘Balakrishnan’ & ‘Aggarwala’, 2000 <ISBN:978-0-8176-4129-0>; ‘Kundu’ & ‘Pradhan’, 2009 <[doi:10.1198/TECH.2009.0019](https://doi.org/10.1198/TECH.2009.0019)>), making it a flexible tool for Bayesian reliability, survival, and lifetime data analysis.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports checkmate, numDeriv, maxLik, rlang, stats, graphics,
grDevices, utils, MASS

Suggests testthat (>= 3.0.0), knitr, rmarkdown, pkgdown, ggplot2,
gridExtra

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 2026-08-07 20:00:02 UTC

Contents

AIC.tkfit	3
BIC.tkfit	4
CAIC	4
coef.tkfit	5
DIC	5
eval_logposterior	6
expected_logposterior	6
HQIC	7
likelihood	7
logLik.tkfit	8
loglik_complete	8
loglik_dispatcher	9
loglik_doubly_censored	9
loglik_hybrid_censored	10
loglik_interval_censored	11
loglik_left_censored	12
loglik_progressive_type2_censored	12
loglik_right_censored	13
loglik_type1_censored	14
loglik_type2_censored	14
logprior_dispatcher	15
logprior_independent	16
model_comparison	16
model_comparison_table	17
neglogLik	17
plot.tkfit	18
plot.tksensitivity	18
plot_convergence	19
plot_credible_intervals	19
plot_likelihood_surface	20
plot_posterior_approximation	20
plot_prior_posterior	21
plot_residuals	21
plot_sensitivity_estimates	22
plot_sensitivity_logposterior	22
plot_sensitivity_risk	23
predict.tkfit	23
print.tkexpectation	24
print.tkfit	24
print.tkhessian	25
print.tkmode	25
print.tksensitivity	26

print_model_comparison	26
priors	27
residuals.tkfit	28
summary.tkfit	28
tk_expectation	29
tk_expectations	30
tk_fit	31
tk_hessian	33
tk_loss	34
tk_loss_custom	35
tk_loss_gel	36
tk_loss_linex	37
tk_loss_precautionary	38
tk_loss_sel	39
tk_loss_weighted_sel	40
tk_mode	41
tk_plot	42
tk_posterior	43
tk_sensitivity	44
vcov.tkfit	45
Index	46

AIC.tkfit	<i>AIC Method for tkfit Objects</i>
-----------	-------------------------------------

Description

Computes the Akaike Information Criterion.

Usage

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

Arguments

- object A tkfit object
- ... Additional arguments (ignored)

Value

AIC value

BIC.tkfit

BIC Method for tkfit Objects

Description

Computes the Bayesian Information Criterion.

Usage

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

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

BIC value

CAIC

Compute CAIC (Consistent AIC)

Description

Compute CAIC (Consistent AIC)

Usage

```
CAIC(object, ...)

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

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Numeric scalar value of Consistent Akaike Information Criterion (CAIC).

Methods (by class)

- CAIC(tkfit): CAIC method for tkfit objects

coef.tkfit	<i>Coefficients Method for tkfit Objects</i>
------------	--

Description

Extracts the Bayes estimates from a tkfit object.

Usage

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

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Named vector of Bayes estimates

DIC	<i>Compute DIC (Deviance Information Criterion) Approximation</i>
-----	---

Description

Computes an approximation to the DIC using the TK framework.

Usage

```
DIC(object, ...)

## S3 method for class 'tkfit'
DIC(object, ...)
```

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Numeric scalar value of Deviance Information Criterion (DIC) approximation.

Methods (by class)

- DIC(tkfit): DIC method for tkfit objects

eval_logposterior	<i>Evaluate the Log-Posterior at Given Parameters</i>
-------------------	---

Description

Evaluates the log-posterior function created by tk_posterior.

Usage

```
eval_logposterior(log_post_fn, param)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
param	Parameter vector

Value

Numeric value of the log-posterior

expected_logposterior	<i>Compute Expected Log-Posterior</i>
-----------------------	---------------------------------------

Description

Computes the expected log-posterior using the TK approximation.

Usage

```
expected_logposterior(object, ...)
```

```
## S3 method for class 'tkfit'
expected_logposterior(object, ...)
```

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Numeric scalar value of expected log-posterior.

Methods (by class)

- `expected_logposterior(tkfit)`: Expected log-posterior method for tkfit objects

HQIC	<i>Compute HQIC (Hannan-Quinn Information Criterion)</i>
------	--

Description

Compute HQIC (Hannan-Quinn Information Criterion)

Usage

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

```
## S3 method for class 'tkfit'
```

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

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Numeric scalar value of Hannan-Quinn Information Criterion (HQIC).

Methods (by class)

- `HQIC(tkfit)`: HQIC method for tkfit objects

likelihood	<i>Likelihood Construction for Censored Data</i>
------------	--

Description

Functions to construct log-likelihood contributions for various censoring schemes. These functions are used internally to build the complete log-likelihood from user-supplied density/mass functions and CDFs.

Value

Numeric value of the log-likelihood for the specified data and parameters.

logLik.tkfit	<i>Log-Likelihood Method for tkfit Objects</i>
--------------	--

Description

Extracts the log-likelihood at the posterior mode.

Usage

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

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Log-likelihood value

loglik_complete	<i>Complete (Uncensored) Data Log-Likelihood</i>
-----------------	--

Description

Computes the log-likelihood for complete (uncensored) data.

Usage

```
loglik_complete(data, pdf, param)
```

Arguments

data	Numeric vector of observations
pdf	User-supplied density/mass function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
loglik_complete(c(1.2, 2.3, 0.9), pdf_exp, c(1.0))
```

loglik_dispatcher	<i>Generic Likelihood Dispatcher</i>
-------------------	--------------------------------------

Description

Dispatches to the appropriate likelihood function based on the censoring scheme.

Usage

```
loglik_dispatcher(data, censoring_scheme, pdf, cdf = NULL, param, ...)
```

Arguments

data	Data (format depends on censoring scheme: vector, matrix, or data.frame)
censoring_scheme	Character string specifying the censoring scheme
pdf	User-supplied density/mass function
cdf	User-supplied CDF function (required for censored schemes)
param	Parameter vector
...	Additional arguments specific to censoring schemes (status, censoring_time, n, r, removals, etc.)

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
loglik_dispatcher(c(1.2, 2.3), "complete", pdf_exp, cdf_exp, c(1.0))
```

loglik_doubly_censored

Doubly Censored Data Log-Likelihood

Description

Computes the log-likelihood for doubly censored data.

Usage

```
loglik_doubly_censored(data, status, pdf, cdf, param)
```

Arguments

data	Numeric vector of observations or censoring times
status	Numeric vector of censoring indicators (-1 = left-censored, 0 = observed, 1 = right-censored)
pdf	User-supplied density function
cdf	User-supplied CDF function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
loglik_doubly_censored(c(0.5, 1.2, 2.0), c(-1, 0, 1), pdf_exp, cdf_exp, c(1.0))
```

loglik_hybrid_censored

Hybrid Censored Data Log-Likelihood

Description

Computes the log-likelihood for hybrid censored data.

Usage

```
loglik_hybrid_censored(data, censoring_time, r, n, pdf, cdf, param)
```

Arguments

data	Numeric vector of observed failure times
censoring_time	Type-I censoring time T
r	Target number of failures for Type-II censoring
n	Total number of items
pdf	User-supplied density function
cdf	User-supplied CDF function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
loglik_hybrid_censored(c(0.5, 1.2), 2.0, 3, 5, pdf_exp, cdf_exp, c(1.0))
```

loglik_interval_censored

Interval-Censored Data Log-Likelihood

Description

Computes the log-likelihood for interval-censored data. Data should be a matrix or data.frame with columns: lower, upper.

Usage

```
loglik_interval_censored(data, pdf, cdf, param)
```

Arguments

data	Matrix or data.frame with 2 columns: lower, upper
pdf	User-supplied density function
cdf	User-supplied CDF function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
dat <- cbind(c(1, 2), c(1, 3))
loglik_interval_censored(dat, pdf_exp, cdf_exp, c(1.0))
```

loglik_left_censored *Left-Censored Data Log-Likelihood*

Description

Computes the log-likelihood for left-censored data. Status: 1 = observed, 0 = left-censored.

Usage

```
loglik_left_censored(data, status, pdf, cdf, param)
```

Arguments

data	Numeric vector of observations or censoring times
status	Numeric vector of censoring indicators (1 = observed, 0 = censored)
pdf	User-supplied density function
cdf	User-supplied CDF function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
loglik_left_censored(c(1.2, 2.3), c(1, 0), pdf_exp, cdf_exp, c(1.0))
```

loglik_progressive_type2_censored
 Progressive Type-II Censored Data Log-Likelihood

Description

Computes the log-likelihood for progressive Type-II censored data.

Usage

```
loglik_progressive_type2_censored(data, removals, n, pdf, cdf, param)
```

Arguments

data	Numeric vector of ordered failure times
removals	Numeric vector of removal counts at each failure time
n	Initial sample size
pdf	User-supplied density function
cdf	User-supplied CDF function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
loglik_progressive_type2_censored(c(0.5, 1.2), c(1, 2), 5, pdf_exp, cdf_exp, c(1.0))
```

loglik_right_censored *Right-Censored Data Log-Likelihood*

Description

Computes the log-likelihood for right-censored data. Status: 1 = observed, 0 = right-censored.

Usage

```
loglik_right_censored(data, status, pdf, cdf, param)
```

Arguments

data	Numeric vector of observations or censoring times
status	Numeric vector of censoring indicators (1 = observed, 0 = censored)
pdf	User-supplied density function
cdf	User-supplied CDF function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
loglik_right_censored(c(1.2, 2.3), c(1, 0), pdf_exp, cdf_exp, c(1.0))
```

loglik_type1_censored *Type-I Censored Data Log-Likelihood*

Description

Computes the log-likelihood for Type-I (time) censored data.

Usage

```
loglik_type1_censored(data, censoring_time, pdf, cdf, param)
```

Arguments

data	Numeric vector of observations
censoring_time	Fixed censoring time T
pdf	User-supplied density function
cdf	User-supplied CDF function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
loglik_type1_censored(c(1.2, 2.0, 2.0), 2.0, pdf_exp, cdf_exp, c(1.0))
```

loglik_type2_censored *Type-II Censored Data Log-Likelihood*

Description

Computes the log-likelihood for Type-II censored data.

Usage

```
loglik_type2_censored(data, n, r, pdf, cdf, param)
```

Arguments

data	Numeric vector of the r ordered failure times
n	Total number of items
r	Number of observed failures
pdf	User-supplied density function
cdf	User-supplied CDF function
param	Parameter vector

Value

Numeric value of the log-likelihood

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
loglik_type2_censored(c(0.5, 1.2), 5, 2, pdf_exp, cdf_exp, c(1.0))
```

logprior_dispatcher *Generic Prior Function Dispatcher*

Description

Dispatches to the appropriate prior function based on the prior family name.

Usage

```
logprior_dispatcher(param, prior_family, hyperparameters)
```

Arguments

param	Numeric vector of parameters
prior_family	Character string specifying the prior family
hyperparameters	List of hyperparameters for the prior

Value

Numeric value of the log-prior density

Examples

```
logprior_dispatcher(c(2, 3), "gamma", list(shape = 2, rate = 1))
```

logprior_independent *Independent Priors for Multiple Parameters*

Description

Computes the log-prior density when each parameter has an independent prior.

Usage

```
logprior_independent(param, prior_spec)
```

Arguments

param	Named numeric vector of parameters
prior_spec	List specifying prior for each parameter

Value

Numeric value of the total log-prior density

Examples

```
param_spec <- c(alpha = 2, beta = 3)
prior_spec <- list(
  alpha = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)),
  beta = list(family = "normal", hyperparameters = list(mean = 0, sd = 10))
)
logprior_independent(param_spec, prior_spec)
```

model_comparison *Model Comparison Statistics for tkfit Objects*

Description

Computes various model comparison statistics for a fitted tkfit object.

Value

Numeric value or data frame containing model comparison statistics.

model_comparison_table

Comprehensive Model Comparison Table

Description

Computes a comprehensive table of model comparison statistics.

Usage

```
model_comparison_table(object)
```

Arguments

object A tkfit object

Value

A data.frame containing model comparison statistics (Statistic, Value).

neglogLik

Compute Negative Log-Likelihood

Description

Compute Negative Log-Likelihood

Usage

```
neglogLik(object, ...)
```

```
## S3 method for class 'tkfit'
```

```
neglogLik(object, ...)
```

Arguments

object A tkfit object
 ... Additional arguments (ignored)

Value

Numeric scalar value of negative log-likelihood at posterior mode.

Methods (by class)

- neglogLik(tkfit): Negative log-likelihood method for tkfit objects

plot.tkfit	<i>Plot Method for tkfit Objects</i>
------------	--------------------------------------

Description

Creates diagnostic plots for a fitted tkfit object.

Usage

```
## S3 method for class 'tkfit'
plot(x, which = c(1, 2, 3, 4), ask = FALSE, ...)
```

Arguments

x	A tkfit object
which	Which plots to display (1-6, or "all")
ask	Whether to prompt before showing each plot
...	Additional graphical parameters

Value

Invisibly returns the input x object of class "tkfit", called for side effects.

plot.tksensitivity	<i>Plot Method for tksensitivity Objects</i>
--------------------	--

Description

Creates plots showing the sensitivity of estimates to prior hyperparameters.

Usage

```
## S3 method for class 'tksensitivity'
plot(x, which = c(1, 2), ask = FALSE, ...)
```

Arguments

x	A tksensitivity object
which	Which plots to display (1-3, or "all")
ask	Whether to prompt before showing each plot
...	Additional graphical parameters

Value

Invisibly returns the input x object of class "tksensitivity", called for side effects.

plot_convergence	<i>Convergence Diagnostic Plot</i>
------------------	------------------------------------

Description

Plots optimization convergence diagnostics.

Usage

```
plot_convergence(fit, ...)
```

Arguments

fit	A tkfit object
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

plot_credible_intervals	<i>Credible Intervals Plot</i>
-------------------------	--------------------------------

Description

Plots credible intervals for each parameter.

Usage

```
plot_credible_intervals(fit, parameter_index = NULL, ...)
```

Arguments

fit	A tkfit object
parameter_index	Which parameter(s) to plot (NULL for all)
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

plot_likelihood_surface

Likelihood Surface Plot

Description

Plots the likelihood surface or profile for model parameters.

Usage

```
plot_likelihood_surface(fit, n_points = 50, ...)
```

Arguments

fit	A tkfit object
n_points	Number of points for each dimension
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

plot_posterior_approximation

Posterior Approximation Plot

Description

Plots the approximate posterior distribution for each parameter.

Usage

```
plot_posterior_approximation(fit, parameter_index = NULL, n_points = 100, ...)
```

Arguments

fit	A tkfit object
parameter_index	Which parameter(s) to plot (NULL for all)
n_points	Number of points for the curve
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

plot_prior_posterior	<i>Prior vs Posterior Comparison Plot</i>
----------------------	---

Description

Compares the prior and posterior distributions.

Usage

```
plot_prior_posterior(fit, parameter_index = NULL, n_points = 100, ...)
```

Arguments

fit	A tkfit object
parameter_index	Which parameter(s) to plot (NULL for all)
n_points	Number of points for the curve
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

plot_residuals	<i>Residuals Plot</i>
----------------	-----------------------

Description

Plots residuals from the fitted model.

Usage

```
plot_residuals(fit, type = "pearson", ...)
```

Arguments

fit	A tkfit object
type	Type of residuals ("pearson", "deviance")
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

plot_sensitivity_estimates
Plot Sensitivity of Estimates

Description

Plot Sensitivity of Estimates

Usage

```
plot_sensitivity_estimates(sensitivity, ...)
```

Arguments

sensitivity	A tksensitivity object
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

plot_sensitivity_logposterior
Plot Sensitivity of Log-Posterior

Description

Plot Sensitivity of Log-Posterior

Usage

```
plot_sensitivity_logposterior(sensitivity, ...)
```

Arguments

sensitivity	A tksensitivity object
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

plot_sensitivity_risk *Plot Sensitivity of Risk*

Description

Plot Sensitivity of Risk

Usage

```
plot_sensitivity_risk(sensitivity, ...)
```

Arguments

sensitivity	A tksensitivity object
...	Additional graphical parameters

Value

Invisibly returns NULL, called for side effects.

predict.tkfit *Predict Method for tkfit Objects*

Description

Generates predictions from a fitted tkfit object.

Usage

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

Arguments

object	A tkfit object
newdata	New data for prediction
type	Type of prediction ("density", "cdf", "survival")
...	Additional arguments (ignored)

Value

Predicted values

print.tkexpectation *Print Method for tkexpectation Objects*

Description

Print Method for tkexpectation Objects

Usage

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

Arguments

x	A tkexpectation object
...	Additional arguments (ignored)

Value

Invisibly returns the input x object of class "tkexpectation", called for side effects.

print.tkfit *Print Method for tkfit Objects*

Description

Print Method for tkfit Objects

Usage

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

Arguments

x	A tkfit object
...	Additional arguments (ignored)

Value

Invisibly returns the input x object of class "tkfit", called for side effects.

print.tkhessian	<i>Print Method for tkhessian Objects</i>
-----------------	---

Description

Print Method for tkhessian Objects

Usage

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

Arguments

x	A tkhessian object
...	Additional arguments (ignored)

Value

Invisibly returns the input x object of class "tkhessian", called for side effects.

print.tkmode	<i>Print Method for tkmode Objects</i>
--------------	--

Description

Print Method for tkmode Objects

Usage

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

Arguments

x	A tkmode object
...	Additional arguments (ignored)

Value

Invisibly returns the input x object of class "tkmode", called for side effects.

```
print.tksensitivity
```

Print Method for tksensitivity Objects

Description

Print Method for tksensitivity Objects

Usage

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

Arguments

x	A tksensitivity object
...	Additional arguments (ignored)

Value

Invisibly returns the input x object of class "tksensitivity", called for side effects.

```
print_model_comparison
```

Print Model Comparison Table

Description

Print Model Comparison Table

Usage

```
print_model_comparison(object, ...)
```

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Invisibly returns NULL, called for side effects.

priors

Prior Distribution Functions

Description

Functions for computing log-prior densities for various prior distributions. These functions are used internally by `tk_posterior` to construct the log-posterior.

Usage

```
logprior_gamma(param, hyperparameters)
logprior_normal(param, hyperparameters)
logprior_beta(param, hyperparameters)
logprior_uniform(param, hyperparameters)
logprior_exponential(param, hyperparameters)
logprior_lognormal(param, hyperparameters)
logprior_weibull(param, hyperparameters)
logprior_invgamma(param, hyperparameters)
```

Arguments

<code>param</code>	Numeric vector of parameters
<code>hyperparameters</code>	List of hyperparameters for the prior

Value

Numeric value of the log-prior density

Functions

- `logprior_gamma()`: Gamma prior log-density
- `logprior_normal()`: Normal prior log-density
- `logprior_beta()`: Beta prior log-density
- `logprior_uniform()`: Uniform prior log-density
- `logprior_exponential()`: Exponential prior log-density
- `logprior_lognormal()`: Log-Normal prior log-density
- `logprior_weibull()`: Weibull prior log-density
- `logprior_invgamma()`: Inverse Gamma prior log-density

residuals.tkfit	<i>Residuals Method for tkfit Objects</i>
-----------------	---

Description

Computes residuals from a fitted tkfit object.

Usage

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

Arguments

object	A tkfit object
type	Type of residuals ("pearson", "deviance")
...	Additional arguments (ignored)

Value

Residual values

summary.tkfit	<i>Summary Method for tkfit Objects</i>
---------------	---

Description

Summary Method for tkfit Objects

Usage

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

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Invisibly returns the input object object of class "tkfit", called for side effects.

tk_expectation	<i>Tierney-Kadane Approximation of Posterior Expectations</i>
----------------	---

Description

Computes the Tierney-Kadane approximation of $E(g(\theta) \mid x)$ for a given function g . This is the core computational engine for all Bayes estimators.

Usage

```
tk_expectation(
  log_post_fn,
  g,
  mode_result,
  hessian_result,
  initial_values_star = NULL,
  ...
)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
g	Function $g(\theta)$ for which to compute the expectation
mode_result	Result from tk_mode containing posterior mode
hessian_result	Result from tk_hessian containing Hessian at mode
initial_values_star	Initial values for optimizing $h^*(\theta)$ (defaults to mode)
...	Additional arguments passed to tk_mode

Value

An object of class "tkexpectation" (a list) containing:

expectation	Numeric scalar representing the Tierney-Kadane approximation of $E(g(\theta) \mid x)$.
theta_hat	Numeric vector, posterior mode of $h(\theta)$.
theta_star	Numeric vector, posterior mode of modified log-posterior $h^*(\theta)$.
h_theta_hat	Numeric scalar, $h(\theta)$ evaluated at theta_hat.
h_star_theta_star	Numeric scalar, $h^*(\theta)$ evaluated at theta_star.
Sigma	Curvature matrix $(-H^{-1})$ at theta_hat.
Sigma_star	Curvature matrix $(-H^{*-1})$ at theta_star.
det_Sigma	Determinant of Sigma.
det_Sigma_star	Determinant of Sigma_star.

ratio	Square root determinant ratio $\sqrt{ \Sigma / \hat{\Sigma} }$.
exponent	Exponential factor $n * (h^*(\theta_{\text{star}}) - h(\theta_{\text{hat}}))$.
n	Effective sample size.
mode_result	Full output from <code>tk_mode</code> for $h(\theta)$.
mode_star_result	Full output from <code>tk_mode</code> for $h^*(\theta)$.
hessian_result	Full output from <code>tk_hessian</code> for $h(\theta)$.
hessian_star_result	Full output from <code>tk_hessian</code> for $h^*(\theta)$.

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
mode_result <- tk_mode(log_post, initial_values = c(1))
hessian_result <- tk_hessian(log_post, mode_result$mode)
g_fn <- function(param) param[1]
tk_exp <- tk_expectation(log_post, g_fn, mode_result, hessian_result)
```

tk_expectations	<i>Compute Multiple Expectations Efficiently</i>
-----------------	--

Description

Computes TK approximations for multiple functions $g_i(\theta)$ efficiently by reusing the posterior mode and Hessian.

Usage

```
tk_expectations(log_post_fn, g_list, mode_result, hessian_result, ...)
```

Arguments

log_post_fn	Log-posterior function from <code>tk_posterior</code>
g_list	List of functions $g_i(\theta)$ for which to compute expectations
mode_result	Result from <code>tk_mode</code>
hessian_result	Result from <code>tk_hessian</code>
...	Additional arguments passed to <code>tk_expectation</code>

Value

Named vector of expectations

tk_fit	<i>Main Function for Bayesian Estimation via Tierney-Kadane Approximation</i>
--------	---

Description

Performs Bayesian parameter estimation for user-specified univariate probability distributions using the Tierney-Kadane approximation.

Usage

```
tk_fit(
  data,
  censoring_scheme,
  pdf,
  cdf = NULL,
  prior_spec = NULL,
  loglik = NULL,
  initial_values,
  lower_bounds = NULL,
  upper_bounds = NULL,
  loss_function = "sel",
  loss_params = NULL,
  custom_g = NULL,
  optimization_method = "BFGS",
  hessian_method = "Richardson",
  analytic_hessian = NULL,
  true_parameter = NULL,
  ...
)
```

Arguments

data	Data (format depends on censoring scheme: vector, matrix, or data.frame)
censoring_scheme	Character string specifying the censoring scheme
pdf	User-supplied density/mass function
cdf	User-supplied CDF function (required for censored schemes)
prior_spec	List specifying prior for each parameter, or a single prior function
loglik	Optional user-supplied log-likelihood function
initial_values	Initial parameter values for optimization
lower_bounds	Lower bounds for parameters (optional)
upper_bounds	Upper bounds for parameters (optional)
loss_function	Loss function for Bayes estimation ("sel", "linex", "gel", "precautionary", "weighted-sel", "custom")

loss_params	List of parameters for the loss function (e.g., c for LINEX, q for GEL)
custom_g	Custom g(theta) function for custom loss
optimization_method	Optimization method ("BFGS", "L-BFGS-B", "Nelder-Mead", "nlminb", "max-Lik", "trust")
hessian_method	Method for numerical differentiation ("Richardson", "simple")
analytic_hessian	Optional user-supplied analytic Hessian function
true_parameter	True parameter values (for bias calculation in simulation studies)
...	Additional arguments for censoring schemes (status, censoring_time, n, r, removals, etc.)

Value

An object of class "tkfit" (a list) containing:

posterior_mode	Named numeric vector of posterior MAP estimates.
bayes_estimates	Named numeric vector of Tierney-Kadane Bayes estimates.
loss_function	Character string specifying the loss function used.
loss_params	List of specific parameters for the loss function (if any).
covariance	Numeric matrix of approximate posterior covariances.
standard_errors	Named numeric vector of posterior standard errors.
credible_intervals	Matrix containing 95% normal-approximation credible intervals.
true_parameter	Numeric vector of true parameter values if provided.
bias	Named numeric vector of estimation bias if true_parameter was provided.
relative_bias	Named numeric vector of relative bias if true_parameter was provided.
log_posterior_at_mode	Numeric scalar value of log-posterior at mode.
log_likelihood_at_mode	Numeric scalar value of log-likelihood at mode.
prior_at_mode	Numeric scalar value of log-prior density at mode.
optimization_method	Character string indicating optimizer used.
convergence	Integer status code from optimizer (0 = success).
iterations	Integer, number of iterations taken by optimizer.
gradient_norm	Numeric scalar, norm of gradient at posterior mode.
hessian	Hessian matrix of scaled log-posterior h(theta) at mode.
Sigma	Curvature matrix (-H ⁻¹).
hessian_result	Full output list from tk_hessian .

data	Input data.
censoring_scheme	Character string, censoring scheme used.
pdf	Probability density function used.
cdf	Cumulative distribution function used.
prior_spec	Prior specification supplied.
loglik	User-supplied log-likelihood function used (if any).
initial_values	Initial parameter values used.
lower_bounds	Lower bounds specified.
upper_bounds	Upper bounds specified.
extra_args	List of extra censoring arguments.
n	Effective sample size.
execution_time	Execution time in seconds.
session_info	Output from sessionInfo().
call	Matched function call.

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
fit <- tk_fit(data, "complete", pdf_exp, cdf_exp, prior_spec,
             initial_values = c(rate = 1), loss_function = "sel")
summary(fit)
```

tk_hessian

Compute the Hessian and Posterior Covariance Matrix

Description

Computes the Hessian of the log-posterior at a given parameter value and derives the approximate posterior covariance matrix.

Usage

```
tk_hessian(
  log_post_fn,
  param,
  method = "Richardson",
  analytic_hessian = NULL,
  validate_analytic = TRUE
)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
param	Parameter vector at which to evaluate the Hessian
method	Method for numerical differentiation ("Richardson", "simple")
analytic_hessian	Optional user-supplied analytic Hessian function
validate_analytic	Whether to validate analytic Hessian against numerical

Value

An object of class "tkhessian" (a list) containing:

hessian	Hessian matrix of scaled log-posterior $h(\theta)$ evaluated at param.
Sigma	Curvature matrix $(-H^{-1})$.
covariance	Approximate posterior covariance matrix (Sigma / n).
standard_errors	Named numeric vector of posterior standard errors.
parameter	Parameter vector evaluated.
method	Numerical differentiation method used.
validation	Analytic Hessian validation results list, or NULL.

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
mode_result <- tk_mode(log_post, initial_values = c(1))
hess_result <- tk_hessian(log_post, mode_result$mode)
```

tk_loss

Bayes Estimators Under Various Loss Functions

Description

Computes Bayes estimators under different loss functions using the Tierney-Kadane approximation framework.

Dispatches to the appropriate loss function based on the loss type.

Usage

```
tk_loss(loss_type, log_post_fn, mode_result, hessian_result, ...)
```

Arguments

loss_type	Character string specifying the loss function ("sel", "linex", "gel", "precautionary", "weighted-sel", "custom")
log_post_fn	Log-posterior function from tk_posterior
mode_result	Result from tk_mode
hessian_result	Result from tk_hessian
...	Additional arguments specific to the loss function (e.g. c for LINEX, q for GEL, g for custom)

Value

Named numeric vector of Bayes parameter estimates under specified loss.
 Bayes estimate(s) under the specified loss function

tk_loss_custom	<i>Custom Loss Function Estimator</i>
----------------	---------------------------------------

Description

Computes a Bayes estimator for a user-specified loss function.

Usage

```
tk_loss_custom(
  log_post_fn,
  mode_result,
  hessian_result,
  g,
  parameter_index = NULL,
  ...
)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
mode_result	Result from tk_mode
hessian_result	Result from tk_hessian
g	Function $g(\theta)$ for the expectation
parameter_index	Index of parameter to estimate (NULL for all parameters)
...	Additional arguments passed to tk_expectation

Value

Bayes estimate(s) under custom loss

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
mode_result <- tk_mode(log_post, initial_values = c(1))
hessian_result <- tk_hessian(log_post, mode_result$mode)
g_fn <- function(param) log(param[1])
custom_estimate <- tk_loss_custom(log_post, mode_result, hessian_result, g_fn)
```

tk_loss_gel	<i>General Entropy Loss (GEL) Estimator</i>
-------------	---

Description

Computes the Bayes estimator under General Entropy Loss. Estimator: $\theta_{\text{hat}} = (E(\theta^{-q}) / x)^{-1/q}$

Usage

```
tk_loss_gel(
  log_post_fn,
  mode_result,
  hessian_result,
  q = 0.5,
  parameter_index = NULL,
  ...
)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
mode_result	Result from tk_mode
hessian_result	Result from tk_hessian
q	GEL shape parameter ($q \neq 0$)
parameter_index	Index of parameter to estimate (NULL for all parameters)
...	Additional arguments passed to tk_expectation

Value

Bayes estimate(s) under GEL

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
mode_result <- tk_mode(log_post, initial_values = c(1))
hessian_result <- tk_hessian(log_post, mode_result$mode)
gel_estimate <- tk_loss_gel(log_post, mode_result, hessian_result, q = 0.5)
```

tk_loss_linex	<i>LINEX Loss Estimator</i>
---------------	-----------------------------

Description

Computes the Bayes estimator under LINEX (Linear-Exponential) loss. Estimator: $\theta_{\text{hat}} = -(1/c) * \log E(\exp(-c * \theta) | x)$

Usage

```
tk_loss_linex(
  log_post_fn,
  mode_result,
  hessian_result,
  c = 0.5,
  parameter_index = NULL,
  ...
)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
mode_result	Result from tk_mode
hessian_result	Result from tk_hessian
c	LINEX shape parameter (c != 0)
parameter_index	Index of parameter to estimate (NULL for all parameters)
...	Additional arguments passed to tk_expectation

Value

Bayes estimate(s) under LINEX loss

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
mode_result <- tk_mode(log_post, initial_values = c(1))
hessian_result <- tk_hessian(log_post, mode_result$mode)
linex_estimate <- tk_loss_linex(log_post, mode_result, hessian_result, c = 0.5)
```

tk_loss_precautionary *Precautionary Loss Estimator*

Description

Computes the Bayes estimator under precautionary loss. Estimator: $\theta_{\text{hat}} = \sqrt{E(\theta^2 | x)}$

Usage

```
tk_loss_precautionary(
  log_post_fn,
  mode_result,
  hessian_result,
  parameter_index = NULL,
  ...
)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
mode_result	Result from tk_mode
hessian_result	Result from tk_hessian
parameter_index	Index of parameter to estimate (NULL for all parameters)
...	Additional arguments passed to tk_expectation

Value

Bayes estimate(s) under precautionary loss

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
```

```

mode_result <- tk_mode(log_post, initial_values = c(1))
hessian_result <- tk_hessian(log_post, mode_result$mode)
precautionary_estimate <- tk_loss_precautionary(log_post, mode_result, hessian_result)

```

tk_loss_sel

*Squared Error Loss (SEL) Estimator***Description**

Computes the posterior mean (Bayes estimator under squared error loss).

Usage

```

tk_loss_sel(
  log_post_fn,
  mode_result,
  hessian_result,
  parameter_index = NULL,
  ...
)

```

Arguments

log_post_fn	Log-posterior function from tk_posterior
mode_result	Result from tk_mode
hessian_result	Result from tk_hessian
parameter_index	Index of parameter to estimate (NULL for all parameters)
...	Additional arguments passed to tk_expectation

Value

Bayes estimate(s) under SEL

Examples

```

pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
mode_result <- tk_mode(log_post, initial_values = c(1))
hessian_result <- tk_hessian(log_post, mode_result$mode)
sel_estimate <- tk_loss_sel(log_post, mode_result, hessian_result)

```

tk_loss_weighted_sel *Weighted Squared Error Loss Estimator*

Description

Computes the Bayes estimator under weighted squared error loss. Estimator: $\theta_{\text{hat}} = 1 / E(\theta^{-1} | x)$

Usage

```
tk_loss_weighted_sel(
  log_post_fn,
  mode_result,
  hessian_result,
  parameter_index = NULL,
  ...
)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
mode_result	Result from tk_mode
hessian_result	Result from tk_hessian
parameter_index	Index of parameter to estimate (NULL for all parameters)
...	Additional arguments passed to tk_expectation

Value

Bayes estimate(s) under weighted SEL

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
mode_result <- tk_mode(log_post, initial_values = c(1))
hessian_result <- tk_hessian(log_post, mode_result$mode)
weighted_estimate <- tk_loss_weighted_sel(log_post, mode_result, hessian_result)
```

tk_mode

*Find the Posterior Mode via Optimization***Description**

Locates the posterior mode (MAP estimate) using numerical optimization. Supports multiple optimization backends with automatic fallback.

Usage

```
tk_mode(
  log_post_fn,
  initial_values,
  lower_bounds = NULL,
  upper_bounds = NULL,
  method = "BFGS",
  control = NULL,
  max_attempts = 3
)
```

Arguments

log_post_fn	Log-posterior function from tk_posterior
initial_values	Initial parameter values for optimization
lower_bounds	Lower bounds for parameters (optional)
upper_bounds	Upper bounds for parameters (optional)
method	Optimization method ("BFGS", "L-BFGS-B", "Nelder-Mead", "nlminb", "max-Lik", "trust")
control	List of control parameters for the optimizer
max_attempts	Maximum number of optimization attempts with different methods

Value

An object of class "tkmode" (a list) containing:

mode	Named numeric vector of MAP parameter estimates.
convergence	Integer status code from optimizer (0 = success).
log_posterior	Numeric scalar, log-posterior value evaluated at mode.
iterations	Integer, number of iterations taken.
method	Character string, optimization method used.
message	Warning or informational message string.
gradient_norm	Numeric scalar, norm of gradient at mode.
gradient	Numeric vector, gradient evaluated at mode.

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
mode_result <- tk_mode(log_post, initial_values = c(1), method = "BFGS")
```

tk_plot

*Visualization Functions for tkfit Objects***Description**

Functions to create diagnostic and exploratory plots for Tierney-Kadane Bayesian estimation results.

Dispatches to plot.tkfit for graphical diagnostics.

Usage

```
tk_plot(x, ...)
```

Arguments

x	A tkfit object
...	Additional arguments passed to plot.tkfit

Value

Invisibly returns the input object or NULL, called for side effects.

Invisibly returns the input x object of class "tkfit", called for side effects.

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
fit <- tk_fit(data, "complete", pdf_exp, cdf_exp, prior_spec, initial_values = c(rate = 1))
tk_plot(fit, which = 1)
```

tk_posterior

*Construct the Log-Posterior Function***Description**

Creates a function that computes the log-posterior density for a given parameter vector, combining the log-likelihood and log-prior.

Usage

```
tk_posterior(
  data,
  censoring_scheme,
  pdf,
  cdf = NULL,
  prior_spec = NULL,
  loglik = NULL,
  ...
)
```

Arguments

data	Data (format depends on censoring scheme: vector, matrix, or data.frame)
censoring_scheme	Character string specifying the censoring scheme
pdf	User-supplied density/mass function
cdf	User-supplied CDF function (required for censored schemes)
prior_spec	List specifying prior for each parameter, or a single prior function
loglik	Optional user-supplied log-likelihood function
...	Additional arguments for censoring schemes (status, censoring_time, n, r, removals, etc.)

Value

A function that takes a parameter vector and returns the log-posterior $h(\theta) = (\log L + \log \pi) / n$

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
log_post <- tk_posterior(data, "complete", pdf_exp, cdf_exp, prior_spec)
log_post(c(1.5))
```

tk_sensitivity	<i>Prior Sensitivity Analysis</i>
----------------	-----------------------------------

Description

Performs sensitivity analysis by varying prior hyperparameters and examining the impact on Bayes estimates, bias, variance, and risk.

Usage

```
tk_sensitivity(
  fit,
  parameter_name,
  hyperparameter_name,
  hyperparameter_values,
  loss_function = "sel",
  loss_params = NULL,
  ...
)
```

Arguments

fit	A tkfit object to use as the base model
parameter_name	Name of the parameter whose prior hyperparameters to vary
hyperparameter_name	Name of the hyperparameter to vary
hyperparameter_values	Vector of hyperparameter values to test
loss_function	Loss function to use for estimation
loss_params	Parameters for the loss function
...	Additional arguments passed to tk_fit

Value

An object of class "tksensitivity" (a list) containing:

parameter_name	Character string, target parameter name.
hyperparameter_name	Character string, varied hyperparameter name.
hyperparameter_values	Numeric vector of tested hyperparameter values.
results	Data frame summarizing estimates, standard errors, log-posterior, log-likelihood, and risk values.
plot_data	List containing structured data for diagnostic plotting.
base_fit	The base "tkfit" object.
loss_function	Character string of the loss function used.

Examples

```
pdf_exp <- function(x, param) dexp(x, rate = param[1])
cdf_exp <- function(x, param) pexp(x, rate = param[1])
prior_spec <- list(rate = list(family = "gamma", hyperparameters = list(shape = 2, rate = 1)))
data <- c(1.2, 2.3, 1.8, 3.1, 0.9)
fit <- tk_fit(data, "complete", pdf_exp, cdf_exp, prior_spec, initial_values = c(rate = 1))
sensitivity <- tk_sensitivity(fit, "rate", "shape", c(1, 2, 5))
```

vcov.tkfit

*Variance-Covariance Method for tkfit Objects***Description**

Extracts the approximate posterior covariance matrix from a tkfit object.

Usage

```
## S3 method for class 'tkfit'
vcov(object, ...)
```

Arguments

object	A tkfit object
...	Additional arguments (ignored)

Value

Covariance matrix

Index

AIC.tkfit, 3

BIC.tkfit, 4

CAIC, 4

coef.tkfit, 5

DIC, 5

eval_logposterior, 6

expected_logposterior, 6

HQIC, 7

likelihood, 7

logLik.tkfit, 8

loglik_complete, 8

loglik_dispatcher, 9

loglik_doubly_censored, 9

loglik_hybrid_censored, 10

loglik_interval_censored, 11

loglik_left_censored, 12

loglik_progressive_type2_censored, 12

loglik_right_censored, 13

loglik_type1_censored, 14

loglik_type2_censored, 14

logprior_beta(priors), 27

logprior_dispatcher, 15

logprior_exponential(priors), 27

logprior_gamma(priors), 27

logprior_independent, 16

logprior_invgamma(priors), 27

logprior_lognormal(priors), 27

logprior_normal(priors), 27

logprior_uniform(priors), 27

logprior_weibull(priors), 27

model_comparison, 16

model_comparison_table, 17

neglogLik, 17

plot.tkfit, 18

plot.tksensitivity, 18

plot_convergence, 19

plot_credible_intervals, 19

plot_likelihood_surface, 20

plot_posterior_approximation, 20

plot_prior_posterior, 21

plot_residuals, 21

plot_sensitivity_estimates, 22

plot_sensitivity_logposterior, 22

plot_sensitivity_risk, 23

predict.tkfit, 23

print.tkexpectation, 24

print.tkfit, 24

print.tkhessian, 25

print.tkmode, 25

print.tksensitivity, 26

print_model_comparison, 26

priors, 27

residuals.tkfit, 28

summary.tkfit, 28

tk_expectation, 29

tk_expectations, 30

tk_fit, 31

tk_hessian, 30, 32, 33

tk_loss, 34

tk_loss_custom, 35

tk_loss_gel, 36

tk_loss_linex, 37

tk_loss_precautionary, 38

tk_loss_sel, 39

tk_loss_weighted_sel, 40

tk_mode, 30, 41

tk_plot, 42

tk_posterior, 43

tk_sensitivity, 44

vcov.tkfit, 45