Package 'UnitTrans'

November 26, 2026

Type Package

Title Distributions on the Unit Interval via Transformations of Baseline Laws

Version 0.2.0

Description Builds new probability distributions on (0,1) by applying a monotone transformation (rational, exponential, logit, 'tanh', probit, complementary log-log) to a baseline random variable Y with any available d/p/q/r functions. Provides density, distribution, quantile and random generation functions, maximum likelihood fitting with information criteria, profile-likelihood and bootstrap confidence intervals, automatic model search over baseline/transformation pairs, goodness-of-fit tests with parametric bootstrap, moments, hazard and simulation studies.

License GPL-3

Encoding UTF-8

Language en-US

URL https://github.com/yamenetoo/Statistical-Repository/tree/main/UnitTrans

BugReports https://github.com/yamenetoo/Statistical-Repository/issues

Depends R (>= 3.5.0)

Imports stats, graphics, parallel

Suggests extraDistr, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.1

NeedsCompilation no

Author Mohamad Yamen Al-Mohamad [aut, cre] (ORCID: <https://orcid.org/0009-0009-1665-7203>), Ali Akbar Hydari [aut]

Maintainer Mohamad Yamen Al-Mohamad <yamenmohamad@tabrizu.ac.ir>

Repository CRAN

Date/Publication 2026-11-26 10:30:02 UTC

Contents

baseline_params . . . 2
best_model . . . 3
check_domain . . . 3
confint.transfit . . . 4
detect_transformations . . . 5
dpqr_trans . . . 5
fit_all . . . 6
fit_trans . . . 8
forward_transformation . . . 9
get_transformation . . . 10
gof_trans . . . 10
hazard_trans . . . 11
inverse_transformation . . . 12
loglik_trans . . . 12
moments_trans . . . 13
plot.transfit . . . 13
simulation_study . . . 14
transfit_methods . . . 15
trans_list . . . 15

Index **17**

`baseline_params` *Parameter names of a baseline distribution*

Description

Reads the parameter names from the formals of the ‘d<dist>‘ function, dropping the first argument and arguments such as ‘log‘.

Usage

```
baseline_params(dist)
```

Arguments

`dist`	Name of the baseline distribution, e.g. ‘"gamma"‘.

Value

Character vector of parameter names.

Examples

```
baseline_params("gamma")
```

best_model *Extract the best model from a [fit_all()] result*

Description

Extract the best model from a [fit_all()] result

Usage

```
best_model(object, criterion = object$criterion)
```

Arguments

object	A ‘transfit_all‘ object.
criterion	Criterion to use; defaults to the one used for ranking.

Value

The best ‘transfit‘ object.

check_domain *Check that a baseline/transformation pair is valid*

Description

Verifies that the support of Y is compatible with the transformation. When the baseline cdf is available the probability mass that Y places outside the admissible domain of the transformation is computed exactly (for example ‘P(Y < 0)‘ for ‘"rational"‘); otherwise a grid of quantiles is inspected. The range of V actually reached is also reported.

Usage

```
check_domain(dist, par, trans, tol = 1e-08)
```

Arguments

dist	Baseline distribution name.
par	Baseline parameters (named vector/list or unnamed in formals order).
trans	Transformation name or object.
tol	Largest tolerated probability mass outside the admissible domain.

Value

A list with ‘valid‘ (logical), ‘range‘ (approximate support of V), ‘mass_outside‘ and ‘message‘.

Examples

```
check_domain("norm", c(0, 1), "logit")
check_domain("norm", c(0, 1), "rational")
```

`confint.transfit` *Confidence intervals for a fitted model*

Description

Confidence intervals for a fitted model

Usage

```
## S3 method for class 'transfit'
confint(
  object,
  parm,
  level = 0.95,
  method = c("wald", "bootstrap", "profile"),
  B = 200,
  seed = NULL,
  ...
)
```

Arguments

`object`	A ‘transfit‘ object.
`parm`	Parameter names or indices (default: all).
`level`	Confidence level.
`method`	‘"wald"‘ (from the Hessian), ‘"bootstrap"‘ (parametric percentile bootstrap) or ‘"profile"‘ (profile likelihood, usually better for positive or skewed parameters). A limit returned as ‘NA‘ means that the profile did not cross the cut-off in the searched range.
`B`	Bootstrap replications.
`seed`	Optional seed for the bootstrap (the caller’s random number stream is restored afterwards).
`...`	Unused.

Value

Matrix with lower and upper limits. For the bootstrap the number of failed replications is stored in the attribute ‘"failed"‘.

Examples

```
set.seed(1)
fit <- fit_trans(rtrans(150, "norm", c(0, 1), "logit"), "norm", "logit")
confint(fit)
confint(fit, method = "profile")
```

---

detect_transformations *Detect which transformations suit a baseline distribution*

Description

Detect which transformations suit a baseline distribution

Usage

```
detect_transformations(dist, par)
```

Arguments

dist	Baseline distribution name.
par	Baseline parameters.

Value

A data frame listing each transformation, whether it is valid and the approximate range of V.

Examples

```
detect_transformations("gamma", c(shape = 3, rate = 1))
```

---

dpqr_trans *Density, distribution, quantile and random generation*

Description

Functions for the distribution of $V = g(Y)$ on (0,1), where Y follows a baseline distribution 'dist' with parameters 'par'.

Usage

```
dtrans(v, dist, par, trans, log = FALSE)

ptrans(q, dist, par, trans, lower.tail = TRUE, log.p = FALSE)

qtrans(p, dist, par, trans, lower.tail = TRUE, log.p = FALSE)

rtrans(n, dist, par, trans)
```

Arguments

v, q	Vector of quantiles in (0,1).
dist	Baseline distribution name, e.g. ‘"gamma"‘ (uses ‘dgamma‘, ‘pgamma‘, ‘qgamma‘).
par	Baseline parameters: named vector/list, or unnamed in the order of the baseline’s formals.
trans	Transformation name (see [trans_list()]) or object.
log	Logical; return log-density.
lower.tail	Logical; if ‘FALSE‘, return $P(V > q)$.
log.p	Logical; probabilities are given/returned as ‘log(p)‘.
p	Vector of probabilities.
n	Number of observations.

Details

Tail probabilities are computed directly from the baseline (for a decreasing transformation the upper tail of Y is the lower tail of V), never as ‘1 - p‘, so ‘lower.tail = FALSE‘ and ‘log.p = TRUE‘ stay accurate far into the tails.

Value

Numeric vector. ‘rtrans‘ may return values numerically equal to 0 or 1 for extreme draws.

Examples

```
dtrans(c(.2, .5, .8), "norm", c(0, 1), "logit")
ptrans(0.5, "gamma", c(2, 1), "exponential")
qtrans(c(.25, .5), "norm", c(0, 1), "probit")
set.seed(1); rtrans(5, "gamma", c(2, 1), "rational")
```

---

fit_all *Fit many baseline/transformation combinations and rank them*

Description

Fits every valid combination of baseline distributions and transformations and ranks them by an information criterion. Combinations whose baseline support is incompatible with the transformation, whose optimizer did not converge or whose fit puts mass outside the transformation domain are dropped (reported when ‘verbose = TRUE‘).

Usage

```
fit_all(
  data,
  dists,
  trans = names(.registry()),
  criterion = c("AIC", "BIC", "AICc"),
  verbose = FALSE,
  drop_nonconverged = TRUE,
  seed = NULL,
  cores = 1L,
  adjust = FALSE
)
```

Arguments

data	Numeric vector in (0,1).
dists	Character vector of baseline distribution names.
trans	Character vector of transformation names (default: all).
criterion	‘"AIC"‘, ‘"BIC"‘ or ‘"AICc"‘.
verbose	Print skipped combinations.
drop_nonconverged	Drop fits whose optimizer reported non-convergence.
seed	Optional seed passed to every [fit_trans()] call (the same seed for each combination, so results do not depend on the order of the combinations or on ‘cores‘).
cores	Number of processes (Unix-alike systems only; ignored on Windows).
adjust	Passed to [fit_trans()].

Value

An object of class ‘transfit_all‘: a ranking ‘table‘ (with ‘delta‘ and Akaike ‘weight‘ for the chosen criterion) and the list of ‘fits‘.

Examples

```
set.seed(2)
x <- rtrans(200, "gamma", c(2, 1), "rational")
res <- fit_all(x, c("gamma", "lnorm", "weibull"), c("rational", "exponential"))
res$table
best_model(res)
```

`fit_trans` *Maximum likelihood fit of a transformed distribution*

Description

Fits the distribution of $V = g(Y)$ to data in (0,1) by maximizing the log-likelihood over the baseline parameters. The optimization is carried out on an unconstrained scale (logarithmic for positive parameters), starting from the default values, from a quantile-matching solution and from random perturbations; the best optimum is kept.

Usage

```
fit_trans(
  data,
  dist,
  trans,
  start = NULL,
  lower = NULL,
  upper = NULL,
  n_start = 3,
  seed = NULL,
  adjust = FALSE,
  check_support = TRUE
)
```

Arguments

`data`	Numeric vector with values strictly inside (0,1).
`dist`	Baseline distribution name.
`trans`	Transformation name or object.
`start`	Optional named starting values. Defaults to 0 for location-type parameters and 1 otherwise.
`lower`, `upper`	Optional bounds for the parameters.
`n_start`	Number of starting points (the default or user start, a quantile-matching start, plus random perturbations); the best optimum is kept.
`seed`	Optional seed for the random perturbations. The caller's random number stream is restored afterwards.
`adjust`	If 'TRUE', data are mapped into the open interval with the Smithson-Verkuilen correction '(y (n - 1) + 0.5) / n', which allows observations equal to 0 or 1.
`check_support`	If 'TRUE' (default), stop when the baseline puts mass outside the domain of the transformation (see [check_domain()]).

Value

An object of class ‘transfit‘ with ‘coef‘, ‘se‘, ‘vcov‘, ‘logLik‘, ‘AIC‘, ‘BIC‘, ‘AICc‘, ‘n‘, ‘convergence‘ and ‘hessian_ok‘ (whether the Hessian was positive definite, so that standard errors are available).

Examples

```
set.seed(1)
x <- rtrans(200, "norm", c(0.5, 1), "logit")
fit <- fit_trans(x, "norm", "logit")
fit
```

`forward_transformation` *Forward transformation*

Description

Forward transformation

Usage

```
forward_transformation(y, trans)
```

Arguments

`y`	Numeric vector on the baseline scale.
`trans`	Transformation name or object.

Value

Numeric vector V = g(Y).

Examples

```
forward_transformation(1.2, "rational")
```

get_transformation *Get a transformation object by name*

Description

Get a transformation object by name

Usage

```
get_transformation(name)
```

Arguments

name	One of the names returned by [trans_list()]. The old name ‘"arcsech"‘ is still accepted (with a warning) and refers to ‘"tanh"‘: the map ‘V = 2/(1 + exp(Y))‘ equals ‘1 - tanh(Y/2)‘ and is not the inverse hyperbolic secant.

Value

An object of class ‘unitrans‘ with elements ‘g‘ (forward map), ‘ginv‘ (inverse map), ‘logjac‘ (log |dY/dV|), ‘domain‘ and ‘decreasing‘.

Examples

```
tr <- get_transformation("logit")
tr$g(0)
```

gof_trans *Goodness-of-fit tests with parametric bootstrap*

Description

Computes Kolmogorov-Smirnov, Cramer-von Mises and Anderson-Darling statistics for a fitted model and obtains p-values by a parametric bootstrap that re-estimates the parameters in every replication (so the p-values are valid with estimated parameters). The p-value is ‘(1 + number of T* >= T) / (B + 1)‘, so it is never exactly zero.

Usage

```
gof_trans(fit, B = 200, refit = TRUE, seed = NULL)
```

Arguments

fit	A ‘transfit‘ object.
B	Number of bootstrap replications.
refit	Re-estimate the parameters in each replication.
seed	Optional seed (the caller’s random number stream is restored).

Value

A data frame with statistic and bootstrap p-value for each test, and ‘valid_boot‘, the number of replications that could be fitted. A warning is issued when some replications fail.

Examples

```
set.seed(3)
fit <- fit_trans(rtrans(100, "norm", c(0, 1), "logit"), "norm", "logit")
gof_trans(fit, B = 50)
```

`hazard_trans` *Survival and hazard functions*

Description

The survival function is the upper tail computed directly from the baseline and the hazard is evaluated on the log scale, so both stay accurate near 1.

Usage

```
hazard_trans(v, dist, par, trans)

survival_trans(v, dist, par, trans)
```

Arguments

`v`	Values in (0,1).
`dist`, `par`, `trans`	As in [dtrans()].

Value

Numeric vector.

Examples

```
hazard_trans(c(.2, .5), "gamma", c(2, 1), "rational")
```

inverse_transformation *Inverse transformation*

Description

Maps values of V in (0,1) back to the baseline scale Y.

Usage

```
inverse_transformation(v, trans)
```

Arguments

v	Numeric vector in (0,1).
trans	Transformation name or object.

Value

Numeric vector Y = ginv(V), the inverse of g.

Examples

```
inverse_transformation(0.3, "logit")
```

loglik_trans *Log-likelihood of the transformed distribution*

Description

Log-likelihood of the transformed distribution

Usage

```
loglik_trans(par, data, dist, trans)
```

Arguments

par	Baseline parameters.
data	Observations in (0,1).
dist	Baseline distribution name.
trans	Transformation name or object.

Value

Numeric scalar.

Examples

```
loglik_trans(c(0, 1), c(.2, .4, .7), "norm", "logit")
```

---

moments_trans *Raw moments, mean, variance, skewness and kurtosis*

Description

Computed by numerical integration of the quantile function, $E[V^r] = \int_0^1 Q(p)^r dp$. Central moments in ‘summary_trans‘ are integrated directly (not obtained from raw moments), which avoids cancellation errors.

Usage

```
moments_trans(dist, par, trans, r = 1:4)

summary_trans(dist, par, trans)
```

Arguments

`dist`, `par`, `trans`	As in [dtrans()].
`r`	Order(s) of the raw moments.

Value

‘moments_trans‘ returns a named vector of raw moments; ‘summary_trans‘ returns mean, variance, skewness and excess kurtosis.

Examples

```
moments_trans("gamma", c(2, 1), "rational", r = 1:2)
summary_trans("norm", c(0, 1), "logit")
```

---

plot.transfit *Plot a fitted model*

Description

Histogram with fitted density, plus fitted vs empirical CDF, Q-Q and P-P.

Usage

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

Arguments

x	A ‘transfit‘ object.
which	Which of the four panels to draw (subset of 1:4).
...	Unused.

Value

Invisibly ‘x‘.

Examples

```
set.seed(1)
fit <- fit_trans(rtrans(150, "norm", c(0, 1), "logit"), "norm", "logit")
plot(fit)
```

---

simulation_study *Monte-Carlo simulation study of the MLE*

Description

Repeatedly simulates from a transformed distribution, refits it and summarizes bias, root mean squared error and the coverage of Wald intervals. Coverage is computed over the replications with available standard errors.

Usage

```
simulation_study(dist, par, trans, n = c(50, 100, 200), R = 200, seed = NULL)
```

Arguments

dist, par, trans	True model.
n	Sample size(s).
R	Number of replications.
seed	Optional seed (the caller's random number stream is restored).

Value

A data frame with one row per sample size and parameter, including the number of ‘failed‘ replications.

Examples

```
simulation_study("norm", c(mean = 0, sd = 1), "logit", n = c(50, 100), R = 20, seed = 1)
```

transfit_methods *Residuals and predictions of a fitted model*

Description

'residuals(type = "quantile")' returns normalized quantile residuals $\Phi^{-1}(F(v_i))$, which are approximately standard normal when the model is adequate; 'type = "pit"' returns the probability integral transform $F(v_i)$, approximately uniform.

Usage

```
## S3 method for class 'transfit'
residuals(object, type = c("quantile", "pit"), ...)

## S3 method for class 'transfit'
predict(object, newdata = NULL, type = c("density", "cdf", "quantile"), ...)
```

Arguments

object	A 'transfit' object.
type	For 'predict': '"density"', '"cdf"' or '"quantile"'.
...	Unused.
newdata	Numeric vector of values in (0,1) (or probabilities when 'type = "quantile"'). Defaults to the data used for the fit.

Value

Numeric vector.

Examples

```
set.seed(1)
fit <- fit_trans(rtrans(150, "norm", c(0, 1), "logit"), "norm", "logit")
head(residuals(fit))
predict(fit, c(.2, .5), type = "cdf")
```

trans_list *List the available transformations*

Description

List the available transformations

Usage

```
trans_list()
```

Value

A data frame with the name, forward formula, inverse formula, admissible baseline domain and monotonicity of each transformation.

Examples

```
trans_list()
```

Index

baseline_params, 2
best_model, 3

check_domain, 3
confint.transfit, 4

detect_transformations, 5
dpqr_trans, 5
dtrans *(*dpqr_trans*)*, 5

fit_all, 6
fit_trans, 8
forward_transformation, 9

get_transformation, 10
gof_trans, 10

hazard_trans, 11

inverse_transformation, 12

loglik_trans, 12

moments_trans, 13

plot.transfit, 13
predict.transfit *(*transfit_methods*)*, 15
ptrans *(*dpqr_trans*)*, 5

qtrans *(*dpqr_trans*)*, 5

residuals.transfit *(*transfit_methods*)*, 15
rtrans *(*dpqr_trans*)*, 5

simulation_study, 14
summary_trans *(*moments_trans*)*, 13
survival_trans *(*hazard_trans*)*, 11

trans_list, 15
transfit_methods, 15