Package ‘autorelevate’

September 15, 2026

Type Package

Title The Autorelevated Family of Probability Distributions and Estimation Methods

Version 0.1.0

Description Implements the autorelevated family of probability distributions, obtained by applying the autorelevation transformation of Krakowski (1973) <doi:10.1051/ro/197307V201071> and Dileepkumar and Sankaran (2022) to ten baseline probability distributions: Weibull, Lomax, Burr XII, Gompertz, Log-Logistic, Chen, Exponentiated Exponential, Power Lindley, Log-normal, and Gamma. The Weibull member of the family is studied in detail by Dileep Kumar, Shabeer, and Sankaran (2025) <doi:10.1080/01966324.2026.2665479>. The Lomax member is studied by Sharma, Pal, Bhardwaj, and Tyagi (2026, submitted), who establish its upside-down bathtub hazard shape. Supplies vectorized density, distribution, survival, hazard, quantile (via the negative branch of the Lambert W function), and random-generation functions for all ten members of the family. It also implements Maximum Likelihood, Maximum Product of Spacings, Least Squares, Weighted Least Squares, and Cramer-von Mises estimation methods along with a Kolmogorov-Smirnov goodness-of-fit test, a Total Time on Test plot, and model selection by AIC, BIC, CAIC, and HQIC. It also includes a bundled bladder cancer remission dataset (Lee and Wang, 2003) for illustration.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

URL `https://github.com/vksharma-bhu/autorelevate`

BugReports `https://github.com/vksharma-bhu/autorelevate/issues`

LazyData true

Depends R (>= 3.5)

Imports stats, graphics, utils

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

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Vikas Kumar Sharma [aut, cre]

Maintainer Vikas Kumar Sharma <vksharma@bhu.ac.in>

Repository CRAN

Date/Publication 2026-09-15 14:10:02 UTC

Contents

`autorelevate-package` *autorelevate: The Autorelevated Family of Probability Distributions and Estimation Methods*

Description

Implements the **autorelevated family** of probability distributions – obtained by applying the autorelevation transform of Krakowski (1973) to a baseline lifetime distribution – across ten baseline distributions (Weibull, Lomax, Burr XII, Gompertz, Log-Logistic, Chen, Exponentiated Exponential, Power Lindley, Log-normal, and Gamma). Provides vectorized density, distribution, survival, hazard, exact quantile, and random-generation functions for each distribution, together with five estimation methods (Maximum Likelihood, Maximum Product of Spacings, Least Squares, Weighted Least Squares, and Cramer-von Mises), goodness-of-fit testing, a Total Time on Test diagnostic plot, and cross-family model selection by AIC, BIC, CAIC, and HQIC. See Details below for the mathematical derivation of the transform and the exact role of `p1`/`p2` within each baseline distribution.

Details

The relevation and autorelevation transformations. Let X and Y be non-negative random variables with survival functions $\bar{F}_1$ and $\bar{F}_2$, respectively. The *relevation transform* $\Psi(X)$ of Krakowski (1973) is the survival function of the total lifetime of a system in which an item from population 1 is replaced, at the moment of its failure at age x, by an item of the same age x from population 2. The survival function is

$$\bar{F}(x) = \bar{F}_1(x) - \int_0^x \frac{\bar{F}_2(t)}{\bar{F}_1(t)}\, dF_1(t).$$

When X and Y are identically distributed with common baseline survival function $\bar{F}_0$, the resulting variable is called the *autorelevation* of X, studied systematically by Dileepkumar and Sankaran (2022). Its survival function simplifies to

$$\bar{F}(x) = \bar{F}_0(x)\left(1 - \log \bar{F}_0(x)\right) = q(\bar{F}_0(x)),$$

where $q(t) = t(1 - \log t)$ is a concave distortion function on $[0, 1]$ with $q(0) = 0$ and $q(1) = 1$. Writing the baseline cumulative hazard as $\Lambda_0(x) = -\log \bar{F}_0(x)$, this becomes

$$\bar{F}(x) = e^{-\Lambda_0(x)}\left(1 + \Lambda_0(x)\right), \qquad f(x) = f_0(x)\,\Lambda_0(x),$$

which is the parametrization used throughout this package (`sautorelevate`, `dautorelevate`). Because the transformation adds no extra parameter to the baseline model, every autorelevated member inherits the baseline's parameters `p1`, `p2` with no increase in dimensionality.

Domain. Throughout this package, and in every formula in this documentation, $x > 0$ (support of the autorelevated family), and both baseline parameters $p1 > 0$ and $p2 > 0$ – reflecting that every one of the ten baseline distributions is itself a lifetime distribution with positive support and positive parameters. Values of `x`, `p1`, or `p2` outside this range are not part of the model; see `.validate_ar_inputs` for how invalid `p1`/`p2` are rejected with an explicit error, and the `d`/`p`/`s`/`ha` functions' own documentation for how out-of-range `x` is handled (mapped to the natural boundary value of 0, 1, or the density/hazard being 0, as appropriate, rather than raising an error).

Baseline distributions. Ten baseline probability distributions are supported via the `dist` argument, each with its own interpretation of `p1` and `p2` (both must be single positive, finite scalars – vectorizing over parameter values, the way `dnorm` vectorizes over `mean`, is not supported; only the first argument, e.g. `x` or `q`, is vectorized):

- `"weibull"`: $\Lambda_0(x) = p1\, x^{p2}$. This is the Autorelevated Weibull distribution of Dileep Kumar, Shabeer, and Sankaran (2025).
- `"lomax"`: $\Lambda_0(x) = p2 \log(1 + p1\, x)$. The autorelevated Lomax distribution is studied by Sharma, Pal, Bhardwaj, and Tyagi (2026, submitted), who establish its upside-down bathtub hazard shape with applications to cancer survival data.
- `"burr"` (Burr XII): $\Lambda_0(x) = p2 \log(1 + x^{p1})$.
- `"gompertz"`: $\Lambda_0(x) = p1\left(e^{p2x} - 1\right)$.
- `"loglogistic"`: $\Lambda_0(x) = \log\left(1 + (x/p1)^{p2}\right)$.
- `"chen"`: $\Lambda_0(x) = p1\left(e^{x^{p2}} - 1\right)$ (Chen, 2000).
- `"expexp"` (Exponentiated Exponential): $\Lambda_0(x) = -\log\left(1 - (1 - e^{-p1x})^{p2}\right)$ (Gupta & Kundu, 1999).

- "powerlindley": $\Lambda_0(x) = p1\, x^{p2} - \log\left(1 + \frac{p1\, x^{p2}}{p1+1}\right)$ (Ghitany, Al-Mutairi, Balakrishnan, & Al-Enezi, 2013); inverted in closed form using the same Lambert W_{-1} solver used elsewhere in this package (see qautorelevate).
- "lognormal": $\Lambda_0(x) = -\log \Phi\left(-\frac{\log x - p1}{p2}\right)$, p1 = log-mean, p2 = log-standard-deviation.
- "gamma": p1 = rate, p2 = shape, using pgamma/qgamma internally.

Estimation. fit_autorelevate fits any of the ten members by Maximum Likelihood (MLE), Maximum Product of Spacings (MPS), Least Squares (LS), Weighted Least Squares (WLS), or Cramer-von Mises (CvM) minimum-distance estimation, and reports AIC, BIC, CAIC, and HQIC for method = "mle". fit_all_methods and compare_families compare methods and baseline distributions, respectively. ttt_plot produces a Total Time on Test plot, a pre-fitting diagnostic for hazard shape (Aarset, 1987).

Data. bladder_cancer bundles a widely used real lifetime dataset (Lee & Wang, 2003) for illustration.

Author(s)

Maintainer: Vikas Kumar Sharma <vksharma@bhu.ac.in>

Authors:

- Vikas Kumar Sharma <vksharma@bhu.ac.in>

References

Krakowski, M. (1973). The relevation transform and a generalization of the gamma distribution function. *Revue francaise d'automatique, informatique, recherche operationnelle. Recherche operationnelle*, 7(V2), 107-120. doi:10.1051/ro/197307V201071

Dileepkumar, M., & Sankaran, P. G. (2022). Some results of auto-relevation transform in reliability analysis. *Statistics and Applications*, 20(2), 251-263.

Dileep Kumar, M., Shabeer, A. M., & Sankaran, P. G. (2025). Reliability properties and applications of autorelevated Weibull distribution. *American Journal of Mathematical and Management Sciences*, 44(3-4), 215-237. doi:10.1080/01966324.2026.2665479

Sharma, V. K., Pal, S., Bhardwaj, H., & Tyagi, V. (2026). The autorelevated Lomax distribution: An upside-down bathtub hazard model with properties and applications to cancer survival data. Submitted.

Chen, Z. (2000). A new two-parameter lifetime distribution with bathtub shape or increasing failure rate function. *Statistics & Probability Letters*, 49(2), 155-161.

Gupta, R. D., & Kundu, D. (1999). Generalized exponential distributions. *Australian & New Zealand Journal of Statistics*, 41(2), 173-188.

Ghitany, M. E., Al-Mutairi, D. K., Balakrishnan, N., & Al-Enezi, L. J. (2013). Power Lindley distribution and associated inference. *Computational Statistics & Data Analysis*, 64, 20-33.

Aarset, M. V. (1987). How to identify a bathtub hazard rate. *IEEE Transactions on Reliability*, R-36(1), 106-108.

Lee, E. T., & Wang, J. W. (2003). *Statistical Methods for Survival Data Analysis* (3rd ed.). Wiley.

See Also

Useful links:

- https://github.com/vksharma-bhu/autorelevate
- Report bugs at https://github.com/vksharma-bhu/autorelevate/issues

---

bladder_cancer *Bladder Cancer Remission Times*

Description

Remission times (in months) of 128 bladder cancer patients, a widely used benchmark lifetime dataset in the distribution-theory literature.

Usage

```
bladder_cancer
```

Format

A numeric vector of length 128 (remission times in months).

Details

This is the dataset analyzed by Dileep Kumar, Shabeer, and Sankaran (2025, Sec. 8.1), who use its `ttt_plot` shape to show the hazard rate is upside-down bathtub (UBT) and fit the Autorelevated Weibull distribution to it, outperforming the plain Weibull, exponential, exponentiated exponential, and gamma distributions by AIC, BIC, CAIC, HQIC, and Kolmogorov-Smirnov goodness-of-fit.

Source

Lee, E. T., & Wang, J. W. (2003). *Statistical Methods for Survival Data Analysis* (3rd ed.). Wiley.

References

Dileep Kumar, M., Shabeer, A. M., & Sankaran, P. G. (2025). Reliability properties and applications of autorelevated Weibull distribution. *American Journal of Mathematical and Management Sciences*, 44(3-4), 215-237. doi:10.1080/01966324.2026.2665479

Examples

```
data(bladder_cancer)
summary(bladder_cancer)
ttt_plot(bladder_cancer)
compare_families(bladder_cancer)
```

compare_families *Compare All Baseline Distribution Families for the Autorelevated Family*

Description

Fits a dataset across all ten supported baseline distribution distributions using Maximum Likelihood Estimation (MLE) and returns a ranked comparison table based on AIC, BIC, CAIC, HQIC, and goodness-of-fit statistics.

Usage

```
compare_families(data, method = "mle")
```

Arguments

`data`	Vector of positive sample observations.
`method`	Estimation method, defaults to `"mle"` (required for AIC/BIC/CAIC/HQIC; see `fit_autorelevate`).

Details

Since every autorelevated family has exactly two free parameters regardless of `dist` (the transform adds none), differences in AIC/BIC/CAIC/HQIC across rows of the returned table reflect purely the baseline distribution's shape flexibility for the data at hand, with no penalty-term confound from differing parameter counts. Ranking by AIC follows the model-selection approach used for the Autorelevated Weibull member by Dileep Kumar, Shabeer, and Sankaran (2025).

Value

An object of class `autorelevate_compare`: a data frame with one row per baseline distribution, ranked by ascending AIC.

References

Dileep Kumar, M., Shabeer, A. M., & Sankaran, P. G. (2025). Reliability properties and applications of autorelevated Weibull distribution. *American Journal of Mathematical and Management Sciences*, 44(3-4), 215-237. doi:10.1080/01966324.2026.2665479

See Also

`fit_autorelevate`, `fit_all_methods`

Other autorelevate estimation functions: `fit_all_methods()`, `fit_autorelevate()`

Examples

```
set.seed(1)
x <- rautorelevate(150, dist = "weibull", p1 = 0.5, p2 = 1.5)
comp <- compare_families(x)
print(comp)

# On the bundled real dataset
data(bladder_cancer)
compare_families(bladder_cancer)
```

---

dautorelevate *Probability Density Function (PDF) for the Autorelevated Family*

Description

Density function of the autorelevated family built from a baseline distribution specified by `dist`, `p1`, and `p2`.

Usage

```
dautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 1.5, log = FALSE)
```

Arguments

`x`	Vector of quantiles.
`dist`	Baseline distribution: `"weibull"`, `"lomax"`, `"burr"`, `"gompertz"`, `"loglogistic"`, `"chen"`, `"expexp"`, `"powerlindley"`, `"lognormal"`, or `"gamma"`. See autorelevate-package for the role of `p1` and `p2` within each distribution.
`p1`	Baseline parameter 1 (interpretation depends on `dist`).
`p2`	Baseline parameter 2 (interpretation depends on `dist`).
`log`	Logical; if TRUE, densities are returned as log(f).

Details

The autorelevation transform of Krakowski (1973) and Dileepkumar and Sankaran (2022), applied to a baseline density $f_0(x)$ with cumulative hazard $\Lambda_0(x)$, gives the PDF

$$f(x) = f_0(x)\,\Lambda_0(x), \qquad x > 0.$$

No extra parameter is introduced relative to the baseline model.

For `dist = "weibull"` ($\Lambda_0(x) = p1\,x^{p2}$), this reduces to the Autorelevated Weibull density of Dileep Kumar, Shabeer, and Sankaran (2025, Eq. 2.3),

$$f(x) = p2\,p1^2\,x^{2\,p2-1}\,e^{-p1\,x^{p2}},$$

with `p1` playing the role of the Weibull rate λ and `p2` the shape β. A convenient by-product (their Theorem 2.2) is that if X is Autorelevated Weibull with parameters `p1`, `p2`, then $Y = X^{p2}$ follows a Gamma distribution with shape 2 and scale `1 / p1`; this gives an exact distributional check for `dist = "weibull"`.

See autorelevate-package for the cumulative hazard of all ten supported baseline distributions.

Value

Numeric vector of density values.

References

Krakowski, M. (1973). The relevation transform and a generalization of the gamma distribution function. *Revue francaise d'automatique, informatique, recherche operationnelle. Recherche operationnelle*, 7(V2), 107-120. doi:10.1051/ro/197307V201071

Dileepkumar, M., & Sankaran, P. G. (2022). Some results of auto-relevation transform in reliability analysis. *Statistics and Applications*, 20(2), 251-263.

Dileep Kumar, M., Shabeer, A. M., & Sankaran, P. G. (2025). Reliability properties and applications of autorelevated Weibull distribution. *American Journal of Mathematical and Management Sciences*, 44(3-4), 215-237. doi:10.1080/01966324.2026.2665479

See Also

Other autorelevate distribution functions: `haautorelevate()`, `pautorelevate()`, `qautorelevate()`, `rautorelevate()`, `sautorelevate()`

Examples

```
x <- seq(0.1, 5, by = 0.1)

# Weibull baseline (Autorelevated Weibull)
fx <- dautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 1.5)
plot(x, fx, type = "l", ylab = "Density", main = "Autorelevated Weibull")

# Log-density, useful for likelihood-based estimation
dautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 1.5, log = TRUE)

# The five newer baseline distributions
dautorelevate(x, dist = "chen", p1 = 0.3, p2 = 1.4)
dautorelevate(x, dist = "expexp", p1 = 0.8, p2 = 2.3)
dautorelevate(x, dist = "powerlindley", p1 = 1.2, p2 = 1.7)
dautorelevate(x, dist = "lognormal", p1 = 0.5, p2 = 0.8)
dautorelevate(x, dist = "gamma", p1 = 1.1, p2 = 2.4)
```

`fit_all_methods` *Compare All Estimation Methods for an Autorelevated Model*

Description

Fits a given baseline distribution using all five methods (MLE, MPS, LS, WLS, CvM) and returns a comparison matrix to evaluate parameter stability and Kolmogorov-Smirnov goodness-of-fit across methods.

Usage

```
fit_all_methods(data, dist = "weibull")
```

Arguments

data	Vector of positive sample observations.
dist	Baseline distribution.

Details

Each method targets a different discrepancy between the fitted and empirical distribution (likelihood, spacing products, or a weighted distance between the fitted CDF and the empirical CDF; see `fit_autorelevate` for the exact objective of each method). A method that fails to converge for the data at hand contributes a row of `NA` (via `.robust_optim`'s error, caught here) rather than stopping the comparison for the other methods.

Value

A matrix comparing parameter estimates and KS statistics across methods (rows: `MLE`, `MPS`, `LS`, `WLS`, `CVM`; columns: `Estimate_p1`, `Estimate_p2`, `KS_Statistic`).

See Also

`fit_autorelevate`, `compare_families`

Other autorelevate estimation functions: `compare_families()`, `fit_autorelevate()`

Examples

```
set.seed(1)
x <- rautorelevate(100, dist = "weibull", p1 = 0.5, p2 = 1.5)
fit_all_methods(x, dist = "weibull")
```

`fit_autorelevate` *Fit Autorelevated Family Parameters*

Description

Fits the parameters `p1`, `p2` of an autorelevated distribution to data by Maximum Likelihood (MLE), Maximum Product of Spacings (MPS), Least Squares (LS), Weighted Least Squares (WLS), or Cramer-von Mises (CvM) minimum-distance estimation.

Usage

```
fit_autorelevate(data, dist = "weibull", method = "mle")
```

Arguments

`data`	Vector of positive sample observations. Missing, non-finite, or non-positive values are dropped with a warning before fitting.
`dist`	Baseline distribution: `"weibull"`, `"lomax"`, `"burr"`, `"gompertz"`, `"loglogistic"`, `"chen"`, `"expexp"`, `"powerlindley"`, `"lognormal"`, or `"gamma"`.
`method`	Estimation routine: `"mle"`, `"mps"`, `"ls"`, `"wls"`, or `"cvm"`.

Details

Let $x_{(1)} \leq \cdots \leq x_{(n)}$ be the ordered, strictly positive observations. Optimization is over $(\log p1, \log p2)$ (via `optim`, with an automatic Nelder-Mead/restart fallback if the primary optimizer fails to converge; see `.robust_optim`), which enforces positivity without constrained optimization. The starting point itself is chosen adaptively from the data via a coarse log-scale grid search (see `.mle_seed_grid`), rather than a single fixed default, since a fixed starting point can silently overflow or underflow for several distributions when the data's scale differs greatly from 1.

- **MLE** maximizes the log-likelihood $\sum_i \log f(x_{(i)})$, following the Autorelevated Weibull log-likelihood derivation of Dileep Kumar, Shabeer, and Sankaran (2025, Sec. 6.1), generalized to any baseline via `dautorelevate`. Standard errors are obtained from the observed Fisher information (the numerical Hessian at the optimum, evaluated on the $(\log p1, \log p2)$ scale used internally by the optimizer and mapped back to the $(p1, p2)$ scale via the delta method, $\widehat{\mathrm{Var}}(p) \approx D\,\widehat{\mathrm{Var}}(\log p)\,D$ with $D = \mathrm{diag}(p1, p2)$).
- **LS and WLS** minimize, respectively,
$$\sum_{i=1}^{n}\left(F(x_{(i)}) - \frac{i}{n+1}\right)^2 \quad \text{and} \quad \sum_{i=1}^{n}\frac{(n+1)^2(n+2)}{i(n+1-i)}\left(F(x_{(i)}) - \frac{i}{n+1}\right)^2,$$
following Swain, Venkatraman, and Wilson (1988).
- **CvM** minimizes the Cramer-von Mises statistic of Choi and Bulgren (1968),
$$\frac{1}{12n} + \sum_{i=1}^{n}\left(F(x_{(i)}) - \frac{2i-1}{2n}\right)^2.$$
- **MPS** maximizes the geometric mean of the spacings $D_i = F(x_{(i)}) - F(x_{(i-1)})$ (with $F(x_{(0)}) = 0$, $F(x_{(n+1)}) = 1$), i.e., minimizes $-\sum_i \log D_i$.

For MPS/LS/WLS/CvM, the optimizer is seeded at the MLE solution.

Model-selection criteria (`method = "mle"` only, since all four require the maximized log-likelihood $\hat{\ell}$ and are only asymptotically justified through it), with $k = 2$ parameters for every distribution in this package:

$$\mathrm{AIC} = 2k - 2\hat{\ell}, \quad \mathrm{BIC} = k\log n - 2\hat{\ell}, \quad \mathrm{CAIC} = \frac{2kn}{n-k-1} - 2\hat{\ell}, \quad \mathrm{HQIC} = 2k\log(\log n) - 2\hat{\ell},$$

matching the criteria reported by Dileep Kumar, Shabeer, and Sankaran (2025, Sec. 8.1). `CAIC` is `NA` when $n \leq k + 1$ (too few observations for the correction term to be defined); note that this particular `CAIC` formula is algebraically identical to the corrected AIC (AICc) of Hurvich and Tsai (1989) — some sources use "CAIC" for a different formula (Bozdogan, 1987), so compare against the exact formula above, not just the acronym, when cross-referencing other software.

A Kolmogorov-Smirnov goodness-of-fit test (`ks.test`) comparing the data to the fitted CDF is returned for every method.

Value

Object of class `autorelevate_fit`: a list with components `par` (named numeric vector `p1`, `p2`), `vcov`, `value`, `convergence`, `data`, `dist`, `method`, `n`, `log_lik`, `aic`, `bic`, `caic`, `hqic`, `ks_stat`, and `ks_pval`.

References

Choi, K., & Bulgren, W. (1968). An estimation procedure for mixtures of distributions. *Journal of the Royal Statistical Society Series B*, 30(3), 444-460.

Swain, J. J., Venkatraman, S., & Wilson, J. R. (1988). Least-squares estimation of distribution functions in Johnson's translation system. *Journal of Statistical Computation and Simulation*, 29(4), 271-297.

Hurvich, C. M., & Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, 76(2), 297-307.

Dileep Kumar, M., Shabeer, A. M., & Sankaran, P. G. (2025). Reliability properties and applications of autorelevated Weibull distribution. *American Journal of Mathematical and Management Sciences*, 44(3-4), 215-237. doi:10.1080/01966324.2026.2665479

See Also

`fit_all_methods`, `compare_families`

Other autorelevate estimation functions: `compare_families()`, `fit_all_methods()`

Examples

```
set.seed(1)
x <- rautorelevate(100, dist = "weibull", p1 = 0.5, p2 = 1.5)

fit_mle <- fit_autorelevate(x, dist = "weibull", method = "mle")
print(fit_mle)
summary(fit_mle)

fit_ls <- fit_autorelevate(x, dist = "weibull", method = "ls")
fit_ls$par

# A real dataset
data(bladder_cancer)
fit_bc <- fit_autorelevate(bladder_cancer, dist = "weibull", method = "mle")
summary(fit_bc)
```

`haautorelevate` *Hazard Rate Function for the Autorelevated Family*

Description

Hazard (failure) rate function of the autorelevated family built from a baseline distribution specified by `dist`, `p1`, and `p2`.

Usage

```
haautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 1.5)
```

Arguments

x	Vector of quantiles.
dist	Baseline distribution.
p1	Baseline parameter 1.
p2	Baseline parameter 2.

Details

The hazard rate is the ratio of the PDF to the survival function, $h(x) = f(x)/\bar{F}(x)$. Substituting $f(x) = f_0(x)\Lambda_0(x)$ and $\bar{F}(x) = e^{-\Lambda_0(x)}(1 + \Lambda_0(x))$, and using $f_0(x) = h_0(x)\, e^{-\Lambda_0(x)}$ for the baseline hazard $h_0(x)$, gives the compact identity

$$h(x) = h_0(x)\, \frac{\Lambda_0(x)}{1 + \Lambda_0(x)}.$$

Since $\Lambda_0/(1 + \Lambda_0) \in (0, 1)$ is increasing in Λ_0, the autorelevated hazard is always a *damped* version of the baseline hazard, but the damping factor grows with x, which is what allows a monotone baseline hazard to become non-monotone after autorelevation. Dileep Kumar, Shabeer, and Sankaran (2025, Theorem 4.1) show that for `dist = "weibull"` with shape `p2` = β: the hazard is increasing (IHR) for $\beta > 1$, upside-down bathtub (UBT) for $1/2 < \beta < 1$ (a strict, open interval – at $\beta = 1$ exactly, direct calculus on $h(x) = p1^2 x/(1 + p1x)$ shows $h'(x) = p1^2/(1 + p1x)^2 > 0$ for all x, i.e. strictly increasing with no decreasing phase, so $\beta = 1$ belongs with the IHR case, not UBT), and decreasing (DHR) for $0 < \beta \leq 1/2$.

Value

Numeric vector of hazard rate values.

References

Dileep Kumar, M., Shabeer, A. M., & Sankaran, P. G. (2025). Reliability properties and applications of autorelevated Weibull distribution. *American Journal of Mathematical and Management Sciences*, 44(3-4), 215-237. doi:10.1080/01966324.2026.2665479

See Also

Other autorelevate distribution functions: `dautorelevate()`, `pautorelevate()`, `qautorelevate()`, `rautorelevate()`, `sautorelevate()`

Examples

```
x <- seq(0.1, 5, by = 0.1)

# Upside-down bathtub hazard (1/2 < beta <= 1)
h_ubt <- haautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 0.8)
```

```
plot(x, h_ubt, type = "l", ylab = "Hazard rate", main = "UBT-shaped hazard")

# Increasing hazard (beta > 1)
h_ihr <- haautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 1.5)
lines(x, h_ihr, col = "blue")
```

`pautorelevate` *Cumulative Distribution Function (CDF) for the Autorelevated Family*

Description

Distribution function of the autorelevated family built from a baseline distribution specified by `dist`, `p1`, and `p2`.

Usage

```
pautorelevate(
  q,
  dist = "weibull",
  p1 = 0.5,
  p2 = 1.5,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

`q`	Vector of quantiles.
`dist`	Baseline distribution.
`p1`	Baseline parameter 1.
`p2`	Baseline parameter 2.
`lower.tail`	Logical; if TRUE (default), probabilities are P[X <= x], otherwise P[X > x].
`log.p`	Logical; if TRUE, probabilities are returned as log(p).

Details

Using the baseline cumulative hazard $\Lambda_0(x)$, the survival function of the autorelevation transform is

$$\bar{F}(x) = e^{-\Lambda_0(x)}\,(1 + \Lambda_0(x))$$

(see `sautorelevate` for the derivation), so the CDF is

$$F(x) = 1 - e^{-\Lambda_0(x)}\,(1 + \Lambda_0(x))\,.$$

Equivalently, $\bar{F}(x) = q(\bar{F}_0(x))$ for the concave distortion function $q(t) = t(1 - \log t)$ on $[0, 1]$: the autorelevated family is, for every choice of `dist`, a distorted version of its own baseline distribution (Dileepkumar & Sankaran, 2022).

Value

Numeric vector of cumulative probability values.

References

Dileepkumar, M., & Sankaran, P. G. (2022). Some results of auto-relevation transform in reliability analysis. *Statistics and Applications*, 20(2), 251-263.

See Also

Other autorelevate distribution functions: `dautorelevate()`, `haautorelevate()`, `qautorelevate()`, `rautorelevate()`, `sautorelevate()`

Examples

```
q <- seq(0.1, 5, by = 0.5)
pautorelevate(q, dist = "weibull", p1 = 0.5, p2 = 1.5)

# Upper tail P[X > x]
pautorelevate(q, dist = "weibull", p1 = 0.5, p2 = 1.5, lower.tail = FALSE)

# CDF + survival function sum to 1, for every baseline distribution
cdf <- pautorelevate(q, dist = "gamma", p1 = 1.1, p2 = 2.4)
surv <- sautorelevate(q, dist = "gamma", p1 = 1.1, p2 = 2.4)
all.equal(cdf + surv, rep(1, length(q)))
```

`plot.autorelevate_fit` *Plot Diagnostics for an Autorelevated Fit*

Description

Two-panel diagnostic plot: a histogram of the data with the fitted density overlaid, and the empirical CDF with the fitted CDF overlaid.

Usage

```
## S3 method for class 'autorelevate_fit'
plot(x, ...)
```

Arguments

`x`	An autorelevate_fit object.
`...`	Unused additional arguments.

Value

Invisibly, x.

Examples

```
set.seed(1)
x <- rautorelevate(200, dist = "weibull", p1 = 0.5, p2 = 1.5)
fit <- fit_autorelevate(x, dist = "weibull", method = "mle")
plot(fit)
```

`print.autorelevate_compare`

Print Method for autorelevate_compare Objects

Description

Print Method for autorelevate_compare Objects

Usage

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

Arguments

`x`	An autorelevate_compare object.
`...`	Additional arguments (unused).

Value

Invisibly, x.

Examples

```
set.seed(1)
x <- rautorelevate(100, dist = "weibull", p1 = 0.5, p2 = 1.5)
print(compare_families(x))
```

`print.autorelevate_fit`

Print Summary of an Autorelevated Fit

Description

Print Summary of an Autorelevated Fit

Usage

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

Arguments

`x`	An autorelevate_fit object.
`...`	Unused additional arguments.

Value

Invisibly, x.

Examples

```
set.seed(1)
x <- rautorelevate(100, dist = "weibull", p1 = 0.5, p2 = 1.5)
print(fit_autorelevate(x, dist = "weibull", method = "mle"))
```

`qautorelevate` *Quantile Function via Exact Lambert W_-1 Inversion*

Description

Quantile function of the autorelevated family, obtained analytically via the negative branch of the Lambert W function rather than numerical root-finding.

Usage

```
qautorelevate(p, dist = "weibull", p1 = 0.5, p2 = 1.5, lower.tail = TRUE)
```

Arguments

`p`	Vector of probabilities.
`dist`	Baseline distribution.
`p1`	Baseline parameter 1.
`p2`	Baseline parameter 2.
`lower.tail`	Logical; if TRUE (default), probabilities are P[X <= x].

Details

We want x_p solving $F(x_p) = p$, i.e., $\bar{F}(x_p) = 1 - p$. Writing $S = 1 - p$ and $y = 1 + \Lambda_0(x_p)$, the survival identity $\bar{F}(x) = e^{-\Lambda_0(x)}(1 + \Lambda_0(x))$ becomes $y\,e^{-y} = S/e$. Since $W_{-1}(z)$ is defined by $W_{-1}(z)\,e^{W_{-1}(z)} = z$ and $y \geq 1$, setting $w = -y$ gives $we^w = -S/e$, so

$$w = W_{-1}(-S/e), \qquad \Lambda_0(x_p) = y - 1 = -1 - w.$$

Applying the baseline inverse cumulative hazard then gives the exact quantile $x_p = \Lambda_0^{-1}(-1 - W_{-1}(-S/e))$. For most baselines Λ_0^{-1} is elementary algebra; for `"powerlindley"` it is itself obtained via a second application of W_{-1} (see baseline_internal), so no numerical root-finding is used anywhere in this function for any baseline. This generalizes Theorem 2.3 of Dileep Kumar, Shabeer, and Sankaran (2025), which gives the corresponding identity for the Weibull baseline.

Value

Numeric vector of evaluated quantiles.

References

Dileep Kumar, M., Shabeer, A. M., & Sankaran, P. G. (2025). Reliability properties and applications of autorelevated Weibull distribution. *American Journal of Mathematical and Management Sciences*, 44(3-4), 215-237. doi:10.1080/01966324.2026.2665479

Ghitany, M. E., Al-Mutairi, D. K., Balakrishnan, N., & Al-Enezi, L. J. (2013). Power Lindley distribution and associated inference. *Computational Statistics & Data Analysis*, 64, 20-33.

See Also

Other autorelevate distribution functions: `dautorelevate()`, `haautorelevate()`, `pautorelevate()`, `rautorelevate()`, `sautorelevate()`

Examples

```
p <- c(0.1, 0.25, 0.5, 0.75, 0.9)
qautorelevate(p, dist = "weibull", p1 = 0.5, p2 = 1.5)

# Round trip: CDF then quantile recovers the original x, for every distribution
x <- seq(0.5, 3, by = 0.5)
cdf <- pautorelevate(x, dist = "powerlindley", p1 = 1.2, p2 = 1.7)
qautorelevate(cdf, dist = "powerlindley", p1 = 1.2, p2 = 1.7)
```

`rautorelevate` *Random Generation for the Autorelevated Family*

Description

Draws random variates from the autorelevated family via the inverse transform method, using the exact quantile function `qautorelevate`.

Usage

```
rautorelevate(n, dist = "weibull", p1 = 0.5, p2 = 1.5)
```

Arguments

`n`	Number of observations.
`dist`	Baseline distribution.
`p1`	Baseline parameter 1.
`p2`	Baseline parameter 2.

Details

Because `qautorelevate` is available in closed form (no numerical inversion), generation is exact and fast: a single uniform draw $U \sim \mathrm{Unif}(0, 1)$ per variate is transformed via $X = Q(U)$, where Q is the autorelevated quantile function.

Value

Numeric vector of simulated random variates.

See Also

Other autorelevate distribution functions: `dautorelevate()`, `haautorelevate()`, `pautorelevate()`, `qautorelevate()`, `sautorelevate()`

Examples

```
set.seed(1)
x <- rautorelevate(500, dist = "weibull", p1 = 0.5, p2 = 1.5)
hist(x, breaks = 30, probability = TRUE, main = "Simulated Autorelevated Weibull sample")
curve(dautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 1.5),
      add = TRUE, col = "blue", lwd = 2)
```

`sautorelevate` *Survival Function for the Autorelevated Family*

Description

Survival (reliability) function of the autorelevated family built from a baseline distribution specified by `dist`, `p1`, and `p2`.

Usage

```
sautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 1.5, log = FALSE)
```

Arguments

`x`	Vector of quantiles.
`dist`	Baseline distribution.
`p1`	Baseline parameter 1.
`p2`	Baseline parameter 2.
`log`	Logical; if TRUE, survival values are returned as log(S).

Details

Let $\bar{F}_0(x)$ and $\Lambda_0(x) = -\log \bar{F}_0(x)$ be the baseline survival function and cumulative hazard. For identically distributed populations, the relevation transform of Krakowski (1973) reduces to the autorelevation survival function

$$\bar{F}(x) = \bar{F}_0(x) \left(1 - \log \bar{F}_0(x)\right) = e^{-\Lambda_0(x)} \left(1 + \Lambda_0(x)\right).$$

For `dist = "weibull"` this is the Autorelevated Weibull survival function of Dileep Kumar, Shabeer, and Sankaran (2025, Eq. 2.2), $\bar{F}(x) = e^{-p1\, x^{p2}} \left(1 + p1\, x^{p2}\right)$.

Value

Numeric vector of survival values.

References

Krakowski, M. (1973). The relevation transform and a generalization of the gamma distribution function. *Revue francaise d'automatique, informatique, recherche operationnelle. Recherche operationnelle*, 7(V2), 107-120. doi:10.1051/ro/197307V201071

See Also

Other autorelevate distribution functions: `dautorelevate()`, `haautorelevate()`, `pautorelevate()`, `qautorelevate()`, `rautorelevate()`

Examples

```
x <- seq(0.1, 5, by = 0.5)
sautorelevate(x, dist = "weibull", p1 = 0.5, p2 = 1.5)
sautorelevate(x, dist = "gompertz", p1 = 0.3, p2 = 0.5, log = TRUE)
```

`summary.autorelevate_fit` *Summary Table for an Autorelevated Fit*

Description

Reports parameter estimates and, for `method = "mle"`, standard errors, log-likelihood, AIC, BIC, CAIC, and HQIC (see Details in `fit_autorelevate` for why these are MLE-only), together with the Kolmogorov-Smirnov goodness-of-fit statistic for every method.

Usage

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

Arguments

`object`	An autorelevate_fit object.
`...`	Unused additional arguments.

Value

Matrix of estimates and standard errors (invisibly).

Examples

```
set.seed(1)
x <- rautorelevate(100, dist = "weibull", p1 = 0.5, p2 = 1.5)
fit <- fit_autorelevate(x, dist = "weibull", method = "mle")
summary(fit)
```

`ttt_plot` *Total Time on Test (TTT) Plot*

Description

Produces the empirical Total Time on Test (TTT) plot of Aarset (1987), a standard graphical diagnostic for identifying the shape of a lifetime data set's hazard rate before fitting any model.

Usage

```
ttt_plot(data, plot = TRUE, ...)
```

Arguments

`data`	Vector of positive sample observations. Missing, non-finite, or non-positive values are dropped with a warning before computing the plot.
`plot`	Logical; if TRUE (default) draws the plot. If FALSE, only the underlying data frame is returned (invisibly either way).
`...`	Additional arguments passed to `plot`.

Details

For ordered data $x_{(1)} \leq \cdots \leq x_{(n)}$, the empirical TTT statistic is

$$G(r/n) = \left(\sum_{i=1}^{r} x_{(i)} + (n-r)x_{(r)} \right) \Big/ \sum_{i=1}^{n} x_{(i)}, \qquad r = 1, \ldots, n.$$

Plotting $G(r/n)$ against r/n and comparing to the 45-degree line indicates the hazard shape: concave indicates an increasing hazard (IHR), convex indicates a decreasing hazard (DHR), convex then concave indicates a bathtub hazard, and concave then convex indicates an upside-down bathtub (UBT) hazard (Aarset, 1987).

Value

Invisibly, a data frame with columns `r_over_n` and `TTT`.

References

Aarset, M. V. (1987). How to identify a bathtub hazard rate. *IEEE Transactions on Reliability*, R-36(1), 106-108.

Examples

```
data(bladder_cancer)
ttt_plot(bladder_cancer)   # concave-then-convex: UBT hazard
```

Index