Package 'tweedie'

November 14, 2026

Version 3.2.2

Date 2026-11-15

Title Evaluation of Tweedie Exponential Family Models

Depends R (>= 2.8.0)

Encoding UTF-8

Imports methods, stats, graphics, lifecycle (>= 1.0.0), statmod (>= 1.4.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Description Maximum likelihood computations for Tweedie families, including the series expansion (Dunn and Smyth, 2005; <doi:10.1007/s11222-005-4070-y>) and the Fourier inversion (Dunn and Smyth, 2008; <doi:10.1007/s11222-007-9039-6>), and related methods.

License GPL (>= 2)

NeedsCompilation yes

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.1.1

Author Peter K. Dunn [cre, aut]

Maintainer Peter K. Dunn <pdunn2@usc.edu.au>

Repository CRAN

Date/Publication 2026-11-14 22:20:02 UTC

Contents

tweedie-package 2
dtweedie_inversion 2
dtweedie_saddle 4
dtweedie_series 5
logLiktweedie 6
ptweedie_inversion 7
ptweedie_saddle 9

ptweedie_series 10
ptweedie_tail_integrate 11
Tweedie 12
tweedie_AIC 14
tweedie_convert 16
tweedie_dev 17
tweedie_integrand 18
tweedie_lambda 20
tweedie_plot 21
tweedie_profile 22

Index **26**

`tweedie-package` *Evaluation of Tweedie Exponential Family Models*

Description

This package provides maximum likelihood computations for Tweedie families, including the series expansion (Dunn and Smyth, 2005) and the Fourier inversion (Dunn and Smyth, 2008), and related methods.

Author(s)

Maintainer: Peter K. Dunn <pdunn2@usc.edu.au>

Authors:

- Peter K. Dunn <pdunn2@usc.edu.au>

`dtweedie_inversion` *Fourier Inversion Evaluation for the Tweedie Probability Function*

Description

Evaluates the probability density function (PDF) for Tweedie distributions using Fourier inversion, for given values of the dependent variable `y`, the mean `mu`, dispersion `phi`, and power parameter `power`. *Not usually called by general users*, but can be used in the case of evaluation problems.

Usage

```
dtweedie_inversion(y, mu, phi, power, method = 3, verbose = FALSE,
                           details = FALSE, IGexact = TRUE)

dtweedie.inversion(y, power, mu, phi, method = 3, verbose, details)
```

Arguments

y	vector of quantiles.
mu	the mean parameter μ.
phi	the dispersion parameter ϕ.
power	scalar; the power parameter p.
method	the method to use; one of 1, 2, or 3 (the default), or NULL to choose, for each value, the method with the smallest multiplier (Dunn and Smyth, 2008). Method 3 (rescaling y to one) keeps the relative accuracy in the tails, where Methods 1 and 2 can lose it entirely. (Before version 3.2.0, this argument was ignored, and the method was always chosen as for NULL.)
verbose	logical; if TRUE, display some internal computation details. The default is FALSE.
details	logical; if TRUE, return a list with basic details of the integration. The default is FALSE.
IGexact	logical; if TRUE (the default), evaluate the inverse Gaussian distribution using the 'exact' values, otherwise uses inversion.

Value

A numeric vector of densities if details=FALSE; if details = TRUE, a list containing density (a vector of the values of the density), logdensity (its logarithm, computed without underflow), regions (a vector of the number of integration regions used),method (a vector giving the evaluation method used; see the Note below on the three methods), and exitstatus (a vector, where a 1 for any value means a computational problem or target relative accuracy not reached, for the corresponding observation).

Note

The 'exact' values for the inverse Gaussian distribution are not really exact, but evaluated using statmod::[pdpqr]invgauss, all of which are very accurate (Giner & Smyth, 2016). For special cases of p (i.e., $p = 0, 1, 2, 3$), where no inversion is needed, regions and method are set to NA for all values of y. For special cases of y for other values of p (i.e., $P(Y = 0)$), regions and method are set to NA.

The three methods are described in Dunn & Smyth (2008).

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Giner G., Smyth G. K. (2016). statmod: probability calculations for the inverse Gaussian distribution. *The R Journal*, **8**(1), 339–351. doi:10.32614/RJ2016024

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Examples

```
# Plot a Tweedie density
y <- seq(0.02, 4, length = 50)
fy <- dtweedie_inversion(y, mu = 1, phi = 1, power = 1.1)
plot(y, fy, type = "l", lwd = 2, ylab = "Density")
```

`dtweedie_saddle` *Tweedie densities evaluation using the saddlepoint approximation*

Description

Density function for the Tweedie EMDs using a saddlepoint approximation.

Usage

```
dtweedie_saddle(y, xi = NULL, mu, phi, eps = 1/6, power = NULL)

dtweedie.saddle(y, xi = NULL, mu, phi, eps = 1/6, power = NULL)
```

Arguments

`y`	vector of quantiles.
`xi`	scalar; the value of ξ such that the variance is var$[Y] = \phi\mu^{\xi}$. A synonym for `power`.
`mu`	vector of mean μ.
`phi`	vector of dispersion parameters ϕ.
`eps`	the offset in computing the variance function; the default is `eps=1/6` (as suggested by Nelder and Pregibon, 1987).
`power`	scalar; a synonym for ξ, the Tweedie index parameter.

Value

A numeric vector of densities.

Accuracy

The saddlepoint approximation is an approximation, not an exact evaluation, and no warning is given when it is inaccurate. Its relative error grows with $z = \phi y^{p-2}$, and is roughly of order $z/10$ or smaller (for example, exactly $z/12$ to first order for the gamma distribution, $p = 2$); it is small only when z is small. For $1 < p < 2$, the default offset `eps = 1/6` (which keeps the density finite near $y = 0$) adds a further relative error of about $(1 + \epsilon/y)^{-p/2} - 1$, which does not shrink as ϕ becomes small (about 11% at $y = 1$ when $p = 1.5$).

`dtweedie()` uses a corrected saddlepoint approximation only where its estimated relative error is below 10^{-13}; use `dtweedie()` for accurate densities.

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Nelder, J. A. and Pregibon, D. (1987). An extended quasi-likelihood function *Biometrika*, **74**(2), 221–232. doi:10.1093/biomet/74.2.221

Examples

```
# Plot a Tweedie density
y <- seq(0.01, 4, length = 50)
fy <- dtweedie_saddle(y, power = 1.1, mu = 1, phi = 1)
plot(y, fy, type = "l", lwd = 2, ylab = "Density")
```

`dtweedie_series` *Series Evaluation for the Tweedie Probability Function*

Description

Evaluates the probability density function (PDF) for Tweedie distributions using an infinite series, for given values of the dependent variable `y`, the mean `mu`, dispersion `phi`, and power parameter `power`. *Not usually called by general users*, but can be used in the case of evaluation problems.

Usage

```
dtweedie_series(y, power, mu, phi, details = FALSE)

dtweedie.series(y, power, mu, phi)
```

Arguments

`y`	vector of quantiles.
`power`	scalar; the value of p such that the variance is $\text{var}[Y] = \phi\mu^p$.
`mu`	vector of mean μ.
`phi`	vector of dispersion parameters ϕ.
`details`	logical; if `TRUE`, returns a list: the density (`density`), its logarithm (`logf`, computed without underflow), an estimate of its relative error from rounding (`relerr`; for $p > 2$, where the series alternates in sign; zero otherwise), and the range of terms summed (`lo`, `hi`).

Value

A numeric vector of probabilities.

Accuracy

For $1 < p < 2$, the terms of the series are all positive, and the density keeps full relative accuracy. For $p > 2$, the terms alternate in sign, and rounding error can cancel many digits. An estimate of the resulting relative error is returned (as `relerr`) when `details = TRUE`; no warning is given for a large estimated error. Where the series cannot be summed at all, `NaN` is returned with a warning. `dtweedie()` uses the series for $p > 2$ only where this estimated relative error is below 10^{-12}.

References

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Examples

```
# Plot a Tweedie density
y <- seq(0.01, 4, length = 50)
fy <- dtweedie_series(y, power = 1.1, mu = 1, phi = 1)
plot(y, fy, type = "l", lwd = 2, ylab = "Density")
```

---

logLiktweedie *Log-likelihood for Tweedie distributions*

Description

Evaluates the log-likelihood for a fitted Tweedie GLM.

Usage

```
logLiktweedie(glm.obj, dispersion = NULL)
```

Arguments

`glm.obj`	a fitted `glm` object, fitted using the `tweedie` family.
`dispersion`	the dispersion parameter, usually extracted from `glm.obj`; however, occasionally a specified value of the dispersion may be needed.

Details

The log-Likelihood is computed by evaluating the density function.

Value

The value of the computed log-likelihood.

Note

Evaluating the likelihood can be time consuming, so the function may take some time for large data sets.

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Sakamoto, Y., Ishiguro, M., and Kitagawa G. (1986). *Akaike Information Criterion Statistics*. D. Reidel Publishing Company.

See Also

dtweedie

Examples

```
# Fit a Tweedie density using  tweedie  family function from  statmod
pretend <- data.frame( y = stats::rgamma(20, shape = 1, rate = 1) )
fit <- glm(y ~ 1, data = pretend,
           family = statmod::tweedie(link.power = 0, var.power = 2.1))

# Compute the AIC
logLiktweedie(fit)
```

---

ptweedie_inversion *Fourier Inversion Evaluation for the Tweedie Distribution Function*

Description

Evaluates the distribution function (DF) for Tweedie distributions using Fourier inversion, for given values of the dependent variable `y`, the mean `mu`, dispersion `phi`, and power parameter `power`. *Not usually called by general users*, but can be in the case of evaluation problems.

Usage

```
ptweedie_inversion(q, mu, phi, power, lower.tail = TRUE, log.p = FALSE, IGexact = TRUE,
                           verbose = FALSE, details = FALSE)

ptweedie.inversion(q, power, mu, phi, verbose, details)
```

Arguments

`q`	vector of quantiles.
`mu`	the mean parameter.
`phi`	the dispersion parameter.
`power`	the power parameter p.

`lower.tail`	logical; if `TRUE` (the default) computes the distribution function $F(y)$; if `FALSE`, computes $1 - F(y)$.
`log.p`	logical; if `TRUE`, probabilities are returned as $\log(p)$, computed directly from the already-correctly-computed tail probability rather than as `log(ptweedie_inversion(...))` after the fact. The default is `FALSE`.
`IGexact`	logical; if `TRUE` (the default), evaluate the inverse Gaussian distribution using the 'exact' values, otherwise uses inversion.
`verbose`	logical; if `TRUE`, displays some internal computation details. The default is `FALSE`.
`details`	logical; if `TRUE`, returns the value of the distribution and some information about the integration. The default is `FALSE`.

Value

If `details = FALSE`, a numeric vector of the distribution function values; if `details = TRUE`, a list containing `cdf` (a vector of the values of the distribution function), `regions` (a vector of the number of integration regions used), `exitstatus` (a vector, where a 1 for any value means a computational problem or target relative accuracy not reached, for the corresponding observation), and `small_tail` (a logical vector; `TRUE` where the inversion returned a tail probability below 10^{-8}, whose relative accuracy may be poor; see the Accuracy section).

For special cases of p (i.e., $p = 0, 1, 2, 3$), where no inversion is needed, `regions` is set to `NA` for all values of `q`. For special cases of `q` for other values of p (i.e., $P(Y = 0)$), `regions` is set to `NA`.

Accuracy

The inversion computes $F(y) = 1/2 - I/\pi$ (and $1 - F(y) = 1/2 + I/\pi$ for the upper tail), where I is the integral evaluated numerically (for $1 < p < 2$, the constant $1/2$ is replaced by $(1 \pm e^{-\lambda})/2$). When the requested tail probability P is small, I/π is close to $1/2$ and most of the digits cancel, so the error is *absolute* (a few times 10^{-16}) rather than relative: the relative error is about $10^{-16}/P$ or a few times larger. Tail probabilities below about 10^{-8} may therefore have poor relative accuracy, and those far below 10^{-16} may have no correct digits. These values are flagged by `small_tail` when `details = TRUE`; no warning is given.

`ptweedie()` does not have this limitation: it recomputes small tail probabilities from the inversion by integrating the density, which keeps its relative accuracy far into the tails. Use `ptweedie()` for small tail probabilities.

Note

The 'exact' values for the inverse Gaussian distribution are not really exact, but evaluated using statmod::[pdpqr]invgauss, all of which are very accurate (Giner & Smyth, 2016).

References

Dunn, Peter K and Smyth, Gordon K (Submitted). Evaluation of the Tweedie exponential dispersion model distribution function. Submitted.

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Giner G., Smyth G. K. (2016). statmod: probability calculations for the inverse Gaussian distribution. *The R Journal*, **8**(1), 339–351. doi:10.32614/RJ2016024

Examples

```
# Plot a Tweedie distribution function
y <- seq(0.01, 4, length = 50)
Fy <- ptweedie_inversion(y, mu = 1, phi = 1, power = 1.1)
plot(y, Fy, type = "l", lwd = 2, ylab = "Distribution function")
```

`ptweedie_saddle` *Tweedie CDF evaluation using the saddlepoint approximation*

Description

Distribution function for the Tweedie EMDs using a saddlepoint approximation.

Usage

```
ptweedie_saddle(q, mu, phi, power, lower.tail = TRUE, log.p = FALSE)
```

Arguments

`q`	vector of quantiles.
`mu`	vector of mean μ.
`phi`	vector of dispersion parameters ϕ.
`power`	scalar; the value of p such that the variance is $\text{var}[Y] = \phi\mu^p$.
`lower.tail`	logical; if `TRUE` (the default), computes the lower tail $F(y)$; otherwise computes $S(y) = 1 - F(y)$.
`log.p`	logical; if `TRUE`, probabilities are returned as $\log(p)$. The default is `FALSE`.

Value

A numeric vector of densities.

References

Dunn, Peter K and Smyth, Gordon K (Submitted). Evaluation of the Tweedie exponential dispersion model distribution function. Submitted.

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Nelder, J. A. and Pregibon, D. (1987). An extended quasi-likelihood function *Biometrika*, **74**(2), 221–232. doi:10.1093/biomet/74.2.221

Examples

```
# Plot a Tweedie distribution function
y <- seq(0.01, 4, length = 50)
Fy <- ptweedie_saddle(y, power = 2.1, mu = 1, phi = 1)
plot(y, Fy, type = "l", lwd = 2, ylab = "CDF")
```

---

ptweedie_series *Series Evaluation for the Tweedie Distribution Function*

Description

Evaluates the distribution function (DF) for Tweedie distributions with $1 < p < 2$ using an infinite series, for given values of the dependent variable y, the mean mu, dispersion phi, and power parameter power. *Not usually called by general users*, but can be in the case of evaluation problems.

Usage

```
ptweedie_series(q, power, mu, phi, lower.tail = TRUE, log.p = FALSE,
                        verbose = FALSE, details = FALSE, max.terms = 1e6)

ptweedie.series(q, power, mu, phi, verbose = FALSE, details = FALSE)
```

Arguments

q	vector of quantiles.
power	the power parameter p.
mu	the mean parameter μ.
phi	the dispersion parameter ϕ.
lower.tail	logical; if TRUE (the default), computes $F(y) = \Pr(Y \leq y)$; if FALSE, computes the upper tail $S(y) = \Pr(Y > y)$ directly, rather than as 1 - ptweedie_series(...).
log.p	logical; if TRUE, the logarithm of the probability is returned, computed directly via log-sum-exp accumulation over the series terms rather than as log(ptweedie_series(...)). This retains accuracy even when the probability itself would underflow to zero.
verbose	logical; if TRUE, displays some internal computation details. The default is FALSE.
details	logical; if TRUE, returns the value of the distribution function and some details.
max.terms	integer; the maximum number of terms summed for each value before the summing is truncated, and a warning produced. The default is 1e6, enough for λ up to about 4×10^9; each extra million terms takes about half a second per value.

Value

A numeric vector of probabilities (or log-probabilities, if log.p = TRUE).

Accuracy

The terms of the series are all positive, and are summed on the log scale, so both tails keep full relative accuracy, far into the tails. The only limitation is the number of terms: if more than `max.terms` are needed (when λ is very large), the series is truncated and a warning is given.

Note

The 'exact' values for the inverse Gaussian distribution are not really exact, but evaluated using inverse normal distributions, for which very good numerical approximation are available in R.

References

Dunn, Peter K and Smyth, Gordon K (Submitted). Evaluation of the Tweedie exponential dispersion model distribution function. Submitted.

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Examples

```
# Plot a Tweedie distribution function
y <- seq(0.01, 4, length = 50)
Fy <- ptweedie_series(y, power = 1.1, mu = 1, phi = 1)
plot(y, Fy, type = "l", lwd = 2, ylab = "Distribution function")
```

---

ptweedie_tail_integrate

Tweedie CDF evaluation using direct integration (in the tails only)

Description

Distribution function evaluation in the distribution tails using direct integration of the density function.

Usage

```
ptweedie_tail_integrate(q, mu, phi, power, lower.tail = TRUE, IGexact = TRUE,
                        fallback = TRUE)
```

Arguments

`q`	vector of quantiles.
`mu`	vector of mean μ.
`phi`	vector of dispersion parameters ϕ.
`power`	scalar; a synonym for ξ, the Tweedie index parameter. Here, $p > 2$.

lower.tail	logical; if TRUE (the default), computes the lower tail $F(y)$; otherwise computes $S(y) = 1 - F(y)$.
IGexact	logical; if TRUE (the default), evaluate the inverse Gaussian distribution using the 'exact' values, otherwise uses inversion.
fallback	logical; if]FALSE,only the Gauss-Laguerre rules are tried, which is cheap; the slower piecewise adaptive integration is skipped, and ok = FALSE returned if they fail. The default is TRUE

References

Dunn, Peter K and Smyth, Gordon K (Submitted). Evaluation of the Tweedie exponential dispersion model distribution function. Submitted.

Examples

```
ptweedie_tail_integrate(0.00001, power = 2.1, mu = 1, phi = 1)
```

Tweedie *Tweedie distributions*

Description

Density, distribution function, quantile function and random generation for the the Tweedie family of distributions, with mean mu, dispersion parameter phi and variance power power (or xi, a synonym for power).

Usage

```
dtweedie(y, xi = NULL, mu, phi, power = NULL, verbose = FALSE, log = FALSE)
ptweedie(q, xi = NULL, mu, phi, power = NULL, lower.tail = TRUE,
  log.p = FALSE, verbose = FALSE)
qtweedie(p, xi = NULL, mu, phi, power = NULL)
rtweedie(n, xi = NULL, mu, phi, power = NULL)
```

Arguments

y	vector of quantiles.
xi	scalar; the value of ξ such that the variance is var$[Y] = \phi\mu^{\xi}$. A synonym for power.
mu	vector of mean μ.
phi	vector of dispersion parameters ϕ.
power	scalar; a synonym for ξ, the Tweedie index parameter.
verbose	logical; if TRUE, some details of the algorithms used is shown. The default is FALSE.

log	logical; if TRUE, dtweedie() returns the log of the density, computed without underflow (so it stays finite far into the tails, where the density itself underflows to zero). The default is FALSE.
q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required (as for rnorm).
lower.tail	logical; if TRUE (the default), computes the lower tail $F(y)$; otherwise computes $S(y) = 1 - F(y)$.
log.p	logical; if TRUE, probabilities are returned as $\log(p)$. The default is FALSE.

Details

Tweedie EDMs belong to the class of exponential dispersion models (EDMs), known for their role in generalized linear models (GLMs). The Tweedie distributions are the EDMs with a variance of the form $\text{var}[Y] = \phi\mu^p$ where $p \geq 1$. *This function only evaluates for* $p \geq 1$.

Special cases are the Poisson ($p = 1$ with $\phi = 1$), gamma ($p = 2$), and inverse Gaussian ($p = 3$) distributions. Evaluation is difficult for p outside of $p = 0, 1, 2, 3$. For the density, dtweedie() uses one of three methods for each value, depending on p and $z = \phi y^{p-2}$ (the dispersion after rescaling y to one):

1. the saddlepoint approximation with its asymptotic correction, where z is small enough for its estimated relative error to be below 10^{-13};
2. otherwise, the series (dtweedie_series) for $1 < p < 2$, and for $p > 2$ where its estimated relative error is below 10^{-12};
3. otherwise, the Fourier inversion (dtweedie_inversion).

Against high-precision reference values, the relative error of the density is below about 4×10^{-13}. (Before version 3.2.0, dtweedie() also interpolated from stored values of the Fourier inversion, which was less accurate.)

When $1 < p < 2$, the density function include a positive probability for $Y = 0$.

Value

dtweedie() gives the density, ptweedie() gives the distribution function, qtweedie() gives the quantile function, and rtweedie generates random deviates.

As for R's own functions for distributions (such as dnorm), the result of dtweedie(), ptweedie() and qtweedie() has the same length and form as the first input (y, q or p): its attributes are kept, so (for example) a matrix gives a matrix, and a named vector gives a named vector. (mu and phi must be scalars, or the same length as the first input.) rtweedie() returns a vector of length n, without attributes.

Note

dtweedie() and ptweedie() are the only functions generally to be called by users. Consequently, all checks on the function inputs are only performed in these functions.

References

Dunn, Peter K and Smyth, Gordon K (Submitted). Evaluation of the Tweedie exponential dispersion model distribution function. Submitted.

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

See Also

dtweedie_series, dtweedie_inversion, ptweedie_series, ptweedie_inversion, dtweedie_saddle, tweedie_lambda

Examples

```
# Compute a Tweedie density
power <- 1.1
mu <- 1
phi <- 1
y <- seq(0, 5, by = 0.5)
dtweedie(y, power = power, mu = mu, phi = phi)

# Compare to the saddlepoint density
dtweedie_saddle(y = y, power = power, mu = mu, phi = phi)

# The DF:
ptweedie(y, power = power, mu = mu, phi = phi)
```

---

tweedie_AIC *AIC for Tweedie Glms*

Description

Evaluates the AIC for a fitted Tweedie GLM. The Tweedie family of distributions belong to the class of exponential dispersion models (EDMs), famous for their role in generalized linear models. The Tweedie distributions are the EDMs with a variance of the form $\text{var}[Y] = \phi\mu^p$ where $p \geq 1$. *This function only evaluates for* $p \geq 1$.

Usage

```
tweedie_AIC(glm.obj, dispersion = NULL, k = 2, verbose = TRUE)

AICtweedie(glm.obj, dispersion = NULL, k = 2, verbose = TRUE)
```

Arguments

glm.obj	a fitted glm object, fitted using the tweedie family.
dispersion	the dispersion parameter, usually extracted from glm.obj; however, occasionally a specified value of the dispersion may be needed.
k	the AIC penalty; k = 2 (the default) produces the AIC.
verbose	logical; if TRUE, display details of the internal process. The default is FALSE.

Details

The AIC is computed by evaluating the density function.

Value

The value of the computed AIC.

Note

Evaluating the likelihood can be time consuming, so the function may take some time for large data sets.

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Sakamoto, Y., Ishiguro, M., and Kitagawa G. (1986). *Akaike Information Criterion Statistics*. D. Reidel Publishing Company.

See Also

dtweedie

Examples

```
# Fit a Tweedie density using  tweedie  family function from  statmod
pretend <- data.frame( y = stats::rgamma(20, shape = 1, rate = 1) )
fit <- glm(y ~ 1, data = pretend,
           family = statmod::tweedie(link.power = 0, var.power = 2.1))

# Compute the AIC
tweedie_AIC(fit)
```

`tweedie_convert` *Tweedie Distribution: Convert Between Parameter Formats*

Description

For $1 < p < 2$, a Tweedie random variable is a Poisson sum of gamma random variables: $Y = \sum_{i=1}^{N} Z_i$, where $N \sim \text{Pois}(\lambda)$ and the Z_i are independent gamma variables with shape α and scale γ.

`tweedie_convert()` converts from the Tweedie (GLM) parameters p, μ and ϕ to the underlying Poisson parameter λ and gamma parameters α and γ.

`tweedie_convert_back()` converts the other way: from λ, α and γ to p, μ and ϕ.

Usage

```
tweedie_convert(xi = NULL, mu, phi, power = NULL)

tweedie_convert_back(lambda, alpha, gamma)

tweedie.convert(xi = NULL, mu, phi, power = NULL)
```

Arguments

`xi`	a synonym for `power`.
`mu`	the mean parameter μ.
`phi`	the dispersion parameter ϕ.
`power`	the power parameter p; a synonym for ξ.
`lambda`	the mean λ of the underlying Poisson distribution.
`alpha`	the shape parameter α of the underlying gamma distribution.
`gamma`	the scale parameter γ of the underlying gamma distribution (the reciprocal of the rate parameter).

Details

The relationships are

$$\lambda = \frac{\mu^{2-p}}{\phi(2-p)}, \quad \alpha = \frac{2-p}{p-1}, \quad \gamma = \phi(p-1)\mu^{p-1}$$

and, in reverse,

$$p = \frac{\alpha+2}{\alpha+1}, \quad \mu = \lambda\alpha\gamma, \quad \phi = \frac{\lambda\alpha(\alpha+1)\gamma^2}{\mu^p}.$$

Here γ is the gamma *scale* parameter; the corresponding *rate* parameter (sometimes written β) is $\beta = 1/\gamma$. In R's notation, the gamma variables are `rgamma(n, shape = alpha, scale = gamma)`.

Value

`tweedie_convert()` returns a list of the parameters of the corresponding underlying Poisson and gamma distributions: `poisson.lambda` (λ from the underlying Poisson distribution), `gamma.shape`, `gamma.scale` (the shape α and scale γ parameters from the underlying gamma distribution), `p0` (the probability that $Y = 0$), `gamma.mean` and `gamma.phi` (the gamma mean $\alpha\gamma$ and dispersion parameter $1/\alpha$ values).

`tweedie_convert_back()` returns a list containing `mu` (the Tweedie mean μ), `phi` (the Tweedie dispersion parameter ϕ), `power` (the Tweedie index parameter p), and `p0` (the probability that $Y = 0$).

Examples

```
### Fit a Tweedie density
pretend <- data.frame( y = rgamma(20, shape = 1, rate = 1) )
fit <- glm(y ~ 1, data = pretend,
           family = statmod::tweedie(link.power = 0, var.power = 1.4))

# Convert parameters
tweedie_convert(mu = fitted(fit, type="response"), phi = 1, power = 1.4)

### Convert, then convert back
pg <- tweedie_convert(mu = 2, phi = 0.5, power = 1.4)
tweedie_convert_back(lambda = pg$poisson.lambda,
                     alpha  = pg$gamma.shape,
                     gamma  = pg$gamma.scale)
```

---

tweedie_dev *Unit Deviance for a Tweedie Distribution*

Description

Computes the unit deviance for Tweedie distributions. Since version 3.2.0 it is computed without cancellation, so it has full relative accuracy even when `y` is very close to `mu`.

Usage

```
tweedie_dev(y, mu, power)

tweedie.dev(y, mu, power)
```

Arguments

`y`	vector of quantiles.
`mu`	the mean parameter μ.
`power`	the power parameter p.

Value

A numeric vector containing the unit deviance.

References

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Examples

```
# Unit deviance is not symmetric in general:
round( tweedie_dev(0:6, mu = 3, power = 1.1), 3)
```

`tweedie_integrand` *Display Integrand Information for Tweedie Fourier inversion*

Description

Plots the integrand for Fourier inversion and the real and imaginary parts separately.

Usage

```
tweedie_integrand(y, power, mu, phi, t = seq(0, 5, length = 200),
                  type = "PDF", whichPlots = 1:4,
                  plot_args  = list())
```

Arguments

`y`	vector of quantiles.
`power`	a synonym for ξ; the Tweedie power-index on the variance.
`mu`	the mean parameter μ.
`phi`	the dispersion parameter ϕ.
`t`	the values of the variable over which to integrate; the default is `t = seq(0, 5, length = 200)`.
`type`	either `"PDF"` (the default) for the (probability) density function, or `"CDF"` for the (cumulative) distribution function.
`whichPlots`	which combination of the four plots (described below) are produced; by default, all four are produced (i.e., `whichPlots = 1:4`).
`plot_args`	A named list of arguments controlling the main plot. These are passed to `plot`.

Details

The Tweedie family of distributions belong to the class of exponential dispersion models (EDMs), famous for their role in generalized linear models. The Tweedie distributions are the EDMs with a variance of the form $\text{var}[Y] = \phi\mu^p$ where p is greater than or equal to one, or less than or equal to zero.

This function only evaluates for p greater than or equal to one.

Special cases include the normal ($p = 0$), Poisson ($p = 1$ with $\phi = 1$), gamma ($p = 2$) and inverse Gaussian ($p = 3$) distributions. For other values of `power`, the distributions are still defined but cannot be written in closed form, and hence evaluation is very difficult.

When $1 < p < 2$, the distribution are continuous for Y greater than zero, with a positive mass at $Y = 0$. For $p > 2$, the distributions are continuous for Y greater than zero.

This function displays the integrand that is evaluated for computing the Fourier inversion, for the PDF or CDF.

Value

A list containing the real and imaginary parts of $k(t)$, `Real` and `Imag` respectively, plus the values of the integrand as `IG`. The main purpose of the function is the side-effect of producing a 2×2 grid of plots. The first is the imaginary parts of $k(t)$. The second is $\cos \Im k(t)$ (for `type=="PDF"`) or $\sin \Im k(t)$ (for `type=="CDF"`). The third is the real part of $\Re k(t)$ The fourth is the integrand, with the envelope shown as a dashed line.

Author(s)

Peter Dunn (<pdunn2@usc.edu.au>)

References

Dunn, P. K. and Smyth, G. K. (2008). Evaluation of Tweedie exponential dispersion model densities by Fourier inversion. *Statistics and Computing*, **18**, 73–86. doi:10.1007/s1122200790396

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Dunn, Peter K and Smyth, Gordon K (2001). Tweedie family densities: methods of evaluation. *Proceedings of the 16th International Workshop on Statistical Modelling*, Odense, Denmark, 2–6 July

Jorgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society*, B, **49**, 127–162.

Jorgensen, B. (1997). *Theory of Dispersion Models*. Chapman and Hall, London.

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

See Also

`dtweedie`

Examples

```
tweedie_integrand(2, power = 3, mu = 1, phi = 1,
                  plot_args = list(lwd = 2))
```

`tweedie_lambda` *The Probability of Observing a Zero Value for a Tweedie Density*

Description

The probability that the variable takes the value of zero.

Usage

```
tweedie_lambda(mu, phi, power)
```

Arguments

`mu`	the mean parameter μ.
`phi`	the dispersion parameter ϕ.
`power`	the power parameter p (sometimes denoted ξ).

Value

The value of λ when $1 < p < 2$ such that $P(Y = 0) = \exp(-\lambda)$. When $p > 2$, a vector of zeros is returned.

References

Dunn, Peter K and Smyth, Gordon K (2005). Series evaluation of Tweedie exponential dispersion model densities *Statistics and Computing*, **15**(4). 267–280. doi:10.1007/s112220054070y

Examples

```
lambda <- tweedie_lambda(mu = 1:3, phi = 1, power = 1.1)
exp( -lambda)

# When p > 2, there is zero probability that Y = 0:
lambda  <- tweedie_lambda(mu = 1, phi = 1, power = 3.1)
```

tweedie_plot *Plot Tweedie Models*

Description

This function produced a plot of the specified Tweedie distribution.

Usage

```
tweedie_plot(y, xi = NULL, mu, phi, type = "pdf", power = NULL,
                    add = FALSE,
                   plot_args = list(), point_args = list(), line_args = list())

tweedie.plot(
  y,
  xi = NULL,
  mu,
  phi,
  type = "pdf",
  power = NULL,
  add = FALSE,
  ...
)
```

Arguments

y	the values for y in the plot.
xi	a synonym for power.
mu	the mean of the distribution μ.
phi	the dispersion parameter ϕ.
type	the type of plot, either pdf (the default) or cdf.
power	the variance power p.
add	logical; if TRUE, the plot is added to the current plot; if FALSE (the default) the plot is produced on a fresh plot.
plot_args	A named list of arguments controlling the main plot. These are passed to plot.
point_args	A named list of graphical parameters for plotted points. These are passed to points.
line_args	A named list of graphical parameters for plotted lines. These are passed to lines.
...	Additional graphical arguments, passed to plot_args for backward compatibility. Deprecated; use plot_args in tweedie_plot instead.

Details

If $1 < p < 2$, the mass at $Y = 0$ is automatically added.

Examples

```
y <- seq(0, 4, length = 50)
tweedie_plot(y, power = 1.1, mu = 1, phi = 1,
             line_args = list(lwd = 2))
```

---

tweedie_profile *Profile Likelihood Estimate of Tweedie Variance Index Parameter*

Description

This function profiles the (log-)likelihood over a vector of Tweedie power-index parameter (denoted p or ξ) to find the maximum likelihood estimate (MLE) of the index parameter p (or equivalently ξ).

Usage

```
tweedie_profile(formula, p.vec = NULL, xi.vec = NULL, link.power = 0,
  data, weights = 1, offset = 0, fit.glm = FALSE, do.smooth = TRUE,
  do.plot = FALSE, do.ci = do.smooth, eps = 1/6,
  control = list( epsilon = 1e-09, maxit = stats::glm.control()$maxit,
  trace = glm.control()$trace ),
  do.points = do.plot, method = "inversion", conf.level = 0.95,
  phi.method = ifelse(method == "saddlepoint", "saddlepoint", "mle"),
  verbose = FALSE, add0 = FALSE,
  plot_args = list(), point_args = list(), line_args = list())

tweedie.profile(
  formula,
  p.vec = NULL,
  xi.vec = NULL,
  link.power = 0,
  data,
  weights = 1,
  offset = 0,
  fit.glm = FALSE,
  do.smooth = TRUE,
  do.plot = FALSE,
  do.ci = do.smooth,
  eps = 1/6,
  control = list(epsilon = 1e-09, maxit = stats::glm.control()$maxit, trace =
    glm.control()$trace),
  do.points = do.plot,
  method = "inversion",
  conf.level = 0.95,
  phi.method = ifelse(method == "saddlepoint", "saddlepoint", "mle"),
```

```
  verbose = FALSE,
  add0 = FALSE,
  plot_args = list(),
  point_args = list(),
  line_args = list()
)
```

Arguments

formula	a formula expression as for other regression models and generalized linear models, of the form response ~ predictors.
p.vec	a vector of p values for consideration. The values must all be larger than one. If the response has zeros, values must be $1 < p < 2$. If NULL (default), p.vec is set automatically.
xi.vec	a synonym for p.vec, as some authors use the ξ notation.
link.power	the power link function to use in the tweedie GLM family. These link functions $g(\cdot)$ are of the form $g(\eta) = \eta^{link.power}$, where link.power = 0 (default) refers to the logarithm link function.
data	an optional data frame, list or environment containing the variables.
weights	an optional vector of prior weights to be used in the fitting process. As for glm, observation i has dispersion ϕ/w_i; observations with zero weight are dropped.
offset	an *a priori* known component included in the linear predictor. See model.offset.
fit.glm	logical; if TRUE, the Tweedie GLM is fitted using the value of p found by the profiling function. The default is FALSE.
do.smooth	logical; if TRUE (default), a spline is fitted to the data to smooth the profile likelihood plot. **Note** that p.vec must contain *at least five points* for smoothing.
do.plot	logical; if TRUE, a plot of the profile likelihood is produced. The default is FALSE.
do.ci	logical; if TRUE, the nominal 100*conf.level is computed. Defaults to the value of do.smooth. Confidence intervals are only computed if do.smooth = TRUE.
eps	the offset in computing the variance function. Default is 1/6 (as recommended by Nelder and Pregibon, 1987). eps is ignored unless method = "saddlepoint".
control	a list of parameters for controlling the fitting process;
do.points	logical; if TRUE, the points used to compute the likelihood as given by p.vec (or equivalently, xi.vec) are explicitly shown by points. The defaults is the value of do.plot.
method	the method of evaluation; one of saddlepoint, interpolation, series or inversion (the default). When ϕ is found by maximum likelihood (phi.method = "mle"), the log-likelihood found along the way (using dtweedie) is used, so method only matters for method = "saddlepoint".
conf.level	the level of confidence for the confidence intervals; the default is 0.95 (for 95% confidence intervals).
phi.method	the method used to estimate ϕ; one of saddlepoint, mle (the default).

verbose	logical; if TRUE, some details of the calculations are shown. The default is FALSE.
add0	logical; if TRUE, adds $P(Y = 0)$ to the plot. The default is FALSE.
plot_args	A named list of arguments controlling the main plot. These are passed to plot.
point_args	A named list of graphical parameters for plotted points. These are passed to points.
line_args	A named list of graphical parameters for plotted lines. These are passed to lines.

Details

For each value in p.vec, the function computes an estimate of ϕ and then computes the value of the log-likelihood for these parameters. The plot of the log-likelihood against p.vec allows the maximum likelihood value of p to be found. Once p is found, the distribution within the class of Tweedie distributions is identified.

Note

The estimates of p and ϕ are printed invisibly. If the response variable has any exact zeros, the values in p.vec must all be between one and two (or perhaps zero).

The function can be temperamental (for theoretical reasons involved in numerically computing the density; see Dunn and Smyth (2005)) and may be very slow or fail. One solution is to change the method. The default is method = "inversion"; then try "series", "interpolation", and "saddlepoint" in that order. Note that method = "saddlepoint" is an approximate method only.

It is recommended that for the first use with a data set, use p.vec with only a small number of values and set do.smooth = FALSE, do.ci = FALSE. If this is successful, a larger vector p.vec and smoothing can be used.

References

Dunn, P. K. and Smyth, G. K. (2018). Generalized linear models with examples in R. Springer. doi:10.1007/9781441901187

Examples

```
data(rock)
# A quick profile, over a few values of p (no smoothing)
out <- tweedie_profile(perm ~ 1, data = rock, do.smooth = FALSE,
                       xi.vec = seq(1.8, 2.4, by = 0.2))
out$xi.max


# A finer profile, smoothed, with a confidence interval (slower)
out <- tweedie_profile(perm ~ 1, data = rock, do.plot = TRUE,
                       xi.vec = seq(1.5, 2.75, length = 11),
                       line_args = list(lwd = 2))

# The estimate for the variance power index (p, or xi), and its CI:
out$xi.max
```

```
out$ci
```

Index

∗ **distribution**
  dtweedie_inversion, 2
  dtweedie_saddle, 4
  dtweedie_series, 5
  ptweedie_inversion, 7
  ptweedie_saddle, 9
  ptweedie_series, 10
  ptweedie_tail_integrate, 11
  Tweedie, 12
  tweedie-package, 2
∗ **models**
  tweedie-package, 2
  tweedie_integrand, 18
  tweedie_profile, 22
∗ **regression**
  tweedie-package, 2

AICtweedie *(*tweedie_AIC*)*, 14

dnorm, *13*
dtweedie, *4*, *6*, *7*, *15*, *19*, *23*
dtweedie *(*Tweedie*)*, 12
dtweedie.inversion
        *(*dtweedie_inversion*)*, 2
dtweedie.saddle *(*dtweedie_saddle*)*, 4
dtweedie.series *(*dtweedie_series*)*, 5
dtweedie_inversion, 2, *14*
dtweedie_saddle, 4, *14*
dtweedie_series, 5, *14*

glm, *23*

lines, *21*, *24*
logLiktweedie, 6

model.offset, *23*

plot, *18*, *21*, *24*
points, *21*, *24*
ptweedie, *8*
ptweedie *(*Tweedie*)*, 12
ptweedie.inversion
        *(*ptweedie_inversion*)*, 7
ptweedie.series *(*ptweedie_series*)*, 10
ptweedie_inversion, 7, *14*
ptweedie_saddle, 9
ptweedie_series, 10, *14*
ptweedie_tail_integrate, 11

qtweedie *(*Tweedie*)*, 12

rnorm, *13*
rtweedie *(*Tweedie*)*, 12

Tweedie, 12
tweedie *(*tweedie-package*)*, 2
tweedie-package, 2
tweedie.convert *(*tweedie_convert*)*, 16
tweedie.dev *(*tweedie_dev*)*, 17
tweedie.plot *(*tweedie_plot*)*, 21
tweedie.profile *(*tweedie_profile*)*, 22
tweedie_AIC, 14
tweedie_convert, 16
tweedie_convert_back *(*tweedie_convert*)*,
        16
tweedie_dev, 17
tweedie_integrand, 18
tweedie_lambda, *14*, 20
tweedie_plot, *21*, 21
tweedie_profile, 22