

# Package ‘neftest’

July 19, 2021

**Type** Package

**Title** Goodness of fit tests based on zero regression characterizations of Tweedie, Bar-Lev and Enis class of distributions

**Version** 1.0.1

**Author** Author [aut],  
Author [aut],  
Author [aut],  
Author [aut,cre],  
Author [aut,cre]

**Maintainer** Xu Liu <liu.xu@sufe.edu.cn>

**Description** Provide test statistics for the goodness of fit tests based on zero regression characterizations of Tweedie, Bar-Lev and Enis class of distributions, and p-value of test statistic according to several specified distributions, which is based on the proposed method.

**License** GPL (>= 2)

**Depends** R (>= 3.2.0)

**LazyData** true

**NeedsCompilation** yes

**Repository** CRAN

**URL** <https://github.com/xliusufe/neftest>

**Encoding** UTF-8

## R topics documented:

neftest-package . . . . .	2
pvals . . . . .	2
rIGauss . . . . .	4
Tnw . . . . .	5
<b>Index</b>	<b>7</b>

---

neftest-package	<i>Goodness of fit tests based on zero regression characterizations of Tweedie, Bar-Lev and Enis class of distributions</i>
-----------------	---

---

### Description

Provide test statistics for the goodness of fit tests based on zero regression characterizations of Tweedie, Bar-Lev and Enis class of distributions, and p-value of test statistic according to several specified distributions, which is based on the proposed method.

### Details

Package: neftest  
 Type: Package  
 Version: 1.0.1  
 Date: 2021-07-18  
 License: GPL (>= 2)

### References

Authors (2021). Goodness of fit tests based on zero regression characterizations of Tweedie, Bar-Lev and Enis class of distributions. Manuscript.

---

pvals	<i>The p-value of the test based on the test statistic <math>T_{nw}</math></i>
-------	--

---

### Description

Compute the p-value of the test based on the test statistic  $T_{nw}$ .

### Usage

```
pvals(x, distr="Poisson", bootstrap = FALSE, B = 1000,
      weight="normal", a = 1.0, max.iter = 100, tol = 1e-8)
```

### Arguments

x	A length $n$ vector of input data.
distr	The true distribution including Poisson distribution <code>distr = "Poisson"</code> , Gamma distribution <code>distr = "Gamma"</code> and Inverse Gaussian distribution <code>distr = "Inverse Gaussian"</code> . Default is <code>distr = "Poisson"</code> .
bootstrap	logical. Bootstrap method is used to compute the p-value if FALSE (default), and the maximum likelihood method otherwise.
B	Number of bootstrap samples. Default is 1000.

weight	The weight functions including normal distribution weight = "normal" and Laplace distribution weight = "laplace". Default is weight = "normal".
a	The parameter of the weight function. Default is 1.0. See details in the paper
max.iter	The maximum number of iterations in Newton method. Default is 100.
tol	The precision of the Newton method. Default is 1e-8.

**Value**

pval	The p-value of the test.
------	--------------------------

**References**

Authors (2021). Goodness of fit tests based on zero regression characterizations of Tweedie, Bar-Lev and Enis class of distributions. Manuscript.

**Examples**

```
# Poisson
n <- 100
distr <- "Poisson"
x <- rpois(n, lambda = 1)
pval = pvals(x, distr)
```

```
pval
```

```
# Poisson with Bootstrap
n <- 100
B <- 500
distr <- "Poisson"
x <- rpois(n, lambda = 1)
pval = pvals(x, distr, bootstrap = TRUE, B = B)
```

```
pval
```

```
# Gamma
n <- 100
distr <- "Gamma"
x <- rgamma(n, shape = 1, rate = 1)
pval = pvals(x, distr)
```

```
pval
```

```
# Gamma with Bootstrap
n <- 100
B <- 500
distr <- "Gamma"
x <- rgamma(n, shape = 1, rate = 1)
pval = pvals(x, distr, bootstrap = TRUE, B = B)
```

```
pval
```

```
# Inverse Gaussian
n    <- 100
distr <- "Inverse Gaussian"
x    <- rIGauss(n, mu = 1, lambda = 1)
pval = pvals(x, distr)

pval

# Inverse Gaussian with Bootstrap
n    <- 100
B    <- 500
distr <- "Inverse Gaussian"
x    <- rIGauss(n, mu = 1, lambda = 1)
pval = pvals(x, distr, bootstrap = TRUE, B = B)

pval
```

---

rIGauss

---

Generates random numbers from Inverse Gaussian distribution.

---

## Description

Generates random numbers from Inverse Gaussian distribution.

## Usage

```
rIGauss(n, mu = 1.0, lambda = 1.0)
```

## Arguments

n	Number of random numbers to be generated.
mu	Shape parameter $\mu$ . Default is 1.
lambda	Scale parameter $\lambda$ . Default is 1.

## Details

The probability density function:

$$f(x; \mu, \lambda) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right), x > 0.$$

## Value

x	The random numbers from Inverse Gaussian distribution.
---	--

## Examples

```
x = rIGauss(n = 10, mu = 1.0, lambda = 1.0)
x
```

---

Tnw	<i>The test statistic for testing if a distribution is a TBE(<math>\gamma_0</math>)</i>
-----	---

---

## Description

Compute the test statistic  $T_{nw}$ .

## Usage

```
Tnw(x, gamma0 = 1, weight = "normal", a = 1.0)
```

## Arguments

x	A length $n$ vector of input data.
gamma0	The power parameter in TBE( $\gamma_0$ ). Default is 1.
weight	The weight functions including normal distribution weight = "normal" and Laplace distribution weight = "laplace" . Default is weight = "normal".
a	The parameter of the weight function. Default is 1.0.

## Value

Tn                      The test statistic  $T_{nw}$ .

## References

Authors (2021). Goodness of fit tests based on zero regression characterizations of Tweedie, Bar-Lev and Enis class of distributions. Manuscript.

## Examples

```
# Poisson
n <- 100
x <- rpois(n, lambda = 1)
B <- 1000
Tn <- Tnw(x, gamma0 = 1, weight = "normal", a = 1.0)
Tb = rep(NA, B)
lambdahat = mean(x)
for(b in 1:B){
  xb <- rpois(n, lambda = lambdahat)
  Tb[b] <- Tnw(xb, gamma0 = 1, weight = "normal", a = 1.0)
}
pval = mean(Tb > Tn)

# Gamma
n <- 100
x <- rgamma(n, shape = 1, rate = 1)
Tn <- Tnw(x, gamma0 = 1, weight = "normal", a = 1.0)

# Inverse Gaussian
n <- 100
x <- rIGauss(n, mu = 1, lambda = 1)
B <- 1000
```

```
Tn <- Tnw(x, gamma0 = 1, weight = "normal", a = 1.0)
Tb = rep(NA, B)
nuhat = mean(x)
lambdahat = (mean(1/x)-nuhat^(-1))^(-1)
for(b in 1:B){
  xb <- rIGauss(n, mu = nuhat, lambda = lambdahat)
  Tb[b] <- Tnw(xb, gamma0 = 1, weight = "normal", a = 1.0)
}
pval = mean(Tb > Tn)
```

# Index

## \* **package**

neftest-package, [2](#)

neftest (neftest-package), [2](#)

neftest-package, [2](#)

pvals, [2](#)

rIGauss, [4](#)

Tnw, [5](#)