

# Package: vsgoftest (via r-universe)

September 7, 2024

**Type** Package

**Title** Goodness-of-Fit Tests Based on Kullback-Leibler Divergence

**Version** 1.0-1

**Date** 2020-12-17

**Author** Justine Lequesne [aut], Philippe Regnault [aut, cre]

**Maintainer** Philippe Regnault <philipperegnault@hotmail.com>

**Description** An implementation of Vasicek and Song goodness-of-fit tests. Several functions are provided to estimate differential Shannon entropy, i.e., estimate Shannon entropy of real random variables with density, and test the goodness-of-fit of some family of distributions, including uniform, Gaussian, log-normal, exponential, gamma, Weibull, Pareto, Fisher, Laplace and beta distributions; see Lequesne and Regnault (2020) <[doi:10.18637/jss.v096.c01](https://doi.org/10.18637/jss.v096.c01)>.

**Depends** stats, fitdistrplus

**Imports** Rcpp (>= 0.12.1)

**Suggests** knitr

**VignetteBuilder** knitr

**LinkingTo** Rcpp

**Encoding** UTF-8

**License** GPL (>= 2)

**NeedsCompilation** yes

**Date/Publication** 2020-12-17 16:30:02 UTC

**Repository** <https://testpregnault.r-universe.dev>

**RemoteUrl** <https://github.com/cran/vsgoftest>

**RemoteRef** HEAD

**RemoteSha** 4d091f66be09886f437f673a47f238ec9a5e6a39

## Contents

vsgoftest-package . . . . .	2
contaminants . . . . .	4
dlaplace . . . . .	4
dpareto . . . . .	5
entropy.estimate . . . . .	7
vs.test . . . . .	8

<b>Index</b>	<b>11</b>
--------------	-----------

---

vsgoftest-package	<i>Goodness-of-Fit Tests Based on Kullback-Leibler Divergence</i>
-------------------	---

---

### Description

An implementation of Vasicek and Song goodness-of-fit tests. Several functions are provided to estimate differential Shannon entropy, i.e., estimate Shannon entropy of real random variables with density, and test the goodness-of-fit of some family of distributions, including uniform, Gaussian, log-normal, exponential, gamma, Weibull, Pareto, Fisher, Laplace and beta distributions; see Lequesne and Regnault (2020) <doi:10.18637/jss.v096.c01>.

### Details

The DESCRIPTION file:

```

Package:      vsgoftest
Type:         Package
Title:        Goodness-of-Fit Tests Based on Kullback-Leibler Divergence
Version:      1.0-1
Date:         2020-12-17
Author:       Justine Lequesne [aut], Philippe Regnault [aut, cre]
Maintainer:  Philippe Regnault <philipperegnault@hotmail.com>
Description:  An implementation of Vasicek and Song goodness-of-fit tests. Several functions are provided to estimate d
Depends:      stats, fitdistrplus
Imports:      Rcpp (>= 0.12.1)
Suggests:    knitr
VignetteBuilder: knitr
LinkingTo:    Rcpp
Encoding:     UTF-8
License:      GPL (>=2)
Packaged:    2020-12-17 12:12:51 UTC; philippe

```

Index of help topics:

contaminants	Organic and inorganic contaminant concentration data
dlaplace	The Laplace distribution
dpareto	The Pareto distribution
entropy.estimate	Vasicek estimate of differential Shannon Entropy
vs.test	Vasicek-Song goodness-of-fit test for various distributions
vsgoftest-package	Goodness-of-Fit Tests Based on Kullback-Leibler Divergence

Further information is available in the following vignettes:

vsgoftest\_tutorial    Tutorial (source, pdf)

### Author(s)

Justine Lequesne [aut], Philippe Regnault [aut, cre]

Maintainer: Philippe Regnault <philipperegnault@hotmail.com>

### References

Vasicek, O., A test for normality based on sample entropy, *Journal of the Royal Statistical Society*, **38(1)**, 54-59 (1976).

Song, K. S., Goodness-of-fit tests based on Kullback-Leibler discrimination information, *Information Theory, IEEE Transactions on*, **48(5)**, 1103-1117 (2002).

Girardin, V., Lequesne, J. Entropy-based goodness-of-fit tests - a unifying framework. Application to DNA replication. *Communications in Statistics: Theory and Methods* (2017). <https://doi.org/10.1080/03610926.2017.1401>

Lequesne, J., Regnault, P. vsgoftest: An R Package for Goodness-of-Fit Testing Based on Kullback-Leibler Divergence. *Journal of Statistical Software*, **96** (2020). doi:10.18637/jss.v096.c01

### Examples

```
set.seed(1)
samp <- rnorm(50, mean = 2, s = 3)

##Estimating entropy
entropy.estimate(x = samp, window = 8)
log(2*pi*exp(1))/2 #true value of entropy of normal distribution

##Testing normality
vs.test(x = samp, densfun = 'dnorm', param = c(2,3), B = 500) #Simple null hypothesis
vs.test(x = samp, densfun='dnorm', B = 500) #Composite null hypothesis
```

---

 contaminants

*Organic and inorganic contaminant concentration data*


---

### Description

Organic and inorganic contaminant concentration data from Superfund sites; see Singh *et al.* (1997).

### Usage

```
data(contaminants)
```

### Format

Four numeric vectors of respective lengths 17, 17, 23 and 23.

### Details

aluminium1 and manganese are groundwater concentration measurements of aluminium and manganese from seventeen wells at the Naval Construction Battalion Center Superfound Site in Rhode Island.

aluminium2 and toluene are concentration measurements of aluminium and toluene compiled from two waste piles at Elmara School Superfound site in Washington County, PA.

### Source

Singh, A K., Singh, A., Engelhardt, M. The lognormal distribution in environmental applications, Technology Support Center Issue Paper, US EPA (1997).

---

 dlaplace

*The Laplace distribution*


---

### Description

Density, cumulative distribution function, quantile function and random generation for the laplace distribution.

### Usage

```
dlaplace(x, mu, b, log = FALSE)
plaplace(q, mu, b, lower.tail = TRUE, log.p = FALSE)
qlaplace(p, mu, b, lower.tail = TRUE, log.p = FALSE)
rlaplace(n, mu, b)
```

**Arguments**

x, q	(numeric, vector) a vector of quantiles.
p	(numeric, vector) a vector of probabilities.
n	(numeric, vector) sample size to be generated.
mu	(numeric, single value) the location parameter.
b	(numeric, single value) the scale parameter.
log, log.p	(logical, single value) if TRUE, probabilities are given as $\log(p)$ . Default is FALSE.
lower.tail	(logical, single value) if TRUE (default), probabilities are $P(X \leq x)$ ; otherwise $P(X > x)$ .

**Details**

The laplace distribution with shape parameter  $\mu > 0$  and scale parameter  $b > 0$  has density

$$f(x) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right), \quad x \in R.$$

**Value**

dlaplace gives the density, plaplace gives the distribution function, qlaplace gives the quantile function, and rlaplace generates random deviates.

The length of the result is determined by n for rnorm, and is the maximum of the lengths of the numerical arguments for the other functions.

**Author(s)**

J. Lequesne <justine.lequesne@unicaen.fr>

**Examples**

```
set.seed(1)
rlaplace(100, mu=2, b=1)
```

---

dpareto

*The Pareto distribution*


---

**Description**

Density, cumulative distribution function, quantile function and random generation for the Pareto distribution.

**Usage**

```
dpareto(x, mu, c, log = FALSE)
ppareto(q, mu, c, lower.tail = TRUE, log.p = FALSE)
qpareto(p, mu, c, lower.tail = TRUE, log.p = FALSE)
rpareto(n, mu, c)
```

**Arguments**

x, q	(numeric, vector) a vector of quantiles.
p	(numeric, vector) a vector of probabilities.
n	(numeric, vector) sample size to be generated.
mu	(numeric, single value) the shape parameter.
c	(numeric, single value) the scale parameter.
log, log.p	(logical, single value) if TRUE, probabilities are given as $\log(p)$ . Default is FALSE.
lower.tail	(logical, single value) if TRUE (default), probabilities are $P(X \leq x)$ ; otherwise $P(X > x)$ .

**Details**

The pareto distribution with shape parameter  $\mu > 0$  and scale parameter  $c > 0$  has density

$$f(x) = \mu c^\mu x^{-1-\mu},$$

for  $x \geq 0$ .

**Value**

dpareto gives the density, ppareto gives the distribution function, qpareto gives the quantile function, and rpareto generates random deviates.

The length of the result is determined by n for rnorm, and is the maximum of the lengths of the numerical arguments for the other functions.

**Author(s)**

J. Lequesne <justine.lequesne@unicaen.fr>

**References**

- Arnold, B.C. Pareto distribution, *International Cooperative Publishing House, Fairland* (1983).  
 Philbrick, S.W. A practical guide to the single parameter Pareto distribution. *Proceedings of the Casualty Actuarial Society LXXII*, **44**, 44-85 (1985).

**Examples**

```
n<- 100
rpareto(n,mu=2,c=1)
```

---

entropy.estimate      *Vasicek estimate of differential Shannon Entropy*

---

**Description**

Computes Vasicek estimate of differential Shannon entropy from a numeric sample.

**Usage**

```
entropy.estimate(x,window)
```

**Arguments**

**x** (numeric, vector) the numeric sample.  
**window** (numeric, single value) an integer between 1 and half on the sample size specifying the window size for computing Vasicek estimate. See Details for additional information.

**Details**

Vasicek estimator of Shannon entropy is defined, for a random sample  $X_1, \dots, X_n$ , by

$$\frac{1}{n} \sum_{i=1}^n \log\left(\frac{n}{2m} [X_{(i+m)} - X_{(i-m)}]\right),$$

where  $X_{(i)}$  is the order statistic,  $m < (n/2)$  is the window size, and  $X_{(i)} = X_{(1)}$  for  $i < 1$  and  $X_{(i)} = X_{(n)}$  for  $i > n$ .

**Value**

A single numeric value representing the Vasicek estimate of entropy of the sample

**Author(s)**

J. Lequesne <justine.lequesne@unicaen.fr>

**References**

Vasicek, O., A test for normality based on sample entropy, *Journal of the Royal Statistical Society*, **38(1)**, 54-59 (1976).

**See Also**

[vs.test](#) which performs Vasicek-Song goodness-of-fit tests to the specified maximum entropy distribution family.

**Examples**

```
set.seed(2)
samp <- rnorm(100, mean = 0, s = 1)
entropy.estimate(x = samp, window = 8)
log(2*pi*exp(1))/2 #true value of entropy of normal distribution
```

vs.test

*Vasicek-Song goodness-of-fit test for various distributions***Description**

Performs Vasicek-Song goodness-of-fit test to the specified distribution family.

**Usage**

```
vs.test(x, densfun, param = NULL,
        simulate.p.value = NULL, B = 5000,
        delta = NULL, extend = FALSE, relax = FALSE)
```

**Arguments**

x	(numeric, vector) the numeric sample.
densfun	A character string specifying the fitted distribution. Possible values are "dunif", "dnorm", "dlnorm", "dexp", "dgamma", "dweibull", "dpareto", "df", "dlaplace" and "dbeta".
param	(numeric, vector) specifies the parameter(s) of the fitted distribution. If NULL (default), a GOF test to the parametric family of distributions specified by densfun is performed.
simulate.p.value	(logical, single value) if TRUE, the p-value of the sample is estimated by means of Monte Carlo methods. If NULL (the default), the p-value is simulated if the sample size is smaller than 80; otherwise, an asymptotic p-value is computed.
B	(numeric, single value) a numeric value specifying the number of simulations to perform in Monte-Carlo estimation of the p-value.
delta	(numeric, single value) a numeric value smaller than 1/3 specifying the upper bound $n^{1/3} - \delta$ for window size, where $n$ is the sample size. The default depends on densfun; see Vignettes for details.
extend	(logical, single value). If FALSE (the default), the bound for the window is $n^{1/3} - \delta$ ; if TRUE, the bound is $n/2$ .
relax	(logical, single value) avoids the constraint $V_{mn} \leq -\frac{1}{n} \sum_{i=1}^n \log p_0(X_i, \hat{\theta}_n)$ when computing the optimal window; see details. Default is FALSE.



## Details

The test statistic is

$$I_{mn} = -V_{mn} - \frac{1}{n} \sum_{i=1}^n \log p_0(X_i, \theta),$$

where  $V_{mn}$  is the Vasicek estimator of Shannon entropy computed from the numeric sample  $x$  with window size  $m$  and  $p_0(x, \theta)$  is the density function of the specified distribution `densfun` to be tested, with  $\theta$  the parameter of the null for a simple hypothesis or its maximum likelihood estimate for a composite null hypothesis (`param=NULL`); See Song (2002), Girardin and Lequesne (2017) and Lequesne and Regnault (2018).

An optimal window size  $m$  is automatically computed; see Song (2002).

An exact p-value is computed if the sample size is less than 100. Otherwise, asymptotic distribution is used whose approximation may be inaccurate for small samples; see Lequesne and Regnault (2018).

## Value

A list with class "htest" containing the following components:

<code>observed</code>	The sample under study.
<code>data.name</code>	The name (as an R object) of the sample.
<code>null.value</code>	A character string specifying the name of the fitted distribution.
<code>method</code>	The character string "Vasicek GOF test to" followed by the name of the fitted distribution.
<code>statistic</code>	Vasicek test statistic; see Details below.
<code>parameter</code>	The optimal window for Vasicek test statistic
<code>estimate</code>	Parameter(s) of the fitted distribution. If <code>param</code> is <code>NULL</code> , parameters are estimated. If <code>param</code> is suitably filled out by the user, it is returned.
<code>p.value</code>	The p-value of the test.

## Author(s)

J. Lequesne <justine.lequesne@unicaen.fr>

## References

Vasicek, O., A test for normality based on sample entropy, *Journal of the Royal Statistical Society*, **38(1)**, 54-59 (1976).

Song, K. S., Goodness-of-fit tests based on Kullback-Leibler discrimination information, *Information Theory, IEEE Transactions on*, **48(5)**, 1103-1117 (2002).

Girardin, V., Lequesne, J. Entropy-based goodness-of-fit tests - a unifying framework. Application to DNA replication. *Communications in Statistics: Theory and Methods* (2017). <https://doi.org/10.1080/03610926.2017.1401>

Lequesne, J., Regnault, P. vsqoftest: An R Package for Goodness-of-Fit Testing Based on Kullback-Leibler Divergence. *Journal of Statistical Software*, **96** (2020). doi:10.18637/jss.v096.c01

**See Also**

[entropy.estimate](#) which computes the Vasicek estimator of Shannon entropy.

**Examples**

```
set.seed(1)
samp <- rnorm(50,2,3)
vs.test(x = samp, densfun = 'dnorm', param = c(2,3), B = 500) #Simple null hypothesis
vs.test(x = samp, densfun='dnorm', B = 500) #Composite null hypothesis
## Using asymptotic distribution to compute the p-value
vs.test(x = samp, densfun='dnorm', simulate.p.value = FALSE) #Composite null hypothesis
```

# Index

- \* **Differential Shannon entropy**
  - entropy.estimate, 7
- \* **datasets**
  - contaminants, 4
- \* **distribution**
  - dlaplace, 4
  - dpareto, 5
- \* **htest**
  - vs. test, 8
- \* **package**
  - vsgof test-package, 2

aluminium1 (contaminants), 4  
aluminium2 (contaminants), 4

contaminants, 4

dlaplace, 4  
dpareto, 5

entropy.estimate, 7, 10

manganese (contaminants), 4

plaplace (dlaplace), 4  
ppareto (dpareto), 5

qlaplace (dlaplace), 4  
qpareto (dpareto), 5

rlaplace (dlaplace), 4  
rpareto (dpareto), 5

toluene (contaminants), 4

vs. test, 7, 8  
vsgof test (vsgof test-package), 2  
vsgof test-package, 2