# Package **vsgoftest** for R: goodness-of-fit tests based on Kullback-Leibler divergence

Justine Lequesne*and Philippe Regnault†

January 29, 2018

*Service de Recherche Clinique, Centre Henri Becquerel, 76038 Rouen Cedex1, France. E-mail: justine.lequesne@chb.unicancer.fr
†Laboratoire de Mathématiques de Reims, FRE 2011, Université de Reims Champagne-Ardenne, BP 1039, 51687 Reims cedex 2, France. E-mail: philippe.regnault@univ-reims.fr

The package **vsgoftest** provides functions to estimate Shannon entropy of continuous random variables and to test the goodness-of-fit of a vector of real numbers to some prescribed family of distributions. As a by-product, it also provides functions to compute the density, cumulative density and quantile functions of Pareto and Laplace distributions, as well as to generate samples from Pareto and Laplace distributions.

The latest (under development) version of the package **vsgoftest** is available and can be installed in R from the github repository of the project as follows:

```r
#Package devtools must be installed
devtools::install_github('pregnault/vsgoftest', dependencies = TRUE)
```

The package is structured around two functions, `entropy.estimate` and `vs.test`. While the first one aims at computing Vasicek estimator of the differential entropy $\mathbb{S}(P) = -\int p(x) \log p(x) \mathrm{d}x$ of a distribution $P$ on $\mathbb{R}$ with density $p$ from a numeric sample $X_1, \ldots, X_n$ drawn from $P$, the second one performs Vasicek-Song Goodness-of-fit test to usual parametric families of distributions. A comprehensive presentation of their usage is proposed in Sections 1 and 2, with numerous examples. An application to environmental data is presented in Section 3.

Details about entropy estimation, Vasicek-Song goodness-of-fit tests and the contents and features of the package are available in the listed references at the end of the present document. Particularly, see [1] and [2].

## 1 Function entropy.estimate for estimating differential entropy

The function `entropy.estimate` computes the spacing based Vasicek estimator

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

of the differential Shannon entropy of a numeric sample $X_1, \ldots, X_n$, with $X_{(1)} \leq \cdots \leq X_{(n)}$ denoting the order statistics of the sample and $m \in \mathbb{N}^*$ being a window size smaller than $n/2$. Two arguments must be provided:

- `x`: the numeric sample;
- `window`: an integer between 1 and half of the sample size, specifying the window size of the spacing-based estimator (1).

It returns a single value which is the estimate of Shannon differential entropy of the sample. Right below is an example for a sample drawn from a normal distribution with parameters $\mu = 0$ and $\sigma^2 = 1$.

```
library('vsgoftest')
set.seed(2)      #set seed of PRNG
samp <- rnorm(n = 100, mean = 0, sd = 1) #sampling from normal distribution
entropy.estimate(x = samp, window = 8) #estimating entropy with window = 8

[1] 1.394728

log(2*pi*exp(1))/2 #the true value of entropy

[1] 1.418939
```

One may wonder about the choice of the window size. Following the maximum entropy paradigm, one may choose the window size that maximizes the entropy estimate, as described by the following instructions.

```
n <- 100 #sample size
V <- numeric(n/2 -1)
for (i in 1:(n/2 -1)) { #make vary window size from 1 to 49
  #compute estimate for each window
  V[i] <- entropy.estimate(x = samp, window = i)
}
which.max(V) #Choose window that maximizes entropy

[1] 8
```

Consider as a second example, a sample drawn from a Pareto distribution whose density is

$$p(x; c, \mu) = \frac{\mu c^{\mu}}{x^{\mu+1}}, \quad x \geq c,$$

where $c > 0$ and $\mu > 0$. The closed form expression of its Shannon entropy is

$$\mathbb{S}(p(.; c, \mu)) = -\ln \mu + \ln c + \frac{1}{\mu} + 1.$$

Such a sample can be obtained by making use of the function `rpareto`, as illustrated by the following instructions.

```
set.seed(5)
n <- 100 #Sample size
samp <- rpareto(n, c = 1, mu = 2) #sampling from Pareto distribution
entropy.estimate(x = samp, window = 3)

[1] 0.8480204

-log(2) + 3/2 #True value of entropy

[1] 0.8068528
```

## 2 Function vs.test for testing GOF to a specified model

The function `vs.test` performs the Vasicek-Song procedure for testing goodness-of-fit of a numerical sample to whether a prescribed distribution $P = P_0(\theta)$, the so-called simple null hypothesis test

$$H_0 : P = P_0(\theta) \quad \text{against} \quad H_1 : P \neq P_0(\theta), \tag{2}$$

or to a parametric family $\mathcal{P}_0(\Theta)$, the so-called composite null hypothesis test

$$H_0 : P \in \mathcal{P}_0(\Theta) \quad \text{against} \quad H_1 : P \notin \mathcal{P}_0(\Theta); \tag{3}$$

see Appendix below for details. In its shortest call, it requires two arguments to be provided:

- `x`: the numeric sample;
- `densfun`: a character string specifying the targeted family of distributions. Available families of distributions are: uniform, normal, log-normal, exponential, gamma, Weibull, Pareto, Fisher and Laplace distributions. They are referred to by the symbolic name in R of their density function. For example, set `densfun = 'dnorm'` to test GOF to the family of normal distributions.

It returns an object of class `htest`, i.e., a list whose main components are:

- `statistic`: the value of the VS test statistic (12) of the sample, for the optimal window size, as defined in (11);
- `parameter`: the optimal window size;
- `estimate`: the maximum likelihood estimate of the parameters of the distribution to which the GOF is tested;
- `p.value`: the p-value associated to the sample.

By default, once provided the arguments `x` and `densfun`, the function `vs.test` performs the composite GOF VS test to the prescribed family `densfun`. Depending on the sample size, the p-value is estimated by means of Monte-Carlo methods (if the sample size is smaller than 80), or through the asymptotic distribution (7) of the VS test statistic.

In the following example, a normally distributed sample is simulated. Its goodness-of-fit to the family of Laplace distributions is rejected while its normality is accepted.

```
set.seed(2)
samp <- rnorm(50,2,3)
vs.test(x = samp, densfun = 'dlaplace')


Vasicek-Song GOF test to Laplace distribution family

data:  samp
Test statistic = 0.26145, Optimal window = 2, p-value = 0.1274
sample estimates:
   Shape    Scale
2.207414 2.682950
```

An additional argument can be provided to perform a simple null hypothesis test. By setting `param` to a suitable numeric vector (adapted to the family of distribution given in `densfun`), the function `vs.test` performs a GOF test to the single prescribed distribution.

```
set.seed(26)
vs.test(x = samp, densfun = 'dnorm', param = c(2,3))


Vasicek-Song GOF test to normal distribution with Mean=2, St.
dev.=3

data:  samp
Test statistic = 0.22666, Optimal window = 2, p-value = 0.2992
```

Note that when `param` is provided, the MLE of the parameter(s) of the null distribution is not computed, hence the component `estimate` of the result is not available.

One may prefer estimating the p-value of the sample by Monte-Carlo simulations, even when sample size is larger than 80. The optional argument `simulate.p.value` has then to be turned to `TRUE` (`NULL` by default). Note that, it is also possible to choose the number of Monte-Carlo replicates by providing to the optional argument `B` a positive integer (default is `B = 5000`).

```
set.seed(1)
samp <- rweibull(200, shape = 1.05, scale = 1)
vs.test(samp, densfun = 'dexp')


Vasicek-Song GOF test to exponential distribution family

data:  samp
Test statistic = 0.10907, Optimal window = 3, p-value = 0.3461
sample estimates:
   Rate
1.15047
```

```
set.seed(2)
vs.test(samp, densfun = 'dexp', simulate.p.value = TRUE, B = 10000)


Vasicek-Song GOF test to exponential distribution family

data:  samp
Test statistic = 0.10907, Optimal window = 3, p-value = 0.3504
sample estimates:
   Rate
1.15047
```

Whether `simulate.p.value` is turned to `TRUE` or not, Vasicek estimates $V_{mn}$ are computed for all $m$ from 1 to $n^{1/3-\delta}$, where $\delta < 1/3$; hence the test statistic is $I_{\widehat{m}n}$, given by (12) for $\widehat{m}$ the optimal window size, as defined in (11). The choice of $\delta$ depends on the family the goodness-of-fit is tested to. Precisely, for Weibull, Pareto, Fisher, Laplace and Beta families, $\delta$ is set to $2/15$ while for uniform, normal, log-normal, exponential and gamma families, it is set to $1/12$. These default settings result from numerous experimentations. Still, if needed, the user can tune this parameter by providing a numeric value to the optional argument `delta`.

```
vs.test(samp, densfun = 'dexp', delta = 5/30)


Vasicek-Song GOF test to exponential distribution family

data:  samp
Test statistic = 0.16517, Optimal window = 2, p-value = 0.1538
sample estimates:
   Rate
1.15047
```

In addition, when estimating the p-value by means of Monte-Carlo simulations, upper-bounding the window size by $n^{1/3-\delta}$ is not necessary (it is a necessary requirement only for using the asymptotic normality of $I_{mn}$ and hence for computing asymptotic p-values from (7)). This upper-bound can be relaxed so that $m$ ranges from 1 to $n/2$ by adding `extend = TRUE`. The interests are multiple. First, enlarging the range for $m$ may lead to a most reliable test, as illustrated below.

```
set.seed(8)
samp <- rexp(30, rate = 3)
vs.test(x = samp, densfun = "dlnorm")


Vasicek-Song GOF test to log-normal distribution family

data:  samp
Test statistic = 0.30717, Optimal window = 2, p-value = 0.1206
sample estimates:
 Location     Scale
-2.162290  1.683868
```

```
vs.test(x = samp, densfun = "dlnorm", extend = TRUE)


Vasicek-Song GOF test to log-normal distribution family

data:  samp
Test statistic = 0.3029, Optimal window = 3, p-value = 0.007
sample estimates:
 Location     Scale
-2.162290  1.683868
```

Second, enlarging the range for $m$ is also interesting when the sample size is moderate, especially if ties are present. Indeed, the presence of ties is particularly non-appropriate for performing VS tests: if ties are present, it may happen that some spacings $X_{(i+m)} - X_{(i-m)}$ vanish and hence Vasicek estimate is not well-defined. From a computational view point, it requires the window size $m$ to be greater than the maximal number of ties in the sample. Hence, if the upper-bound $n^{1/3-\delta}$ is less than the maximal number of ties, the test statistic can not be computed. Turning `extend` to `TRUE` may avoid this behaviour, as illustrated below.

```
samp <- c(samp, rep(4,3)) #add ties in the previous sample
vs.test(x = samp, densfun = "dexp")

Warning in vs.estimate(x, densfun, ESTIM, extend, delta, relax):  Ties should not be
present for Vasicek-Song test
Error in vs.estimate(x, densfun, ESTIM, extend, delta, relax):  Too many ties to compute
Vasicek estimate.
```

```
vs.test(x = samp, densfun = "dexp", extend = TRUE)

Warning in vs.estimate(x, densfun, ESTIM, extend, delta, relax):  Ties should not be
present for Vasicek-Song test


Vasicek-Song GOF test to exponential distribution family

data:  samp
Test statistic = 0.025702, Optimal window = 16, p-value = 0.9052
sample estimates:
    Rate
1.683785
```

Finally, it may happen that for all $m$ between 1 and $n^{1/3-\delta}$, Vasicek's estimate $V_{mn}$ exceeds the parametric estimate of the entropy of the null distribution, hence no window size exists satisfying (11), as illustrated below.

```
set.seed(84)
ech <- rpareto(20, mu = 1/2, c = 1)
vs.test(x = ech, densfun = 'dpareto', param = c(1/2, 1))
```

By turning `extend` to `TRUE`, the possible values for the window size is enlarged, possibly enabling the existence of Vasicek estimates smaller than empirical entropy.

Note that when estimating the p-value by Monte-Carlo methods, it may happen that for some replicates, the constraint (10) is not satisfied whatever be the window size. These replicates are then ignored and the p-value is computed from remaining replicates. A warning message is added to the output, informing the user on the number of ignored replicates.

```
data(contaminants) #load data from package vsgoftest; see ?contaminants
set.seed(1)
vs.test(x = aluminium2, densfun = 'dpareto')

Warning in vs.test(x = aluminium2, densfun = "dpareto"):  For 176 simulations (over 5000
), entropy estimate is greater than empirical maximum entropy for all window size
```

```
Vasicek-Song GOF test to Pareto distribution family

data:  aluminium2
Test statistic = 1.3676, Optimal window = 2, p-value < 2.2e-16
sample estimates:
         mu           c
  0.3288148 360.0000000
```

A large proportion of such ignored replicates may indicate that the original sample is too small or obviously does not fit the null distribution.

# 3 Application to real data

This Section is devoted to the application to real data of Vasicek-Song GOF tests. The package **vs.test** provides environmental data originating from a guidance report edited by the Technology Support Center of the United States Environmental Protection Agency; see [3]. According to [3], environmental scientists may have to take remediation decisions at suspected sites based on organic and inorganic contaminant concentration measurements. From a statistical point of view, these decisions are usually derived from the computation of confidence upper bounds for contaminant concentrations. Testing the goodness-of-fit of the distribution of data to specified models hence appears of prior importance. [3] also point out that contaminant concentration data from sites quite often appear to follow a skewed probability distribution, making the log-normal family a frequently-used model. They illustrate their purpose by applying Shapiro-Wilk test to the log-transformed of the samples `aluminium1`, `manganese`, `aluminium2` and `toluene` (stored into the present package)[1].

The following code chunks intend to illustrate the applicability and behaviour of the function `vs.test` on these environmental data. The significant level is fixed to 0.1 as in [3]. Note that, because the four samples are small (at most 23 observations), all p-values presented below are estimated by means of Monte-Carlo methods. Note also that warning messages notifying that there are ties in the samples have been dropped out from outputs.

```
set.seed(1)
vs.test(x = aluminium1, densfun = 'dlnorm')


Vasicek-Song GOF test to log-normal distribution family

data:  aluminium1
Test statistic = 0.31232, Optimal window = 2, p-value = 0.3372
sample estimates:
Location    Scale
6.225681 1.609719
```

The log-normality hypothesis is accepted for `aluminium1`. Similar results are obtained for `manganese`. Log-normality is rejected for `aluminium2`.

[1] A succinct description of these data is available by evaluating the following R command: ?contaminants

```r
set.seed(1)
vs.test(x = aluminium2, densfun = 'dlnorm')


Vasicek-Song GOF test to log-normal distribution family

data:  aluminium2
Test statistic = 0.48369, Optimal window = 2, p-value = 0.0256
sample estimates:
 Location     Scale
8.9273293 0.8264409
```

Due to numerous ties in `toluene`, `vs.test` can not compute Vasicek entropy estimate unless `extend` is turned to `TRUE`. Still, `vs.test` notifies that the constraint (10) is violated for all window sizes, which suggests that data are not likely to be drawn from log-normal distribution. Turning `relax` to `TRUE` yields the following result.

```r
set.seed(1)
vs.test(x = toluene, densfun = 'dlnorm', extend = TRUE, relax = TRUE)


Vasicek-Song GOF test to log-normal distribution family

data:  toluene
Test statistic = -2.4984, Optimal window = 11, p-value = 0.7308
sample estimates:
Location    Scale
4.651002 3.579041
```

Still, this last result looks spurious as the test statistic is negative (resulting from the constraint (10) to be disabled by `relax = TRUE`). Alternatively, it is possible to test normality of the log-transformed sample as follows.

```r
set.seed(1)
vs.test(x = log(toluene), densfun ='dnorm', extend = TRUE)


Vasicek-Song GOF test to normal distribution family

data:  log(toluene)
Test statistic = 0.6536, Optimal window = 11, p-value = 2e-04
sample estimates:
    Mean St. dev.
4.651002 3.579041
```

In summary, log-normality is accepted for `aluminium1` and `manganese` while it is rejected for `aluminium2` and `toluene`. These results are consistent with those obtained by [3]. To go a step further, the goodness-of-fit to Pareto distribution family can also be performed for `aluminium2` and `toluene`. Log-normal and Pareto distributions have historically competed with sometimes closely related generating processes and hard-to-distinguish tail properties. Goodness-of-fit of `aluminium2` to Pareto distribution family is rejected.

```
set.seed(1)
vs.test(x = aluminium2, densfun = 'dpareto')


Vasicek-Song GOF test to Pareto distribution family

data:  aluminium2
Test statistic = 1.3676, Optimal window = 2, p-value < 2.2e-16
sample estimates:
         mu           c
  0.3288148 360.0000000
```

Applying `vs.test` to `toluene` with default settings does not yield any result because of numerous ties and the constraint (10) being violated. Again, turning `extend` and `relax` to `TRUE` yields the following spurious result.

```
set.seed(12)
vs.test(x = toluene, densfun = 'dpareto', extend = TRUE, relax = TRUE)


Vasicek-Song GOF test to Pareto distribution family

data:  toluene
Test statistic = -2.9248, Optimal window = 11, p-value = 0.4426
sample estimates:
       mu         c
0.2815007 3.0000000
```

Finally, it is possible to test uniformity of the sample transformed by the cumulative density function of the Pareto distribution, as follows, yielding to accept the goodness-of-fit of `toluene` to the Pareto distribution family.

```
#First, compute the MLE of parameters of Pareto dist.
res.test <- vs.test(x = toluene,
                    densfun = 'dpareto',
                    extend = TRUE, relax = TRUE)
#Then, test uniformity of transformed data
set.seed(5)
vs.test(x = ppareto(toluene,
                    mu = res.test$estimate[1],
                    c = res.test$estimate[2]),
        densfun ='dunif', param = c(0,1), extend = TRUE)


Vasicek-Song GOF test to uniform distribution with Min=0, Max=1

data:  ppareto(toluene, mu = res.test$estimate[1], c = res.test$estimate[2])
Test statistic = 0.25383, Optimal window = 10, p-value = 0.2496
```

# Appendix: Vasicek-Song tests, theoretical background

[4] proposes a goodness-of-fit test based on Kullback-Leibler divergence for either simple (2) or composite (3) null hypotheses. Precisely, the test statistic $I_{mn}$ is an estimator of the Kullback-Leibler divergence $\mathbb{K}(P|P_0(\theta)) = -\mathbb{S}(P) - \int p_0(x;\theta)p(x)\mathrm{d}x$ of the sampled distribution $P$, with respect to the null distribution $P_0(\theta)$ (with respective densities $p$ and $p_0(.;\theta)$) in case of a simple hypothesis or some estimate $P_0(\widehat{\theta}_n)$ otherwise:

$$I_{mn} := -V_{mn} - \frac{1}{n}\sum_{i=1}^{n} \log p_0(X_i, \widehat{\theta}_n), \tag{4}$$

where $V_{mn}$, given by (1), estimates $\mathbb{S}(P)$ while $-\frac{1}{n}\sum_{i=1}^{n} \log p_0(X_i, \widehat{\theta}_n)$ estimates $-\int_{\mathbb{R}} \log p_0(x;\theta)p(x)\mathrm{d}x$. For the test (2) with simple null hypothesis, set $\widehat{\theta}_n = \theta$, where $\theta$ is the null parameter. Otherwise, $\widehat{\theta}_n$ is the maximum likelihood estimator (MLE) of $\theta$, i.e., it satisfies

$$\frac{1}{n}\sum_{i=1}^{n} \log p_0(X_i, \widehat{\theta}_n) = \max_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \log p_0(X_i, \theta).$$

[4] establishes the asymptotic behaviour of $I_{mn}$, independently of the null hypothesis. Precisely, $I_{mn}$ is consistent and asymptotically normally distributed, provided the distribution of the sample belongs to the following class of distributions:

$$\mathcal{F} = \left\{ P \in \mathcal{D} : \sup_{x:\ 0<F(x)<1} \frac{|p'(x)|}{p^2(x)} F(x)[1 - F(x)] < \gamma \right\}, \tag{5}$$

for some $\gamma > 0$, where $F$ is the cumulative density function of $P$, with density $p$ whose derivative is $p'$ (almost every where). The class $\mathcal{F}$ contains the most classical distributions such as uniform ($\gamma = 0$), normal, exponential and gamma ($\gamma = 1$), Fisher ($\gamma = (2 + \nu_2)/\nu_2$ where $\nu_2$ is the second degree of freedom), Pareto ($\gamma = (\mu + 1)/\mu$ where $\mu$ is the shape parameter), etc. If $\mathcal{P}_0(\Theta) \subset \mathcal{F}$, and if

$$m/\log n \xrightarrow[n\to\infty]{} 0 \quad \text{and} \quad m(\log n)^{2/3}/n^{1/3} \xrightarrow[n\to\infty]{} 0, \tag{6}$$

then

$$\sqrt{6mn}[I_{mn} - \log(2m) + \psi(2m)] \xrightarrow{\mathcal{D}} \mathcal{N}(0,1), \tag{7}$$

where $\psi(m)$ is the digamma function. The asymptotic bias $\log(2m) - \psi(2m)$ of $I_{mn}$ is that of $-V_{mn}$. [4] points out that $I_{mn}$ may have an additional substantial bias for small samples and suggests the following bias correction in the asymptotic distribution (7), from which decision rule can be consistently derived for moderate and large sample sizes:

$$\sqrt{6mn}\,[I_{mn} - b_{mn}] \xrightarrow{\mathcal{D}} \mathcal{N}(0,1), \tag{8}$$

where

$$b_{mn} = \log(2m) - \log(n) - \psi(2m) + \psi(n+1) + \frac{2m}{n}R_{2m-1} - \frac{2}{n}\sum_{i=1}^{m} R_{i+m-2},$$

with $R_m = \sum_{j=1}^{m} 1/j$. Through (8), an asymptotic p-value for the related VS test is derived, given by

$$p = 1 - \Phi^{-1}\left(\sqrt{6mn}\,[I_{mn}(x_1^n) - b_{mn}]\right), \tag{9}$$

where $I_{mn}(x_1^n)$ denotes the value of the statistic $I_{mn}$ for the observations $x_1^n = (x_1, \ldots, x_n)$ and $\Phi$ denotes the cumulative density function of the normal distribution. According to [4], the asymptotic p-value (9) provides accurate results when the sample size $n$ is at least 80.

For small sample sizes, Monte Carlo simulations may be preferred for computing p-values, as follows. A large number $N$ of replications of the sample $X_1^n$ drawn from the distribution $P_0(\widehat{\theta}_n)$ (or $P_0(\theta)$ in case of simple null hypothesis) are generated. The test statistic $I_{mn}^i$ is computed for each replication $i, 1 \leq i \leq N$. The p-value is then given by the empirical mean $(\sum_{i=1}^N \mathbb{1}_{\{I_{mn}^i > I_{mn}(x_1^n)\}})/N$.

For choosing $m$, [4] proposes to minimize $I_{mn}$ – that is maximize $V_{mn}$, with respect to $m$, yielding the most conservative test. The author notes also that the values of $m$ for which $I_{mn}$ is negative have to be excluded. Indeed, such negative values for $I_{mn}$ constitute poor estimates of the non-negative divergence $\mathbb{K}(P|P_0(\theta))$. Hence, $m$ has to be chosen subject to the constraint

$$V_{mn} \leq -\frac{1}{n}\sum_{i=1}^n \log(p_0(.;\widehat{\theta}_n)). \tag{10}$$

Finally, the window size selected by Song – say the optimal window size, is

$$\widehat{m} = \min\left\{m^* \in \underset{m\in\mathbb{N}^*}{\operatorname{argmax}}\left\{V_{mn} : \ V_{mn} \leq -\frac{1}{n}\sum_{i=1}^n \log p_0(X_i, \widehat{\theta}_n)\right\} : 1 \leq m^* < \lfloor n^{1/3-\delta}\rfloor\right\}, \tag{11}$$

for some $\delta \in \mathbb{R}$ such that $1/3 - \delta > 0$ and the VS test statistic is

$$I_{\widehat{m}n} = -V_{\widehat{m}n} - \frac{1}{n}\sum_{i=1}^n \log p_0(X_i;\widehat{\theta}_n). \tag{12}$$

The upper bound $n^{1/3-\delta}$ for the window size $m$ is chosen so that conditions (6) are fulfilled and the asymptotic normality (7) holds. No systematic optimal choice for $\delta$ exists; it can depend on the family of distributions the GOF is tested to.

# References

[1] Girardin V, Lequesne J (2017). *Entropy-based goodness-of-fit test, a unifying framework. Application to DNA replication.* Communications in Statistics – Theory and Methods. doi:10.1080/03610926.2017.1401084

[2] Lequesne J, Regnault P (2018) *Package vsgoftest for R: goodness-of-fit tests based on Kullback-Leibler divergence.* URL:

[3] Singh AK, Singh A, Engelhardt M (1997). The lognormal distribution in environmental applications. In *Technology Support Center Issue Paper.* Citeseer.

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