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

Justine Lequesne*and Philippe Regnault†

June 7, 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 **vsgoftest** package provides functions for estimating Shannon entropy of absolutely continuous distributions and testing the goodness-of-fit of some theoretical family of distributions to a vector of real numbers. It also provides functions for computing the density, cumulative density and quantile functions of Pareto and Laplace distributions, as well as for generating samples from these 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:

```
#Package devtools must be installed
devtools::install_github('pregnault/vsgoftest')
```

The package is structured around two functions, `entropy.estimate` and `vs.test`. The first one computes the spacing based estimator of the differential entropy

$$\mathbb{S}(P) := -\int_{\mathbb{R}} p(x)\log p(x)\mathrm{d}x \tag{1}$$

of a distribution P on $\mathbb{R}$ with density p from a numeric sample $X_1,\ldots,X_n$ drawn from P from a numeric sample. The second one performs Vasicek-Song GOF test for 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. The theoretical background attached to entropy estimation and Vaiscek-Song tests is presented in the appendix. In addition, 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 estimate (5) of Shannon entropy (1) from a numeric sample. Two arguments have to 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 (5).

It returns, as a single value, the estimate of Shannon entropy of the sample. Here 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 exact value of entropy

[1] 1.418939
```

The estimate returned by `entropy.estimate` obviously depends on the window selected by the user, as illustrated by the following chunck.

```
sapply(1:10, function(w) entropy.estimate(x = samp, window =w))

 [1] 1.205018 1.346352 1.378732 1.387337 1.391691 1.393512 1.394428
 [8] 1.394728 1.394486 1.392669
```

One may select the window size that maximizes the entropy estimate, as follows.

```
n <- 100 #sample size
V <- sapply(1:(n/2 - 1), function(w) entropy.estimate(x = samp, window =w))
which.max(V) #Choose window that maximizes entropy

[1] 8
```

Let us consider a sample drawn from a Pareto distribution with density

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

where $c > 0$ and $\mu > 0$, which can be obtained by making use of the function `rpareto` as illustrated below. Its Shannon entropy is

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

```
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 #Exact 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 GOF test (VS test) of a numerical sample for 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. Setting two non-optional arguments is required:

- `x`: the numeric sample;
- `densfun`: a character string specifying the theoretical family of distributions of the null hypothesis. 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 ofthe family of normal distributions.

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

- `statistic`: the value of VS test statistic (13) for the sample, with optimal window size defined by (12);
- `parameter`: the optimal window size;
- `estimate`: the maximum likelihood estimate of the parameters of the null distribution;
- `p.value`: the p-value associated to the sample.

By default, `vs.test` performs the composite VS test of `x` for the family of distributions `densfun`. The p-value is estimated by Monte-Carlo method if the sample size is smaller than 80, or through the asymptotic distribution (8) of the VS test statistic otherwise.

In the following example, a normally distributed sample is simulated. VS test rejects the null hypothesis that this sample is drawn from a Laplace distribution, but does not reject the normality hypothesis (for a significant level set to 0.05).

```
set.seed(5)
samp <- rnorm(50,2,3)
vs.test(x = samp, densfun = 'dlaplace')


Vasicek-Song GOF test for the Laplace distribution

data:  samp
Test statistic = 0.32437, Optimal window = 2, p-value = 0.0248
sample estimates:
   Shape    Scale
2.194803 2.687321
```

```
set.seed(4)
vs.test(x = samp, densfun = 'dnorm')


Vasicek-Song GOF test for the normal distribution

data:  samp
Test statistic = 0.21655, Optimal window = 2, p-value = 0.3704
sample estimates:
    Mean St. dev.
2.194803 3.173824
```

For performing a simple null hypothesis GOF test, the additional argument **param** has to be set to a numeric vector, consistent with the parameter requirements for the null distribution. In such case, the MLE of the parameter(s) of the null distribution has not to be computed and hence the component **estimate** in results is not available.

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


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

data:  samp
Test statistic = 0.22196, Optimal window = 2, p-value = 0.331
```

If **param** is not consistent with the specified distribution – e.g., standard deviation for testing a normal distribution is missing or negative, the execution is stopped and an error message is returned.

```
set.seed(2)
samp <- rnorm(50, -2, 1)
vs.test(samp, densfun = 'dnorm', param = -2)

Error in vs.test(samp, densfun = "dnorm", param = -2):  "param":  invalid parameter (not
consistent with the specified distribution)
```

One can estimate the p-value of the sample by Monte-Carlo simulations, even when sample size is larger than 80, by setting the optional argument **simulate.p.value** to **TRUE** (**NULL** by default). The number of Monte-Carlo replicates can be fixed through the optional argument **B** (default is **B = 5000**).

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


Vasicek-Song GOF test for the exponential distribution

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 for the exponential distribution

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

Vasicek's estimates V_{mn} are computed for all m from 1 to $n^{1/3-\delta}$, where $\delta < 1/3$; the test statistic is $I_{\widehat{m}n}$ for $\widehat{m}$ the optimal window size, as defined in (12). The choice of δ depends on the family of distributions of the null hypothesis. Precisely, for Weibull, Pareto, Fisher, Laplace and Beta, δ is set by default to 2/15, while for uniform, normal, log-normal, exponential and gamma, it is set to 1/12. These default settings result from numerous experimentations. Still, the user can choose another value through the optional argument `delta`.

```
set.seed(63)
vs.test(samp, densfun = 'dexp', delta = 5/30)


Vasicek-Song GOF test for the exponential distribution

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

Note that upper-bounding the window size by $n^{1/3-\delta}$ is only required when the asymptotic normality of I_{mn} is used to compute asymptotic p-values from (8). When the p-value are computed by means of Monte-Carlo simulation, this upper-bound can be extended to $n/2$ by adding `extend = TRUE`, which may lead to a more reliable test, as illustrated below.

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


Vasicek-Song GOF test for the log-normal distribution
```

```
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 for the log-normal distribution

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

Enlarging the range of m is also pertinent if ties are present in the sample. Indeed, the presence of ties is particularly inappropriate for performing VS tests, because some spacings $X_{(i+m)} - X_{(i-m)}$ can be zero. The window size m has thus 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. Setting `extend` to `TRUE` can avoid this behavior, 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 for the exponential distribution

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

Finally, Vasicek's estimate V_{mn} may exceed the parametric estimate of the entropy of the null distribution for all m between 1 and $n^{1/3-\delta}$. Then, no window size exists satisfying (12), 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))

Error in vs.estimate(x, densfun, ESTIM, extend, delta, relax):  The sample entropy is
greater than empirical maximal entropy for all possible window sizes; the sample may be
too small or is unlikely to be drawn from the null distribution.
```

Enlarging the possible window sizes by setting `extend` to `TRUE` may enable Vasicek estimates to be smaller than empirical entropy.

Note that when computing the p-value by Monte-Carlo simulation, the constraint (11) may not be satisfied for some replicates, whatever be the window size. These replicates are then ignored and the p-value is computed from the remaining replicates. A warning message is added to the output, informing 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 sizes.


Vasicek-Song GOF test for the Pareto distribution

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 does not fit the null distribution.

3 Application to real data

The **vs.test** package contains 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 take remediation decisions at suspected sites based on organic and inorganic contaminant concentration measurements. These decisions usually derive 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 interest. [3] also points out that contaminant concentration data from sites often appear to follow a skewed probability distribution, making the log-normal family a frequently-used model. The authors illustrate their purpose by applying Shapiro-Wilk test to the log-transformed of the samples `aluminium1`, `manganese`, `aluminium2` and `toluene` (stored in the present package)[1]; see the empirical skewness computed in the following chunk.

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

```
data(contaminants) #Load environmental data from package
#Package DescTools required for this chunk
unlist(lapply(X = list(aluminium1, manganese, aluminium2, toluene),
       FUN = DescTools::Skew))

[1] 2.323343 1.698686 1.996607 3.961129
```

The following code chunks intend to illustrate the use and behaviour of the function `vs.test` for these environmental data. The significant level is fixed to 0.1 as in [3]. 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 for the log-normal distribution

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

The log-normal hypothesis is not rejected for `aluminium1`. Similar results are obtained for `manganese`. Log-normality is rejected for `aluminium2`.

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


Vasicek-Song GOF test for the log-normal distribution

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 set to `TRUE`. Still, `vs.test` notifies that the constraint (11) is violated for all window sizes, which suggests that data are not likely to be drawn from log-normal distribution; see Section **??**. Turning `relax` to `TRUE` yields the following result.

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


Vasicek-Song GOF test for the log-normal distribution
```

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

Again, this last result looks spurious because the test statistic is negative – resulting from (11) not being satisfied by setting `relax = TRUE`. An alternative is to test normality of the log-transformed sample as follows.

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


Vasicek-Song GOF test for the normal distribution

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

The log-normal hypothesis is not rejected for `aluminium1` and `manganese` while it is rejected for `aluminium2` and `toluene`. These results are consistent with those obtained by [3]. Further, the goodness-of-fit to the Pareto distributionsis performed for `aluminium2` and `toluene`. Log-normal and Pareto distributions usually compete with closely related generating processes and hard to distinguish tail properties. Goodness-of-fit of Pareto distribution is rejected for `aluminium2`.

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


Vasicek-Song GOF test for the Pareto distribution

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 yields no result because of numerous ties and the violation of (11). Uniformity of the sample transformed by the cumulative density function of the Pareto distribution can be tested as follows. Goodness-of-fit of the Pareto distribution is not rejected for `toluene`.

```
#Compute the MLE of parameters of Pareto dist.
res.test <- vs.test(x = toluene,
                    densfun = 'dpareto',
                    extend = TRUE, relax = TRUE)
#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 for the 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

$$V_{mn} := \frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{n}{2m}\left[X_{(i+m)} - X_{(i-m)}\right]\right) \tag{5}$$

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 θ is the null parameter. Otherwise, $\widehat{\theta}_n$ is the maximum likelihood estimator (MLE) of θ, 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:

$$\mathcal{F} = \left\{ P \in \mathcal{D} : \sup_{x:\, 0<F(x)<1} \frac{|p'(x)|}{p^2(x)} F(x)[1-F(x)] < \gamma \right\}, \tag{6}$$

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 ν_2 is the second degree of freedom), Pareto ($\gamma = (\mu+1)/\mu$ where μ is the shape parameter), etc. If $\mathcal{P}_0(\Theta) \subset \mathcal{F}$, and if

$$m/\log n \xrightarrow[n\to\infty]{} 0 \quad \text{and} \quad m(\log n)^{2/3}/n^{1/3} \xrightarrow[n\to\infty]{} 0, \tag{7}$$

then

$$\sqrt{6mn}[I_{mn} - \log(2m) + \psi(2m)] \xrightarrow{\mathcal{D}} \mathcal{N}(0,1), \tag{8}$$

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 (8), from which decision rule can be consistently derived for moderate and large sample sizes:

$$\sqrt{6mn}\,[I_{mn} - b_{mn}] \xrightarrow{\mathcal{D}} \mathcal{N}(0,1), \tag{9}$$

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 (9), an asymptotic p-value for the related VS test is derived, given by

$$p = 1 - \Phi^{-1}\left(\sqrt{6mn}\,[I_{mn}(x_1^n) - b_{mn}]\right), \tag{10}$$

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 Φ denotes the cumulative density function of the normal distribution. According to [4], the asymptotic p-value (10) 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

$$V_{mn} \leq -\frac{1}{n}\sum_{i=1}^{n} \log(p_0(.;\widehat{\theta}_n)). \tag{11}$$

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

$$\widehat{m} = \min\left\{ m^* \in \underset{m \in \mathbb{N}^*}{\operatorname{argmax}} \left\{ V_{mn} :\ V_{mn} \leq -\frac{1}{n}\sum_{i=1}^{n} \log p_0(X_i, \widehat{\theta}_n) \right\} : 1 \leq m^* < \lfloor n^{1/3-\delta} \rfloor \right\}, \tag{12}$$

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

$$I_{\widehat{m}n} = -V_{\widehat{m}n} - \frac{1}{n}\sum_{i=1}^{n} \log p_0(X_i; \widehat{\theta}_n). \tag{13}$$

The upper bound $n^{1/3-\delta}$ for the window size m is chosen so that conditions (7) are fulfilled and the asymptotic normality (8) holds. No systematic optimal choice for δ 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.