Journal of Statistical Software

August 2015, Volume 66, Code Snippet 1. *http://www.jstatsoft.org/*

Testing Goodness-of-Fit with the Kernel Density Estimator: GoFKernel

Jose M. Pavia
Universitat de Valencia

Abstract

To assess the goodness-of-fit of a sample to a continuous random distribution, the most popular approach has been based on measuring, using either L_∞- or L_2-norms, the distance between the null hypothesis cumulative distribution function and the empirical cumulative distribution function. Indeed, as far as I know, almost all the tests currently available in R related to this issue (`ks.test` in package **stats**, `ad.test` in package **ADGofTest**, and `ad.test`, `ad2.test`, `ks.test`, `v.test` and `w2.test` in package **truncgof**) use one of these two distances on cumulative distribution functions. This paper (i) proposes `dgeometric.test`, a new implementation of the test that measures the discrepancy between a sample kernel estimate of the density function and the null hypothesis density function on the L_1-norm, (ii) introduces the **GoFKernel** package, and (iii) performs a large simulation exercise to assess the calibration and sensitivity of the above listed tests as well as the Fan's test (Fan 1994), `fan.test`, also implemented in the **GoFKernel** package. In addition to `dgeometric.test` and `fan.test`, the **GoFKernel** package adds a couple of functions that R users might also find of interest: `density.reflected` extends `density`, allowing the computation of consistent kernel density estimates for bounded random variables, and `random.function` offers an ad-hoc and universal (although computational expensive and potentially inaccurate for long tail distributions) sampling method. In light of the simulation results, we can conclude that (i) the tests implemented in the **truncgof** package should not be used to assess goodness-of-fit (at least for non-truncated distributions), (ii) the test `fan.test` shows an over-tendency to not reject the null hypothesis, being visibly miscalibrated (at least in its default option, where the bandwidth parameter is estimated using `dpik` from package **KernSmooth**), (iii) the tests `ks.test` and `ad.test` show similar power, with `ad.test` being slightly preferable in large samples, and (iv) `dgeometric.test` represents a good alternative given its satisfactory calibration and its, in general, superior power in samples of medium and large sizes. As a counterpart it entails more computational burden when the random generator of the null hypothesis density function is not available in R and `random.function` must be used.

Keywords: non-parametric tests, Fan's test, L_1-distance, R, **truncgof**, **ADGofTest**.

1. Introduction

In the literature there are a number of non-parametric tests to assess whether a sample of a continuous random variable comes from a specified distribution. In goodness-of-fit tests the usual statistics are based on measuring, in some way, the discrepancy between either the empirical cumulative distribution or density function and the corresponding theoretical function. L_∞-norm and L_2-norm have been the most popular distance measures employed. Indeed, the tests currently available in R (R Core Team 2015) to this issue (such as `ks.test` from package **stats**, `ad.test` from package **ADGofTest** (Bellosta 2011), `kuiper.test` from package **circular**[1] and the tests `ad.test`, `ad2.test`, `ks.test`, `v.test` and `w2.test` available in the package **truncgof** [2,3]) use one of these two distances on cumulative distribution functions.

These are not however the unique dissimilarity criteria suggested in the literature to deal with this problem. Some proposals can also be found using likelihood ratios, Kullback-Leibler divergence and Renyi distance via entropy measures, or other closeness measures – see, e.g., Fan and Gencay (1993), Zhang (2002), Mattheou and Karagrigoriou (2010), Vexler and Gurevich (2010), or, Mashhadi (2011). Indeed, within these approaches, it is still possible to find in R the **dbEmplikeGOF** package which provides a function, `dbEmplikeGOF`, for density based empirical likelihood goodness-of-fit tests based on sample entropy (Miecznikowski, Vexler, and Shepherd 2013). Unfortunately, the `dbEmplikeGOF` function currently only performs tests of normality and uniformity and, therefore, does not offer a general solution.

In this paper, two tests operating on the density function are made available to R users through the **GoFKernel** package. In particular, the **GoFKernel** package contains an implementation of the Fan's test (Fan 1994), which is based on the L_2-distance, and a practical approximation to compare, using the L_1-norm, the discrepancy between a theoretical density function and a sample kernel estimate of the density function (Cao and Lugosi 2005).[4] The p values of a test based on this distance can be easily computed by Monte Carlo simulation using statistical software. The R functions collected in the **GoFKernel** package are designed to run this test for (almost) any one-dimensional continuous random variable.

Although this paper deals exclusively with one-dimensional continuous random variables, the test could be easily generalized to discrete variables. Its generalization to the useful case of multivariate random variables entails the use of high dimensional density estimation and it is less straightforward. Lindsay, Markatou, and Ray (2014) provide a discussion of the issues entailed and an optimal method for identifying the appropriate bandwidth for use in goodness-of-fit problems. Some proposals for goodness-of-fit tests for multivariate normal vectors, using L_2-distance and entropy measures, can be found in Bowman and Foster (1993) and Anderson, Hall, and Titterington (1994). The relevance of these references also rests on

[1]It should be noted, however, that according to the circular reference manual the `kuiper.test` function only "performs Kuiper's one-sample test of uniformity on the circle" (Lund and Agostinelli 2013).

[2]Although all the **truncgof** tests are designed to deal with left-truncated data, they can be used with non-truncated data without defining the threshold argument.

[3]The tests `adup.test` and `ad2up.test` also available in the **truncgof** package are especially designed to test whether a distribution fits the data well in the upper tail and, therefore, they are tail biased – see, Chernobai, Rachev, and Fabozzi (2005).

[4]The study of testing based on the L_1-distance is not new and has its roots in Hoeffding and Wolfowitz (1958) and LeCam (1973). Later, Györfi and van der Meulen (1991) proved the universal consistency of a similar test based on the histogram estimator, Cao and Lugosi (2005) found general non-asymptotic bounds for the power of the test, and Albers and Schaafsma (2008) provided a general table of critical values recommendable in certain circumstances.

the fact that both papers discuss bandwidth selection; an issue that, as we shall see when we analyze the outcomes for Fan's test, can be crucial in performance of L_2-distance based tests.

The paper is structured as follows. A brief review of the main goodness-of-fit tests based on comparing the distance between the empirical cumulative distribution function and the corresponding theoretical function is performed in Section 2. This review is focused on those tests currently programmed in R. Section 3 introduces theoretically the tests implemented in the R package **GoFKernel** to measure the discrepancies between the null hypothesis density function and an empirical kernel estimate. Section 4 describes and exemplifies the main functions available in **GoFKernel**. In Section 5, a large simulation exercise comparing the calibration (size), power (sensitivity), and speed of the different general goodness-of-fit tests available in R is carried out for several distributions and sample sizes. Section 6 investigates convergence of the `dgeometric.test`. Finally, Section 7 provides conclusions.

2. Tests based on the cumulative distribution function

The Kolmogorov-Smirnov test (KS) is probably the most widely known non-parametric goodness of fit test (`ks.test` from package **stats**, `ks.test` from package **truncgof**). The KS statistic, Equation 1, quantifies at the sampled values, x_i, and with the L_∞-norm the maximum (supremum) distance in absolute values between the empirical cumulative distribution function (ECDF) of the sample, $F_n(x_i)$, and the cumulative function of the reference distribution, $F(x)$. A distance that, as is well-known, converges to 0 if the sample comes from the reference distribution.

$$KS = \max_{x_i} |F_n(x_i) - F(x_i)| \tag{1}$$

The KS test is however not considered to have good power (e.g., Stephens 1974), requiring a relatively large number of data points to properly reject the null hypothesis (Frampton 2010), and is moreover considered more sensitive to the part of the cumulative distribution above the median (Johnson, Miller, and Freund 2011).

In addition to the KS test, within the ECDF testing strategy, we can find the Kuiper test (`v.test`) and the family of Anderson-Darling tests (`ad.test` and `ad2.test` from package **truncgof** and `ad.test` from package **ADGofTest**). The Kuiper test (Kuiper 1962) is closely related to the KS test, being test statistic KP, Equation 2, defined as the sum of the maximum positive and negative deviation between the empirical and theoretical cumulative distributions.

$$KP = \max_{x_i}(i/n - F(x_i)) + \max_{x_i}(F(x_i) - (i-1)/n) \tag{2}$$

This test is equally sensitive to differences along the entire range of the distribution and compared to KS test has the advantage of being invariant under cyclic transformations[5]. It shares however the limitations of the KS test.

The Anderson-Darling tests are another alternative to the KS test. In the same way as the KP test, these tests provide equal sensitivity at both tails (although not maintaining the cyclic invariance). In its simplest version (`ad.test` from package **truncgof**), it is a variance weighted KS statistic based on the L_∞-norm, Equation 3, and in its L_2-norm variant (Anderson and

[5]This makes this test especially suitable to test circular probability distributions (Jammalamadaka and Sengupta 2001) or to identify differences in spread (Press, Flannery, Teukolsky, and Vetterling 1992).

Darling 1952), it is a generalization of the Cramer-von Mises test (`w2.test` from package **truncgof**). In particular, the Anderson-Darling L_2-norm variant is based on a quadratic ECDF statistic measure, Equation 4, and has the advantage of taking into account the differences between the empirical and theoretical cumulative distributions at all the sampled points.

$$AD = \max_{x_i} \left| \frac{F_n(x_i) - F(x_i)}{F(x) \left(1 - F(x)\right)} \right|, \tag{3}$$

$$AD2 = n \int_{-\infty}^{\infty} \left(F_n(x) - F(x)\right)^2 w(x) dF(x). \tag{4}$$

More specifically, the Cramer-von Mises test uses $w(x) = 1$ as weighting function, while the Anderson-Darling test employs $w(x) = [F(x)(1 - F(x))]^{-1}$; a weighting function that places more weight on observations in the tails of the distribution.

Other alternative tests also based on comparing empirical and theoretical cumulative functions have been proposed by Watson (1961), Stephens (1964), and Pearson and Stephens (1962), among others – see also Marsaglia and Marsaglia (2004).

3. Density function based tests

A large number of goodness-of-fit tests have also been proposed using empirical approximations to the density function. Indeed, the idea of using non-parametric empirical (kernel) density estimators for goodness-of-fit tests goes back to Bickel and Rosenblatt (1973) and Rosenblatt (1975). More recent work includes Bowman (1992), Ahmad and Cerrito (1993), Fan (1994, 1998) and Fan and Ullah (1999), among those papers that base their tests on the L_2-error, and Cao and Lugosi (2005) and Albers and Schaafsma (2008), among the approaches employing L_1-distances. In particular, following Fan (1994) and Cao and Lugosi (2005), the statistical discrepancy measures used are based respectively on the integral of the squared difference between an empirical kernel function estimate (EKF) of the unknown density function and the null hypothesis density function to be tested, $I^g_{n,h}$, Equation 5, and the integral of the absolute value difference, $T_{n,h}$, Equation 6.

$$I^g_{n,h} = \int_{-\infty}^{\infty} \left(f_{n,h}(x) - f(x)\right)^2 dx, \tag{5}$$

$$T_{n,h} = \int_{-\infty}^{\infty} \left|f_{n,h}(x) - f(x)\right| dx, \tag{6}$$

where $f_{n,h}(x)$ is a sample empirical h-bandwidth kernel estimation and $f(x)$ is the density function under the null hypothesis. Note that the values of the statistics depend on the bandwidth selected.

After developing the sum of squares of Equation 5 and employing some empirical approximations, Fan (1994) proposed a bias-corrected test, later simplified in Li and Racine (2007, pp. 380–381), which follows an asymptotic Gaussian distribution under some standard smoothness assumptions and the condition that the kernel bandwidth tends to zero when its product with the sample size tends to infinity. According to Cao and Lugosi (2005, p. 600), however, L_1 approaches are preferable to L_2 ones since they allow one "to drop unnecessary assumptions as well as to obtain non-asymptotic performance bounds".

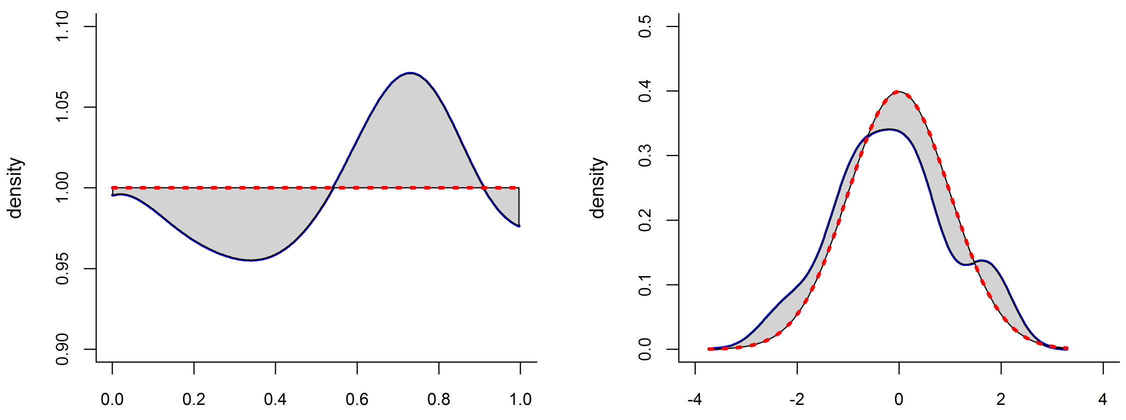

Figure 1: The shaded areas are the values that are computed with the $T_{n,h}$ statistic, Equation 6. The left figure has been obtained from a sample of size 500 from a $U(0,1)$, and the right one from a sample of size 50 from a $N(0,1)$. Reference densities are marked in red using dashed lines and empirical densities in blue with continuous lines.

To compute the p value of the test based on Equation 6, we can use numerical integration and Monte Carlo simulation. After computing by numerical integration the area between the density function under the null hypothesis and a sample empirical kernel estimator (see Figure 1), we can obtain the p value of the test by simulation as follows: (i) drawing S samples from $f(x)$ with the same size n as our actual sample; (ii) estimating the kernel density function $f_{n,h_s}(x)$ for each of these new samples; (iii) computing the area between the theoretical density and each of the estimates of (ii); and, (iv) calculating the p value as the proportion of times the sample S areas computed in (iii) exceed the value of $T_{n,h}$ obtained from the observed sample. This approach resembles the strategy implemented in the package **truncgof** and is quite close to the one suggested in Cao and Lugosi (2005) for computing the critical region of the $T_{n,h}$ statistic, given a significance level. On the one hand, the tests implemented in the **truncgof** package compute the p value by resampling in a transformation of the (truncated) distribution (Chernobai *et al.* 2005, p. 5), which under the null hypothesis follows a uniform distribution. On the other hand, Cao and Lugosi (2005, p. 609) suggested a similar algorithm to compute the threshold of the critical region of the test, but with a fixed bandwidth properly chosen to minimize $T_{n,h}$ under the null hypothesis. Indeed, Cao and Lugosi (2005, p. 599) consider that the choice of the bandwidth "plays a crucial role in the performance of the test". Regarding the number of S samples needed to obtain a relatively accurate approximation of the p value, although it depends on the particular null hypothesis density considered, it seems that as a rule good results are obtained with just a hundred of samples (see Section 6). More accurate approximations that require the increase of the number of samples can be obtained at the expense of increased computational time.

4. Package GoFKernel

The algorithm described above to implement a test based on the $T_{n,h}$ statistic as well as the Fan's test (with the simplification suggested in Li and Racine 2007, pp. 380–381 which "should have better finite-sample properties" since it has an asymptotic zero center term) has been programmed in the **GoFKernel** package. These two tests are implemented in the two main functions of the package: `fan.test` and `dgeometric.test`. Section 4.1 explains the `fan.test` function and Section 4.2 describes how the `dgeometric.test` works. Furthermore, in Section 4.3 the rest of the functions of the package, a couple of which (`density.reflected` and `random.function`) could be of interest for R practitioners, are presented.

4.1. Function `fan.test`

The function `fan.test` performs the Fan's test (Fan 1994) in the variant proposed in Li and Racine (2007, pp. 380–381). The programmed test is based on an asymptotic approximation of $I^g_{n,h}$, Equation 5, to a normal distribution and analyzes the goodness-of-fit of a sample, via a kernel density estimate, to a theoretical density function. In its default option, `fan.test` uses the function `dpik` included in the package **KernSmooth** (Wand 2013) to estimate the bandwidth[6]. Hence it requires that package to be available to run correctly. The function is used as:

```
fan.test(x, fun.den, par = NULL, lower = -Inf, upper = Inf,
  kernel = "normal", bw = NULL)
```

The arguments of the function are described as follows:

- `x`: a numeric vector of data values.
- `fun.den`: an actual density distribution function, such as `dnorm`. Only continuous densities are valid.
- `par`: a list of additional parameters of the distribution specified, default `NULL`.
- `lower`: lower end point of the support of the variable characterized by `fun.den`, default `-Inf`.
- `upper`: upper end point of the support of the variable characterized by `fun.den`, default `Inf`.
- `kernel`: a character string with the kernel to be used, either `"normal"` (a $N(0,1)$ density), `"box"` (a $U(-1,1)$ density) or `"epanech"` (an Epanechnikov quadratic kernel density), default `"normal"`.
- `bw`: a number indicating the bandwidth (parameter h in Equation 5) to be used in the empirical kernel estimate of the data, default `NULL`. In its default option, the bandwidth is estimated using the function `dpik` included in the package **KernSmooth**.

The output is an object of class '`htest`' like for the Kolmogorov-Smirnov test, `ks.test` from package **stats**. The output is a list containing the standardized value of the $I^g_{n,h}$ statistic (`statistic`), the p value of the test (`p.value`), the character string `"Fan's test"` (`method`), a character string giving the kernel used (`kernel`), and a character string giving the name of the data (`data.name`).

[6]The method implemented in `dpik` for selecting the bandwidth of a kernel density estimate was proposed by Sheather and Jones (1991) and is described in Section 3.6 of Wand and Jones (1995).

As an example, firstly, a test is carried out to see if the null hypothesis of uniform distribution can be accepted for a random sample of size 100 from a uniform distribution and, secondly, if the object `risk76.1929` available in **GoFKernel** follows the particular density function defined by $f(x) = 2 - 2x$ for $0 < x < 1$.

```
R> library("GoFKernel")
R> set.seed(125)
R> fan.test(runif(100), dunif, lower = 0, upper = 1)

 Fan's test
data:  runif(100)
Ig = -3.2061, p-value = 0.9993
```

In this example, the function `fan.test` computes for the simulated sample a value of -3.0261 for the standarization of $I^g_{n,h}$ statistic and a p value of 0.9993.

```
R> f0 <- function(x) ifelse(x >= 0 & x <= 1, 2 - 2 * x, 0)
R> fan.test(risk76.1929, f0, lower = 0, upper = 1, kernel = "epanech")

 Fan's test
data:  risk76.1929
Ig = -3.8156, p-value = 0.9999
```

According to this output the Fan's test points clearly to not reject – with a p value of 0.9999 – the null hypothesis for the `risk76.1929` dataset.

4.2. Function `dgeometric.test`

The function `dgeometric.test` performs the test described in the last paragraph of Section 3. The test is based on computing the area defined by the $T_{n,h}$ statistic for the observed data. To obtain the p value, the value of the statistic is then compared to the same area for a simulation of samples, with the same size than the observed sample, drawn from the distribution stated in the null hypothesis. The function uses a Gaussian kernel to estimate the kernel density functions and, when available, employs the random generators programed in R. It is used as:

```
dgeometric.test(x, fun.den, par = NULL, lower = -Inf, upper = Inf,
  n.sim = 101, bw = NULL)
```

The arguments of the function are described as follows:

- `x`: a numeric vector of data values.
- `fun.den`: an actual density distribution function, such as `dnorm`. Only continuous densities are valid.
- `par`: a list of additional parameters of the distribution specified, default `NULL`.
- `lower`: lower end point of the support of the variable characterized by `fun.den`, default `-Inf`.

- `upper`: upper end point of the support of the variable characterized by `fun.den`, default `Inf`.
- `n.sim`: number of iterations performed to calculate the p value, default `101`.
- `bw`: a number indicating the bandwidth (parameter h in Equation 6) to be used in the empirical kernel estimates, default `NULL`. In its default option, the bandwidth varies in each simulated dataset and is the one provided by the function `density` under hypothesis of a Gaussian kernel.

The output is an object of class '`htest`' like for the Kolmogorov-Smirnov test, `ks.test` from package **stats**. The output is a list containing the value of the $T_{n,h}$ statistic (`statistic`), the p value of the test (`p.value`), the character string `"Geometric test"` (`method`), the number of simulations performed to calculate the p value (`iterations`), and a character string giving the name of the data (`data.name`).

As an example, firstly, a test is carried out to see if the object `risk76.1929` available in **GoFKernel** follows the particular density function defined by $f(x) = 2 - 2x$ for $0 < x < 1$, and, secondly, for a random sample of size 200 from a logNormal distribution, a test is carried out to see if the null hypothesis of a Gamma distribution with sample MLE as parameters can be accepted.

```
R> set.seed(158)
R> f0 <- function(x) ifelse(x >= 0 & x <= 1, 2 - 2 * x, 0)
R> dgeometric.test(risk76.1929, f0, lower = 0, upper = 1, n.sim = 51)

 Geometric test
data:  risk76.1929
Tn = 0.1143, p-value = 0.1373
```

According to this output, at the usual significance levels, the null hypothesis is not rejected with an approximate p value of 0.1373. It is worth comparing this p value with the really different p value obtained for the same dataset with the Fan's test. As we shall see in Section 5, it seems that Fan's test tends to inflate p values, at least when the bandwidth is estimated using the default option.

```
R> library("MASS")
R> set.seed(1)
R> x <- rlnorm(200, meanlog = 1, sdlog = 1)
R> dgeometric.test(x, dgamma, par = lapply(fitdistr(x, "gamma")$estimate,
+    function(i) i), lower = 0, upper = Inf, n.sim = 125)

 Geometric test
data:  x
Tn = 0.2080, p-value = 0.016
```

According to this output, at the usual significance level of 0.05, the null hypothesis of a Gamma density is rejected for this simulated data with an approximate p value of 0.016.

4.3. Other functions in GoFKernel

In addition to the functions `fan.test` and `dgeometric.test`, which perform respectively Fan's test and the L_1 geometric test in the simulation form described at the end of Section 3, the **GoFKernel** package includes the functions `inverse`, `support.facto`, `random.function`, `area.between` and `density.reflected`. These five functions are (internal) functions necessary to implement the geometric test. More specifically, (i) `inverse` computes the inverse function of any given cumulative distribution function; (ii) `support.facto` determines for a random variable with an infinity theoretical support its numerical de facto support; (iii) `random.function` generates draws of any random variable given its density (or cumulative) function; (iv) `area.between` numerically calculates the area between a theoretical density function and an empirical kernel estimate (see Figure 1); and, (v) `density.reflected` computes an empirical kernel estimate of a sample using, for bounded variables, reflection in the borders – see, e.g., Silverman (1986). In addition, the `risk76.1929` object contains a vector with the annual fraction of the time exposed to risk of death with an age of 76 years for people born in 1929 that immigrated to Spain during 2006. Under the null hypothesis of uniform distribution of dates of birth and dates of immigration, the above time exposed to risk has as density function $f(x) = 2 - 2x$ for $0 < x < 1$ (Pavía, Morillas, and Lledó 2012).

Of the above functions, the two of most interest for R users are: `density.reflected` and `random.function`. The function `density.reflected` is based on the function `density` and produces an output of the same class as `density` for bounded variables and the same output for unbounded variables.[7] For bounded variables, `density.reflected` avoids via reflection the inconsistencies that `density` shows in the boundaries of the support of the random variable. The `random.function` function offers an ad-hoc and universal (although computational expensive and potentially inaccurate for long tail distributions) sampling method that allows the drawing of samples of (almost) any one-dimensional continuous random variable. This function makes it possible to implement the `dgeometric.test` even when the null hypothesis density function is not available in the R environment and it must be directly provided by the user. They are used as follows:

```
density.reflected(x, lower = -Inf, upper = Inf, ...)
```

The arguments of the function `density.reflected` are described as follows:

- `x`: a numeric vector of data values.
- `lower`: lower end point of the support of the variable from which `x` is supposed to come, default `-Inf`.
- `upper`: upper end point of the support of the variable from which `x` is supposed to come, default `Inf`.
- `...`: further `density` arguments.

The arguments of the function `random.function` are described as follows:

```
random.function(n = 1, f, lower = -Inf, upper = Inf, kind = "density")
```

[7] The latter is true except for two small changes. The `density.reflected` function (i) always omits `NA`s and (ii) significantly concentrates the kernel density around the unique observed value in degenerate samples.

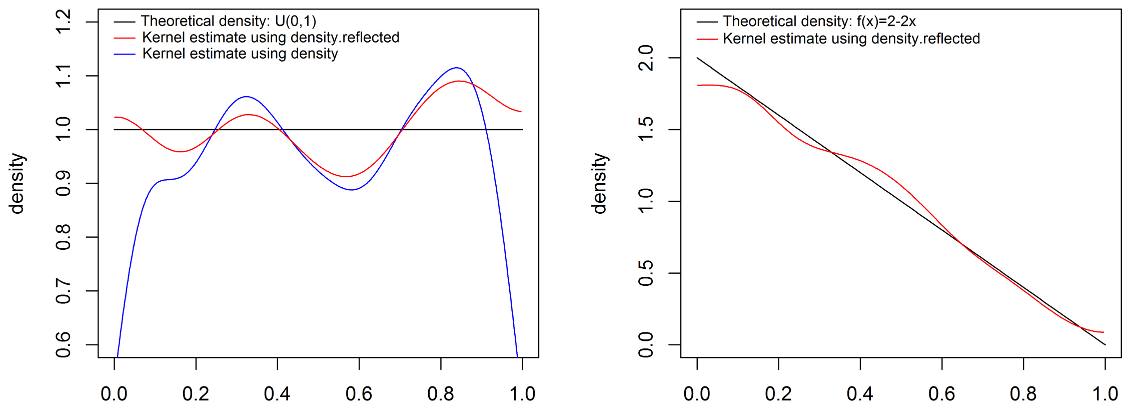

Figure 2: Left panel: Example of differences between using `density` and `density.reflected` in computing a kernel density estimate for a bounded distribution for a $U(0,1)$ sample of size 1000. Right panel: Graphical representation of a sample generated using `random.function`. A kernel density estimate (using `density.reflected`) of a sample of size 300 of the theoretical function $f(x) = 2 - 2x$ for $0 < x < 1$ is displayed in the panel.

- `n`: number of samples to be drawn, default 1.
- `f`: either a density (default) or cumulative distribution function of the random variable to be sampled.
- `lower`: lower end point of the support of the random variable characterized by `f`, default `-Inf`.
- `upper`: upper end point of the support of the random characterized by `f`, default `Inf`.
- `kind`: a character string identifying the kind of function used to identify the distribution, either `"density"` (default) or `"cumulative"`, as alternative.

To exemplify how these functions work and to show their usefulness, Figure 2 displays graphically (i) the different kernel estimates that are obtained using functions `density` and `density.reflected` for a bounded variable (left panel) and (ii) how a sample can be drawn using `random.function` for a random variable even though its random generator is not available in R (right panel). The code used to generate Figure 2 is also provided.

```
R> dev.new(width = 40, height = 15)
R> par(mfcol = c(1, 2), mai = c(0.5, 0.9, .3, .3))
R> set.seed(789)
R> x <- runif(1000)
R> curve(dunif, from = 0, to = 1, ylab = "density", ylim = c(0.6, 1.2),
+    xlab = "")
R> points(density(x, from = 0, to = 1), type = "l", col = "blue")
R> points(density.reflected(x, lower = 0, upper = 1), type = "l",
+    col = "red")
R> segments(0, 1.2, 0.05, 1.2, col = "black")
```

```
R> text(0.28, 1.2, "Theoretical density: U(0,1)", cex = 0.75)
R> segments(0, 1.17, 0.05, 1.17, col = "red")
R> text(0.374, 1.17, "Kernel estimate using density.reflected", cex = 0.75)
R> segments(0, 1.14, 0.05, 1.14, col = "blue")
R> text(0.309, 1.14, "Kernel estimate using density", cex = 0.75)
R> set.seed(942)
R> f0 <- function(x) ifelse(x >= 0 & x <= 1, 2 - 2 * x, 0)
R> curve(f0, from = 0, to = 1, ylab = "density", ylim = c(0,2.25),
+    xlab = "")
R> x <- random.function(300, f0, 0, 1)
R> points(density.reflected(x, lower = 0, upper = 1), type = "l",
+    col = "red")
R> segments(0, 2.25, 0.05, 2.25, col = "black")
R> text(0.30, 2.25, "Theoretical density: f(x)=2-2x", cex = 0.75)
R> segments(0, 2.13, 0.05, 2.13, col = "red")
R> text(0.379, 2.13, "Kernel estimate using density.reflected",
+    cex = 0.75)
```

5. A comparison of the ECDF and EKF approaches

The `fan.test` and `dgeometric.test` functions available in **GoFKernel** offer EKF goodness-of-fit alternatives to the ECDF tests currently available in R (`ks.test` in **stats**, `ad.test` in **ADGofTest**, and `ad.test`, `ad2.test`, `ks.test`, `v.test` and `w2.test` in **truncgof**). In this section, we assess through simulation the calibration (Section 5.1) and sensitivity power (Section 5.2) of all these functions. The speed of the default options of these functions is also studied (Section 5.3). To analyze the size and power of the tests, the significance level is fixed at $\alpha = 0.05$ and the proportion of times that the stated null hypothesis is incorrectly (correctly) rejected is examined for several sample sizes (10, 20, 50, 100, 200 and 500) and random distributions. A thousand samples are simulated for each combination of sample size and random distribution and the above nine tests applied to these simulated samples.

As just presenting the proportion of times the p values are under 0.05 in each scenario[8] can hide relevant issues, the actual p value distributions have been displayed for a sample of scenarios in order to provide further insights and to reinforce the comments that are derived from the chosen summary statistic. In particular, as a consequence of the specifications employed in the simulation exercises, within the calibration scenarios a test is considered to have a good size when its p value distribution is approximately uniform. On the other hand, in sensitivity scenarios the clustering of p values around zero will be an indicator of a proper power of the corresponding test.

5.1. Calibration of the tests

To analyze the calibration of the tests, we study for some instances to what extend the size of the test (the actual proportion of times that a true null hypothesis is rejected) equals the nominal significance level of the test. This is a standard procedure that in the current study

[8]Each combination of a null hypothesis, an actual distribution, and a sample size defines a scenario.

Reference value 0.050		Sample size					
The closer to 0.050 the better		10	20	50	100	200	500
	Test function	Null hypothesis: $U(0,1)$					
Actual distribution $U(0,1)$	`ks.test` {**stats**}	0.045	0.059	0.046	0.043	0.051	0.053
	`ad.test` {**ADGofTest**}	0.053	0.058	0.050	0.055	0.058	0.045
	`ad.test` {**truncgof**}	1.000	1.000	1.000	1.000	1.000	1.000
	`ad2.test` {**truncgof**}	0.000	0.000	0.000	0.000	0.000	0.000
	`ks.test` {**truncgof**}	0.000	0.000	0.000	0.000	0.000	0.000
	`v.test` {**truncgof**}	0.000	0.000	0.000	0.000	0.000	0.000
	`w2.test` {**truncgof**}	0.000	0.000	0.000	0.000	0.000	0.000
	`fan.test`	0.004	0.019	0.006	0.000	0.000	0.000
	`dgeometric.test`	0.046	0.071	0.049	0.058	0.058	0.062
	Test function	Null hypothesis: $N(0,1)$					
Actual distribution $N(0,1)$	`ks.test` {**stats**}	0.038	0.051	0.033	0.046	0.047	0.048
	`ad.test` {**ADGofTest**}	0.037	0.055	0.046	0.042	0.052	0.051
	`ad.test` {**truncgof**}	0.184	0.125	0.078	0.071	0.073	0.051
	`ad2.test` {**truncgof**}	0.482	0.505	0.482	0.502	0.471	0.433
	`ks.test` {**truncgof**}	0.360	0.379	0.373	0.364	0.329	0.305
	`v.test` {**truncgof**}	0.156	0.157	0.128	0.150	0.154	0.140
	`w2.test` {**truncgof**}	0.443	0.465	0.438	0.438	0.422	0.399
	`fan.test`	0.016	0.027	0.014	0.016	0.004	0.000
	`dgeometric.test`	0.065	0.076	0.068	0.063	0.072	0.063
	Test function	Null hypothesis: $Exp(1)$					
Actual distribution $Exp(1)$	`ks.test` {**stats**}	0.041	0.052	0.047	0.038	0.053	0.053
	`ad.test` {**ADGofTest**}	0.035	0.054	0.049	0.042	0.056	0.049
	`ad.test` {**truncgof**}	0.071	0.082	0.054	0.052	0.054	0.049
	`ad2.test` {**truncgof**}	0.212	0.221	0.217	0.226	0.221	0.233
	`ks.test` {**truncgof**}	0.173	0.172	0.175	0.187	0.177	0.182
	`v.test` {**truncgof**}	0.064	0.079	0.065	0.082	0.079	0.065
	`w2.test` {**truncgof**}	0.239	0.222	0.198	0.244	0.232	0.229
	`fan.test`	0.013	0.008	0.001	0.000	0.000	0.000
	`dgeometric.test`	0.055	0.070	0.073	0.074	0.069	0.065
	Test function	Null hypothesis: $logN(1,1)$					
Actual distribution $logN(1,1)$	`ks.test` {**stats**}	0.033	0.056	0.051	0.045	0.043	0.063
	`ad.test` {**ADGofTest**}	0.037	0.050	0.059	0.055	0.050	0.062
	`ad.test` {**truncgof**}	0.201	0.132	0.092	0.073	0.063	0.058
	`ad2.test` {**truncgof**}	0.477	0.497	0.478	0.498	0.487	0.466
	`ks.test` {**truncgof**}	0.348	0.370	0.358	0.355	0.337	0.330
	`v.test` {**truncgof**}	0.162	0.153	0.162	0.141	0.134	0.148
	`w2.test` {**truncgof**}	0.443	0.447	0.456	0.442	0.431	0.437
	`fan.test`	0.012	0.006	0.005	0.001	0.000	0.000
	`dgeometric.test`	0.085	0.073	0.074	0.052	0.065	0.064

Table 1: Proportion of times the true null hypothesis is rejected at a significance level of 0.05. Elaborated using R version 3.1.0 (R Core Team 2015), **ADGofTest** 0.3 (Bellosta 2011), **GoFKernel** 2.0-3 (Pavía 2015), **KernSmooth** 2.23-12 (Wand 2013), and **truncgof** 0.6-0 (Wolter 2012).

has been performed in scenarios when the samples come from (i) a uniform distribution on the interval $[0,1]$, $U(0,1)$, (ii) a standard normal distribution, $N(0,1)$, (iii) an exponential distribution of rate (and mean) equal to 1, $Exp(1)$, and (iv) a logNormal distribution whose logarithm has mean equal to 1 and standard deviation equal to 1, $logN(1,1)$.

Focusing on the outputs of Table 1[9] the first issue that comes to our attention is the bad behavior of **truncgof** functions as a whole. Indeed, analyzing the p values attained for the $U(0,1)$ scenarios (see also Figure 3, where the default `density.reflected` kernel density estimates of p values for a group of $U(0,1)$ scenarios are presented), it seems that these functions are not useful for bounded variables. The **truncgof** package produces (almost) systematically p values equal to 1 for three of its functions (`ks.test` from package **truncgof**, `v.test` and `w2.test`) and really clustered distributions, around 0 and 1, for, respectively, `ad.test` and `ad2.test` functions from package **truncgof**. In the remaining scenarios the picture for the **truncgof** functions is also quite demoralizing, their frequency in rejecting the null hypotheses is clearly above the expected figure of 0.05. The only function of this package that shows reasonable figures for several of the (unbounded) variables is `ad.test` from package **truncgof**, which registers acceptable rejecting rates for large samples in $N(0,1)$ and $logN(1,1)$ scenarios and adequate rates in the $Exp(1)$ scenarios. In $Exp(1)$ scenarios, there is another function of the package, `v.test`, that also shows acceptable figures. The same conclusions can be extracted observing Figures 4, 5 and 6, where the default `density.reflected` kernel estimates of the p value distributions have been presented for a sample of scenarios.

The above results for the **truncgof** package functions are really unexpected. Although the **truncgof** package (Wolter 2012) is intended to deal with the issue for left truncated distributions, looking at the theoretical paper that supports this package (Chernobai *et al.* 2005) there are no apparent reasons for this weird behavior. Indeed, except for a constant factor (which is a function of the number of observations, n, of the sample to be tested), the statistics that are (should be) implemented in the **truncgof** collapse to the non-truncated ones introduced in Equations 1–4 when the threshold is kept at its default option.[10]

For the rest of the functions, the results highlight that the `fan.test` function clearly has an over-tendency to accept the null hypothesis, an unambiguous sign of miscalibration of the test (at least in its default options). This miscalibration is even exacerbated in the scenarios with right-asymmetric distributions (see Figures 3 to 6). Finally, as results in Table 1 show and pictures in Figures 3 to 6 confirm, the outcomes for the rest of the functions are adequate. As a rule, nevertheless, it could be said that the `ks.test` function from package **stats** and the `ad.test` function from package **ADGofTest** show a test size slightly smaller than expected and that `dgeometric.test` shows a size slightly greater.

[9]The numbers in the table must be observed as approximations to the actual sizes of the tests. Obviously, different figures would have been obtained with a different set of simulated samples. As a reference, and by comparison with a different set of simulations (not presented here), a variation as large as ± 0.02 in the estimation of the size of a test might be considered as non-unusual. Hence, those tests whose values in the tables vary within the range $[.03, .07]$ could be observed, in general, as well-calibrated.

[10]In its default option, the **truncgof** tests set the threshold in `-Inf`. (Although this is not documented in the help file of the package, it can be easily observed looking inside the functions.) This issue entails the estimated cumulative distribution function to be zero in the threshold and, consequently, the stated equivalence, see Tables 1 and 2 in Chernobai *et al.* (2005, pp. 20–21).

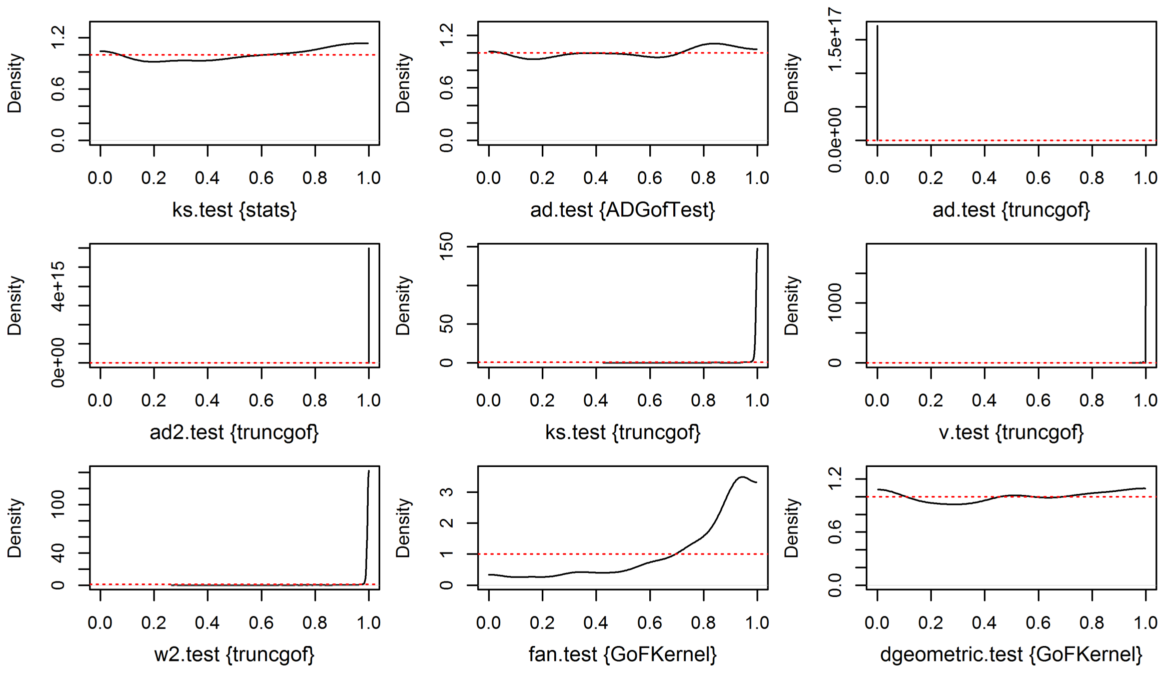

Figure 3: Kernel estimates of the p values distributions for different hypothesis tests for the scenario defined by a sample size of 20 and $U(0, 1)$ for both null hypothesis and actual distribution. Reference distribution marked with a red dashed line.

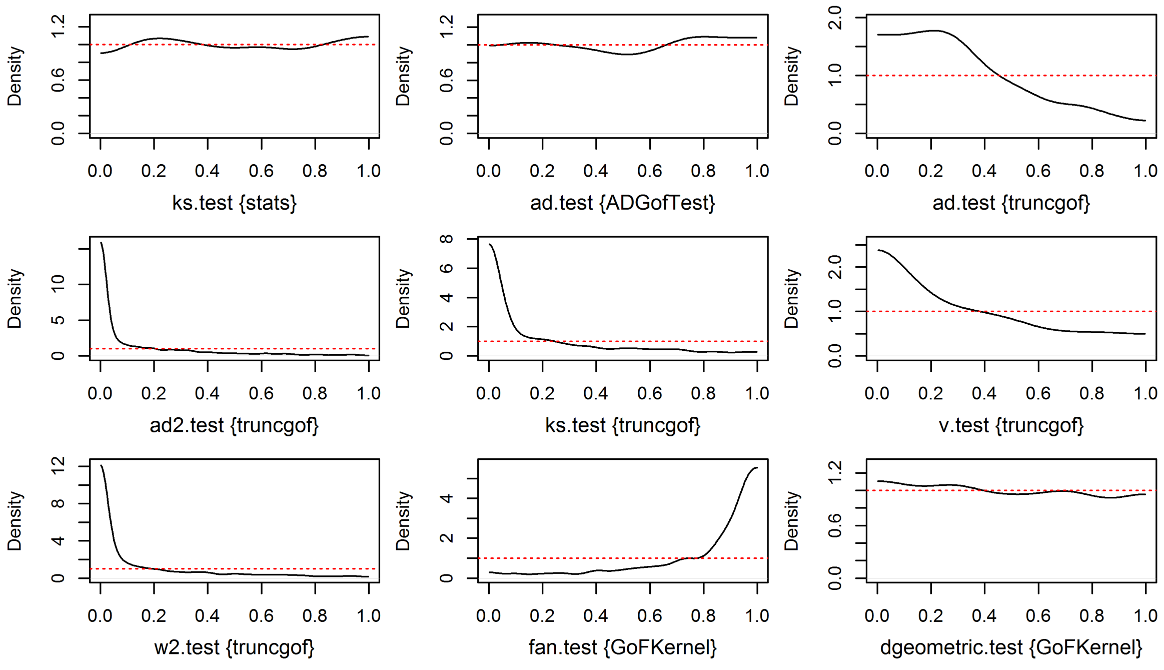

Figure 4: Kernel estimates of the p values distributions for different hypothesis tests for the scenario defined by a sample size of 50 and $N(0, 1)$ for both null hypothesis and actual distribution. Reference distribution marked with a red dashed line.

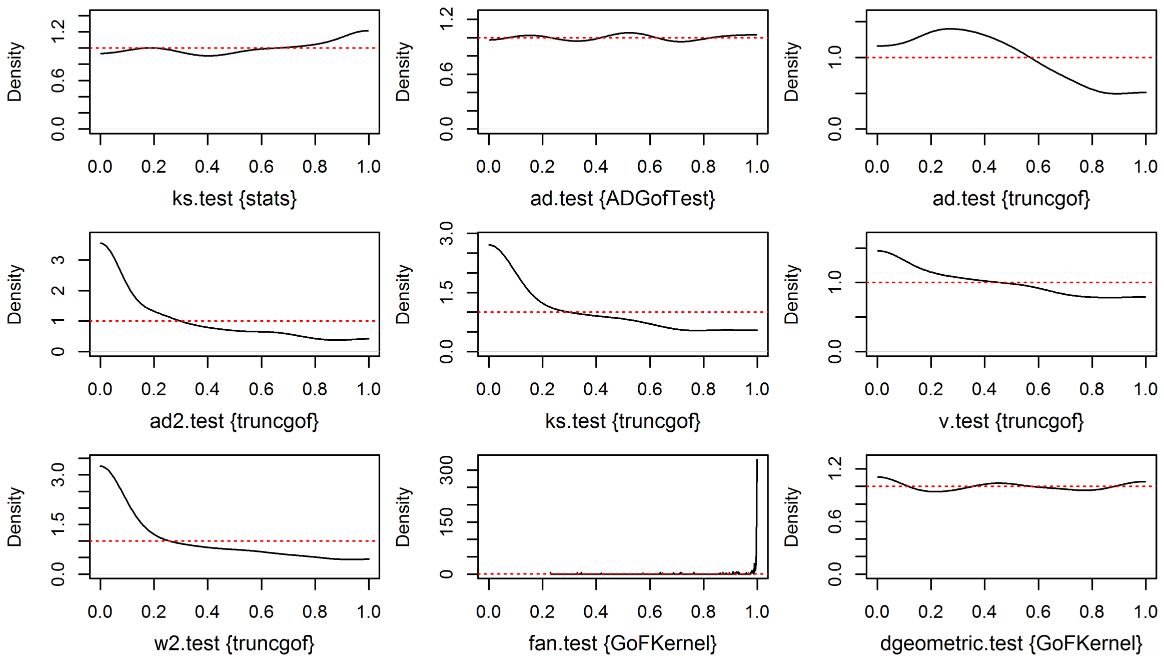

Figure 5: Kernel estimates of the p values distributions for different hypothesis tests for the scenario defined by a sample size of 100 and $Exp(1)$ for both null hypothesis and actual distribution. Reference distribution marked with a red dashed line.

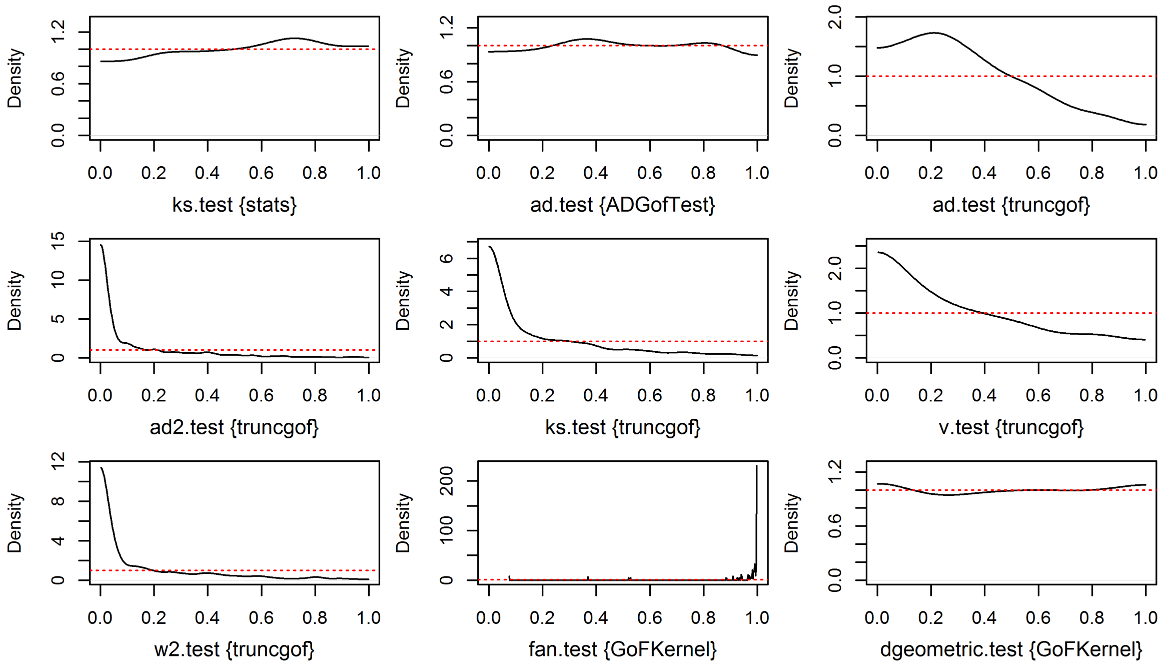

Figure 6: Kernel estimates of the p values distributions for different hypothesis tests for the scenario defined by a sample size of 200 and $logN(1, 1)$ for both null hypothesis and actual distribution. Reference distribution marked with a red dashed line.

5.2. Sensitivity of the tests

The power of a statistical test is the probability that the test rejects the null hypothesis when the null hypothesis is indeed false. To study how sensitive the rejection rules of the tests are to false null hypotheses, we study some scenarios where their discriminant power is challenged. That is, we consider situations in which the random distributions stated in the null hypotheses are chosen to be close (in shape) to the actual distributions generating the data. More specifically, we consider couples where the actual and null hypothesis densities are, respectively: (i) a Beta with both shape parameters equal to 1.3, $Be(1.3, 1.3)$, and a $[0, 1]$ uniform distribution, $U(0, 1)$; (ii) a standard Cauchy[11], $Ca(0, 1)$, and a $N(0, \widehat{\sigma})$, where $\widehat{\sigma}$ is the standard deviation of the sample, (iii) a Gamma with both shape and rate equal to 0.9, $Ga(0, 9, 0.9)$ and a $Exp(1)$, and (iv) a $logN(1, 1)$ and a $Ga(\widehat{\alpha}, \widehat{\beta})$, where $\widehat{\alpha}$ and $\widehat{\beta}$ are respectively the shape and rate MLE.

The summary outcomes of the sensitivity analysis are presented in Table 2. Focusing firstly on the $Be(1.3, 1.3) - U(0, 1)$ scenarios, the non-usefulness of **truncgof** tests for bounded random variables is confirmed (see also Figure 7). In line with previous results, they produce systematically extreme p values irrespectively of the sample. Regarding the rest of the tests, we observe (both in Table 2 and in Figure 7) the Fan test showing again the really conservative behavior demonstrated in the calibration analysis. This conservative behavior also occurs for small-size samples in the remaining tests, although at a lower level. These three tests are not equivalent, however. The `ad.test` function from package **ADGofTest** looks preferable to `ks.test` from package **stats**, and `dgeometric.test` is the function showing the higher power in this case, although its figures for really small sample sizes are indeed modest.

The $Ca(0, 1) - N(0, \widehat{\sigma})$ scenarios are the ones with the greatest differences between the actual and null hypothesis distributions. This is reflected in the rejection rates of all the tests that show really high figures in all the cases, except for the smallest sample sizes. The `ks.test` from package **stats** and `ad.test` from package **ADGofTest** functions are the more reluctant to correctly reject the null hypothesis with small sizes, whereas the **truncgof** functions seem to have the greatest powers now (see Table 2 and Figure 8). This superiority of **truncgof** functions is, however, more apparent than real, observing $N(0, 1)$ outcomes in Table 1. It highlights also the amount of simulations for which no solution is offered by some functions of, mainly, the **truncgof** package. The functions `ad.test` and `ad2.test` from package **truncgof** produced an error 221, 546 and 635 times for, respectively, samples of sizes 100, 200 and 500. Likewise, the `fan.test` function generated a total of seven errors.[12]

Focusing now on the $Ga(0.9, 0.9) - Exp(1)$ scenarios we observe that in general all the tests experience great difficulties in discriminating between the actual distribution generating the sample and the null hypothesis distribution. This is especially true if we take into account (and discount) the results obtained in Section 5.1 for $Exp(1)$ scenarios (see Table 1 and Figure 5) for some of the **truncgof** functions (`ad2.test`, `ks.test` and `w2.test`), which show a clear over-tendency to reject the real $Exp(1)$ null hypothesis. EKF tests do not offer this time an alternative to be considered. In this case the EKF tests are the ones showing the lower powers. In practice, `fan.test` is unable to discriminate between both models and `dgeometric.test` is, unlike the rest of the functions, not able to significantly improve their power in the largest

[11] A Cauchy distribution with location parameter equal to 0 and scale parameter equal to 1.

[12] It could be tracked that the errors registered with the `fan.test` function had their origin in the computation of the integral of the convolution of the density in the sampled values. This computation is made using the `integrate` function.

Reference value 1.000 The closer to 1.000 the better		Sample size 10	 20	 50	 100	 200	 500
	Test function	Null hypothesis: $U(0,1)$					
Actual distribution $Be(1.3,1.3)$	`ks.test` {**stats**}	0.028	0.043	0.057	0.070	0.141	0.457
	`ad.test` {**ADGofTest**}	0.018	0.025	0.043	0.091	0.259	0.835
	`ad.test` {**truncgof**}	1.000	1.000	1.000	1.000	1.000	1.000
	`ad2.test` {**truncgof**}	0.000	0.000	0.000	0.000	0.000	0.000
	`ks.test` {**truncgof**}	0.000	0.000	0.000	0.000	0.000	0.000
	`v.test` {**truncgof**}	0.000	0.000	0.000	0.000	0.000	0.000
	`w2.test` {**truncgof**}	0.000	0.000	0.000	0.000	0.000	0.000
	`fan.test`	0.008	0.021	0.013	0.029	0.019	0.023
	`dgeometric.test`	0.046	0.069	0.131	0.243	0.478	0.923
	Test function	Null hypothesis: $N(0,\widehat{\sigma})$					
Actual distribution $Ca(0,1)$	`ks.test` {**stats**}	0.147	0.513	0.947	0.997	1.000	1.000
	`ad.test` {**ADGofTest**}	0.107	0.502	0.955	0.999	1.000	1.000
	`ad.test` {**truncgof**} (1)	0.646	0.844	0.988	*0.997*	*1.000*	*1.000*
	`ad2.test` {**truncgof**} (1)	0.747	0.902	0.999	*1.000*	*1.000*	*1.000*
	`ks.test` {**truncgof**}	0.665	0.870	0.998	1.000	1.000	1.000
	`v.test` {**truncgof**}	0.553	0.838	0.996	1.000	1.000	1.000
	`w2.test` {**truncgof**}	0.720	0.894	0.998	1.000	1.000	1.000
	`fan.test` {**GoFKernel**} (2)	*0.285*	*0.703*	0.988	*1.000*	1.000	1.000
	`dgeometric.test`	0.253	0.704	0.995	1.000	1.000	1.000
	Test function	Null hypothesis: $Exp(1)$					
Actual distribution $Ga(0.9,0.9)$	`ks.test` {**stats**}	0.058	0.063	0.068	0.076	0.086	0.162
	`ad.test` {**ADGofTest**}	0.058	0.069	0.087	0.108	0.113	0.236
	`ad.test` {**truncgof**}	0.145	0.122	0.111	0.124	0.119	0.122
	`ad2.test` {**truncgof**}	0.257	0.285	0.281	0.349	0.369	0.567
	`ks.test` {**truncgof**}	0.186	0.208	0.203	0.234	0.245	0.403
	`v.test` {**truncgof**}	0.085	0.098	0.111	0.131	0.159	0.281
	`w2.test` {**truncgof**}	0.240	0.261	0.246	0.303	0.324	0.471
	`fan.test`	0.009	0.010	0.001	0.000	0.000	0.000
	`dgeometric.test`	0.056	0.060	0.061	0.042	0.032	0.046
	Test function	Null hypothesis: $Ga(\widehat{\alpha},\widehat{\beta})$					
Actual distribution $logN(1,1)$	`ks.test` {**stats**}	0.002	0.014	0.046	0.147	0.464	0.980
	`ad.test` {**ADGofTest**}	0.000	0.003	0.023	0.141	0.527	0.997
	`ad.test` {**truncgof**} (3)	0.113	0.196	0.375	0.533	0.735	*0.934*
	`ad2.test` {**truncgof**} (3)	0.101	0.235	0.504	0.816	0.989	*1.000*
	`ks.test` {**truncgof**}	0.099	0.190	0.381	0.662	0.933	0.999
	`v.test` {**truncgof**}	0.080	0.174	0.340	0.616	0.927	1.000
	`w2.test` {**truncgof**}	0.098	0.214	0.465	0.786	0.984	1.000
	`fan.test`	0.005	0.043	0.118	0.143	0.182	0.469
	`dgeometric.test`	0.021	0.042	0.104	0.220	0.489	0.949

Table 2: Proportion of times the false null hypothesis is rejected at a significance level of 0.05. Elaborated using R version 3.1.0 (R Core Team 2015), **ADGofTest** 0.3 (Bellosta 2011), **GoFKernel** 2.0-3 (Pavía 2015), **KernSmooth** 2.23-10 (Wand 2013), **MASS** 7.3-31 (Venables and Ripley 2002), and **truncgof** 0.6-0 (Wolter 2012). (1) An error is produced 221, 446 and 635 times for, respectively, samples of sizes 100, 200 and 500. (2) This function produced an error 4, 2 and 1 times for, respectively, samples of sizes 10, 20 and 100. (3) These functions produced an error 11 times for samples of size 500.

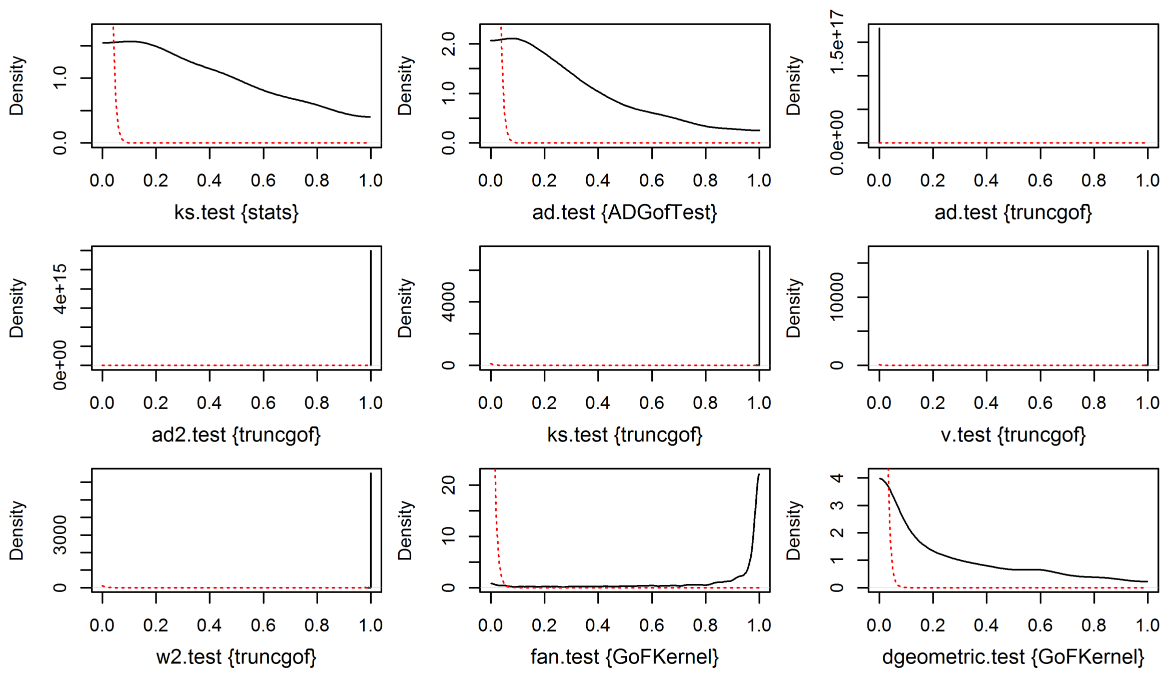

Figure 7: Kernel estimates of the p values distributions for different hypothesis tests for the scenario defined by a $U(0, 1)$ as null hypothesis, a $Be(1.3, 1.3)$ as actual distribution and a sample size of 100. Reference distribution marked with a red dashed line.

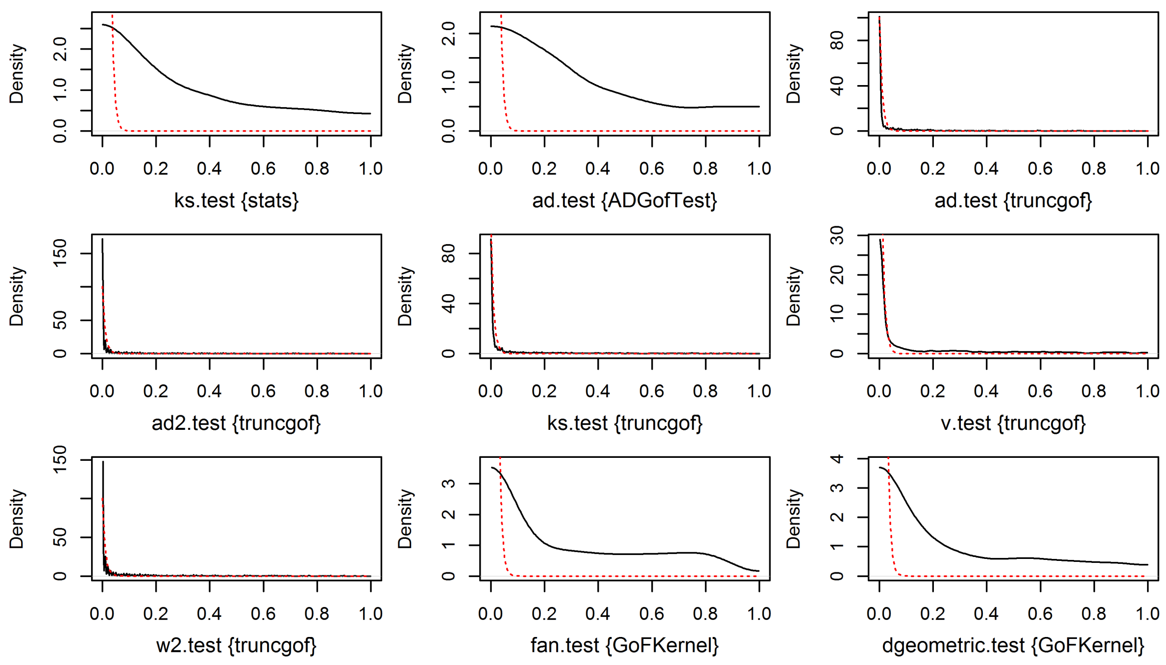

Figure 8: Kernel estimates of the p values distributions for different hypothesis tests for the scenario defined by a $N(0, \widehat{\sigma})$ as null hypothesis, a $Ca(0, 1)$ as actual distribution and a sample size of 10. Reference distribution marked with a red dashed line.

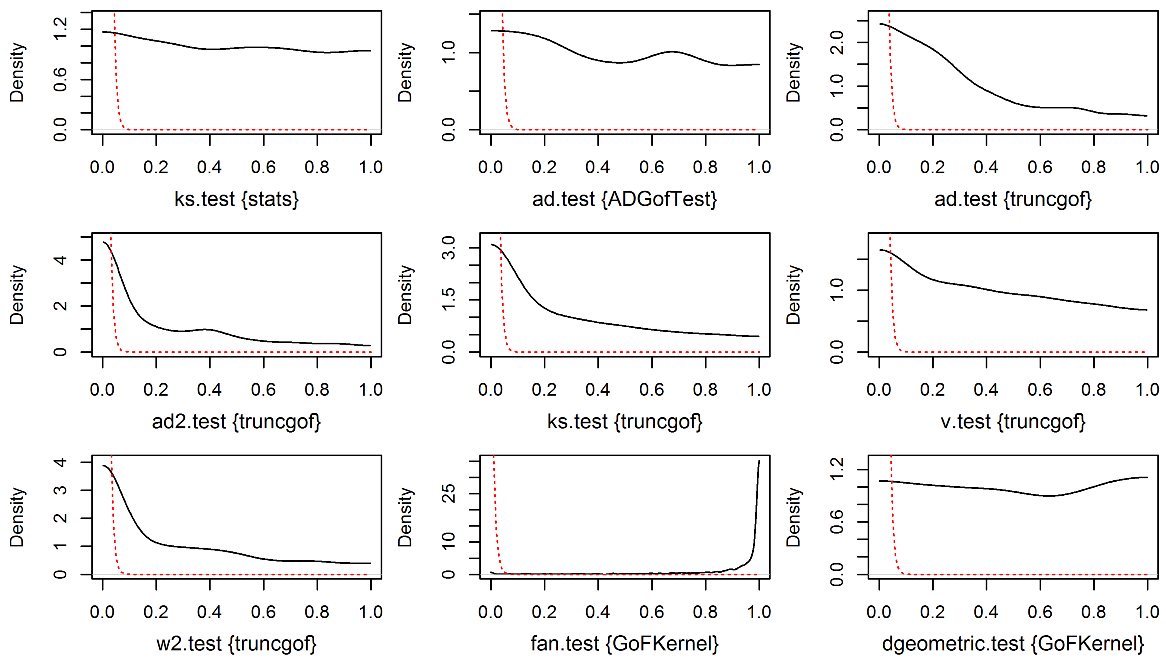

Figure 9: Kernel estimates of the p values distributions for different hypothesis tests for the scenario defined by a $Exp(1)$ as null hypothesis, a $Ga(0.9, 0.9)$ as actual distribution and a sample size of 20. Reference distribution marked with a red dashed line.

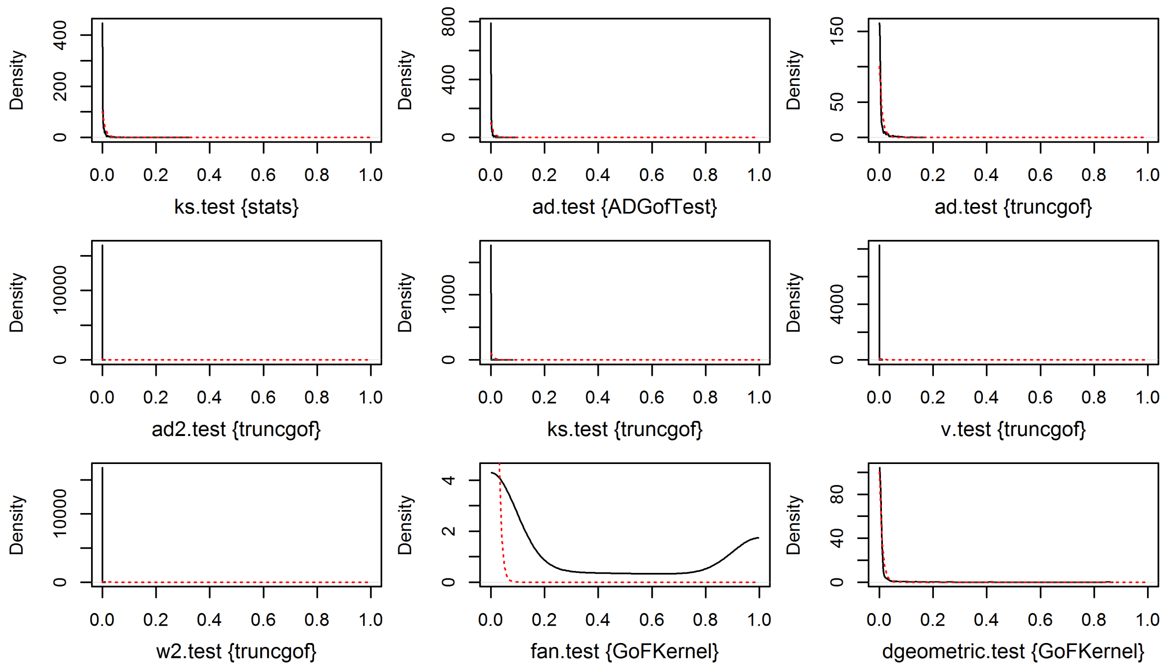

Figure 10: Kernel estimates of the p values distributions for different hypothesis tests for the scenario defined by a $logN(1, 1)$ as null hypothesis, a $Ga(\widehat{\alpha}, \widehat{\beta})$ as actual distribution and a sample size of 500. Reference distribution marked with a red dashed line.

sample size case.

The picture for the $logN(1,1)-Ga(\widehat{\alpha},\widehat{\beta})$ scenarios serves to reinforce the conclusions reached after analyzing $Be(1.3,1.3)-U(0,1)$ scenarios and $Ca(0,1)-N(0,\widehat{\sigma})$ scenarios. Discounting again for the over-tendency of **truncgof** functions to reject the true null hypothesis, massively present in $logN(1,1)$ scenarios (see Table 1), it becomes evident that the superior power that **truncgof** tests exhibit in small samples is a matter of illusion. The possibilities that the Gamma distribution offers to mimic logNormal models is reflected in the lower powers that all the tests show in small samples, which nevertheless grow significantly as sample sizes increase. They all need a relatively large number of observations to properly reject the null hypothesis. The worst power is registered by the `fan.test` function, without doubt a consequence of its over-tendency to accept null hypotheses. On the other hand, regarding the outcomes for `ks.test` from package **stats**, `ad.test` from package **ADGofTest** and `dgeometric.test`, we observe that the three functions show similar power, with `dgeometric.test` presenting nonetheless greater power in medium-size samples. Finally, in these scenarios, it also highlights the fact that some of the functions, (`ad.test` and `ad2.test` from package **truncgof**), experience an error in some of the simulations.[13]

5.3. Speeds of the tests

Although the speed of the studied functions is not of concern when just a bunch of tests must be performed[14], the computational burden could become an issue when we are interested in carrying out hundreds of thousands (like in our simulation exercise) or millions of tests. So, this section is devoted to analyzing the time spent by the different functions during the simulation exercise.

The first issue that clearly emerges in this analysis is that the analyzed functions, when used with their default options, can be clustered in four groups in terms of velocity. A group of `ks.test` from package **stats** and `ad.test` from package **ADGofTest** (so called `sup-tests`), a group with all the **truncgof** functions (`truncgof-tests`) and two more groups, each one with just a function: `dgeometric.test` and `fan.test`. The group with the fastest functions is `sup-tests`, whereas the **truncgof** functions are as a rule the slowest. The `fan.test` and `dgeometric.test` are in an intermediate position, with `dgeometric.test` faster with large sample sizes and slower with small sample sizes. In particular, considering all the scenarios as a whole, the time spent per sample has been on average 0.0015, 0.0714, 0.3027 and 0.5392 secs for, respectively, `sup-tests`, `dgeometric.test`, `fan.test` and `truncgof-tests` functions.[15]

In an analysis by scenarios (see Figure 11), the first result that stands out is the regularity that speeds of `sup-tests` and `dgeometric.test` functions show, not significantly varying among scenarios and sample sizes[16]; when, on the contrary, `truncgof-tests` and `fan.test` functions' velocities vary both across scenarios and sample sizes.

[13] As happened with the errors recorded in $Ca(0,1)-N(0,\widehat{\sigma})$ scenarios for the **truncgof** functions, the problem seems to be in the internal function `mctest` of the **truncgof** package.

[14] In our simulation exercise, none of the tests needed more than a few seconds to produce its output and the vast majority of them spent much less than a second.

[15] These numbers have been obtained after running our simulations in a I7 3,4GHz 6 cores processor computer, with 16GB (DDR3) of RAM memory and a SSD 250GB hard disk, using R version 3.1.0 (2014-04-10) (R Core Team 2015) for Windows (platform x86_64-w64-mingw32/x64 (64-bit)) and, as packages, **ADGofTest** 0.3 (Bellosta 2011), **GoFKernel** 2.0-3 (Pavía 2015), **KernSmooth** 2.23-12 (Wand 2013), **MASS** 7.3-31 (Venables and Ripley 2002), and **truncgof** 0.6-0 (Wolter 2012).

[16] The only small exceptions to this rule among the scenarios studied (not observable in Figure 11 as a

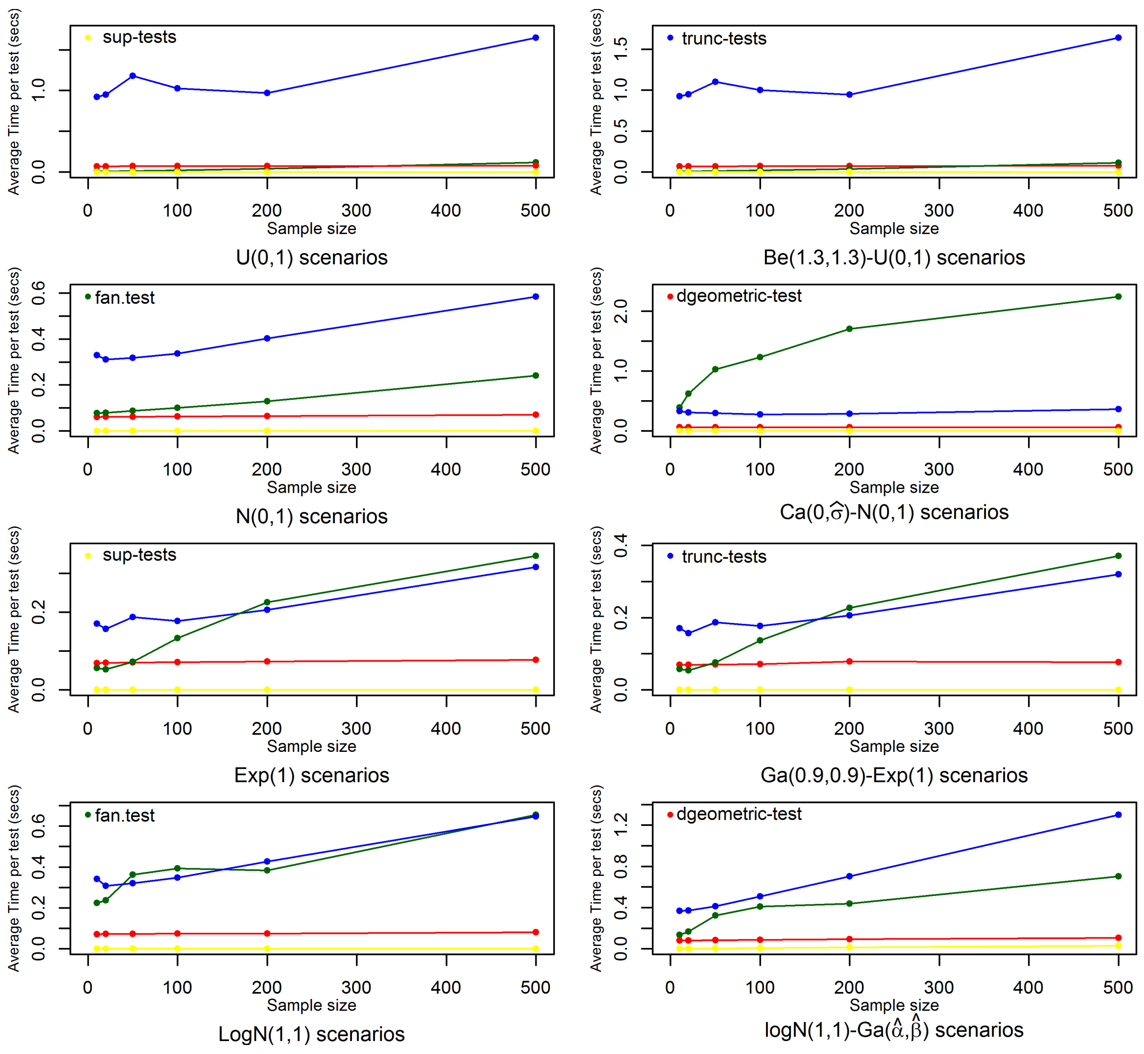

Figure 11: Average time spent per sample, in seconds, for `sup-group` (`ks.test` {**stats**} and `ad.test` {**ADGofTest**}), `fan.test`, `truncgof-group` (`ad.test` {**truncgof**}, `ad2.test`, `ks.test` {**truncgof**}, `v.test` and `w2.test`), and `dgeometric.test` functions (employed with their default options). For more details, see Footnote 15.

Focusing on the `truncgof-tests` and `fan.test` groups of functions, Figure 11 highlights the great similarities that their speed curves show between scenarios with the same null hypothesis. The only exception to this rule seems to be in the $Ca(0,1) - N(0,\widehat{\sigma})$ case, where the `fan.test` function needs a significant extra-time to properly deal with the Cauchy samples. Indeed, with the exception of this case, conditioned to the sample size, the time spent

matter of scale) occur in $logN(1,1) - Ga(\widehat{\alpha},\widehat{\beta})$ scenarios, where the time spent for the `sup-tests` functions and for the `dgeometric.test` function grows with sample size. In the latter case, however, the relatively slight increases recorded for the `dgeometric.test` function suggest that the reason behind this behavior is more in the MLE routines than in the test itself. Finally, it should also be noted that the average time spent per test by `sup-tests` functions in $logN(1,1) - Ga(\widehat{\alpha},\widehat{\beta})$ scenarios also grows with regards to the other scenarios. For example, it grows from an average of 0.0004 (0.0002) secs per test to an average of 0.0291 (0.0022) secs for the largest (smallest) sample size. Anyway, these tests still remain by large the fastest.

per test by the `fan.test` function does not vary that much across scenarios. The behavior of the `truncgof-tests` functions is, nevertheless, more volatile. They show variability both across scenarios and sample sizes.

6. On convergence of `dgeometric.test`

The `dgeometric.test` function uses `n.sim = 101` as default option to approximate p values. The convergence of the estimation algorithm is, however, asymptotic. So, the question of whether 101 simulated samples are enough as to obtain a relative accurate approximation of p values emerges in a natural way.

To try to give an answer to this question, the *actual* p values corresponding to the samples generated in our simulation exercise have also been computed[17] and compared to the ones obtained with the default option.

A summary of the comparisons made between approximate and *actual* p values is offered in Table 3. In particular, in each scenario, Table 3 (i) counts the proportion of times that the default option approximation offers the *correct* solution at the usual significant levels ($\alpha = 0.1$, $\alpha = 0.05$ and $\alpha = 0.01$), (ii) the correlation between *actual* and approximate p values and (iii) the mean and standard deviation of the absolute differences between *actual* and approximate p values.

Looking at the numbers in Table 3, it could be argued that the outcomes that produces the `dgeometric.test` function with default options can be considered adequate. The differences observed, both in absolute values and in terms of the proportion of times that the `dgeometric.test` function with default options does not hold the correct decision, are indeed within the limits of the uncertainty linked with the process. For example, with a significant level of $\alpha = 0.05$, the percentage of decision coincidences in calibration scenarios is as large as 98 per cent, when as we noted in Section 5.1 the volatility in the estimation of the p values is as large as ± 0.02, even for well-established test like the KS test (`ks.test` from package **stats**). Some deviations like the ones observed are therefore reasonable.

As a final note of interest, it is worth mentioning that as expected when the null hypothesis is clearly rejected by the data (as happened in the $Ca(0,1) - N(0,\widehat{\sigma})$ scenarios) the estimated p values are less sensitive to the number of simulated samples employed.

7. Conclusions

This paper introduces a new solution for the non-parametric goodness-of-fit test based on the test statistic that measures the distance in absolute value (i.e., the area) between an empirical kernel estimate of the observations and a null hypothesis (one-dimensional continuous) density function. This test in addition to the Fan's test (Fan 1994) has been implemented in the **GoFKernel** package, which is presented in this paper. A comparative analysis with the other general non-parametric goodness-of-fit tests currently available in R reveals that: (i) the tests implemented in the **truncgof** package are non-usefulness for bounded variables, presenting moreover in unbounded variables a clear over-tendency to reject the null hypothesis; and

[17]To do that, `n.sim = 1000` has been used in the `dgeometric.function` and the attained p values considered as reference p values.

Scenarios	Sample size	Proportion of coincidences at sig. level, $\alpha =$			Correlation	Differences in absolute values		Scenarios	Sample size	Proportion of coincidences at sig. level, $\alpha =$			Correlation	Differences in absolute values	
		0.1	0.05	0.01		μ	σ			0.1	0.05	0.01		μ	σ
$U(0,1)$ Scenarios	10	0.973	0.991	0.993	0.988	0.034	0.028	$Be(1.3,1.3)$ Scenarios	10	0.967	0.970	0.993	0.989	0.033	0.026
	20	0.981	0.980	0.979	0.990	0.033	0.027		20	0.954	0.979	0.987	0.989	0.033	0.028
	50	0.966	0.988	0.983	0.989	0.033	0.027		50	0.960	0.963	0.968	0.989	0.032	0.027
	100	0.969	0.976	0.988	0.989	0.034	0.028		100	0.946	0.956	0.938	0.988	0.029	0.028
	200	0.977	0.981	0.984	0.991	0.031	0.026		200	0.945	0.920	0.901	0.988	0.021	0.023
	500	0.971	0.976	0.991	0.988	0.034	0.027		500	0.985	0.978	0.910	0.969	0.006	0.010
$N(0,1)$ Scenarios	10	0.977	0.979	0.987	0.990	0.032	0.027	$Ca(0,1)$ Scenarios	10	0.941	0.942	0.950	0.992	0.027	0.025
	20	0.966	0.977	0.983	0.989	0.033	0.027		20	0.983	0.965	0.912	0.995	0.013	0.020
	50	0.977	0.971	0.986	0.989	0.032	0.027		50	0.999	1.000	0.992	0.990	0.000	0.004
	100	0.978	0.984	0.988	0.989	0.032	0.028		100	1.000	1.000	1.000	0.949	0.000	0.000
	200	0.967	0.978	0.987	0.990	0.032	0.028		200	1.000	1.000	1.000	1.000	0.000	0.000
	500	0.967	0.983	0.984	0.990	0.032	0.027		500	1.000	1.000	1.000	1.000	0.000	0.000
$Exp(1)$ Scenarios	10	0.964	0.984	0.989	0.990	0.032	0.028	$Ga(0.9,0.9)$ Scenarios	10	0.974	0.982	0.988	0.990	0.033	0.027
	20	0.971	0.981	0.986	0.991	0.031	0.027		20	0.964	0.986	0.988	0.990	0.032	0.026
	50	0.972	0.980	0.986	0.990	0.032	0.028		50	0.975	0.980	0.982	0.991	0.031	0.027
	100	0.974	0.980	0.986	0.990	0.031	0.027		100	0.973	0.975	0.991	0.991	0.031	0.026
	200	0.972	0.980	0.989	0.988	0.034	0.029		200	0.979	0.988	0.993	0.988	0.034	0.028
	500	0.976	0.977	0.985	0.990	0.032	0.027		500	0.984	0.984	0.995	0.990	0.032	0.026
$logN(1,1)$ Scenarios	10	0.995	0.997	0.991	0.984	0.034	0.028	$Ga(\widehat{\alpha},\widehat{\beta})$ Scenarios	10	0.979	0.988	0.993	0.989	0.036	0.028
	20	0.978	0.975	0.983	0.989	0.033	0.027		20	0.983	0.983	0.992	0.984	0.033	0.028
	50	0.977	0.980	0.981	0.990	0.032	0.026		50	0.973	0.980	0.990	0.991	0.032	0.027
	100	0.966	0.984	0.985	0.990	0.031	0.027		100	0.960	0.955	0.955	0.992	0.029	0.026
	200	0.976	0.975	0.985	0.990	0.033	0.027		200	0.947	0.928	0.910	0.988	0.021	0.024
	500	0.977	0.986	0.985	0.989	0.033	0.029		500	0.992	0.982	0.931	0.973	0.004	0.009

Table 3: Differences in performance of `dgeometric.test` function with default options and with `n.sim = 1000`. Elaborated using R version 3.1.0 (R Core Team 2015), **GoFKernel** 2.0-3 (Pavía 2015) and **MASS** 7.3-31 (Venables and Ripley 2002).

that, on the contrary, (ii) the `fan.test` function shows an over-tendency to accept the null hypothesis, at least in its default option (where the bandwidth is computed using the `dpik` function of **KernSmooth** package). The analysis also show that (iii) the tests implemented in `ks.test` in package **stats**, `ad.test` in package **ADGofTest** and `dgeometric.test` functions represent the best alternatives, with the last function showing more frequently superior power in samples of medium and large sizes, although at the cost of a higher computational burden (see Section 5.3).

In summary, taking aside computational costs and combining the statistical conclusions of Sections 5.1 and 5.2, it seems that: (i) to test uniformity, the `dgeometric.test` function offers the preferable test for samples of small and medium size, with `ks.test` from package **stats** and `ad.test` from package **ADGofTest** being competitive as soon as the sample size grows; (ii) to test normality, all the three recommended tests offer good performance, with the `dgeometric.test` function being superior in samples of small size; (iii) to test goodness-of-fit to an exponential distribution, no solution is adequate for small sample sizes, with both versions of the *AD* test (`ad.test` from package **ADGofTest** and `ad.test` from package **truncgof**) improving their numbers starting from samples of medium size; and (iv) to test goodness-of-fit to a logNormal distribution, the `dgeometric.test` function is slightly preferable to `ks.test` from package **stats** and `ad.test` from package **ADGofTest**, with no test offering good discriminant power between the logNormal distribution and the Gamma distribution in samples of small size.

The **GoFKernel** package is of course incomplete and as part of future work it should grow incorporating other EKF based tests and/or improving the flexibility of their methods.

Acknowledgments

The author wish to thank Marie Hodkinson for revising the English of the paper and Carlos Ivorra for his help with LaTeX. Acknowledgments must also be extended to two anonymous referees and the editors for their really valuable comments and recommendations. They have significantly improved both the paper and the package, suggesting ideas to speed up its algorithms and to make it more flexible. The support of the Spanish Ministry of Economics and Competitiveness through the project CSO2013-43054-R is also acknowledged. The usual disclaimers apply.

References

Ahmad IA, Cerrito PB (1993). "Goodness of Fit Tests Based on the L_2-Norm of Multivariate Probability Density Functions." *Journal of Nonparametric Statistics*, **2**(2), 169–181.

Albers CJ, Schaafsma W (2008). "Goodness of Fit Testing Using a Specific Density Estimate." *Statistics & Decisions*, **26**(1), 3–23.

Anderson NH, Hall P, Titterington DM (1994). "Two-Sample Test Statistics for Measuring Discrepancy between Two Multivariate Probability Density Functions Using Kernel-Based Density Estimates." *Journal of Multivariate Analysis*, **50**, 41–54.

Anderson TW, Darling DA (1952). "Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes." *The Annals of Mathematical Statistics*, **23**, 193–212.

Bellosta CJG (2011). ***ADGofTest****: Anderson-Darling GoF test.* R package version 0.3, URL `http://CRAN.R-project.org/package=ADGofTest`.

Bickel PJ, Rosenblatt M (1973). "On Some Global Measures of the Deviations of Density Function Estimates." *The Annals of Statistics*, **1**(6), 1071–1095.

Bowman AW (1992). "Density Based Tests for Goodness-of-Fit." *Journal of Statistical Computation and Simulation*, **40**(1–2), 1–13.

Bowman AW, Foster PJ (1993). "Adaptive Smoothing and Density-Based Tests for Multivariate Normality." *Journal of the American Statistical Association*, **88**(422), 529–537.

Cao R, Lugosi G (2005). "Goodness-of-Fit Tests Based on the Kernel Density Estimator." *Scandinavian Journal of Statistics*, **32**(4), 599–616.

Chernobai A, Rachev S, Fabozzi F (2005). "Composites Goodness-of-Fit Tests for Left-Truncated Loss Samples." *Technical report*, University of California, Santa Barbara. URL `https://statistik.ets.kit.edu/download/doc_secure1/tr_composite_goodness_tests.pdf`.

Fan Y (1994). "Testing the Goodness of Fit of a Parametric Density Function by Kernel Method." *Econometric Theory*, **10**(2), 316–356.

Fan Y (1998). "Goodness-of-Fit Tests Based on Kernel Density Estimators with Fixed Smoothing Parameters." *Econometric Theory*, **14**(5), 604–621.

Fan Y, Gencay R (1993). "Hypotheses Testing Based on Modified Nonparametric Estimation of an Affinity Measure between Two Distributions." *Journal of Nonparametric Statistics*, **2**(4), 389–403.

Fan Y, Ullah A (1999). "On Goodness-of-Fit Tests for Weakly Dependent Processes Using Kernel Method." *Journal of Nonparametric Statistics*, **11**(1–3), 337–360.

Frampton A (2010). *Stochastic Analysis of Fluid Flow and Tracer Pathways in Crystalline Fracture Networks*. Ph.D. thesis, Royal Institute of Technology.

Györfi L, van der Meulen EC (1991). "A Consistent Goodness of Fit Test Based on the Total Variation Distance." In G Roussas (ed.), *Nonparametric Functional Estimation and Related Topics*, NATO ASI Series, pp. 631–646. Kluwer Academic Publishers.

Hoeffding W, Wolfowitz J (1958). "Distinguishability of Sets of Distributions." *The Annals of Mathematical Statistics*, **29**(3), 700–718.

Jammalamadaka SR, Sengupta A (2001). *Topics in Circular Statistics*. World Scientific Pub Co Inc.

Johnson RA, Miller I, Freund JE (2011). *Miller & Freund's Probability and Statistics for Engineers*. Prentice Hall PTR.

Kuiper NH (1962). "Tests Concerning Random Points on a Circle." In *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A, 63*, pp. 38–67.

LeCam L (1973). "Convergence of Estimates Under Dimensionality Restrictions." *The Annals of Statistics*, **1**(1), 38–53.

Li Q, Racine JS (2007). *Nonparametric Econometrics: Theory and Practice*. Princeton University Press.

Lindsay BG, Markatou M, Ray S (2014). "Kernels, Degrees of Freedom, and Power Properties of Quadratic Distance Goodness-of-Fit Tests." *Journal of the American Statistical Association*, **109**, 395–410.

Lund U, Agostinelli C (2013). ***circular**: Circular Statistics*. R package version 0.4-7, URL `http://CRAN.R-project.org/package=circular`.

Marsaglia G, Marsaglia J (2004). "Evaluating the Anderson-Darling Distribution." *Journal of Statistical Software*, **9**(2), 1–5. URL `http://www.jstatsoft.org/v09/i02/`.

Mashhadi MA (2011). "Some Goodness of Fit Tests Based on Renyi Information." *Applied Mathematical Sciences*, **39**(5), 1921–1934.

Mattheou K, Karagrigoriou A (2010). "A New Family of Divergence Measures for Tests of Fit." *Australian & New Zealand Journal of Statistics*, **52**(2), 187–200.

Miecznikowski J, Vexler A, Shepherd L (2013). "**dbEmpLikeGOF**: An R Package for Nonparametric Likelihood Ratio Tests for Goodness-of-Fit and Two-Sample Comparisons Based on Sample Entropy." *Journal of Statistical Software*, **54**(3), 1–19. URL `http://www.jstatsoft.org/v54/i03/`.

Pavía JM (2015). ***GoFKernel**: Testing Goodness-of-Fit with the Kernel Density Estimator.* R package version 2.0-6, URL `http://CRAN.R-project.org/package=GoFKernel`.

Pavía JM, Morillas FG, Lledó J (2012). "Introducing Migratory Flows in Life Table Construction." *Sort: Statistics and Operations Research Transactions*, **36**(1), 103–114.

Pearson ES, Stephens MA (1962). "The Goodness-of-Fit Tests Based on W_N^2 and U_N^2." *Biometrika*, **49**(3/4), 397–402.

Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1992). *Numerical Recipes in Fortran: The Art of Scientific Computing*. 2nd edition. Cambridge University Press.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org/`.

Rosenblatt M (1975). "A Quadratic Measure of Deviation of Two-Dimensional Density Estimates and a Test of Independence." *The Annals of Statistics*, **3**(1), 1–14.

Sheather SJ, Jones MC (1991). "A Reliable Data-Based Bandwidth Selection Method for Kernel Density Estimation." *Journal of the Royal Statistical Society B*, **53**, 683–690.

Silverman BW (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC Monographs on Statistics & Applied Probability. Chapman and Hall/CRC.

Stephens MA (1964). "The Distribution of the Goodness-of-Fit Statistic, U_N^2. II." *Biometrika*, **51**(3/4), 393–397.

Stephens MA (1974). "EDF Statistics for Goodness of Fit and Some Comparisons." *Journal of the American Statistical Association*, **69**(347), 730–737.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.

Vexler A, Gurevich G (2010). "Empirical Likelihood Ratios Applied to Goodness-of-Fit Tests Based on Sample Entropy." *Computational Statistics & Data Analysis*, **54**(2), 531–545.

Wand M (2013). ***KernSmooth**: Functions for Kernel Smoothing for Wand & Jones (1995).* R package version 2.23-10, URL `http://CRAN.R-project.org/package=KernSmooth`.

Wand MP, Jones MC (1995). *Kernel Smoothing*. Chapman and Hall, London.

Watson GS (1961). "Goodness-of-Fit Tests on a Circle." *Biometrika*, **48**(1/2), 109–114.

Wolter T (2012). ***truncgof**: GoF Tests Allowing for Left Truncated Data*. R package version 0.6-0, URL `http://CRAN.R-project.org/package=truncgof`.

Zhang J (2002). "Powerful Goodness-of-Fit Tests Based on the Likelihood Ratio." *Journal of the Royal Statistical Society B*, **64**(2), 281–294.

Affiliation:

Jose M. Pavia
Department of Applied Economics
Facultat d'Economia
Universitat de Valencia
46022 Valencia, Spain
E-mail: `pavia@uv.es`
URL: `http://www.uv.es/pavia/`

Journal of Statistical Software `http://www.jstatsoft.org/`
published by the American Statistical Association `http://www.amstat.org/`

Volume 66, Code Snippet 1 *Submitted:* 2013-05-03
August 2015 *Accepted:* 2014-10-19