Package ‘dgof’

October 13, 2022

Version 1.4

Date 2022-07-16

Title Discrete Goodness-of-Fit Tests

Author Taylor B. Arnold, John W. Emerson, R Core Team and contributors worldwide

Maintainer Taylor B. Arnold <taylor.b.arnold@gmail.com>

Description A revision to the stats::ks.test() function and the associated ks.test.Rd help page. With one minor exception, it does not change the existing behavior of ks.test(), and it adds features necessary for doing one-sample tests with hypothesized discrete distributions. The package also contains cvm.test(), for doing one-sample Cramer-von Mises goodness-of-fit tests.

License GPL (>= 2.0)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-06-16 16:50:02 UTC

R topics documented:

`cvm.test` *Discrete Cramer-von Mises Goodness-of-Fit Tests*

Description

Computes the test statistics for doing one-sample Cramer-von Mises goodness-of-fit tests and calculates asymptotic p-values.

Usage

```
cvm.test(x, y, type = c("W2", "U2", "A2"),
         simulate.p.value=FALSE, B=2000, tol=1e-8)
```

Arguments

`x` a numerical vector of data values.

`y` an `ecdf` or step-function (`stepfun`) for specifying the hypothesized model.

`type` the variant of the Cramer-von Mises test; `"W2"` is the default and most common method, `"U2"` is for cyclical data, and `"A2"` is the Anderson-Darling alternative. For details see references.

`simulate.p.value` a logical indicating whether to compute p-values by Monte Carlo simulation.

`B` an integer specifying the number of replicates used in the Monte Carlo test (for discrete goodness-of-fit tests only).

`tol` used as an upper bound for possible rounding error in values (say, `a` and `b`) when needing to check for equality (`a==b`) (for discrete goodness-of-fit tests only).

Details

While the Kolmogorov-Smirnov test may be the most popular of the nonparametric goodness-of-fit tests, Cramer-von Mises tests have been shown to be more powerful against a large class of alternatives hypotheses. The original test was developed by Harald Cramer and Richard von Mises (Cramer, 1928; von Mises, 1928) and further adapted by Anderson and Darling (1952), and Watson (1961).

Value

An object of class `htest`.

Note

Additional notes?

Author(s)

Taylor B. Arnold and John W. Emerson

Maintainer: Taylor B. Arnold <taylor.arnold@yale.edu>

References

T. W. Anderson and D. A. Darling (1952). *Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes.* Annals of Mathematical Statistics, 23:193-212.

V. Choulakian, R. A. Lockhart, and M. A. Stephens (1994). *Cramer-von Mises statistics for discrete distributions*. The Canadian Journal of Statistics, 22(1): 125-137.

H. Cramer (1928). *On the composition of elementary errors.* Skand. Akt., 11:141-180.

M. A. Stephens (1974). *Edf statistics for goodness of fit and some comparisons*. Journal of the American Statistical Association, 69(347): 730-737.

R. E. von Mises (1928). *Wahrscheinlichkeit, Statistik und Wahrheit.* Julius Springer, Vienna, Austria.

G. S. Watson (1961). *Goodness of fit tests on the circle.* Biometrika, 48:109-114.

See Also

`ks.test`, `ecdf`, `stepfun`

Examples

```
require(dgof)

x3 <- sample(1:10, 25, replace=TRUE)

# Using ecdf() to specify a discrete distribution:
ks.test(x3, ecdf(1:10))
cvm.test(x3, ecdf(1:10))

# Using step() to specify the same discrete distribution:
myfun <- stepfun(1:10, cumsum(c(0, rep(0.1, 10))))
ks.test(x3, myfun)
cvm.test(x3, myfun)

# Usage of U2 for cyclical distributions (note U2 unchanged, but W2 not)

set.seed(1)
y <- sample(1:4, 20, replace=TRUE)
cvm.test(y, ecdf(1:4), type='W2')
cvm.test(y, ecdf(1:4), type='U2')
z <- y
cvm.test(z, ecdf(1:4), type='W2')
cvm.test(z, ecdf(1:4), type = 'U2')

# Compare analytic results to simulation results
set.seed(1)
y <- sample(1:3, 10, replace=TRUE)
cvm.test(y, ecdf(1:6), simulate.p.value=FALSE)
cvm.test(y, ecdf(1:6), simulate.p.value=TRUE)
```

`ks.test` *Kolmogorov-Smirnov Tests*

Description

Performs one or two sample Kolmogorov-Smirnov tests.

Usage

```
ks.test(x, y, ...,
        alternative = c("two.sided", "less", "greater"),
        exact = NULL, tol=1e-8, simulate.p.value=FALSE, B=2000)
```

Arguments

`x` a numeric vector of data values.

`y` a numeric vector of data values, or a character string naming a cumulative distribution function or an actual cumulative distribution function such as `pnorm`. Alternatively, `y` can be an `ecdf` function (or an object of class `stepfun`) for specifying a discrete distribution.

`...` parameters of the distribution specified (as a character string) by `y`.

`alternative` indicates the alternative hypothesis and must be one of `"two.sided"` (default), `"less"`, or `"greater"`. You can specify just the initial letter of the value, but the argument name must be give in full. See 'Details' for the meanings of the possible values.

`exact` `NULL` or a logical indicating whether an exact p-value should be computed. See 'Details' for the meaning of `NULL`. Not used for the one-sided two-sample case.

`tol` used as an upper bound for possible rounding error in values (say, `a` and `b`) when needing to check for equality (`a==b`); value of `NA` or `0` does exact comparisons but risks making errors due to numerical imprecisions.

`simulate.p.value` a logical indicating whether to compute p-values by Monte Carlo simulation, for discrete goodness-of-fit tests only.

`B` an integer specifying the number of replicates used in the Monte Carlo test (for discrete goodness-of-fit tests only).

Details

If `y` is numeric, a two-sample test of the null hypothesis that `x` and `y` were drawn from the same *continuous* distribution is performed.

Alternatively, `y` can be a character string naming a continuous (cumulative) distribution function (or such a function), or an `ecdf` function (or object of class `stepfun`) giving a discrete distribution. In these cases, a one-sample test is carried out of the null that the distribution function which generated `x` is distribution `y` with parameters specified by `...`.

The presence of ties generates a warning unless `y` describes a discrete distribution (see above), since continuous distributions do not generate them.

The possible values `"two.sided"`, `"less"` and `"greater"` of `alternative` specify the null hypothesis that the true distribution function of `x` is equal to, not less than or not greater than the hypothesized distribution function (one-sample case) or the distribution function of `y` (two-sample case), respectively. This is a comparison of cumulative distribution functions, and the test statistic is the maximum difference in value, with the statistic in the `"greater"` alternative being $D^+ = \max_u[F_x(u) - F_y(u)]$. Thus in the two-sample case `alternative="greater"` includes distributions for which `x` is stochastically *smaller* than `y` (the CDF of `x` lies above and hence to the left of that for `y`), in contrast to `t.test` or `wilcox.test`.

Exact p-values are not available for the one-sided two-sample case, or in the case of ties if `y` is continuous. If `exact = NULL` (the default), an exact p-value is computed if the sample size is less than 100 in the one-sample case with `y` continuous or if the sample size is less than or equal to 30 with `y` discrete; or if the product of the sample sizes is less than 10000 in the two-sample case for continuous `y`. Otherwise, asymptotic distributions are used whose approximations may be inaccurate in small samples. With `y` continuous, the one-sample two-sided case, exact p-values are obtained as described in Marsaglia, Tsang & Wang (2003); the formula of Birnbaum & Tingey (1951) is used for the one-sample one-sided case.

In the one-sample case with `y` discrete, the methods presented in Conover (1972) and Gleser (1985) are used when `exact=TRUE` (or when `exact=NULL`) and `length(x)<=30` as described above. When `exact=FALSE` or `exact=NULL` with `length(x)>30`, the test is not exact and the resulting p-values are known to be conservative. Usage of `exact=TRUE` with sample sizes greater than 30 is not advised due to numerical instabilities; in such cases, simulated p-values may be desirable.

If a single-sample test is used with `y` continuous, the parameters specified in `...` must be prespecified and not estimated from the data. There is some more refined distribution theory for the KS test with estimated parameters (see Durbin, 1973), but that is not implemented in `ks.test`.

Value

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

`statistic`	the value of the test statistic.
`p.value`	the p-value of the test.
`alternative`	a character string describing the alternative hypothesis.
`method`	a character string indicating what type of test was performed.
`data.name`	a character string giving the name(s) of the data.

Author(s)

Modified by Taylor B. Arnold and John W. Emerson to include one-sample testing with a discrete distribution (as presented in Conover's 1972 paper – see references).

References

Z. W. Birnbaum and Fred H. Tingey (1951), One-sided confidence contours for probability distribution functions. *The Annals of Mathematical Statistics*, **22**/4, 592–596.

William J. Conover (1971), *Practical Nonparametric Statistics*. New York: John Wiley & Sons. Pages 295–301 (one-sample Kolmogorov test), 309–314 (two-sample Smirnov test).

William J. Conover (1972), A Kolmogorov Goodness-of-Fit Test for Discontinuous Distributions. *Journal of American Statistical Association*, Vol. 67, No. 339, 591–596.

Leon Jay Gleser (1985), Exact Power of Goodness-of-Fit Tests of Kolmogorov Type for Discontinuous Distributions. *Journal of American Statistical Association*, Vol. 80, No. 392, 954–958.

Durbin, J. (1973) *Distribution theory for tests based on the sample distribution function*. SIAM.

George Marsaglia, Wai Wan Tsang and Jingbo Wang (2003), Evaluating Kolmogorov's distribution. *Journal of Statistical Software*, **8**/18. https://www.jstatsoft.org/v08/i18/.

See Also

shapiro.test which performs the Shapiro-Wilk test for normality; cvm.test for Cramer-von Mises type tests.

Examples

```
require(graphics)
require(dgof)

set.seed(1)

x <- rnorm(50)
y <- runif(30)
# Do x and y come from the same distribution?
ks.test(x, y)
# Does x come from a shifted gamma distribution with shape 3 and rate 2?
ks.test(x+2, "pgamma", 3, 2) # two-sided, exact
ks.test(x+2, "pgamma", 3, 2, exact = FALSE)
ks.test(x+2, "pgamma", 3, 2, alternative = "gr")

# test if x is stochastically larger than x2
x2 <- rnorm(50, -1)
plot(ecdf(x), xlim=range(c(x, x2)))
plot(ecdf(x2), add=TRUE, lty="dashed")
t.test(x, x2, alternative="g")
wilcox.test(x, x2, alternative="g")
ks.test(x, x2, alternative="l")

#########################################################
# TBA, JWE new examples added for discrete distributions:

x3 <- sample(1:10, 25, replace=TRUE)

# Using ecdf() to specify a discrete distribution:
ks.test(x3, ecdf(1:10))
```

```
# Using step() to specify the same discrete distribution:
myfun <- stepfun(1:10, cumsum(c(0, rep(0.1, 10))))
ks.test(x3, myfun)

# The previous R ks.test() does not correctly calculate the
# test statistic for discrete distributions (gives warning):
# stats::ks.test(c(0, 1), ecdf(c(0, 1)))
# ks.test(c(0, 1), ecdf(c(0, 1)))

# Even when the correct test statistic is given, the
# previous R ks.test() gives conservative p-values:
stats::ks.test(rep(1, 3), ecdf(1:3))
ks.test(rep(1, 3), ecdf(1:3))
ks.test(rep(1, 3), ecdf(1:3), simulate=TRUE, B=10000)
```

Index