

Package ‘ddst’

April 17, 2009

Type Package

Title Data driven smooth test

Version 1.01

Date 2008-08-08

Author Przemyslaw Biecek (R code), Teresa Ledwina (support, descriptions)

Maintainer Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

Description Support for data driven testing

License GPL-2

Depends R (>= 2.1.0), orthopolynom, evd

Repository CRAN

Date/Publication 2008-09-28 19:26:52

R topics documented:

ddst-package	1
ddst.exp.test	5
ddst.extr.test	6
ddst.norm.test	8
ddst.uniform.test	10

Index	13
--------------	-----------

ddst-package *Data Driven Smooth Tests*

Description

Set of data driven smooth goodness of fit tests.

Details

Package: ddst
 Type: Package
 Version: 1.0
 Date: 2008-07-01
 License: GPL-2

General Description

Smooth test was introduced by Neyman (1937) to verify simple null hypothesis asserting that observations obey completely known continuous distribution function F . Smooth test statistic (with k components) can be interpreted as score statistic in an appropriate class of auxiliary models indexed by a vector of parameters $\theta \in \mathbb{R}^k$, $k \geq 1$.

Pertaining auxiliary null hypothesis asserts $\theta = \theta_0 = 0$. Therefore, in this case, the smooth test statistic based on n i.i.d. observations Z_1, \dots, Z_n has the form $W_k = [1/\sqrt{n} \sum_{i=1}^n l(Z_i)]' I^{-1} [1/\sqrt{n} \sum_{i=1}^n l(Z_i)]$ where $l(Z_i)$, $i=1, \dots, n$, is k -dimensional (row) score vector, the symbol $'$ denotes transposition while $I = Cov_{\theta_0}[l(Z_1)]'[l(Z_1)]$. Following Neyman's idea of modelling underlying distributions one gets $l(Z_i) = (\phi_1(F(Z_i)), \dots, \phi_k(F(Z_i)))$ and I being the identity matrix, where ϕ_j 's, $j \geq 1$, are zero mean orthonormal functions on $[0,1]$, while F is the completely specified null distribution function.

In case of composite null hypothesis there is also unspecified vector of nuisance parameters γ defining the distribution of observations. Smooth statistic (with k components) in such applications is understood as efficient score statistic for some class of models indexed by an auxiliary parameter $\theta \in \mathbb{R}^k$, $k \geq 1$. Pertaining efficient score vector $l^*(Z_i; \gamma)$ is defined as the residual from projection the score vector for θ onto the space spanned by score vector for γ . As such, smooth test is alternative name for $C(\alpha)$ Neyman's test. See Neyman (1959), Bühler and Puri (1966) as well as Javitz (1975) for details. Hence, smooth test, based on n i.i.d. variables Z_1, \dots, Z_n rejects hypothesis $\theta = \theta_0 = 0$ for large values of

$W_k^*(\tilde{\gamma}) = [1/\sqrt{n} \sum_{i=1}^n l^*(Z_i; \tilde{\gamma})]' [I^*(\tilde{\gamma})]^{-1} [1/\sqrt{n} \sum_{i=1}^n l^*(Z_i; \tilde{\gamma})]'$, where $\tilde{\gamma}$ is an appropriate estimator of γ while $I^*(\gamma) = Cov_{\theta_0}[l^*(Z_1; \gamma)]'[l^*(Z_1; \gamma)]$. More details can be found in Janic and Ledwina (2008), Kallenberg and Ledwina (1997 a,b) as well as Inglot and Ledwina (2006 a,b).

Auxiliary models, mentioned above, aim to mimic the unknown underlying model for the data at hand. To choose the dimension k of the auxiliary model we apply some model selection criteria. Among several solutions already considered, we decided to implement two following ones, pertaining to the two above described problems and resulting W_k and $W_k^*(\tilde{\gamma})$. The selection rules in the two cases are briefly denoted by T and T^* , respectively, and given by

$$T = \min_{1 \leq k \leq d} W_k - \pi(k, n, c) > \min_{j=1, \dots, d} W_j - \pi(j, n, c),$$

and

$$T^* = \min_{1 \leq k \leq d} W_k^*(\tilde{\gamma}) - \pi^*(k, n, c) > \min_{j=1, \dots, d} W_j^*(\tilde{\gamma}) - \pi^*(j, n, c).$$

Both criteria are based on approximations of penalized loglikelihoods, where loglikelihoods are replaced by W_k and $W_k^*(\tilde{\gamma})$, respectively. The penalties for the dimension j in case of simple and composite null hypothesis are defined as follows

$$\pi(j, n, c) = j \log n, \text{ if } \max_{1 \leq k \leq d} |Y_{-k}| < \sqrt{c \log(n)}, 2j, \text{ if } \max_{1 \leq k \leq d} |Y_{-k}| > \sqrt{c \log(n)}$$

and

$\pi^*(j, n, c) = j \log n$, if $\max 1 < = k < = d |Y_{k^*}| < = \text{sqrt}(c \log(n))$, 2j if $\max(1 < = k < = d) |Y_{k^*}| > \text{sqrt}(c \log(n))$

respectively, where c is some calibrating constant, d is maximal dimension taken into account,

$(Y_1, \dots, Y_k) = [1/\text{sqrt}(n) \sum_{i=1}^k I^*(Z_i)] I^{-1/2}$

while

$(Y_1^*, \dots, Y_k^*) = [1/\text{sqrt}(n) \sum_{i=1}^k I^*(Z_i; \tilde{\gamma})] [I^*(\tilde{\gamma})]^{-1/2}$.

In consequence, data driven smooth tests for the simple and composite null hypothesis reject for large values of W_T and $W_{T^*} = W_{T^*}(\tilde{\gamma})$, respectively. For details see Inglot and Ledwina (2006 a,b,c).

The choice of c in T and T^* is decisive to finite sample behaviour of the selection rules and pertaining statistics W_T and $W_{T^*}(\tilde{\gamma})$. In particular, under large c 's the rules behave similarly as Schwarz's (1978) BIC while for $c=0$ they mimic Akaike's (1973) AIC. For moderate sample sizes, values c in (2,2.5) guarantee, under 'smooth' departures, only slightly smaller power as in case BIC were used and simultaneously give much higher power than BIC under multimodal alternatives. In general, large c 's are recommended if changes in location, scale, skewness and kurtosis are in principle aimed to be detected. For evidence and discussion see Inglot and Ledwina (2006 c).

If $c > 0$ then the limiting null distribution of W_T and $W_{T^*}(\tilde{\gamma})$ is central chi-squared with one degree of freedom. In our implementation, for given n , both critical values and p -values are computed by MC method.

Empirical distributions of T and T^* as well as W_T and $W_{T^*}(\tilde{\gamma})$ are not essentially influenced by the choice of reasonably large d 's, provided that sample size is at least moderate.

For more details see: <http://www.biecek.pl/R/ddst/description.pdf>.

Author(s)

Przemyslaw Biecek and Teresa Ledwina

Maintainer: You should complain to Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

References

- Akaike, H. (1973). Information theory and the maximum likelihood principle. In: *2nd International Symposium on Information Theory*, (eds. B. N. Petrov and F. Csaki), 267-281. Akademiai Kiado, Budapest.
- Buhler, W.J., Puri, P.S. (1966). On optimal asymptotic tests of composite hypotheses with several constraints. *Z. Wahrsch. verw. Geb.* **5**, 71-88.
- Inglot, T., Ledwina, T. (2006 a). Data-driven score tests for homoscedastic linear regression model: asymptotic results. *Probab. Math. Statist.* **26**, 41-61.
- Inglot, T., Ledwina, T. (2006 b). Data-driven score tests for homoscedastic linear regression model: the construction and simulations. In *Prague Stochastics 2006. Proceedings*, (eds. M. Hušková, M. Janžura), 124-137. Matfyzpress, Prague.
- Inglot, T., Ledwina, T. (2006 c). Towards data driven selection of a penalty function for data driven Neyman tests. *Linear Algebra and its Appl.* **417**, 579-590.
- Javitz, H.S. (1975). Generalized smooth tests of goodness of fit, independence and equality of distributions. Ph.D. thesis at University of California, Berkeley.

Janic, A. and Ledwina, T. (2008). Data-driven tests for a location-scale family revisited. *J. Statist. Theory. Pract. Special issue on Modern Goodness of Fit Methods. accepted..*

Kallenberg, W.C.M., Ledwina, T. (1997 a). Data driven smooth tests for composite hypotheses: Comparison of powers. *J. Statist. Comput. Simul.* **59**, 101–121.

Kallenberg, W.C.M., Ledwina, T. (1997 b). Data driven smooth tests when the hypothesis is composite. *J. Amer. Statist. Assoc.* **92**, 1094–1104.

Neyman, J. (1937). ‘Smooth test’ for goodness of fit. *Skand. Aktuarietidskr.* **20**, 149-199.

Neyman, J. (1959). Optimal asymptotic tests of composite statistical hypotheses. In *Probability and Statistics*, (ed. U. Grenander), Harald Cramer Volume, 212–234. Wiley, New York.

Examples

```
# Data Driven Smooth Test for Uniformity
#
# H0 is true
z = runif(80)
ddst.uniform.test(z, compute.p=TRUE)

# H0 is false
z = rbeta(80,4,2)
(t = ddst.uniform.test(z, compute.p=TRUE))
t$p.value

# Data Driven Smooth Test for Normality
#
# H0 is true
z = rnorm(80)
ddst.norm.test(z, compute.p=TRUE)

# H0 is false
z = rexp(80,4)
ddst.norm.test(z, B=5000, compute.p=TRUE)

# Data Driven Smooth Test for Extreme Value Distribution
#
# H0 is true
#library(evd)
#z = -qgumbel(runif(100),-1,1)
#ddst.extr.test(z, compute.p = TRUE)

# H0 is false
z = rexp(80,4)
ddst.extr.test(z, compute.p = TRUE)

# Data Driven Smooth Test for Exponentiality
#
# H0 is true
z = rexp(80,4)
ddst.exp.test(z, compute.p = TRUE)

# H0 is false
```

```
z = rchisq(80,4)
ddst.exp.test (z, compute.p = TRUE)
```

ddst.exp.test *Data Driven Smooth Test for Exponentiality*

Description

Performs data driven smooth test for composite hypothesis of exponentiality.

Usage

```
ddst.exp.test(x, base = ddst.base.legendre, c = 100, B = 1000, compute.p = F, Dmax
```

Arguments

<code>x</code>	a (non-empty) numeric vector of data values.
<code>base</code>	a function which returns orthogonal system, might be <code>ddst.base.legendre</code> for Legendre polynomials or <code>ddst.base.cos</code> for cosine system, see package description.
<code>c</code>	a parameter for model selection rule, see package description.
<code>B</code>	an integer specifying the number of replicates used in p-value computation.
<code>compute.p</code>	a logical value indicating whether to compute a p-value.
<code>Dmax</code>	an integer specifying the maximum number of coordinates, only for advanced users.
<code>...</code>	further arguments.

Details

Null density is given by $f(z; \gamma) = \exp(-z/\gamma)$ for $z \geq 0$ and 0 otherwise.

Modelling alternatives similarly as in Kallenberg and Ledwina (1997 a,b), e.g., and estimating γ by $\tilde{\gamma} = 1/n \sum_{i=1}^n Z_i$ yields the efficient score vector $l^*(Z_i; \tilde{\gamma}) = (\text{phi}_1(F(Z_i; \tilde{\gamma})), \dots, \text{phi}_k(F(Z_i; \tilde{\gamma})))$, where phi_j 's are j th degree orthonormal Legendre polynomials on $[0,1]$ or cosine functions $\text{sqrt}(2)\cos(\pi jx)$, $j > 1$, while $F(z; \gamma)$ is the distribution function pertaining to $f(z; \gamma)$.

The matrix $[I^*(\tilde{\gamma})]^{-1}$ does not depend on $\tilde{\gamma}$ and is calculated for succeeding dimensions k using some recurrent relations for Legendre's polynomials and computed in a numerical way in case of cosine basis. In the implementation the default value of c in T^* is set to be 100.

Therefore, T^* practically coincides with $S1$ considered in Kallenberg and Ledwina (1997 a).

For more details see: <http://www.biecek.pl/R/ddst/description.pdf>.

Value

An object of class `htest`

<code>statistic</code>	the value of the test statistic.
<code>parameter</code>	the number of choosen coordinates (<code>k</code>).
<code>method</code>	a character string indicating the parameters of performed test.
<code>data.name</code>	a character string giving the name(s) of the data.
<code>p.value</code>	the p-value for the test, computed only if <code>compute.p=T</code> .

Author(s)

Przemyslaw Biecek and Teresa Ledwina

References

Kallenberg, W.C.M., Ledwina, T. (1997 a). Data driven smooth tests for composite hypotheses: Comparison of powers. *J. Statist. Comput. Simul.* **59**, 101–121.

Kallenberg, W.C.M., Ledwina, T. (1997 b). Data driven smooth tests when the hypothesis is composite. *J. Amer. Statist. Assoc.* **92**, 1094–1104.

Examples

```
# H0 is true
z = rexp(80,4)
ddst.exp.test (z, compute.p = TRUE)

# H0 is false
z = rchisq(80,4)
(t = ddst.exp.test (z, compute.p = TRUE))
t$p.value
```

`ddst.extr.test`

Data Driven Smooth Test for Extreme Value Distribution

Description

Performs data driven smooth test for composite hypothesis of extreme value distribution.

Usage

```
ddst.extr.test(x, base = ddst.base.legendre, c = 100, B = 1000, compute.p = F, Dmax
```

Arguments

<code>x</code>	a (non-empty) numeric vector of data values.
<code>base</code>	a function which returns orthogonal system, might be <code>ddst.base.legendre</code> for Legendre polynomials or <code>ddst.base.cos</code> for cosine system, see package description.
<code>c</code>	a parameter for model selection rule, see package description.
<code>B</code>	an integer specifying the number of replicates used in p-value computation.
<code>compute.p</code>	a logical value indicating whether to compute a p-value.
<code>Dmax</code>	an integer specifying the maximum number of coordinates, only for advanced users.
<code>...</code>	further arguments.

Details

Null density is given by $f(z; \gamma) = 1/\gamma_2 \exp((z - \gamma_1)/\gamma_2 - \exp((z - \gamma_1)/\gamma_2))$, z in \mathbb{R} .

We model alternatives similarly as in Kallenberg and Ledwina (1997) and Janic-Wroblewska (2004) using Legendre's polynomials or cosines. The parameter $\gamma = (\gamma_1, \gamma_2)$ is estimated by $\tilde{\gamma} = (\tilde{\gamma}_1, \tilde{\gamma}_2)$, where $\tilde{\gamma}_1 = -1/n \sum_{i=1}^n Z_{-i} + \varepsilon G$, where $\varepsilon \approx 0.577216$ is the Euler constant and $G = \tilde{\gamma}_2 = [n(n-1) \ln 2]^{-1} \sum_{1 < j < i < n} (Z_{-n} : i^o - Z_{-n} : j^o)$ while $Z_{-n} : 1^o < \dots < Z_{-n} : n^o$ are ordered variables $-Z_1, \dots, -Z_n$, cf Hosking et al. (1985). The above yields auxiliary test statistic $W_{-k}^*(\tilde{\gamma})$ described in details in Janic and Ledwina (2008), in case when Legendre's basis is applied.

The related matrix $[I^*(\tilde{\gamma})]^{-1}$ does not depend on $\tilde{\gamma}$ and is calculated for succeeding dimensions k using some recurrent relations for Legendre's polynomials and numerical methods for cosine functions. In the implementation the default value of c in T^* was fixed to be 100. Hence, T^* is Schwarz-type model selection rule. The resulting data driven test statistic for extreme value distribution is $W_{-T^*} = W_{-T^*}(\tilde{\gamma})$.

For more details see: <http://www.biecek.pl/R/ddst/description.pdf>.

Value

An object of class `htest`

<code>statistic</code>	the value of the test statistic.
<code>parameter</code>	the number of chosen coordinates (k).
<code>method</code>	a character string indicating the parameters of performed test.
<code>data.name</code>	a character string giving the name(s) of the data.
<code>p.value</code>	the p-value for the test, computed only if <code>compute.p=T</code> .

Author(s)

Przemyslaw Biecek and Teresa Ledwina

References

- Hosking, J.R.M., Wallis, J.R., Wood, E.F. (1985). Estimation of the generalized extreme-value distribution by the method of probability-weighted moments. *Technometrics* 27, 251–261.
- Janic-Wroblewska, A. (2004). Data-driven smooth test for extreme value distribution. *Statistics* 38, 413–426.
- Janic, A. and Ledwina, T. (2008). Data-driven tests for a location-scale family revisited. *J. Statist. Theory. Pract. Special issue on Modern Goodness of Fit Methods. accepted.*
- Kallenberg, W.C.M., Ledwina, T. (1997). Data driven smooth tests for composite hypotheses: Comparison of powers. *J. Statist. Comput. Simul.* 59, 101–121.

Examples

```
#library(evd)
#
## for given vector of 19 numbers
#z = c(13.41, 6.04, 1.26, 3.67, -4.54, 2.92, 0.44, 12.93, 6.77, 10.09,
# 4.10, 4.04, -1.97, 2.17, -5.38, -7.30, 4.75, 5.63, 8.84)
#ddst.extr.test(z, compute.p=TRUE)
#
## H0 is true
#x = -qgumbel(runif(100), -1, 1)
#ddst.extr.test(x, compute.p = TRUE)
#
## H0 is false
#x = rexp(80, 4)
#ddst.extr.test(x, compute.p = TRUE)
```

ddst.norm.test

Data Driven Smooth Test for Normality

Description

Performs data driven smooth test for composite hypothesis of normality.

Usage

```
ddst.norm.test(x, base = ddst.base.legendre, c = 100, B = 1000, compute.p = F, Dmax
```

Arguments

- | | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x | a (non-empty) numeric vector of data values. |
| base | a function which returns orthogonal system, might be <code>ddst.base.legendre</code> for Legendre polynomials or <code>ddst.base.cos</code> for cosine system, see package description. |
| c | a parameter for model selection rule, see package description. |

B	an integer specifying the number of replicates used in p-value computation.
compute.p	a logical value indicating whether to compute a p-value.
Dmax	an integer specifying the maximum number of coordinates, only for advanced users.
...	further arguments.

Details

Null density is given by $f(z; \gamma) = 1/(\text{sqrt}(2\pi)\gamma_2)\exp(-(z - \gamma_1)^2/(2\gamma_2^2))$ for z in \mathbb{R} .

We model alternatives similarly as in Kallenberg and Ledwina (1997 a,b) using Legendre's polynomials or cosine basis. The parameter $\gamma = (\gamma_1, \gamma_2)$ is estimated by $\tilde{\gamma} = (\tilde{\gamma}_1, \tilde{\gamma}_2)$, where $\tilde{\gamma}_1 = 1/n \sum_{i=1}^n Z_i$ and $\tilde{\gamma}_2 = 1/(n-1) \sum_{i=1}^{n-1} (Z_{n:i} + 1 - Z_{n:i})(H_{i+1} - H_i)$, while $Z_{n:1} < \dots < Z_{n:n}$ are ordered values of Z_1, \dots, Z_n and $H_i = \Phi^{-1}((i - 3/8)(n + 1/4))$, cf. Chen and Shapiro (1995).

The above yields auxiliary test statistic $W_{k^*}(\tilde{\gamma})$ described in details in Janic and Ledwina (2008), in case when Legendre's basis is applied. The pertaining matrix $[T^*(\tilde{\gamma})]^{-1}$ does not depend on $\tilde{\gamma}$ and is calculated for succeeding dimensions k using some recurrent relations for Legendre's polynomials and is computed in a numerical way in case of cosine basis. In the implementation of T^* the default value of c is set to be 100. Therefore, in practice, T^* is Schwarz-type criterion. See Inglot and Ledwina (2006) as well as Janic and Ledwina (2008) for comments. The resulting data driven test statistic for normality is $W_{T^*} = W_{T^*}(\tilde{\gamma})$.

For more details see: <http://www.biecek.pl/R/ddst/description.pdf>.

Value

An object of class `htest`

statistic	the value of the test statistic.
parameter	the number of chosen coordinates (k).
method	a character string indicating the parameters of performed test.
data.name	a character string giving the name(s) of the data.
p.value	the p-value for the test, computed only if <code>compute.p=T</code> .

Author(s)

Przemyslaw Biecek and Teresa Ledwina

References

- Chen, L., Shapiro, S.S. (1995). An alternative test for normality based on normalized spacings. *J. Statist. Comput. Simulation* 53, 269–288.
- Inglot, T., Ledwina, T. (2006). Towards data driven selection of a penalty function for data driven Neyman tests. *Linear Algebra and its Appl.* **417**, 579–590.
- Janic, A. and Ledwina, T. (2008). Data-driven tests for a location-scale family revisited. *J. Statist. Theory. Pract. Special issue on Modern Goodness of Fit Methods. accepted..*

Kallenberg, W.C.M., Ledwina, T. (1997 a). Data driven smooth tests for composite hypotheses: Comparison of powers. *J. Statist. Comput. Simul.* **59**, 101–121.

Kallenberg, W.C.M., Ledwina, T. (1997 b). Data driven smooth tests when the hypothesis is composite. *J. Amer. Statist. Assoc.* **92**, 1094–1104.

Examples

```
# for given vector of 19 numbers
z = c(13.41, 6.04, 1.26, 3.67, -4.54, 2.92, 0.44, 12.93, 6.77, 10.09,
      4.10, 4.04, -1.97, 2.17, -5.38, -7.30, 4.75, 5.63, 8.84)
ddst.norm.test(z, compute.p=TRUE)

# H0 is true
z = rnorm(80)
ddst.norm.test(z, compute.p=TRUE)

# H0 is false
z = rexp(80,4)
ddst.norm.test(z, B=5000, compute.p=TRUE)
```

ddst.uniform.test *Data Driven Smooth Test for Uniformity*

Description

Performs data driven smooth tests for simple hypothesis of uniformity on $[0,1]$.

Usage

```
ddst.uniform.test(x, base = ddst.base.legendre, c = 2.4, B = 1000, compute.p = F, D
```

Arguments

<code>x</code>	a (non-empty) numeric vector of data values.
<code>base</code>	a function which returns orthogonal system, might be <code>ddst.base.legendre</code> for Legendre polynomials or <code>ddst.base.cos</code> for cosine system, see package description.
<code>c</code>	a parameter for model selection rule, see package description.
<code>B</code>	an integer specifying the number of replicates used in p-value computation.
<code>compute.p</code>	a logical value indicating whether to compute a p-value.
<code>Dmax</code>	an integer specifying the maximum number of coordinates, only for advanced users.
<code>...</code>	further arguments.

Details

Embedding null model into the original exponential family introduced by Neyman (1937) leads to the information matrix I being identity and smooth test statistic with k components $W_k = [1/\sqrt{n} \sum_{j=1}^k \sum_{i=1}^n \phi_j(Z_i)]^2$, where ϕ_j is j th degree normalized Legendre polynomial on $[0,1]$ (default value of parameter base = 'ddst.base.legendre'). Alternatively, in our implementation, cosine system can be selected (base = 'ddst.base.cos'). For details see Ledwina (1994) and Inglot and Ledwina (2006).

An application of the pertaining selection rule T for choosing k gives related 'ddst.uniform.test()' based on statistic W_T .

Similar approach applies to testing goodness-of-fit to any fully specified continuous distribution function F . For this purpose it is enough to apply the above solution to transformed observations $F(z_1), \dots, F(z_n)$.

For more details see: <http://www.biecek.pl/R/ddst/description.pdf>.

Value

An object of class `htest`

<code>statistic</code>	the value of the test statistic.
<code>parameter</code>	the number of choosen coordinates (k).
<code>method</code>	a character string indicating the parameters of performed test.
<code>data.name</code>	a character string giving the name(s) of the data.
<code>p.value</code>	the p-value for the test, computed only if <code>compute.p=T</code> .

Author(s)

Przemyslaw Biecek and Teresa Ledwina

References

- Inglot, T., Ledwina, T. (2006). Towards data driven selection of a penalty function for data driven Neyman tests. *Linear Algebra and its Appl.* **417**, 579–590.
- Ledwina, T. (1994). Data driven version of Neyman's smooth test of fit. *J. Amer. Statist. Assoc.* **89** 1000-1005.
- Neyman, J. (1937). 'Smooth test' for goodness of fit. *Skand. Aktuarietidskr.* **20**, 149-199.

Examples

```
# H0 is true
z = runif(80)
ddst.uniform.test(z, compute.p=TRUE)

# known fixed alternative
z = rnorm(80,10,16)
ddst.uniform.test(pnorm(z, 10, 16), compute.p=TRUE)

# H0 is false
```

```
z = rbeta(80,4,2)
(t = ddst.uniform.test(z, compute.p=TRUE))
t$p.value
```

Index

*Topic **h**test

- ddst-package, 1
- ddst.exp.test, 4
- ddst.extr.test, 6
- ddst.norm.test, 8
- ddst.uniform.test, 10

*Topic **n**onparametric

- ddst-package, 1
- ddst.exp.test, 4
- ddst.extr.test, 6
- ddst.norm.test, 8
- ddst.uniform.test, 10

*Topic **p**ackage

- ddst-package, 1

- ddst (*ddst-package*), 1
- ddst-package, 1
- ddst.base.cos (*ddst-package*), 1
- ddst.base.legendre
(*ddst-package*), 1
- ddst.exp.Nk (*ddst.exp.test*), 4
- ddst.exp.test, 4
- ddst.extr.Nk (*ddst.extr.test*), 6
- ddst.extr.test, 6
- ddst.IIC (*ddst-package*), 1
- ddst.norm.Nk (*ddst.norm.test*), 8
- ddst.norm.test, 8
- ddst.phi (*ddst-package*), 1
- ddst.uniform.Nk
(*ddst.uniform.test*), 10
- ddst.uniform.test, 10