Package ‘diptest’

December 5, 2016

Version 0.75-7

Date 2015-06-07

Title Hartigan's Dip Test Statistic for Unimodality - Corrected

Description Compute Hartigan's dip test statistic for unimodality /
multimodality and provide a test with simulation based p-values, where
the original public code has been corrected.

Maintainer Martin Maechler <maechler@stat.math.ethz.ch>

BuildResaveData no

Author Martin Maechler (originally from Fortran and S-plus by Dario Ringach, NYU.edu)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-12-05 18:28:47

R topics documented:

dip *Compute Hartigans' Dip Test Statistic for Unimodality*

Description

Computes Hartigans' dip test statistic for testing unimodality, and additionally the modal interval.

Usage

```
dip(x, full.result = FALSE, min.is.0 = FALSE, debug = FALSE)
```

Arguments

`x`	numeric; the data.
`full.result`	logical or string; `dip(., full.result=TRUE)` returns the full result list; if `"all"` it additionally uses the `mn` and `mj` components to compute the initial GCM and LCM, see below.
`min.is.0`	logical indicating if the **min**imal value of the dip statistic D_n can be zero or not. Arguably should be set to `TRUE` for internal consistency reasons, but is false by default both for continuity and backwards compatibility reasons, see the examples below.
`debug`	logical; if true, some tracing information is printed (from the C routine).

Value

depending on `full.result` either a number, the dip statistic, or an object of class `"dip"` which is a `list` with components

`x`	the sorted `unname()`d data.
`n`	`length(x)`.
`dip`	the dip statistic
`lo.hi`	indices into `x` for lower and higher end of modal interval
`xl, xu`	lower and upper end of modal interval
`gcm, lcm`	(last used) indices for **g**reatest **c**onvex **m**inorant and the **l**east **c**oncave **m**ajorant.
`mn, mj`	index vectors of length `n` for the GC minorant and the LC majorant respectively.

For "full" results of class `"dip"`, there are `print` and `plot` methods, the latter with its own manual page.

Note

For $n \leq 3$ where `n <- length(x)`, the dip statistic D_n is always the same minimum value, $1/(2n)$, i.e., there's no possible dip test. Note that up to May 2011, from Hartigan's original Fortran code, `Dn` was set to zero, when all `x` values were identical. However, this entailed discontinuous behavior, where for arbitrarily close data $\tilde{x}$, $D_n(\tilde{x}) = \frac{1}{2n}$.

Yong Lu `<lyongu+@cs.cmu.edu>` found in Oct 2003 that the code was not giving symmetric results for mirrored data (and was giving results of almost 1, and then found the reason, a misplaced '`")"`' in the original Fortran code. This bug has been corrected for diptest version 0.25-0.

Nick Cox (Durham Univ.) said (on March 20, 2008 on the Stata-list):
As it comes from a bimodal husband-wife collaboration, the name perhaps should be *"Hartigan-Hartigan dip test"*, but that does not seem to have caught on. Some of my less statistical colleagues would sniff out the hegemony of patriarchy there, although which Hartigan is being overlooked is not clear.

Martin Maechler, as a Swiss, and politician, would say:
Let's find a compromise, and call it *"Hartigans' dip test"*, so we only have to adapt orthography (:-).

Author(s)

Martin Maechler `<maechler@stat.math.ethz.ch>`, 1994, based on S (S-PLUS) and C code donated from Dario Ringach `<dario@wotan.cns.nyu.edu>` who had applied `f2c` on the original Fortran code available from Statlib.

In Aug.1993, recreated and improved Hartigans' "P-value" table, which later became `qDiptab`.

References

P. M. Hartigan (1985) Computation of the Dip Statistic to Test for Unimodality; *Applied Statistics (JRSS C)* **34**, 320–325.
Corresponding (buggy!) Fortran code of 'AS 217' available from Statlib, `http://lib.stat.cmu.edu/apstat/217`

J. A. Hartigan and P. M. Hartigan (1985) The Dip Test of Unimodality; *Annals of Statistics* **13**, 70–84.

See Also

`dip.test` to compute the dip *and* perform the unimodality test, based on P-values, interpolated from `qDiptab`; `isoreg` for isotonic regression.

Examples

```
data(statfaculty)
plot(density(statfaculty))
rug(statfaculty, col="midnight blue"); abline(h=0, col="gray")
dip(statfaculty)
(dS <- dip(statfaculty, full = TRUE, debug = TRUE))
plot(dS)
## even more output -- + plot showing "global" GCM/LCM:
(dS2 <- dip(statfaculty, full = "all", debug = 3))
```

```
plot(dS2)

data(faithful)
fE <- faithful$eruptions
plot(density(fE))
rug(fE, col="midnight blue"); abline(h=0, col="gray")
dip(fE, debug = 2) ## showing internal work
(dE <- dip(fE, full = TRUE)) ## note the print method
plot(dE, do.points=FALSE)

data(precip)
plot(density(precip))
rug(precip, col="midnight blue"); abline(h=0, col="gray")
str(dip(precip, full = TRUE, debug = TRUE))

##-----------------  The  'min.is.0' option :  ---------------------

##' dip(.) continuity and 'min.is.0' exploration:
dd <- function(x, debug=FALSE) {
   x_ <- x ; x_[1] <- 0.9999999999 * x[1]
   rbind(dip(x , debug=debug),
         dip(x_, debug=debug),
         dip(x , min.is.0=TRUE, debug=debug),
         dip(x_, min.is.0=TRUE, debug=debug), deparse.level=2)
}

dd( rep(1, 8) ) # the 3rd one differs ==> min.is.0=TRUE is *dis*continuous
dd( 1:7 )       # ditto

dd( 1:7, debug=TRUE)
## border-line case ..
dd( 1:2, debug=TRUE)

## Demonstrate that  'min.is.0 = TRUE'  does not change the typical result:
B.sim <- 1000 # or larger
D5  <- {set.seed(1); replicate(B.sim, dip(runif(5)))}
D5. <- {set.seed(1); replicate(B.sim, dip(runif(5), min.is.0=TRUE))}
stopifnot(identical(D5, D5.), all.equal(min(D5), 1/(2*5)))
hist(D5, 64); rug(D5)

D8  <- {set.seed(7); replicate(B.sim, dip(runif(8)))}
D8. <- {set.seed(7); replicate(B.sim, dip(runif(8), min.is.0=TRUE))}
stopifnot(identical(D8, D8.))
```

`dip.test` *Hartigans' Dip Test for Unimodality*

Description

Compute Hartigans' dip statistic D_n, and its p-value for the test for unimodality, by interpolating tabulated quantiles of $\sqrt{n}D_n$.

For $X_i \sim F, i.i.d.$, the null hypothesis is that F is a unimodal distribution. Consequently, the test alternative is non-unimodal, i.e., at least bimodal. Using the language of medical testing, you would call the test "Test for **Multi**modality".

Usage

```
dip.test(x, simulate.p.value = FALSE, B = 2000)
```

Arguments

`x`	numeric vector; sample to be tested for unimodality.
`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.

Details

If `simulate.p.value` is `FALSE`, the p-value is computed via linear interpolation (of $\sqrt{n}D_n$) in the `qDiptab` table. Otherwise the p-value is computed from a Monte Carlo simulation of a uniform distribution (`runif(n)`) with `B` replicates.

Value

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

`statistic`	the dip statistic D_n, i.e., `dip(x)`.
`p.value`	the p-value for the test, see details.
`method`	character string describing the test, and whether Monte Carlo simulation was used.
`data.name`	a character string giving the name(s) of the data.

Note

see also the package vignette, which describes the procedure in more details.

Author(s)

Martin Maechler

References

see those in `dip`.

See Also

For goodness-of-fit testing, notably of continuous distributions, `ks.test`.

Examples

```
## a first non-trivial case
(d.t <- dip.test(c(0,0, 1,1))) # "perfect bi-modal for n=4" --> p-value = 0
stopifnot(d.t$p.value == 0)

data(statfaculty)
plot(density(statfaculty)); rug(statfaculty)
(d.t <- dip.test(statfaculty))

x <- c(rnorm(50), rnorm(50) + 3)
plot(density(x)); rug(x)
## border-line bi-modal ...  BUT (most of the times) not significantly:
dip.test(x)
dip.test(x, simulate=TRUE, B=5000)

## really large n -- get a message
dip.test(runif(4e5))
```

---

exHartigan *Hartigan's Artificial n-modal Example Data Set*

Description

63 (integer) numbers; unimodal or bimodal, that's the question.

This is now *deprecated*. Please use `statfaculty` instead!

Examples

```
data(exHartigan)
plot(dH <- density(exHartigan))
rug(exHartigan)# should jitter
```

---

plot.dip *Plot a dip() Result, i.e., Class "dip" Object*

Description

Plot method for `"dip"` objects, i.e., the result of `dip(., full.result=TRUE)` or similar.

Note: We may decide to enhance the plot in the future, possibly not entirely back-compatibly.

Usage

```
## S3 method for class 'dip'
plot(x, do.points = (n < 20),
     colG = "red3", colL = "blue3", colM = "forest green",
     col.points = par("col"), col.hor = col.points,
     doModal = TRUE, doLegend = TRUE, ...)
```

Arguments

`x` — an R object of `class` `"dip"`, i.e., typically the result of `dip(., full.result= FF)` where `FF` is `TRUE` or a string such as `"all"`.

`do.points` — logical indicating if the ECDF plot should include points; passed to `plot.ecdf`.

`colG, colL, colM` — the colors to be used in the graphics for the **G**reatest convex minorant, the **L**east concave majorant, and the **M**odal interval, respectively.

`col.points, col.hor` — the color of points or horizontal lines, respectively, simply passed to `plot.ecdf`.

`doModal` — logical indicating if the modal interval $[x_L, x_U]$ should be shown.

`doLegend` — logical indicating if a legend should be shown.

`...` — further optional arguments, passed to `plot.ecdf`.

Author(s)

Martin Maechler

See Also

`dip`, also for examples; `plot.ecdf`.

`qDiptab` — *Table of Quantiles from a Large Simulation for Hartigan's Dip Test*

Description

Whereas Hartigan(1985) published a table of empirical percentage points of the dip statistic (see `dip`) based on N=9999 samples of size n from $U[0, 1]$, our table of empirical quantiles is currently based on N=1'000'001 samples for each n.

Format

A numeric matrix where each row corresponds to sample size n, and each column to a probability (percentage) in $[0, 1]$. The dimnames are named `n` and `Pr` and coercable to these values, see the examples. `attr(qDiptab, "N_1")` is $N-1$, such that with `k <- as.numeric(dimnames(qDiptab)$Pr) * attr(qDiptab, "` e.g., `qDiptab[n == 15,]` contains exactly the order statistics $D_{[k]}$ (from the $N+1$ simulated values of `dip(U)`, where `U <- runif(15)`.

Note

Taking N=1'000'001 ensures that all the `quantile(X, p)` used here are exactly order statistics `sort(X)[k]`.

Author(s)

Martin Maechler <maechler@stat.math.ethz.ch>, in its earliest form in August 1994.

See Also

dip, also for the references; dip.test() which performs the hypothesis test, using qDtiptab (and its null hypothesis of a uniform distribution).

Examples

```
data(qDiptab)
str(qDiptab)
## the sample sizes `n' :
dnqd <- dimnames(qDiptab)
(nn <- as.integer(dnqd $n))
## the probabilities:
P.p <- as.numeric(print(dnqd $ Pr))

## This is as "Table 1" in Hartigan & Hartigan (1985) -- but more accurate
ps <- c(1,5,10,50,90,95,99, 99.5, 99.9)/100
tab1 <- qDiptab[nn <= 200,  as.character(ps)]
round(tab1, 4)
```

---

statfaculty *Faculty Quality in Statistics Departments*

Description

Faculty quality in statistics departments was assessed as part of a larger study reported by Scully(1982).

Accidentally, this is also provided as the exHartigan ("**ex**ample of **Hartigan**s"') data set.

Usage

```
data(statfaculty)
```

Format

A numeric vector of 63 (integer) numbers, sorted increasingly, as reported by the reference.

Source

M. G. Scully (1982) Evaluation of 596 programs in mathematics and physical sciences; *Chronicle Higher Educ.* **25** 5, 8–10.

References

J. A. Hartigan and P. M. Hartigan (1985) The Dip Test of Unimodality; *Annals of Statistics* **13**, 70–84.

Examples

```
data(statfaculty)
plot(dH <- density(statfaculty))
rug(jitter(statfaculty))

data(exHartigan)
stopifnot(identical(exHartigan,statfaculty))
```

Index