Package 'GofKmt'

February 21, 2020

Type Package

Title Khmaladze Martingale Transformation Goodness-of-Fit Test

Version 2.1.1

Date 2020-02-13

Author Jiwoong Kim <jwboys26 at gmail.com>

Maintainer Jiwoong Kim <jwboys26@gmail.com>

Description Consider a goodness-of-fit(GOF) problem of testing whether a random sample comes from one sample location-scale model where location and scale parameters are unknown. It is well known that Khmaladze martingale transformation method - which was proposed by Khmaladze (1981) <DOI:10.1137/1126027> - provides asymptotic distribution free test for the GOF problem. This package contains one function: KhmaladzeTrans(). In this version, KhmaladzeTrans() provides test statistic and critical value of GOF test for normal, Cauchy, logistic, Gamma, and Weibull distributions.

Depends R (>= 3.5.0)

License GPL-2

LazyData TRUE

Imports Rcpp (>= 1.0.3), ggplot2, stats, utils, Rsolnp

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.0.2

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-02-21 14:50:02 UTC

R topics documented:

Integration_Tables *Tables of integrations used for fast computation*

Description

A dataset containing one normal and two logistic integration tables

Usage

```
data(Integration_Tables)
```

Details

#'@format A list containing tables of integrations for normal, logistic and Cauchy distributions:

normal.table 769-by-3 table for a normal distribution

logistic.table1 2561-by-3 table for the logistic distribution

logistic.table2 3841-by-1 table for the logistic distribution

cauchy.table 2561-by-3 table for the Cauchy distribution

KhmaladzeTrans *Implementing Khmaladze Martingale Transformation.*

Description

Performs goodness-of-fit test through Khmaladze matringale transformation

Usage

```
KhmaladzeTrans(X, strDist, bGraph = FALSE, nNum = 10)
```

Arguments

X	a random sample of n observations
strDist	the name of the null distribution for the hypothesis test: Normal, Cauchy, or Logistic. Other distributions such as gamma, Weibull and Frechet will be available in later versions.
bGraph	a logical value which specifies whether or not to get the graph of the objective function of the martingale transformation.
nNum	the number of ticks on each segmented interval when drawing the graph of the objective function. The default is 10. Bigger value will result in a smoother graph.

Value

A list of the following values:

opt.x the value of x where the optimum of the objective function - which is also the test statistic - occurs.

test.stat the test statistic obtained through Khmaladze martingale transformation

graph.data a data frame which includes the information of the objective function.

graph a ggplot object which includes the graph of the objective function.

intervals a list of segmented intervals over which the graph of the objective function is defined.

crit.value a vector of critical values for the level of 0.01, 0.025, 0.05, and 0.10

mu the point estimate for the location parameter mu

sigma the point estimate for the scale parameter sigma

References

[1] Khmaladze, E.V., Koul, H.L. (2004). Martingale transforms goodness-of-fit tests in regression models. Ann. Statist., 32. 995-1034

[2] E.V. Khmaladze, H.L. Koul (2009). Goodness-of-fit problem for errors in nonparametric regression: distribution free approach. Ann. Statist., 37(6A) 3165-3185.

[3] Kim, Jiwoong (2020). Implementation of a goodness-of-fit test through Khmaladze martingale transformation.

Examples

```
####################
n = 10
X = rnorm(n, 1,3)    # Generate a random sample of n observations from N(1,3)
strDist = "Normal"
lResult = KhmaladzeTrans(X, strDist, bGraph=TRUE, nNum=10)
KMT_OptimalX = lResult$opt.x
KMT_TestStat = lResult$test.stat
KMT_DM = lResult$graph.data
KMT_Graph = lResult$graph
KMT_Intervals = lResult$intervals
KMT_CriticalValue = lResult$crit.value
KMT_Muhat = lResult$mu
KMT_Sigmahat = lResult$sigma
####################

####################
n = 10
X = rlogis(n, 1,2)  # Generate a random sample of n observations from the logistic distribution
strDist = "Logistic"
lResult = KhmaladzeTrans(X, strDist, bGraph=TRUE, nNum=10)
KMT_OptimalX = lResult$opt.x
KMT_TestStat = lResult$test.stat
KMT_DM = lResult$graph.data
KMT_Graph = lResult$graph
```

```
KMT_Intervals = lResult$intervals
KMT_CriticalValue = lResult$crit.value
KMT_Muhat = lResult$mu
KMT_Sigmahat = lResult$sigma
#####################
#####################
n = 10
X = rcauchy(n, 0,1)  # Generate a random sample of n observations from Cauchy distribution
strDist = "Cauchy"
lResult = KhmaladzeTrans(X, strDist, bGraph=TRUE, nNum=10)
KMT_OptimalX = lResult$opt.x
KMT_TestStat = lResult$test.stat
KMT_DM = lResult$graph.data
KMT_Graph = lResult$graph
KMT_Intervals = lResult$intervals
KMT_CriticalValue = lResult$crit.value
KMT_Muhat = lResult$mu
KMT_Sigmahat = lResult$sigma
#####################
```

Index