

Package: KMT (via r-universe)

May 26, 2026

Type Package

Title Khmaladze Martingale Transformation Goodness-of-Fit Test

Version 1.0.1

Description Consider a goodness-of-fit 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 proposed by Khmaladze (1981) <[doi:10.1137/1126027](https://doi.org/10.1137/1126027)> provides asymptotic distribution free test. This package provides test statistic and critical value of the test for normal, Cauchy, and logistic distributions. This package used the main algorithm proposed by Kim (2020) <[doi:10.1007/s00180-020-00971-7](https://doi.org/10.1007/s00180-020-00971-7)> and tests for other distributions will be available at the later version.

License GPL-2

Encoding UTF-8

SystemRequirements GNU make

Depends R (>= 3.5)

Imports Rcpp (>= 1.0.10), ggplot2, stats, utils, Rsolnp, graphics, gumbel

LinkingTo Rcpp, RcppArmadillo

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.1

Config/pak/sysreqs make

Repository <https://jwboys26.r-universe.dev>

Date/Publication 2026-05-19 17:52:56 UTC

RemoteUrl <https://github.com/jwboys26/kmt>

RemoteRef HEAD

RemoteSha 5f92471985daa8b8c3088713cba3905dbaf9c35b

Contents

| | |
|-----------------------------|-----------|
| AD | 2 |
| BM | 3 |
| CvM | 3 |
| Distr_Information | 4 |
| DrawUnz | 5 |
| KS | 7 |
| Run_KMT | 8 |
| Index | 10 |

| | |
|----|----------------------------|
| AD | <i>Run other GOF tests</i> |
|----|----------------------------|

Description

Run the Anderson-Darling (AD) test.

Usage

AD(Z, Distr)

Arguments

Z a normalized random sample of n observations.
 Distr a null distribution of the GOF test. It should be one of "Normal", "Logistic", "Cauchy", or "Gumbel"

Value

A test statistic of the AD test.

Examples

```
n=10
mu=2
sigma=1

X = rnorm(n, mu, sigma)

muhat = mean(X)
sighat = sd(X)

Z = (X-muhat)/sighat
null_distr="Normal"
AD_teststat = AD(Z, null_distr)
AD_teststat
```

BM *Brownian Motion*

Description

Get a d-dimension Brownian motion B(t) and its graph

Usage

BM(x = 0, t = 1, n = 10, d = 1)

Arguments

x a starting point of the Brownian motion, B(0).
 t end of time, that is, $0 = t_0 < t_1 < \dots < t_n = t$
 n the last index of the time, that is, $t_n = t$
 d a dimension of the Brownian motion: 1 or 2.

Value

A list of

tn t_n

Bt1 a vector of the Brownian motion ($B_1(t_0, \dots, B_1(t_n))$)

Bt2 a vector of the Brownian motion ($B_2(t_0, \dots, B_2(t_n))$)

Examples

```
#### Generate an 2-dimensional Brownian motion and its graph
x0=c(0,0)
lst = BM(x=x0, t=1, n=10, d=2)
```

CvM *Run other GOF tests*

Description

Run the Cramer-von Mises (CvM) and Watson tests.

Usage

CvM(Z, Distr)

Arguments

`Z` a normalized random sample of n observations.
`Distr` a null distribution of the GOF test. It should be one of "Normal", "Logistic", "Cauchy", or "Gumbel"

Value

A vector of the CvM and Watson test statistics.

Examples

```
n=10
mu=2
sigma=1

X = rnorm(n, mu, sigma)

muhat = mean(X)
sighat = sd(X)

Z = (X-muhat)/sighat
null_distr="Normal"
Vec = CvM(Z, null_distr)
CvM_teststat = Vec[1]
Watson_teststat = Vec[2]

CvM_teststat;Watson_teststat
```

Distr_Information *Information of distribution*

Description

Obtain all details regarding the null distribution such as $c1(x)$.

Usage

```
Distr_Information(x, strDistr)
```

Arguments

`x` a real number
`strDistr` a null distribution: "Normal", "Logistic", "Cauchy"

Value

a list of the following values:

fl $f(x)$

F1 $F(x)$

re $r(x)$. Logistic only

Re $R(x)$. Logistic only

phix $\phi(x)$

v0 v_0

v1 $v_1(x)$

v2 $v_2(x)$

c0 $c_0(x)$

c1 $c_1(x)$

c2 $c_2(x)$

Gamma $\Gamma(x)$

s1 $s_1(x)$

s2 $s_2(x)$

s3 $s_3(x)$

S1 $S_1(x)$

S2 $S_2(x)$

S3 $S_3(x)$

Examples

```
x = 1.2
lst= Distr_Information(x, "Normal")
phix = lst$phix
phix
c0 = lst$c0
c0
```

Description

Draw the graph of $|U[n](z)|$.

Usage

```
DrawUnz(
  X,
  strDistr,
  type = "l",
  lty = 1,
  lwd = 1.5,
  col = "red",
  xlim = c(NA, NA),
  ylim = c(NA, NA),
  margin_x = 0.1,
  margin_y = 0.5,
  dGap = 0.01,
  b_abline = FALSE
)
```

Arguments

| | |
|----------|---|
| X | a random sample of n observations |
| strDistr | a null distribution for the hypothesis test: Normal, Cauchy, Logistic, or Gumbel. |
| type | a type of plot. A default is a line. |
| lty | a line type. The default value is a solid. |
| lwd | a line width. |
| col | a line color. |
| xlim | a limit for the x-axis. |
| ylim | a limit for the y-axis. |
| margin_x | a margin of graph in the x-axis. |
| margin_y | a margin of graph in the y-axis. |
| dGap | a length of subintervals of the x-axis. |
| b_abline | a logical value for drawing vertical lines where the discontinuities of the graph happen. |

Value

A list of the following values:

gObj plot of the graph of the supremum of the KMT test statistic, that is, $|U[n](z)|$

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. *Comp. Stat.*, 35(4): 1993-2017

Examples

```
#####
n=20
mu0=2
sigma0=1
X = rnorm(n, mu0, sigma0)
strDistr="Normal"
DrawUnz(X, strDistr, type="l", lty=1, lwd=1.5, col="red",
        xlim = c(-5,5), ylim = c(NA,NA),
        margin_x=0.1, margin_y=0.5, dGap=0.01, b_abline=TRUE)
```

KS

Run other GOF tests

Description

Run the Kolmogorov-Smirnov (KS) and Kuiper tests.

Usage

```
KS(Z, Distr)
```

Arguments

Z a normalized random sample of n observations.
Distr a null distribution of the GOF test. It should be one of "Normal", "Logistic", "Cauchy", or "Gumbel"

Value

A vector of the KS and Kuiper test statistics.

Examples

```
n=10
mu=2
sigma=1

X = rnorm(n, mu, sigma)

muhat = mean(X)
sighat = sd(X)

Z = (X-muhat)/sighat
null_distr="Normal"
Vec = KS(Z, null_distr)
KS_teststat = Vec[1]
Kuiper_teststat = Vec[2]
```

KS_teststat;Kuiper_teststat

Run_KMT

Implementing Khmaladze Martingale Transformation.

Description

Performs goodness-of-fit test through Khmaladze martingale transformation

Usage

```
Run_KMT(
  X,
  strDistr = "Normal",
  bEstimation = FALSE,
  bFast_Estimation = FALSE,
  bParallel = 0,
  nThreads = 16
)
```

Arguments

| | |
|------------------|---|
| X | a random sample of n observations |
| strDistr | a null distribution for the hypothesis test: Normal, Cauchy, Logistic, or Gumbel. |
| bEstimation | a logical value which specifies whether or not to estimate parameters. The default value is TRUE. For FALSE, (μ) and (σ) will be set as 0 and 1, respectively. |
| bFast_Estimation | a logical value which specifies whether or not to use the maximum likelihood estimator ($\hat{\theta}$) for the location and scale parameters. The default value is FALSE. |
| bParallel | a logical value which specifies whether or not to use the parallel computing. The default value is 0 (=FALSE). |
| nThreads | the number of threads when bParallel is TRUE. The default value is 16. |

Value

A list of the following values:

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

test_stat test.stat is the test statistic obtained through Khmaladze martingale transformation.

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. *Comp. Stat.*, 35(4): 1993-2017

Examples

```
#####  
n=20  
mu0=2; sigma0=1  
X = rnorm(n, mu0, sigma0)
```

```
Run_KMT(X, strDistr="Normal")
```

Index

AD, [2](#)

BM, [3](#)

CvM, [3](#)

Distr_Information, [4](#)

DrawUnz, [5](#)

KS, [7](#)

Run_KMT, [8](#)