

Package ‘STFTS’

October 12, 2022

Title Statistical Tests for Functional Time Series

Version 0.1.0

Description A collection of statistical hypothesis tests of functional time series. While it will include more tests when the related literature are enriched, this package contains the following key tests: functional stationarity test, functional trend stationarity test, functional unit root test, to name a few.

License GPL-2

Encoding UTF-8

Language en-US

RoxygenNote 7.1.1

Depends R (>= 2.10)

Imports e1071

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Yichao Chen [aut],
Chi Seng Pun [cre, aut]

Maintainer Chi Seng Pun <cspun@ntu.edu.sg>

Repository CRAN

Date/Publication 2021-08-19 08:10:05 UTC

R topics documented:

dataset_bb	2
dataset_bm	3
dataset_dbb	3
dataset_sbb	4
default_kernel	4
est_eigenvalue	5
fkpsst	6
FKPSS_teststat	7

fst	8
FST_teststat_MN	10
FST_teststat_TN	11
furt	12
partial_sum	14

Index	15
--------------	-----------

dataset_bb	<i>Integrals of Squared Brownian Bridge</i>
------------	---------------------------------------------

Description

Generate a dataset of independent simulated values of $\int_0^1 B^2(t)dt$, where B is a standard Brownian bridge on $[0,1]$.

Generate a dataset of independent simulated values of $\int_0^1 B^2(t)dt$, where B is a standard Brownian bridge on $[0,1]$.

Usage

dataset_bb(NUM)

dataset_bb(NUM)

Arguments

NUM Number of simulated values generated in the dataset.

Value

A vector with length equals to NUM.

A vector with length equals to NUM.

Examples

```
dataset_bb(10)
dataset_bb(10)
```

dataset_bm	<i>Integrals of Squared Brownian Motion</i>
------------	---------------------------------------------

Description

Generate a dataset of independent simulated values of $\int_0^1 W^2(t)dt$, where W is a standard Brownian motion on $[0,1]$.

Usage

```
dataset_bm(NUM)
```

Arguments

NUM Number of simulated values generated in the dataset.

Value

A vector with length equals to NUM.

Examples

```
dataset_bm(10)
```

dataset_dbb	<i>Integrals of Squared De-meaned Brownian Bridge</i>
-------------	-------------------------------------------------------

Description

Generate a dataset of independent simulated values of $\int_0^1 [B(t) - \int_0^1 B(x)dx]^2 dt$, where B is a standard Brownian bridge on $[0,1]$.

Usage

```
dataset_dbb(NUM)
```

Arguments

NUM Number of simulated values generated in the dataset.

Value

A vector with length equals to NUM.

Examples

```
dataset_dbb(10)
```

dataset_sbb	<i>Integrals of Squared Second-level Brownian Bridge</i>
-------------	----------------------------------------------------------

Description

Generate a dataset of independent simulated values of $\int_0^1 [W(t) + (2x - 3x^2)W(1) + (-6x + 6x^2) \int_0^1 W(x)dx]^2 dt$, where W is a standard Brownian motion on $[0,1]$.

Usage

dataset_sbb(NUM)

Arguments

NUM Number of simulated values generated in the dataset.

Value

A vector with length equals to NUM.

Examples

dataset_sbb(10)

default_kernel	<i>Default Kernel Function</i>
----------------	--------------------------------

Description

Default kernel function used in calculating the estimated long-run covariance function.

Usage

default_kernel(x)

Arguments

x The variable.

Value

The kernel value.

Examples

default_kernel(1)
default_kernel(0)

est_eigenvalue	<i>Eigenvalues of An Estimated Long-run Covariance Function</i>
----------------	-----------------------------------------------------------------

Description

Calculate the eigenvalues of an estimated long-run covariance function.

Usage

```
est_eigenvalue(este, K, h_power, estev)
```

Arguments

este	Estimated errors in the long-run covariance function, inputted in a matrix form with each row representing each observation of the functional error values on equidistant points of any prespecified interval.
K	Kernel function in the estimation of the long-run covariance function.
h_power	Power of sample size 'N' (valued in (0,1)) for the smoothing bandwidth in the kernel function.
estev	Number of the largest eigenvalues chosen in the output.

Value

A vector of first 'estev' largest eigenvalues in descending order.

Examples

```
N<-100
EE<-matrix(rep(0,N*100),ncol=100)
set.seed(1)
for (i in 1:N) {
  temp<-rnorm(100,0,1)
  EE[i,1]<-temp[1]
  for (j in 2:100) {
    EE[i,j]<-EE[i,j-1]+temp[j]
  }
}
est_eigenvalue(este=EE,K=default_kernel,h_power=1/2,estev=10)
```

Description

KPSS test for functional time series with different methods on determining the critical values of the test statistic. The Monte Carlo method (for a large sample size) was constructed in Kokoszka and Young (2016) and the bootstrap-based methods (both the simple bootstrap method and moving block bootstrap method, for a small/moderate sample size) were constructed in Chen and Pun (2019).

Usage

```
fkpsst(
  X,
  ALPHA = 0.05,
  METHOD = "MC",
  K = default_kernel,
  h_power = 2/5,
  est_ev = nrow(X),
  MCNsim = 10000,
  sbb_set = NULL,
  M = 1000,
  b = ceiling((2 * nrow(X))^(1/3))
)
```

Arguments

X	The functional time series being tested, inputted in a matrix form with each row representing each observation of the functional data values on equidistant points of any prespecified interval.
ALPHA	Significance level of the test. The default value is 5%.
METHOD	Method to determine the critical value of the test statistic. The default value 'METHOD="MC"' represents the Monte Carlo method. 'METHOD="SB"' represents the simple bootstrap method and 'METHOD="MBB"' represents the moving block bootstrap method.
K	Kernel function in the estimation of the long-run covariance function, which is only effective in the Monte Carlo method. The default function is 'default_kernel' function in this package.
h_power	Power of sample size 'N' (valued in (0,1)) for the smoothing bandwidth, which is only effective in the Monte Carlo method. The default value is 2/5.
est_ev	Number of the largest eigenvalues chosen to estimate the limiting distribution, which is only effective in the Monte Carlo method. The default value is the sample size 'N'.
MCNsim	Number of Monte Carlo datasets generated in the Monte Carlo method, which is only effective in Monte Carlo method. The default value is 10000.

sbb_set	A vector of independent simulated data generated from the function 'dataset_sbb', which is only effective and essential in Monte Carlo method.
M	Number of bootstrap datasets generated in the bootstrap method, which is only effective in bootstrap methods. The default value is 1000.
b	Block length used in the moving block bootstrap method, which is only effective in the moving block bootstrap method. The default value is $\text{ceiling}((2N)^{(1/3)})$, where 'N' is the sample size.

Value

The result of the test is presented with the value of test statistic and its p-value under the null hypothesis of trend stationarity.

References

- Chen, Y., & Pun, C. S. (2019). A bootstrap-based KPSS test for functional time series. *Journal of Multivariate Analysis*, 174, 104535.
- Kokoszka, P., & Young, G. (2016). KPSS test for functional time series. *Statistics*, 50(5), 957-973.

Examples

```
N<-100
EX<-matrix(rep(0,N*100),ncol=100)
EXI<-rep(0,100)
set.seed(1)
EXI[1]<-rnorm(1,0,1)
for (j in 2:100) {EXI[j]<-EXI[j-1]+rnorm(1,0,1)}
for (i in 1:N) {
temp<-rnorm(100,0,1)
EX[i,1]<-temp[1]
for (j in 2:100) {
EX[i,j]<-EX[i,j-1]+temp[j]
}
}
fkpsst(X=EX,METHOD="SB")
```

FKPSS_teststat

Test Statistic in Functional KPSS Test

Description

Calculate test statistic R_N for functional KPSS test, which Was constructed in Kokoszka and Young (2016) and Chen and Pun (2019).

Usage

```
FKPSS_teststat(X)
```

Arguments

`X` The functional time series being tested, inputted in a matrix form with each row representing each observation of the functional data values on equidistant points of any prespecified interval.

Value

The value of test statistic `R_N` for functional KPSS test.

References

Chen, Y., & Pun, C. S. (2019). A bootstrap-based KPSS test for functional time series. *Journal of Multivariate Analysis*, 174, 104535.

Kokoszka, P., & Young, G. (2016). KPSS test for functional time series. *Statistics*, 50(5), 957-973.

Examples

```
N<-100
EX<-matrix(rep(0,N*100),ncol=100)
set.seed(1)
for (i in 1:N) {
  temp<-rnorm(100,0,1)
  EX[i,1]<-temp[1]
  for (j in 2:100) {
    EX[i,j]<-EX[i,j-1]+temp[j]
  }
}
FST_teststat_TN(EX)
```

 fst

Functional Stationarity Test

Description

Stationarity test for functional time series with different methods on determining the critical values of the test statistic. The Monte Carlo method was constructed in Horvath et al. (2014), while the bootstrap-based methods have not been validated in the literature (although such an option is provided, please use them at your own risk).

Usage

```
fst(
  X,
  STAT = 1,
  ALPHA = 0.05,
  METHOD = "MC",
  K = default_kernel,
  h_power = 2/5,
```



```

est_ev = nrow(X),
MCNsim = 10000,
bb_set = NULL,
dbb_set = NULL,
M = 1000,
b = ceiling((2 * nrow(X))^(1/3))
)

```

Arguments

X	The functional time series being tested, inputted in a matrix form with each row representing each observation of the functional data values on equidistant points of any prespecified interval.
STAT	Test statistic used in the stationarity test. The default value 'STAT=1' represents 'T_N' and 'STAT=2' represents 'M_N' in original paper.
ALPHA	Significance level of the test. The default value is 5%.
METHOD	Method to determine the critical value of the test statistic. The default value 'METHOD="MC"' represents the Monte Carlo method. 'METHOD="SB"' represents the simple bootstrap method and 'METHOD="MBB"' represents the moving block bootstrap method.
K	Kernel function in the estimation of the long-run covariance function, which is only effective in the Monte Carlo method. The default function is 'default_kernel' function in this package.
h_power	Power of sample size 'N' (valued in (0,1)) for the smoothing bandwidth, which is only effective in the Monte Carlo method. The default value is 2/5.
est_ev	Number of the largest eigenvalues chosen to estimate the limiting distribution, which is only effective in the Monte Carlo method. The default value is the sample size 'N'.
MCNsim	Number of Monte Carlo datasets generated in the Monte Carlo method, which is only effective in Monte Carlo method. The default value is 10000.
bb_set	A vector of independent simulated data generated from the function 'dataset_bb', which is only effective and essential in Monte Carlo method with 'STAT=1'.
dbb_set	A vector of independent simulated data generated from the function 'dataset_dbb', which is only effective and essential in Monte Carlo method with 'STAT=2'.
M	Number of bootstrap datasets generated in the bootstrap method, which is only effective in bootstrap methods. The default value is 1000.
b	Block length used in the moving block bootstrap method, which is only effective in the moving block bootstrap method. The default value is $\text{ceiling}((2N)^{1/3})$, where 'N' is the sample size.

Value

The result of the test is presented with the value of test statistic and its p-value under the null hypothesis of local stationarity.

References

Horvath, L., Kokoszka, P., & Rice, G. (2014). Testing stationarity of functional time series. *Journal of Econometrics*, 179(1), 66-82.

Examples

```
N<-100
EX<-matrix(rep(0,N*100),ncol=100)
set.seed(1)
for (i in 1:N) {
  temp<-rnorm(100,0,1)
  EX[i,1]<-temp[1]
  for (j in 2:100) {
    EX[i,j]<-EX[i,j-1]+temp[j]
  }
}
fst(X=EX,METHOD="SB")
```

FST_teststat_MN

Test Statistic M_N in Functional Stationarity Test

Description

Calculate test statistic M_N for functional stationarity test, which was constructed in Horvath et al. (2014).

Usage

```
FST_teststat_MN(X)
```

Arguments

X The functional time series being tested, inputted in a matrix form with each row representing each observation of the functional data values on equidistant points of any prespecified interval.

Value

The value of test statistic M_N calculated for functional stationarity test.

References

Horvath, L., Kokoszka, P., & Rice, G. (2014). Testing stationarity of functional time series. *Journal of Econometrics*, 179(1), 66-82.

Examples

```

N<-100
EX<-matrix(rep(0,N*100),ncol=100)
set.seed(1)
for (i in 1:N) {
temp<-rnorm(100,0,1)
EX[i,1]<-temp[1]
for (j in 2:100) {
EX[i,j]<-EX[i,j-1]+temp[j]
}
}
FST_teststat_MN(EX)

```

FST_teststat_TN

Test Statistic T_N in Functional Stationarity Test

Description

Calculate test statistic T_N for functional stationarity test, which was constructed in Horvath et al. (2014).

Usage

```
FST_teststat_TN(X)
```

Arguments

X The functional time series being tested, inputted in a matrix form with each row representing each observation of the functional data values on equidistant points of any prespecified interval.

Value

The value of test statistic T_N calculated for functional stationarity test.

References

Horvath, L., Kokoszka, P., & Rice, G. (2014). Testing stationarity of functional time series. *Journal of Econometrics*, 179(1), 66-82.

Examples

```

N<-100
EX<-matrix(rep(0,N*100),ncol=100)
set.seed(1)
for (i in 1:N) {
temp<-rnorm(100,0,1)
EX[i,1]<-temp[1]
for (j in 2:100) {

```

```

EX[i,j]<-EX[i,j-1]+temp[j]
}
}
FST_teststat_TN(EX)

```

furt

Functional Unit Root Test

Description

Unit root test for functional time series with different methods on determining the critical values of the test statistic. The Monte Carlo method was constructed in Chen and Pun (2021), while the bootstrap-based methods have not been validated in the literature (although such an option is provided, please use them at your own risk).

Usage

```

furt(
  X,
  ALPHA = 0.05,
  TYPE = "R",
  METHOD = "MC",
  K = default_kernel,
  h_power = 2/5,
  est_ev = nrow(X) - 1,
  MCnsim = 10000,
  bm_set = NULL,
  M = 1000,
  b = ceiling((2 * (nrow(X) - 1))^(1/3))
)

```

Arguments

X	The functional time series being tested, inputted in a matrix form with each row representing each observation of the functional data values on equidistant points of any prespecified interval.
ALPHA	Significance level of the test. The default value is 5%.
TYPE	Type of hypothesis test. The default value 'TYPE="R"' represents the right-tailed test, which is used when the alternative hypothesis is trend stationarity model. 'TYPE="L"' represents the left-tailed test, which is used when the alternative hypothesis is simple stationarity model or AR(1) model.
METHOD	Method to determine the critical value of the test statistic. The default value 'METHOD="MC"' represents the Monte Carlo method. 'METHOD="SB"' represents the simple bootstrap method and 'METHOD="MBB"' represents the moving block bootstrap method.

K	Kernel function in the estimation of the long-run covariance function, which is only effective in the Monte Carlo method. The default function is 'default_kernel' function in this package.
h_power	Power of sample size 'N' (valued in (0,1)) for the smoothing bandwidth, which is only effective in the Monte Carlo method. The default value is 2/5.
est_ev	Number of the largest eigenvalues chosen to estimate the limiting distribution, which is only effective in the Monte Carlo method. The default value is the sample size 'N'.
MNsim	Number of Monte Carlo datasets generated in the Monte Carlo method, which is only effective in Monte Carlo method. The default value is 10000.
bm_set	A vector of independent simulated data generated from the function 'dataset_bm', which is only effective and essential in Monte Carlo method.
M	Number of bootstrap datasets generated in the bootstrap method, which is only effective in bootstrap methods. The default value is 1000.
b	Block length used in the moving block bootstrap method, which is only effective in the moving block bootstrap method. The default value is $\text{ceiling}((2N)^{(1/3)})$, where 'N' is the sample size.

Value

The result of the test is presented with the value of test statistic and its p-value under the null hypothesis of functional random walk.

References

Chen, Y., & Pun, C. S. (2021). Functional Unit Root Test. Available at SSRN.

Examples

```

N<-100
EE<-matrix(rep(0,N*100),ncol=100)
EX<-matrix(rep(0,N*100),ncol=100)
set.seed(1)
for (i in 1:N) {
  temp<-rnorm(100,0,1)
  EE[i,1]<-temp[1]
  for (j in 2:100) {
    EE[i,j]<-EE[i,j-1]+temp[j]
  }
}
EX[1,]<-EE[1,]
for (i in 2:N) {EX[i,]<-EX[i-1]+EE[i,]}
furt(X=EX,METHOD="SB")

```

partial_sum

Partial Sum

Description

Calculate the partial sum matrix of a functional time series.

Usage

```
partial_sum(X)
```

Arguments

X The functional time series being calculated, inputted in a matrix form with each row representing each observation of the functional data values on equidistant points of any prespecified interval.

Value

A square matrix of partial sums of the functional time series. The column (row) dimension of the output matrix is equal to the column dimension of the input matrix.

Examples

```
N<-100
EX<-matrix(rep(0,N*100),ncol=100)
set.seed(1)
for (i in 1:N) {
  temp<-rnorm(100,0,1)
  EX[i,1]<-temp[1]
  for (j in 2:100) {
    EX[i,j]<-EX[i,j-1]+temp[j]
  }
}
partial_sum(EX)
```

Index

dataset_bb, [2](#)
dataset_bm, [3](#)
dataset_dbb, [3](#)
dataset_sbb, [4](#)
default_kernel, [4](#)

est_eigenvalue, [5](#)

FKPSS_teststat, [7](#)
fkpsst, [6](#)
fst, [8](#)
FST_teststat_MN, [10](#)
FST_teststat_TN, [11](#)
furt, [12](#)

partial_sum, [14](#)