

# Package ‘TSEntropies’

January 20, 2025

**Title** Time Series Entropies

**Version** 0.9

**Description** Computes various entropies of given time series. This is the initial version that includes ApEn() and SampEn() functions for calculating approximate entropy and sample entropy. Approximate entropy was proposed by S.M. Pincus in "Approximate entropy as a measure of system complexity", Proceedings of the National Academy of Sciences of the United States of America, 88, 2297-2301 (March 1991). Sample entropy was proposed by J. S. Richman and J. R. Moorman in "Physiological time-series analysis using approximate entropy and sample entropy", American Journal of Physiology, Heart and Circulatory Physiology, 278, 2039-2049 (June 2000). This package also contains FastApEn() and FastSampEn() functions for calculating fast approximate entropy and fast sample entropy. These are newly designed very fast algorithms, resulting from the modification of the original algorithms. The calculated values of these entropies are not the same as the original ones, but the entropy trend of the analyzed time series determines equally reliably. Their main advantage is their speed, which is up to a thousand times higher. A scientific article describing their properties has been submitted to The Journal of Supercomputing and in present time it is waiting for the acceptance.

**Depends** R (>= 3.4.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**NeedsCompilation** yes

**RoxygenNote** 6.1.0.9000

**Author** Jiri Tomcala [aut, cre]

**Maintainer** Jiri Tomcala <jiri.tomcala@vsb.cz>

**Repository** CRAN

**Date/Publication** 2018-10-08 11:00:07 UTC

## Contents

ApEn	2
ApEn_C	3

ApEn_R . . . . .	3
FastApEn . . . . .	4
FastApEn_C . . . . .	4
FastApEn_R . . . . .	5
FastSampEn . . . . .	6
FastSampEn_C . . . . .	6
FastSampEn_R . . . . .	7
SampEn . . . . .	7
SampEn_C . . . . .	8
SampEn_R . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

ApEn	<i>ApEn</i>
------	-------------

---

## Description

This function computes approximate entropy of given time series.

## Usage

```
ApEn(TS, dim = 2, lag = 1, r = 0.2 * sd(TS))
```

## Arguments

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is 0.2*sd(TS)

## Examples

```
timser <- rnorm(2000)
ApEn(timser)
ApEn(timser, r = 0.1*sd(timser))
ApEn(timser, dim = 3, r = 0.1*sd(timser))
```

---

ApEn\_C

*ApEn\_C*

---

### Description

This function computes approximate entropy of given time series. It is implemented in C.

### Usage

```
ApEn_C(TS, dim = 2, lag = 1, r = 0.2 * sd(TS))
```

### Arguments

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is 0.2*sd(TS)

### Examples

```
timser <- rnorm(2000)
ApEn_C(timser)
ApEn_C(timser, r = 0.1*sd(timser))
ApEn_C(timser, dim = 3, r = 0.1*sd(timser))
```

---

ApEn\_R

*ApEn\_R*

---

### Description

This function computes approximate entropy of given time series. It is implemented in R.

### Usage

```
ApEn_R(TS, dim = 2, lag = 1, r = 0.2 * sd(TS))
```

### Arguments

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is 0.2*sd(TS)

**Examples**

```

timser <- rnorm(2000)
ApEn_R(timser)
ApEn_R(timser, r = 0.1*sd(timser))
ApEn_R(timser, dim = 3, r = 0.1*sd(timser))

```

---

FastApEn

*FastApEn*


---

**Description**

This function computes fast approximate entropy of given time series.

**Usage**

```
FastApEn(TS, dim = 2, lag = 1, r = 0.15 * sd(TS))
```

**Arguments**

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is 0.15*sd(TS)

**Examples**

```

timser <- rnorm(2000)
FastApEn(timser)
FastApEn(timser, r = 0.1*sd(timser))
FastApEn(timser, dim = 3, r = 0.1*sd(timser))

```

---

FastApEn\_C

*FastApEn\_C*


---

**Description**

This function computes fast approximate entropy of given time series. It is implemented in C.

**Usage**

```
FastApEn_C(TS, dim = 2, lag = 1, r = 0.15 * sd(TS))
```

**Arguments**

TS - given time series  
dim - dimension of given time series, default value is 2  
lag - downsampling, default value is 1  
r - radius of searched areas, default value is  $0.15 * \text{sd}(\text{TS})$

**Examples**

```
timser <- rnorm(2000)
FastApEn_C(timser)
FastApEn_C(timser, r = 0.1*sd(timser))
FastApEn_C(timser, dim = 3, r = 0.1*sd(timser))
```

---

FastApEn\_R

*FastApEn\_R*

---

**Description**

This function computes fast approximate entropy of given time series. It is implemented in R.

**Usage**

```
FastApEn_R(TS, dim = 2, lag = 1, r = 0.15 * sd(TS))
```

**Arguments**

TS - given time series  
dim - dimension of given time series, default value is 2  
lag - downsampling, default value is 1  
r - radius of searched areas, default value is  $0.15 * \text{sd}(\text{TS})$

**Examples**

```
timser <- rnorm(2000)
FastApEn_R(timser)
FastApEn_R(timser, r = 0.1*sd(timser))
FastApEn_R(timser, dim = 3, r = 0.1*sd(timser))
```

FastSampEn

*FastSampEn*

---

**Description**

This function computes fast sample entropy of given time series.

**Usage**

```
FastSampEn(TS, dim = 2, lag = 1, r = 0.15 * sd(TS))
```

**Arguments**

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is 0.15*sd(TS)

**Examples**

```
timser <- rnorm(2000)
FastSampEn(timser)
FastSampEn(timser, r = 0.1*sd(timser))
FastSampEn(timser, dim = 3, r = 0.1*sd(timser))
```

---

FastSampEn\_C*FastSampEn\_C*

---

**Description**

This function computes fast sample entropy of given time series. It is implemented in C.

**Usage**

```
FastSampEn_C(TS, dim = 2, lag = 1, r = 0.15 * sd(TS))
```

**Arguments**

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is 0.15*sd(TS)

**Examples**

```

timser <- rnorm(2000)
FastSampEn_C(timser)
FastSampEn_C(timser, r = 0.1*sd(timser))
FastSampEn_C(timser, dim = 3, r = 0.1*sd(timser))

```

---

FastSampEn\_R

*FastSampEn\_R*


---

**Description**

This function computes fast sample entropy of given time series. It is implemented in R.

**Usage**

```
FastSampEn_R(TS, dim = 2, lag = 1, r = 0.15 * sd(TS))
```

**Arguments**

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is 0.15*sd(TS)

**Examples**

```

timser <- rnorm(2000)
FastSampEn_R(timser)
FastSampEn_R(timser, r = 0.1*sd(timser))
FastSampEn_R(timser, dim = 3, r = 0.1*sd(timser))

```

---

SampEn

*SampEn*


---

**Description**

This function computes sample entropy of given time series.

**Usage**

```
SampEn(TS, dim = 2, lag = 1, r = 0.2 * sd(TS))
```

**Arguments**

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is $0.2 * sd(TS)$

**Examples**

```
timser <- rnorm(2000)
SampEn(timser)
SampEn(timser, r = 0.1*sd(timser))
SampEn(timser, dim = 3, r = 0.1*sd(timser))
```

---

SampEn\_C

*SampEn\_C*

---

**Description**

This function computes sample entropy of given time series. It is implemented in C.

**Usage**

```
SampEn_C(TS, dim = 2, lag = 1, r = 0.2 * sd(TS))
```

**Arguments**

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is $0.2 * sd(TS)$

**Examples**

```
timser <- rnorm(2000)
SampEn_C(timser)
SampEn_C(timser, r = 0.1*sd(timser))
SampEn_C(timser, dim = 3, r = 0.1*sd(timser))
```



---

SampEn\_R

*SampEn\_R*

---

### Description

This function computes sample entropy of given time series. It is implemented in R.

### Usage

```
SampEn_R(TS, dim = 2, lag = 1, r = 0.2 * sd(TS))
```

### Arguments

TS	- given time series
dim	- dimension of given time series, default value is 2
lag	- downsampling, default value is 1
r	- radius of searched areas, default value is $0.2 * \text{sd}(\text{TS})$

### Examples

```
timser <- rnorm(2000)
SampEn_R(timser)
SampEn_R(timser, r = 0.1*sd(timser))
SampEn_R(timser, dim = 3, r = 0.1*sd(timser))
```

# Index

## \* **ApEn**

ApEn, 2  
ApEn\_C, 3  
ApEn\_R, 3  
FastApEn\_R, 5  
SampEn\_R, 9

## \* **C**

ApEn\_C, 3  
FastApEn\_C, 4  
FastSampEn\_C, 6  
SampEn\_C, 8

## \* **FastApEn**

FastApEn, 4  
FastApEn\_C, 4

## \* **FastSampEn**

FastSampEn, 6  
FastSampEn\_C, 6  
FastSampEn\_R, 7

## \* **R**

ApEn\_R, 3  
FastApEn\_R, 5  
FastSampEn\_R, 7  
SampEn\_R, 9

## \* **SampEn**

SampEn, 7  
SampEn\_C, 8

## \* **approximate**

ApEn, 2  
ApEn\_C, 3  
ApEn\_R, 3  
FastApEn, 4  
FastApEn\_C, 4  
FastApEn\_R, 5  
SampEn\_R, 9

## \* **entropy**

ApEn, 2  
ApEn\_C, 3  
ApEn\_R, 3  
FastApEn, 4

FastApEn\_C, 4  
FastApEn\_R, 5  
FastSampEn, 6  
FastSampEn\_C, 6  
FastSampEn\_R, 7  
SampEn, 7  
SampEn\_C, 8  
SampEn\_R, 9

## \* **fast**

FastApEn, 4  
FastApEn\_C, 4  
FastSampEn, 6  
FastSampEn\_C, 6  
FastSampEn\_R, 7

## \* **sample**

FastSampEn, 6  
FastSampEn\_C, 6  
FastSampEn\_R, 7  
SampEn, 7  
SampEn\_C, 8

ApEn, 2  
ApEn\_C, 3  
ApEn\_R, 3

FastApEn, 4  
FastApEn\_C, 4  
FastApEn\_R, 5  
FastSampEn, 6  
FastSampEn\_C, 6  
FastSampEn\_R, 7

SampEn, 7  
SampEn\_C, 8  
SampEn\_R, 9