

Showcasing periodogram(): estimate the spectral density of a time series

Alexander Klaus

The spectral density of a stationary process shows how the variance of the time series is distributed over different frequencies. As a fundamental tool of time series analysis, the `periodogram` function is used to estimate the spectral density of a stationary time series.

Example 1: Basic Usage of the Function

First, generate some random time series data to work with:

```
devtools::load_all()

set.seed(123)
X <- rnorm(100)
```

Next, select a numerical value from the interval $(-\pi, \pi]$, which represents the frequency at which the periodogram is evaluated:

```
lambda <- pi / 4
```

Now, calculate the periodogram for the time series with the specified frequency:

```
periodogram(X, lambda)
```

```
## [1] 1.392404
```

What is happening inside the function:

- *Input Validation:* The function checks whether the time series X has been passed in the correct format and λ is a numerical value from the interval $(-\pi, \pi]$.
- Each value X_t of the time series X is multiplied by $e^{-it\lambda}$, where t is the time index.
- The sum of these products is computed.
- The absolute value of this sum is calculated and raised to the power of 2.
- The result is divided by the length of the time series n and returned as the periodogram value.

Example 2: Periodogram for a Simple Sine Curve

Create a simple sine curve sample:

```
X <- sin(2 * pi * 1:100 / 100)
```

Define the frequency of the sine curve:

```
lambda <- 2 * pi / 100
```

Compute the periodogram:

```
periodogram(X, lambda)
```

```
## [1] 25
```

Example 3: Periodogram for a Complex Exponential Curve

Generate a complex exponential curve sample:

```
X <- exp(2 * pi * 1i * 1:100 / 100)
```

Set the frequency of the complex exponential:

```
lambda <- 2 * pi / 100
```

Compute the periodogram:

```
periodogram(X, lambda)
```

```
## [1] 100
```

Example 4: Incorrect inputs

If the input do not correspond to the correct format, appropriate error messages are printed to inform the user what is wrong with their input. Here are several examples of how incorrect inputs are handled:

The time series data must be passed in an atomic vector:

```
X <- list(1, 2, 3, 4, 5)
lambda <- pi / 2
```

```
periodogram(X, lambda)
```

```
## Error in periodogram(X, lambda): X must be an atomic vector
```

The values of this atomic vector must be numeric or complex:

```
X <- c(1, 2, "3", 4, "5i")
lambda <- pi / 2
```

```
periodogram(X, lambda)
```

```
## Error in periodogram(X, lambda): The values of X must be numeric or complex
```

In addition, the atomic vector may not contain any NAs:

```
X <- c(1, NA, 2, 3, NA, 5)
lambda <- pi / 2
```

```
periodogram(X, lambda)
```

```
## Error in periodogram(X, lambda): X may not contain NAs
```

The lambda parameter must be a numerical value:

```
X <- 1:10
lambda <- "0"
```

```
periodogram(X, lambda)
```

```
## Error in periodogram(X, lambda): lambda must be numeric
```

Also lambda must lie within the interval $(-\pi, \pi]$:

```
X <- 1:10
lambda <- 2*pi
```

```
periodogram(X, lambda)
```

```
## Error in periodogram(X, lambda): lambda must be from the interval (-pi, pi]
```