# Package ‘spectral’

January 14, 2021

**Version** 0.8.1

**Date** 2021-01-13

**Title** A R package for carrying out the spectral analysis of univariate time series

**Author** Peter F. Craigmile

**Maintainer** Peter F. Craigmile <pfc@stat.osu.edu>

**Depends** R (>= 3.0), statmod

**Description** A R package for carrying out the spectral analysis of univariate time series

**License** GPL-3

**URL** http://www.r-project.org, http://www.stat.osu.edu/~pfc/

## R topics documented:


acvs.from.sdf . . . . . . . . . . 2
ar.sdf . . . . . . . . . . 3
ar2.example . . . . . . . . . . 3
ar4.example . . . . . . . . . . 4
arma.sdf . . . . . . . . . . 4
arma.var.from.sdf . . . . . . . . . . 5
closest.Fourier.frequency . . . . . . . . . . 5
closest.Fourier.index . . . . . . . . . . 6
complex.chol . . . . . . . . . . 6
cosine.taper . . . . . . . . . . 7
create.taper . . . . . . . . . . 7
decibels . . . . . . . . . . 8
direct.spectral . . . . . . . . . . 8
Dirichlet.kernel . . . . . . . . . . 9
dpss.taper . . . . . . . . . . 9
Fejer.kernel . . . . . . . . . . 10
fft1 . . . . . . . . . . 11
Fisher.test . . . . . . . . . . 11
Fourier.frequencies . . . . . . . . . . 12
h . . . . . . . . . . 12
Hanning.taper . . . . . . . . . . 13
idecibels . . . . . . . . . . 13
ma.sdf . . . . . . . . . . 14
mean.spect . . . . . . . . . . 14

multitaper.effective.bw . . . . . . 15
periodicity . . . . . . 15
periodogram . . . . . . 16
plot.spect . . . . . . 16
rectangular.taper . . . . . . 17
renormalize.taper . . . . . . 18
sine.taper . . . . . . 18
spect . . . . . . 19
spect.max . . . . . . 19
squared.gain . . . . . . 20
Thomson.crit . . . . . . 20
Thomson.stat . . . . . . 21
transfer . . . . . . 21

**Index** **23**

---

`acvs.from.sdf` *Calculate the acvs via a Fourier transform of the SDF*

---

### Description

Calculate the ACVS from lags 0 to lag 'max.lag', given a function specifying the SDF, 'the.sdf'.

### Usage

```
acvs.from.sdf(max.lag, the.sdf, use.integrate = TRUE, num.weights = 200, gq, ...)
```

### Arguments

| | |
|---|---|
| `max.lag` | calculate the acvs from 0 to the lag 'max.lag'. |
| `the.sdf` | an R function defining the SDF |
| `use.integrate` | if use.integrate is TRUE use the 'integrate' function to calculate the integral. Otherwise use Gauss-Legendre quadrature from the 'statmod' library using the 'gq' object, if available, or creating a new object with 'num.weights' weights. |
| `num.weights` | the number of weights to use for the Gauss-Legendre quadrature |
| `gq` | a Gauss-Legendre quadrature from the 'statmod' library |
| `...` | Other arguments are passed to 'the.sdf' function |

### Value

a vector of max.lag+1 acvs values.

### Author(s)

Peter F. Craigmile

## ar.sdf — *Calculate the spectral density function (SDF) for an AR process*

### Usage

```
ar.sdf(freqs, phi, sigma2=1, delta.t=1)
```

### Arguments

| | |
|---|---|
| `freqs` | frequencies to evaluate the SDF at |
| `phi` | AR parameters |
| `sigma2` | innovation variance |
| `delta.t` | sampling interval |

### Value

vector of SDF values

### Author(s)

Peter F. Craigmile

## ar2.example — *Coefficients for an example AR(2) process*

### Usage

```
ar2.example
```

### Format

A vector containing the coefficients of the AR(2) process.

### Source

Equation 45 of Percival and Walden (1993).

### References

Percival and Walden (1993) *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press.

---

`ar4.example` *Coefficients for an example AR(4) process*

---

### Usage

```
ar4.example
```

### Format

A vector containing the coefficients of the AR(4) process.

### Source

Equation 46a of Percival and Walden (1993).

### References

Percival and Walden (1993) *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press.

---

`arma.sdf` *Calculate the spectral density function (SDF) for an ARMA process*

---

### Usage

```
arma.sdf(freqs, phi, theta, sigma2=1, delta.t=1)
```

### Arguments

| | |
|---|---|
| `freqs` | frequencies to evaluate the SDF at |
| `phi` | AR parameters |
| `theta` | MA parameters |
| `sigma2` | innovation variance |
| `delta.t` | sampling interval |

### Value

vector of SDF values

### Author(s)

Peter F. Craigmile

## arma.var.from.sdf — *Calculate the variance of an ARMA process using the spectral density function (SDF)*

### Usage

```
arma.var.from.sdf(phi, theta, sigma2=1, delta.t=1)
```

### Arguments

| | |
|---|---|
| `phi` | AR parameters (may be omitted) |
| `theta` | MA parameters (may be omitted) |
| `sigma2` | innovation variance |
| `delta.t` | sampling interval |

### Value

variance of the ARMA process

### Author(s)

Peter F. Craigmile

## closest.Fourier.frequency — *Calculate closest Fourier frequency to a given frequency*

### Usage

```
closest.Fourier.frequency(freq, N, delta.t=1)
```

### Arguments

| | |
|---|---|
| `freq` | frequency |
| `N` | length of times series |
| `delta.t` | sampling interval |

### Details

Calculates the closest Fourier frequency to 'freq', obtained when performing a spectral analysis on a time series of length 'N', with sampling interval 'delta.t'.

### Author(s)

Peter F. Craigmile

## closest.Fourier.index *Calculate closest Fourier index to a given frequency*

### Usage

```
closest.Fourier.index(freq, N, delta.t=1)
```

### Arguments

| | |
|---|---|
| `freq` | frequency |
| `N` | length of times series |
| `delta.t` | sampling interval |

### Details

Calculates the closest Fourier index to 'freq', obtained when performing a spectral analysis on a time series of length 'N', with sampling interval 'delta.t'.

### Author(s)

Peter F. Craigmile

## complex.chol *Complex-valued Cholesky decomposition*

### Usage

```
complex.chol(x,upper=TRUE)
```

### Arguments

| | |
|---|---|
| `x` | Matrix to calculate the complex-valued Cholesky of |
| `upper` | If true (the default), calculate the upper triangular form, if false calculate the lower triangular form. |

### Value

A matrix containing the complex-valued Cholesky decomposition in either the upper or lower triangular form.

### Author(s)

Peter F. Craigmile

## cosine.taper — *Calculate a 100p% cosine taper*

### Usage

```
cosine.taper(N, p)
```

### Arguments

| | |
|---|---|
| `N` | The length of the time series |
| `p` | A single numeric value between 0 and 1, inclusive |

### Value

A vector of length N containing the cosine (100p)% taper

### Author(s)

Peter F. Craigmile

### References

Percival and Walden (1993) *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press, Equation 209.

## create.taper — *Create a spectral taper of a given type*

### Usage

```
create.taper(taper.name, N, ...)
```

### Arguments

| | |
|---|---|
| `taper.name` | Name of the taper |
| `N` | Length of the taper |
| `...` | Extra arguments to pass to the taper |

### Value

A taper as a vector or matrix (if a multiple taper)

### Author(s)

Peter F. Craigmile

decibels *Convert to decibels*

## Description

Converts a numerical vector to the decibels value

## Usage

```
decibels(x)
```

## Arguments

| | |
|---|---|
| x | a numerical vector |

## Value

The vector of x values converted to decibels

## Author(s)

Peter F. Craigmile

## Examples

```
decibels(1)  # return 0
decibels(10) # returns 10
decibels(0)  # return -Inf
```

direct.spectral *Direct spectral estimate*

## Description

Calculate the tapered direct spectral estimate

## Usage

```
direct.spectral(x, taper.name, delta.t=deltat(x), taper=create.taper(taper.name, length(x), ...),
```

## Arguments

| | |
|---|---|
| x | Time series object (the data) |
| taper.name | The name of the taper |
| delta.t | Sampling interval |
| taper | The taper |
| ... | Optional arguments used to create the taper (if necessary) |

### Details

Calculate the tapered direct spectral estimate of a univariate time series of 'x'.

### Value

An object of class 'spect' containing the direct spectral estimate of 'x' using the taper, 'taper'.

### Author(s)

Peter F. Craigmile

---

`Dirichlet.kernel` *Calculate Dirichlet's kernel*

---

### Usage

```
Dirichlet.kernel(freqs, N)
```

### Arguments

| | |
|---|---|
| `freqs` | frequencies to evaluate at |
| `N` | length of time series |

### Value

Dirichlet's kernel at the evaluated frequencies

### Author(s)

Peter F. Craigmile

### References

Equation 1.3c of Percival and Walden (1993)

---

`dpss.taper` *Calculate a set of dpss tapers*

---

### Usage

```
dpss.taper(N, n.tapers, NW = 4, W = NW/N)
```

### Arguments

| | |
|---|---|
| `N` | The length of the time series |
| `n.tapers` | The number of tapers to calculate |
| `NW` | N times by the bandwidth, W |
| `W` | The bandwidth of the tapers, W |

## Value

If n.tapers is one then the first dpss taper of bandwidth W is returned as a vector. If n.tapers is greater than one, this function returns a matrix of dimension N by n.tapers. Each column of the matrix contains a different dpss taper.

## Author(s)

Peter F. Craigmile

## References

Percival and Walden (1993) *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press.

---

# `Fejer.kernel` *Calculate Fejer's kernel*

---

## Usage

```
Fejer.kernel(freqs, N, delta.t=1)
```

## Arguments

| | |
|---|---|
| `freqs` | frequencies to evaluate at |
| `N` | length of time series |
| `delta.t` | sampling interval |

## Value

Fejer's kernel at the evaluated frequencies

## Author(s)

Peter F. Craigmile

## References

Equation 198b of Percival and Walden (1993)

## fft1 — *Calculate fft of a vector using a starting time index of 1*

### Usage

```
fft1(x)
```

### Arguments

| | |
|---|---|
| x | a numerical vector |

### Value

Calculates

$$J(k/n) = \sum_{t=1}^{n} x_t \; e^{-i2\pi(k/n)t}$$

for $k = 0, ..., (n-1)$, where n=length(x)

### Author(s)

Peter F. Craigmile

## Fisher.test — *Fisher's test for an unknown periodicity*

### Usage

```
Fisher.test(sp, alpha=.05)
```

### Arguments

| | |
|---|---|
| sp | Spectral object to analyze. |
| alpha | Size of the test. |

### Value

hypothesis test ('h.test') object containing the results of Fisher's test for an unknown periodicity.

### Author(s)

Peter F. Craigmile

### References

Percival and Walden (1993) *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press, Section 10.9.

## `Fourier.frequencies` *Calculate all the Fourier frequencies*

### Usage

```
Fourier.frequencies(N, delta.t=1)
```

### Arguments

| | |
|---|---|
| `N` | length of times series |
| `delta.t` | sampling interval |

### Details

Calculates all the Fourier frequencies obtained when performing spectral analysis on a time series of length 'N', with sampling interval 'delta.tt'.

### Author(s)

Peter F. Craigmile

## `h` *Calculate the Hermitian transpose of a matrix*

### Usage

```
h(x)
```

### Arguments

| | |
|---|---|
| `x` | a matrix |

### Value

a matrix, containing the Hermitian transpose of 'x'.

### Author(s)

Peter F. Craigmile

---

`Hanning.taper` *Calculate a Hanning taper*

---

## Usage

```
Hanning.taper(N)
```

## Arguments

| | |
|---|---|
| `N` | The length of the time series |

## Details

The Hanning taper is the same as a 100% cosine taper

## Value

A vector of length N containing the Hanning taper

## Author(s)

Peter F. Craigmile

## References

Percival and Walden (1993) *Spectral Analysis for Physical Applic ations: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge Univ ersity Press, Page 210.

---

`idecibels` *Convert from decibels*

---

## Description

Converts a numerical vector from the decibels value

## Usage

```
idecibels(x)
```

## Arguments

| | |
|---|---|
| `x` | a numerical vector |

## Value

The vector of x values converted from decibels to the original scale

## Author(s)

Peter F. Craigmile

## Examples

```
idecibels(0)  # return 1
idecibels(10) # returns 10
idecibels(-Inf)  # return 0
```

---

ma.sdf *Calculate the spectral density function (SDF) for an MA process*

---

## Usage

```
ma.sdf(freqs, theta, sigma2=1, delta.t=1)
```

## Arguments

| | |
|---|---|
| freqs | frequencies to evaluate the SDF at |
| theta | MA parameters |
| sigma2 | innovation variance |
| delta.t | sampling interval |

## Value

Vector of SDF values

## Author(s)

Peter F. Craigmile

---

mean.spect *Spectral Mean*

---

## Description

Average multiply tapered spectral estimates

## Usage

```
## S3 method for class 'spect'
mean(x, ...)
```

## Arguments

| | |
|---|---|
| x | an item of class 'spect' |
| ... | Extra arguments from the 'mean' function' |

## Details

Average the different spectra in the spect class 'sp' (normally used for multitapering).

### Value

An object of class 'spect' which contains the averaged spectra.

### Author(s)

Peter F. Craigmile

---

## multitaper.effective.bw — *Calculate the effective bandwidth of a set of multitapers*

---

### Usage

```
multitaper.effective.bw(tapers)
```

### Arguments

| | |
|---|---|
| `tapers` | a matrix of multitapers |

### Author(s)

Peter F. Craigmile

---

## periodicity — *Generate periodicity*

---

### Usage

```
periodicity(N, amps, freqs, phases, delta.t)
```

### Arguments

| | |
|---|---|
| `N` | The length of the time series to generate. |
| `amps` | A vector of amplitudes. |
| `freqs` | A vector of frequencies. |
| `phases` | A vector of phases (default is to generate from Uniform[$-\pi$,$\pi$]) |
| `delta.t` | The sampling interval (The default is 1). |

### Value

The periodocity

### Author(s)

Peter F. Craigmile

---

`periodogram` *Periodogram*

---

## Description

Calculate periodogram

## Usage

```
periodogram(x, delta.t=deltat(x))
```

## Arguments

| | |
|---|---|
| `x` | Time series object (the data) |
| `delta.t` | Sampling interval |

## Details

Calculate the periodogram of a univariate time series of ’x’.

## Value

An object of class ’spect’ containing the periodogram of ’x’.

## Author(s)

Peter F. Craigmile

---

`plot.spect` *plot.spect*

---

## Usage

```
## S3 method for class 'spect'
plot(x, taper.num, xlab="Frequency", ylab="Spectrum",
   log.freq=FALSE, add=FALSE, type="l", ...)
```

## Arguments

| | |
|---|---|
| `x` | an object of class ’spect’ |
| `taper.num` | if ’x’ contains many tapers, which taper to show. |
| `xlab` | x label for the figure |
| `ylab` | y label for the figure |
| `log.freq` | if TRUE present the figure on the log frequency scale |
| `add` | if TRUE add to current plot, otherwise create new plot |
| `type` | line type is used by default in this figure |
| `...` | further arguments that can be passed to the plot |

### Details

Plot the object 'sp' of class 'spect' in decibels, showing frequencies on the log scale if log.freq=T.

### Value

No value is returned

### Author(s)

Peter F. Craigmile

---

## rectangular.taper *Calculate a rectangular taper*

---

### Usage

```
rectangular.taper(N)
```

### Arguments

| | |
|---|---|
| N | The length of the time series |

### Details

The rectangular taper is the same a 0% cosine taper

### Value

A vector of length N containing the rectangular taper

### Author(s)

Peter F. Craigmile

### References

Percival and Walden (1993) *Spectral Analysis for Physical Applic ations: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge Univ ersity Press, Page 209.

`renormalize.taper` *Renormalize the taper so that it has unit norm*

### Usage

```
renormalize.taper(taper)
```

### Arguments

| | |
|---|---|
| `taper` | The taper |

### Value

The taper renormalized to have unit norm

### Author(s)

Peter F. Craigmile

`sine.taper` *Calculate a set of sine tapers*

### Usage

```
sine.taper(N, n.tapers)
```

### Arguments

| | |
|---|---|
| `N` | The length of the time series |
| `n.tapers` | The number of tapers to calculate |

### Value

If n.tapers is one then the single sine taper is returned as a vector. If n.tapers is greater than one, this function returns a matrix of dimension N by n.tapers. Each column of the matrix contains a different sine taper.

### Author(s)

Peter F. Craigmile

### References

Percival and Walden (1993) *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press.

## spect — *Create a object of class 'spect'*

### Usage

```
spect(spec, N, delta.t=1)
```

### Arguments

| | |
|---|---|
| spec | a numerical vector |
| N | length of the original time series |
| delta.t | sampling interval |

### Details

Create a class of type 'spect' from a spectrum 'spec'of a time series of length 'N' regularly sampled at rate 'delta.t'.

### Value

An object of class 'spect' which contains the frequencies 'freq', spectral estimates 'spec', sampling interval 'delta.t', sample size 'N' and number of Fourier frequencies, 'n.Fourier'.

### Author(s)

Peter F. Craigmile

## spect.max — *The frequency of maximum spectrum*

### Usage

```
spect.max(sp, f1, f2)
```

### Arguments

| | |
|---|---|
| sp | object of class 'spect' |
| f1 | lower bound of the frequency |
| f2 | upper bound of the frequency |

### Value

The frequency between 'f1' and 'f2' which has highest spectrum in the 'sp'.

### Author(s)

Peter F. Craigmile

## squared.gain — *Calculate the squared gain function*

### Usage

```
squared.gain(fs, filter, multiplier=1, use.C=TRUE)
```

### Arguments

| | |
|---|---|
| `fs` | a numerical vector of frequencies |
| `filter` | a vector of filter coefficients |
| `multiplier` | a multipler for the frequencies |
| `use.C` | if use.C is TRUE use C code to calculate; otherwise using R code |

### Details

Calculate the squared gain function for the 'filter' at the frequencies 'multiplier * fs'.

### Value

A numerical vector of squared gain function values, the same length as 'fs'.

### Author(s)

Peter F. Craigmile

## Thomson.crit — *Critical value of Thomson's F test*

### Usage

```
Thomson.crit(n.tapers, alpha)
```

### Arguments

| | |
|---|---|
| `n.tapers` | The number of tapers |
| `alpha` | The alpha value to evaluate critical value |

### Value

alpha'th critical value

### Author(s)

Peter F. Craigmile

### References

Percival and Walden (1993) *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press, Section 10.11.

## Thomson.stat *Thomson's F statistic*

### Usage

```
Thomson.stat(x, index, tapers, n.tapers)
```

### Arguments

| | |
|---|---|
| `x` | The time series of interest |
| `index` | The Fourier index to test at |
| `tapers` | The matrix of tapers to use for the test |
| `n.tapers` | The number of tapers to evaluate |

### Details

Perform Thomson's F test to test for a non-zero amplitude at Fourier index 'index' in the time series 'x' using 'n.tapers' tapers from the matrix of tapers 'tapers'.

### Value

The F statistic of Thomson's F test

### Author(s)

Peter F. Craigmile

### References

Percival and Walden (1993) *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press, Section 10.11.

## transfer *Calculate the transfer function*

### Usage

```
transfer(fs, filter)
```

### Arguments

| | |
|---|---|
| `fs` | a numerical vector of frequencies |
| `filter` | a vector of filter coefficients |

### Details

Calculate the transfer function for the 'filter' at the frequencies 'fs'.

### Value

A numerical vector of transfer function values, the same length as 'fs'.

### Author(s)

Peter F. Craigmile

# Index

∗ **datasets**
ar2.example, 3
ar4.example, 4

acvs.from.sdf, 2
ar.sdf, 3
ar2.example, 3
ar4.example, 4
arma.sdf, 4
arma.var.from.sdf, 5

closest.Fourier.frequency, 5
closest.Fourier.index, 6
complex.chol, 6
cosine.taper, 7
create.taper, 7

dB *(*decibels*)*, 8
decibels, 8
direct.spectral, 8
Dirichlet.kernel, 9
dpss.taper, 9

Fejer.kernel, 10
fft1, 11
Fisher.test, 11
Fourier.frequencies, 12

h, 12
Hanning.taper, 13

idB *(*idecibels*)*, 13
idecibels, 13

ma.sdf, 14
mean.spect, 14
multitaper.effective.bw, 15

periodicity, 15
periodogram, 16
pgram *(*periodogram*)*, 16
plot.spect, 16

rectangular.taper, 17
renormalize.taper, 18

sine.taper, 18
spect, 19
spect.max, 19
squared.gain, 20

Thomson.crit, 20
Thomson.stat, 21
transfer, 21