

# Package ‘oceanwaves’

June 2, 2021

**Title** Ocean Wave Statistics

**Version** 0.2.0

**Date** 2021-05-30

**Description** Calculate ocean wave height summary statistics and process data from bottom-mounted pressure sensor data loggers. Derived primarily from MATLAB functions provided by U. Neumeier at <http://neumeier.perso.ch/matlab/waves.html>. Wave number calculation based on the algorithm in Hunt, J. N. (1979, ISSN:0148-9895) "Direct Solution of Wave Dispersion Equation", American Society of Civil Engineers Journal of the Waterway, Port, Coastal, and Ocean Division, Vol 105, pp 457-459.

**URL** <https://github.com/millerlp/oceanwaves>,  
<https://millerlp.github.io/oceanwaves/>

**BugReports** <https://github.com/millerlp/oceanwaves/issues>

**Depends** R (>= 3.3.0), ggplot2

**Imports** bspec, signal

**Suggests** scales, oce, knitr, rmarkdown, testthat (>= 2.1.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Luke Miller [aut, cre] (<https://orcid.org/0000-0002-2009-6981>),  
Urs Neumeier [aut],  
Travis Mason [ctb],  
Magali Lecouturier [ctb],  
George Voulgaris [ctb]

**Maintainer** Luke Miller <contact@lukemiller.org>

**Repository** CRAN

**Date/Publication** 2021-06-02 04:40:14 UTC

## R topics documented:

almostZero . . . . .	2
detrendHeight . . . . .	3
plotWaveSpectrum . . . . .	4
prCorr . . . . .	4
wavedata . . . . .	5
waveNumL . . . . .	6
waveStatsSP . . . . .	6
waveStatsZC . . . . .	8

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

---

almostZero	<i>Test whether vector elements are effectively zero</i>
------------	--

---

### Description

Test whether elements in vector `x` are effectively zero, within the square root of machine tolerance for a floating point number.

### Usage

```
almostZero(x, y = 0, tolerance = sqrt(.Machine$double.eps))
```

### Arguments

<code>x</code>	Numeric vector of values to compared against 0
<code>y</code>	Value to be compared against. Default is 0.
<code>tolerance</code>	The maximum difference between <code>x</code> and <code>y</code> necessary to call a value non-zero

### Details

Returns TRUE for all vector elements that are within rounding error of zero.

### Value

TRUE or FALSE for each element of the vector `x`.

---

detrendHeight	<i>Remove trend from a time series</i>
---------------	--

---

### Description

Fits a straight line to a vector of values using `lm()`, and uses the regression coefficients to subtract off the linear trend from the values.

### Usage

```
detrendHeight(pt)
```

### Arguments

`pt` A vector of numeric values to be detrended

### Details

Typically this is used to remove a tidal trend from a ocean surface height time series before attempting to calculate statistics for waves, which are variations above and below the mean surface height. Returns a series of residuals around the linear trend, in the original units of measurement (typically meters).

### Value

A list containing the following:

`pt` A vector of detrended values

`seg_len` The segment length used

`h` The mean height of the water column

`trend` A two element vector of the intercept and slope from the linear regression.

### Examples

```
data(wavedata)
detrended <- detrendHeight(wavedata$swDepth.m)
pt <- detrended[['pt']]
plot(pt, type = 'l')
abline(h = 0)
```

---

plotWaveSpectrum	<i>Plot a basic spectrum</i>
------------------	------------------------------

---

### Description

Plot a basic spectrum

### Usage

```
plotWaveSpectrum(freqspec, Fs)
```

### Arguments

freqspec	A data frame containing a column of frequencies 'freq' and a column of spectral power values 'spec'
Fs	Frequency of sampled surface heights, units of Hz

---

prCorr	<i>Correct for depth attenuation of a water surface elevation pressure signal.</i>
--------	--

---

### Description

Bottom-mounted pressure transducers suffer from pressure signal attenuation when attempting to estimate surface wave heights. This function corrects water surface elevation time series based on the depth of the water column and height of the sensor above the bottom.

### Usage

```
prCorr(pt, Fs, zpt, M = 512, CorrLim = c(0.05, 0.33), plot = FALSE)
```

### Arguments

pt	A vector of sea surface elevations (units of meters).
Fs	Sampling frequency (units of Hz). Normally 4 Hz for an OWHL logger.
zpt	Height of the pressure sensor above the seabed (units of meters).
M	Length of time series segments that will be used in the detrending and attenuation correction operations. 512 samples is the default, should be an even number.
CorrLim	[min max] frequency for attenuation correction (Hz, optional, default [0.05 0.33], which translate to periods of 20 sec to 3 sec).
plot	Logical value TRUE or FALSE. Displays a plot of the original and corrected time series.

**Value**

A vector of the depth-corrected surface elevations (units of meters usually). Any original trend in the input data (such as tide change) is present in the output data. The returned surface elevation fluctuations will typically be more extreme than the raw input surface elevations.

**References**

Based on original MATLAB function by developed by Travis Mason, M. Lecouturier & Urs Neumeier <http://neumeier.perso.ch/matlab/waves.html>

Each segment of pt will be linearly detrended, corrected for attenuation, and the linear trend will be added back to the returned data.

**Examples**

```
data(wavedata)
corrected = prCorr(wavedata$swPressure.mbar, Fs = 4, zpt = 0.1)
# Plot the results
corrected = prCorr(wavedata$swPressure.mbar, Fs = 4, zpt = 0.1, plot=TRUE)
```

---

wavedata

*Example wave records*


---

**Description**

A dataset containing pressure records and derived sea surface height from a bottom-mounted pressure transducer data logger deployed at Marguerite Reef near Los Angeles, CA, USA. The device was deployed at a depth of approximately 10 meters, and sea level air pressure was 1014 mbar at the time of deployment. The pressure data logger was mounted approximately 10 cm above the benthos. Sampling rate was 4 Hz, so the total record represents 30 minutes of time.

**Usage**

```
wavedata
```

**Format**

A data frame with 7200 rows and 4 variables:

**DateTime** Date and time, with fractional seconds. UTC time zone.

**absPressure.mbar** Absolute pressure recorded by the data logger, in units of millibar (1 mbar = 100 Pascal). This value includes pressure due to the atmosphere above the sea surface, which was 1014 mbar during the deployment.

**swPressure.mbar** Pressure produced by the column of seawater above the data logger. The pressure due to the atmosphere above the sea surface has been subtracted off of the absolute pressure value absPressure.mbar to produce this value. Units of mbar.

**swDepth.m** Estimated depth in meters of the seawater column above the pressure data logger, converted from the swPressure.mbar value.

**Source**

2016 Provided by Kristin Elsmore, UC Davis Bodega Marine Laboratory

---

waveNumL	<i>A function to calculate wave number.</i>
----------	---

---

**Description**

A function to calculate wave number.

**Usage**

```
waveNumL(f, h)
```

**Arguments**

f	A numeric vector of wave frequencies
h	A numeric vector of water depths (usually in units of meters)

**Value**

The wave number.

**References**

Modified from MATLAB function by Urs Neumeier: <http://neumeier.perso.ch/matlab/waves.html>  
 Based on polynomial approximation from Hunt, J. N. 1979. "Direct Solution of Wave Dispersion Equation," ASCE Journal of the Waterway, Port, Coastal and Ocean Division., Vol 105, pp 457-459  
 suggested by George Voulgaris, University of South Carolina

---

waveStatsSP	<i>Calculate ocean wave parameters using spectral analysis methods</i>
-------------	--

---

**Description**

Calculate ocean wave parameters using spectral analysis methods

**Usage**

```
waveStatsSP(
  data,
  Fs,
  method = c("welchPSD", "spec.pgram"),
  plot = FALSE,
  kernel = NULL,
  segments = NULL,
  ...
)
```

**Arguments**

data	A vector of surface heights that constitute a time series of observations. Typical units = meters.
Fs	Sampling frequency of the surface heights data. Units = Hz, i.e. samples per second.
method	A character string indicating which spectral analysis method should be used. Choose one of <code>welchPSD</code> (default) or <code>spec.pgram</code> .
plot	A logical value denoting whether to plot the spectrum. Defaults to <code>FALSE</code> .
kernel	An object of class <code>tskernel</code> that defines a smoother for use with <code>spec.pgram</code> method. If value is <code>NULL</code> , a default Daniell kernel with widths (9,9,9) is used.
segments	Numeric value indicating the number of windowing segments to use with <code>welchPSD</code> method.
...	Additional arguments to be passed to spectral analysis functions, such as the <code>windowfun</code> option for <code>welchPSD</code> .

**Details**

Carries out spectral analysis of ocean wave height time series to estimate common wave height statistics, including peak period, average period, and significant wave height.

**Value**

List of wave parameters based on spectral methods.

- `h` Average water depth. Same units as input surface heights (typically meters).
- `Hm0` Significant wave height based on spectral moment 0. Same units as input surface heights (typically meters). This is approximately equal to the average of the highest 1/3 of the waves.
- `Tp` Peak period, calculated as the frequency with maximum power in the power spectrum. Units of seconds.
- `m0` Estimated variance of time series (moment 0).
- `T_0_1` Average period calculated as  $m0/m1$ , units seconds. Follows National Data Buoy Center's method for average period (APD).
- `T_0_2` Average period calculated as  $(m0/m2)^{0.5}$ , units seconds. Follows Scripps Institution of Oceanography's method for calculating average period (APD) for their buoys.
- `EPS2` Spectral width parameter.
- `EPS4` Spectral width parameter.

**References**

Original MATLAB function by Urs Neumeier: <http://neumeier.perso.ch/matlab/waves.html>, based on code developed by Travis Mason, Magali Lecouturier and Urs Neumeier.

**See Also**

[waveStatsZC](#) for wave statistics determined using a zero-crossing algorithm.

**Examples**

```
data(wavedata)
waveStatsSP(wavedata$swDepth.m, Fs = 4, method = 'spec.pgram', plot = TRUE)
```

---

waveStatsZC

*Calculate wave statistics using zero-crossing method*


---

**Description**

Calculate ocean wave summary statistics, including significant wave height and period.

**Usage**

```
waveStatsZC(data, Fs, threshold = NULL, plot = FALSE)
```

**Arguments**

data	A numeric vector of water surface height data. The data do not need to be detrended prior to use. Typical units = meters
Fs	Sampling frequency of the data, in Hz.
threshold	The minimum height necessary for a zero-crossing event to be considered a wave.
plot	Set to TRUE if summary histograms of wave heights and wave periods are desired.

**Details**

Based on an upward zero-crossing algorithm originally provided by Urs Neumeier, v1.06. However, by default the algorithm is run as a downward zero-crossing process by inverting the data.

**Value**

A list object containing summary statistic values.

- Hsig Mean of the highest 1/3 of waves in the data set. Units = same as input surface heights.
- Hmean Overall mean wave height, for all waves (bigger than threshold).
- H10 Mean height of the upper 10% of all waves.
- Hmax Maximum wave height in the input data.
- Tmean Mean period of all waves (bigger than threshold). Units = seconds.
- Tsig Mean period of Hsig (highest 1/3 of waves).

**References**

Original MATLAB function by Urs Neumeier: <http://neumeier.perso.ch/matlab/waves.html>



**See Also**

[waveStatsSP](#) for wave statistics determined using spectral analysis methods.

**Examples**

```
data(wavedata)
waveStatsZC(data = wavedata$swDepth.m, Fs = 4, plot = TRUE)
```

# Index

## \* datasets

wavedata, 5

almostZero, 2

detrendHeight, 3

plotWaveSpectrum, 4

prCorr, 4

wavedata, 5

waveNumL, 6

waveStatsSP, 6, 9

waveStatsZC, 7, 8