Package 'TideHarmonics'

October 12, 2022

Version 0.1-1

Date 2017-05-04

Title Harmonic Analysis of Tides

Author Alec Stephenson.

Maintainer Alec Stephenson <alec_stephenson@hotmail.com>

Depends R (>= 2.10)

Description Implements harmonic analysis of tidal and sea-level data.
Over 400 harmonic tidal constituents can be estimated, all with
daily nodal corrections. Time-varying mean sea-levels can also
be used.

LazyData yes

License BSD_3_clause + file LICENSE

NeedsCompilation no

Repository CRAN

Date/Publication 2017-05-04 08:16:54 UTC

R topics documented:

`Broome` *Sea-Level Data At Broome*

Description

The `Broome` data frame has 26304 rows and 2 variables. It gives hourly sea-levels in metres (above Tide Gauge Zero) recorded by the tide gauge within the (UTC) years of 2012-2014. The sea-levels are in the second column. The first column is a POSIXct object giving the dates and times in UTC.

Broome is one of the sites currently used for the Australian Baseline Sea-Level Monitoring Project (ABSLMP). Hourly data for these sites is available from the Australian Bureau of Meteorology.

To obtain the sea-level in AHD (Australian Height Datum), you will need to subtract 5.322 metres from the values given in the dataset.

Usage

```
Broome
```

Format

This data frame contains the following columns:

DateTime A POSIXct object where times are in UTC.

SeaLevel The sea-level in metres above Tide Gauge Zero. Any missing or erroneous data points are set to NA.

Source

Australian Bureau of Meteorology.

See Also

`ftide`, `harmonics`

CapeFerguson *Sea-Level Data At Cape Ferguson*

Description

The CapeFerguson data frame has 26304 rows and 2 variables. It gives hourly sea-levels in metres (above Tide Gauge Zero) recorded by the tide gauge within the (UTC) years of 2012-2014. The sea-levels are in the second column. The first column is a POSIXct object giving the dates and times in UTC.

Cape Ferguson is one of the sites currently used for the Australian Baseline Sea-Level Monitoring Project (ABSLMP). Hourly data for these sites is available from the Australian Bureau of Meteorology.

To obtain the sea-level in AHD (Australian Height Datum), you will need to subtract 1.590 metres from the values given in the dataset.

Usage

```
CapeFerguson
```

Format

This data frame contains the following columns:

DateTime A POSIXct object where times are in UTC.

SeaLevel The sea-level in metres above Tide Gauge Zero. Any missing or erroneous data points are set to NA.

Source

Australian Bureau of Meteorology.

See Also

ftide, harmonics

coef.tide *Coefficients For Tidal Object*

Description

Give coefficients and harmonic constituents of a tidal object.

Usage

```
## S3 method for class 'tide'
coef(object, hc = FALSE, mat = hc, utc = 0, ...)
```

Arguments

`object` An object of class `'tide'`.

`hc` If `TRUE`, returns harmonic constituent values, in order of decreasing amplitude. If `mat` is `FALSE`, a vector is returned containing the amplitudes and then the phase lags. If `mat` is `TRUE`, a four column matrix is returned, with the amplitudes, phase lags, sine coefficients and cosine coefficients in the columns.

`mat` If `TRUE`, a matrix is returned.

`utc` The phase lags are given in UTC plus or minus `utc` hours. For example, if Australian Eastern Standard Time is required, use positive 10. If Pacific Standard Time is required, use negative 8.

`...` Not used.

Value

A numeric vector or numeric matrix of sine and cosine coefficients, or of amplitudes and phase lags. The phase lags are with respect to the time zone specified by the `utc` argument, which is UTC by default.

See Also

`ftide`, `predict.tide`

Examples

```
hfit1 <- ftide(Hillarys$SeaLevel, Hillarys$DateTime, hc60)
hfit2 <- ftide(Hillarys$Sea, Hillarys$Date, hc7, smsl=TRUE)
coef(hfit1, hc = TRUE)
coef(hfit2)
coef(hfit2, mat = TRUE)
coef(hfit2, hc = TRUE)
coef(hfit2, hc = TRUE, mat = FALSE)
```

`Darwin` *Sea-Level Data At Darwin*

Description

The `Darwin` data frame has 26304 rows and 2 variables. It gives hourly sea-levels in metres (above Tide Gauge Zero) recorded by the tide gauge within the (UTC) years of 2012-2014. The sea-levels are in the second column. The first column is a POSIXct object giving the dates and times in UTC.

Darwin is one of the sites currently used for the Australian Baseline Sea-Level Monitoring Project (ABSLMP). Hourly data for these sites is available from the Australian Bureau of Meteorology.

To obtain the sea-level in AHD (Australian Height Datum), you will need to subtract 4.105 metres from the values given in the dataset.

Usage

```
Darwin
```

Format

This data frame contains the following columns:

DateTime A POSIXct object where times are in UTC.

SeaLevel The sea-level in metres above Tide Gauge Zero. Any missing or erroneous data points are set to NA.

Source

Australian Bureau of Meteorology.

See Also

`ftide`, `harmonics`

`Esperance` *Sea-Level Data At Esperance*

Description

The `Esperance` data frame has 26304 rows and 2 variables. It gives hourly sea-levels in metres (above Tide Gauge Zero) recorded by the tide gauge within the (UTC) years of 2012-2014. The sea-levels are in the second column. The first column is a POSIXct object giving the dates and times in UTC.

Esperance is one of the sites currently used for the Australian Baseline Sea-Level Monitoring Project (ABSLMP). Hourly data for these sites is available from the Australian Bureau of Meteorology.

To obtain the sea-level in AHD (Australian Height Datum), you will need to subtract 0.707 metres from the values given in the dataset.

Usage

```
Esperance
```

Format

This data frame contains the following columns:

DateTime A POSIXct object where times are in UTC.

SeaLevel The sea-level in metres above Tide Gauge Zero. Any missing or erroneous data points are set to NA.

Source

Australian Bureau of Meteorology.

See Also

ftide, harmonics

---

ftide *Fit Tidal Data*

Description

Fit tidal data using any of over 400 harmonic constituents. Daily nodal corrections and time-varying mean sea-levels can be used.

Usage

```
ftide(x, dto, hcn = TideHarmonics::hc60, astlon = c("task","cartwright"),
   nodal = TRUE, smsl = FALSE, span = 0.75, degree = 1, ...)
```

Arguments

`x` A numeric vector or time series object giving the sea-levels. Missing values are allowed.

`dto` A date/time vector of the same length as `x`, which should be a POSIXct object or something which can be converted to a POSIXct object. If no time zone is given then UTC is assumed. No missing values are allowed.

`hcn` A vector of constituent names. Some in-built vectors of names can be used: hc4, hc7, hc37, hc60 and hc114. If not specified, the vector hc60 is used by default.

`astlon` The longitude formula to be used. The default "task" is the formulas use in the TASK-2000 software. The alternative "cartwright" uses Cartwright(1982).

`nodal` Should nodal corrections be be used?

`smsl` If `TRUE`, then a smooth curve is fitted to the mean sea-level based on the loess function. This is designed to account for smooth changes at periods longer than the lowest speed harmonic, which by default is the solar annual term Sa.

`span, degree` Arguments passed to the loess function.

`...` Passed to the linear model function `lm`.

Details

The function first removes the mean sea-level from the data (which by default is a single number) and the fits a linear model without intercept to sine and cosine terms defined by the specified harmonics. This (by default) includes nodal corrections. The sine and cosine coefficients can then be used to derive the amplitude and phases. See the package vignette for details.

Different names are used by various organizations for identical constituents. This package is designed to be robust, so that any common name can be used. In the output, the name will get converted to the set of names that we employ.

For constituents based on underlying components, we use our own (logical) naming scheme rather than the (totally confusing) historical scheme. See `harmonics` and the package vignette for details. Either scheme should work for the input vector `hcn`.

The UTC time zone is assumed by default. Even if a different time zone is used, the phase lags will be calculated in UTC. The `utc` argument of `coef.tide` can be used to derive phase lags for different time zones.

An error will be produced if two specified harmonic components have virtually identical speeds (specifically, if the first five Doodson numbers are the same), even if the nodal corrections are different. This is to avoid numerical problems in the linear model fit, because these components will be difficult (or impossible) to identify separately.

The slowest harmonics in the default hc60 vector is the annual solar term Sa. If you do not have at least one year of data you should not include Sa. The fastest harmonics in hc60 have periods of just over 4 hours. If your frequency of observation is more than 2 hours you should not include the faster constituents.

Value

A list object of class `c('tide','lm')`. This is exactly the save as a standard `lm` object but with the following additional components

`fval` The form factor, which in terms of amplitudes is calculated as (K1+O1)/(M2+S2). This component and the two components given below are only available if all four of these harmonics are included in the fit.

`features1` A vector of fitted features that are relevant for semi-diurnal sites. MLWS = Mean Low Water Springs. MLWN = Mean Low Water Neaps. MSL = Mean Sea-Level. MHWN = Mean High Water Neaps. MHWS = Mean High Water Springs.

`features2` A vector of fitted features that are relevant for diurnal or mixed semi-diurnal sites. MLLW = Mean Lower Low Water. MHLW = Mean Higher Low Water. MSL = Mean Sea-Level. MLHW = Mean Lower High Water. MHHW = Mean Higher High Water.

`cfmat` A matrix of sine and cosine coefficients, in the same order as the `hcn` vector.

`cfmat` A matrix of amplitudes, phase lags, sine and cosine coefficients, ordered by decreasing amplitude.

`origin` The POSIXct value used as the origin, which is in the centre of the `dto` vector.

`vn0` A named vector containing reference signals (equilibrium phases) in degrees for each harmonic constituent.

`msl` The mean sea level, either a vector (if `smsl` is `TRUE`) or a single number (if `smsl` is `FALSE`).

`lobj` The fitted loess object if `smsl` is `TRUE`.

`nodal` The nodal argument.

`astlon` The astlon argument.

See Also

`harmonics`, `hc114`, `plot.tide`, `predict.tide`

Examples

```
hfit1 <- ftide(Hillarys$SeaLevel, Hillarys$DateTime, hc60)
hfit2 <- ftide(Hillarys$Sea, Hillarys$Date, hc7, smsl=TRUE)
hfit1
hfit2
```

`harmonics` *Table Of All 409 Harmonic Constituents*

Description

The `harmonics` data frame has 409 rows and 12 variables. It shows the standard list of tidal constituents as provided by the IHO tidal committee.

Some corrections and adjustments have been made to the standard list; these are documented in the package vignette. We also use a different naming scheme which is more logical and consistent. Our names are given in the `name` column. The traditional names are given in the `sname` column. Either can be used in the code.

We have retained lower case characters and have used the first three characters for greek letters e.g. the1 for theta1. The code has been written for robustness and will still work if you use upper case characters and/or full names for greek letters.

There are many variations of constituent names and I have attempted to write the software so that it will work whatever names are used. For example, NOAA uses RHO rather than the more common rho1 or RHO1. The names RHO and RHO1 are therefore automatically converted to rho1. TASK-2000 uses LAMDA1 (with the B missing), which is automatically converted to lam1.

Usage

```
harmonics
```

Format

This data frame contains the following columns:

name The name of the constituent, which is unique.

sname The standard name of the constituent, which is also unique.

speed The speed (angular frequency) in degrees per hour. Derived from the Doodson number. To calculate the period in hours, divide 360 by the speed.

code The extended Doodson code.

i1 The first Doodson number.

i2 The second Doodson number.

i3 The third Doodson number.

i4 The forth Doodson number.

i5 The fifth Doodson number (always zero).

i6 The sixth Doodson number (mostly zero).

phi The phase constant in degrees.

nodal The type of nodal correction, as taken from the standard list.

Source

Standard list of tidal constituents of the IHO tidal committee.

See Also

`ftide`, `hc114`

`hc114` *Names Of Commonly Used Harmonic Constituents*

Description

These vectors give names of commonly used harmonic constituents.

The `hc114` vector gives the first 114 of the 115 constituents listed in the TASK-2000 software manual. The order is the same as that given in the manual. The 115th constituent has the same speed (but a different nodal correction) as the constituent L2. It is omitted because it probably not sensible to include both.

The `hc60` vector gives the first 60 of the 115 constituents listed in the TASK-2000 software manual. The order is the same as that given in the manual (which is in order of speed).

The `hc37` vector gives the 37 constituents that are used by the NOAA, in the sense that they are publicly available on the NOAA website (for USA sites). They are listed in NOAA order. Note that NOAA uses constituents S6 and M8, but neither are contained in `hc60`, and S6 is not contained in `hc114`.

The `hc7` vector gives the seven major constituents M2 S2 N2 K2 K1 O1 P1.

The hc4 vector gives the four major constituents M2 S2 K1 O1. Unless you are investigating single constituents, it is recommended that these four are always included because they are fundamental to the classification of a site as semi-diurnal, mixed semi-diurnal or diurnal.

The task vector gives the same constituents as hc114 but using the TASK-2000 names. The noaa vector gives the same constituents as hc37 but using the NOAA names. The package is written to be robust to different naming schemes, and therefore any of these vectors can be used as the hcn argument to the ftide function.

Usage

```
hc114
task
hc60
hc37
noaa
hc7
hc4
```

Format

Character vectors of different lengths.

See Also

ftide, harmonics

Hillarys *Sea-Level Data At Hillarys*

Description

The Hillarys data frame has 26304 rows and 2 variables. It gives hourly sea-levels in metres (above Tide Gauge Zero) recorded by the tide gauge within the (UTC) years of 2012-2014. The sea-levels are in the second column. The first column is a POSIXct object giving the dates and times in UTC.

Hillarys is one of the sites currently used for the Australian Baseline Sea-Level Monitoring Project (ABSLMP). Hourly data for these sites is available from the Australian Bureau of Meteorology.

To obtain the sea-level in AHD (Australian Height Datum), you will need to subtract 0.763 metres from the values given in the dataset.

Hillarys is located in the south-west of Australia, where tides are typically diurnal (i.e. one high tide per day). The amplitudes of K1 and O1 will likely be larger than the amplitudes of M2 and S2.

This dataset is is used for the examples in the help files.

Usage

```
Hillarys
```

Format

This data frame contains the following columns:

DateTime A POSIXct object where times are in UTC.

SeaLevel The sea-level in metres above Tide Gauge Zero. Any missing or erroneous data points are set to NA.

Source

Australian Bureau of Meteorology.

See Also

ftide, harmonics

---

lambdas *Calculates Astronomical Longitudes*

Description

Calculates astronomical longitudes for given days. Mainly for internal use.

Usage

```
lambdas(dvec, astlon = c("task","cartwright"), ...)
```

Arguments

dvec	A vector of days. Should be a Date object or an object that can be converted to a Date object.
astlon	The longitude formula to be used. The default "task" is the formulas use in the TASK-2000 software. The alternative "cartwright" uses Cartwright(1982).
...	Passed to the as.Date function.

Value

A numeric matrix. Values are in degrees and given in the interval [0,360]. The rows represent astronomical periods of increasing length: sidereal month (s), tropical year (h), lunar perigee (p), lunar nodal (N), sun's perihelion (ph). These correspond to the last five Doodson numbers.

See Also

ftide, harmonics

Examples

```
days <- seq(as.Date("2012-12-30"), as.Date("2013-01-08"), 1)
lambdas(days)
```

nodal_adj *Calculate nodal corrections.*

Description

Calculates nodal corrections from astronomical longitudes (lambdas). Mainly for internal use.

Usage

```
nodal_adj(lambp, lambN, lambph, indegree = TRUE, outdegree = TRUE)
```

Arguments

lambp	Lambdas for the lunar perigee (p).
lambN	Lambdas for the lunar nodal (N).
lambph	Lambdas for the sun's perihelion (ph).
indegree	If FALSE, the lambdas are in radians. Degrees are assumed by default.
outdegree	If FALSE, the returned phase corrections are in radians. Degrees are produced by default.

Value

A list with two elements, where both are matrices with 409 rows, which correspond to the 409 harmonic constituents available in the package. The first element fn gives the amplitude corrections. The second element un gives the phase corrections.

See Also

ftide, lambdas

Examples

```
days <- seq(as.Date("2012-12-30"), as.Date("2013-01-08"), 1)
lamb <- lambdas(days)
nodal_adj(lamb[3,], lamb[4,], lamb[5,])
```

plagtz *Convert Phase Lag*

Description

Converts phase lags between different time zones. Mainly for internal use.

Usage

```
plagtz(plag, tzd, indegree = TRUE, outdegree = TRUE)
```

Arguments

`plag`	A named numeric vector of phase lags, where the names must correspond to the harmonic components. The names must be those used by this package, which are in the first column of the `harmonics` object.
`tzd`	The time difference in hours. For example, when converting from UTC to Australian Eastern Standard Time (AEST), this value must be positive 10.
`indegree`	If `FALSE`, the `plag` phase lags are in radians. Degrees are assumed by default.
`outdegree`	If `FALSE`, the returned phase lags are in radians. Degrees are produced by default.

Value

A numeric vector. By convention, phase lags are given in the interval [0,360] for degrees or [0,2pi] for radians.

See Also

`ftide`, `coef.tide`

Examples

```
pvec <- c(M2 = 34.2, S2 = 256.8)
plagtz(pvec, tzd = 10)
```

plot.tide *Plot Tidal Object*

Description

Plot line traces of estimated tide levels against time.

Usage

```
## S3 method for class 'tide'
plot(x, from, to, by = NULL, split = FALSE, which = NULL,
  msl = !split, ask = split && dev.interactive(), main = NULL,
  xlab = "Times", ylab = "Level", ...)
```

Arguments

`x` An object of class `'tide'`.

`from` Time and date from which to plot. Should be a POSIXct object or something which can be converted to a POSIXct object. If no time zone is given then UTC is assumed.

`to` Time and date up to which to plot. Should be a POSIXct object or something which can be converted to a POSIXct object. If no time zone is given then UTC is assumed.

`by` The time interval in hours between calculated tidal predictions. If `NULL`, then the time interval is that which produces 1000 evaluations between `from` and `to`.

`split` If `TRUE`, plot values for harmonic constituents separately. Otherwise, the constituents are summed.

`which` If `NULL`, use all fitted harmonic constituents. If not `NULL`, then should be a character vector giving the names of the selected constituents.

`msl` Add the mean sea-level to the plots? A time-varying mean sea-level will be used if this has been implemented for the tidal object.

`xlab,ylab` Graphical parameters.

`main,ask` Graphical parameters.

`...` Other parameters to be passed through to plotting functions.

Details

Note that nodal corrections and time-varying mean sea-levels will be used if and only if they were implemented for the tidal object. The longitude formulas will also be the same as those used for the tidal object.

The dates/times plotted on the x-axis correspond to the time zone used for the `from` object.

Value

A list of times and predictions is returned invisibly.

See Also

`ftide`, `predict.tide`

Examples

```
hfit1 <- ftide(Hillarys$SeaLevel, Hillarys$DateTime, hc60)
hfit2 <- ftide(Hillarys$Sea, Hillarys$Date, hc7, smsl=TRUE)
t1 <- as.POSIXct("2012-12-31 23:00", tz = "UTC")
t2 <- as.POSIXct("2013-01-02 14:00", tz = "UTC")
plot(hfit1, t1, t2)
plot(hfit2, t1, t2, split = TRUE)
plot(hfit2, t1, t2, split = TRUE, which = c("M2","S2"))
plot(hfit2, t1, t2, which = "M2", msl = FALSE)
```

PortKembla *Sea-Level Data At Port Kembla*

Description

The PortKembla data frame has 26304 rows and 2 variables. It gives hourly sea-levels in metres (above Tide Gauge Zero) recorded by the tide gauge within the (UTC) years of 2012-2014. The sea-levels are in the second column. The first column is a POSIXct object giving the dates and times in UTC.

Port Kembla is one of the sites currently used for the Australian Baseline Sea-Level Monitoring Project (ABSLMP). Hourly data for these sites is available from the Australian Bureau of Meteorology.

To obtain the sea-level in AHD (Australian Height Datum), you will need to subtract 0.872 metres from the values given in the dataset.

Usage

```
PortKembla
```

Format

This data frame contains the following columns:

DateTime A POSIXct object where times are in UTC.

SeaLevel The sea-level in metres above Tide Gauge Zero. Any missing or erroneous data points are set to NA.

Source

Australian Bureau of Meteorology.

See Also

ftide, harmonics

Portland *Sea-Level Data At Portland*

Description

The Portland data frame has 26304 rows and 2 variables. It gives hourly sea-levels in metres (above Tide Gauge Zero) recorded by the tide gauge within the (UTC) years of 2012-2014. The sea-levels are in the second column. The first column is a POSIXct object giving the dates and times in UTC.

Portland is one of the sites currently used for the Australian Baseline Sea-Level Monitoring Project (ABSLMP). Hourly data for these sites is available from the Australian Bureau of Meteorology.

To obtain the sea-level in AHD (Australian Height Datum), you will need to subtract 0.507 metres from the values given in the dataset.

Usage

```
Portland
```

Format

This data frame contains the following columns:

DateTime A POSIXct object where times are in UTC.

SeaLevel The sea-level in metres above Tide Gauge Zero. Any missing or erroneous data points are set to NA.

Source

Australian Bureau of Meteorology.

See Also

ftide, harmonics

predict.tide *Predict The Tide Using Tidal Object*

Description

Predict the tide values at specified times.

Usage

```
## S3 method for class 'tide'
predict(object, from, to, by = NULL, split = FALSE,
  which = NULL, msl = !split, ...)
```

Arguments

`object` An object of class `'tide'`.

`from` Time and date from which to calculate tide values. Should be a POSIXct object or something which can be converted to a POSIXct object. If no time zone is given then UTC is assumed.

`to` Time and date up to which to calculate tide values. Should be a POSIXct object or something which can be converted to a POSIXct object. If no time zone is given then UTC is assumed.

`by` The time interval in hours between calculated tidal predictions. If `NULL`, then the time interval is that which produces 1000 evaluations between `from` and `to`.

`split` If `TRUE`, predictions are given for harmonic constituents separately. Otherwise, the constituents are summed.

`which` If `NULL`, use all fitted harmonic constituents. If not NULL, then should be a character vector giving the names of the selected constituents.

`msl` Add the mean sea-level to the predictions? A time-varying mean sea-level will be used if this has been implemented for the tidal object.

`...` Not used.

Details

Note that nodal corrections and time-varying mean sea-levels will be used if and only if they were implemented for the tidal object. The longitude formulas will also be the same as those used for the tidal object.

Value

A numeric vector of predictions (if `split` is `FALSE`) or a numeric matrix of predictions (if `split` is `TRUE`).

See Also

`ftide`, `plot.tide`

Examples

```
hfit1 <- ftide(Hillarys$SeaLevel, Hillarys$DateTime, hc60)
hfit2 <- ftide(Hillarys$Sea, Hillarys$Date, hc7, smsl=TRUE)
t1 <- as.POSIXct("2012-12-31 23:00", tz = "UTC")
t2 <- as.POSIXct("2013-01-01 14:00", tz = "UTC")
predict(hfit1, t1, t2)
predict(hfit2, t1, t2, split = TRUE)
predict(hfit2, t1, t2, split = TRUE, which = c("M2","S2"))
predict(hfit2, t1, t2, which = "M2", msl = FALSE)
```

print.tide *Print Tidal Object*

Description

Printing a tidal object.

Usage

```
## S3 method for class 'tide'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

x	An object of class 'tide'.
digits	Number of printed digits.
...	Not used.

Details

A different features vector is printed based on whether the form factor is less than 0.5 (indicating a semi-diurnal site) or greater than or equal to 0.5 (indicating a diurnal or mixed semi-diurnal site). If the four harmonic constituents M2 S2 K1 O1 are not included in the fit, then the features vector cannot be calculated and is not printed.

Phase lags are always printed with respect to UTC. The `utc` argument of `coef.tide` can be used to produce phase lags for different time zones.

Value

The tidal object is invisibly returned.

See Also

`ftide`, `coef.tide`

Examples

```
hfit1 <- ftide(Hillarys$SeaLevel, Hillarys$DateTime, hc60)
hfit2 <- ftide(Hillarys$Sea, Hillarys$Date, hc7, smsl=TRUE)
hfit1
hfit2
```

`spdlunar` *Speeds For Basic Astronomical Periods*

Description

The `spdlunar` vector contains speeds (angular frequencies) in degrees per msh (mean solar hour) for the astronomic periods corresponding to the six digits of the (lunar) Doodson number. The periods are given by (i) the mean lunar day (ii) the sidereal month (iii) the tropical year (iv) the perigeal cycle of the moon (v) the nodal cycle of the moon (vi) the perihelion cycle of the sun.

The `spdsolar` vector contains speeds (angular frequencies) in degrees per msh (mean solar hour) for the astronomic periods corresponding to the six digits of the solar Doodson number. It is the same as `spdlunar` except that the first value corresponds to the mean solar day instead of the mean lunar day.

To obtain the astronomic periods in msh, take the inverse of the vector and multiply by 360.

Usage

```
spdlunar
spdsolar
```

Format

Numeric vectors of length six.

See Also

`harmonics`

`Thevenard` *Sea-Level Data At Thevenard*

Description

The `Thevenard` data frame has 26304 rows and 2 variables. It gives hourly sea-levels in metres (above Tide Gauge Zero) recorded by the tide gauge within the (UTC) years of 2012-2014. The sea-levels are in the second column. The first column is a POSIXct object giving the dates and times in UTC.

Thevenard is one of the sites currently used for the Australian Baseline Sea-Level Monitoring Project (ABSLMP). Hourly data for these sites is available from the Australian Bureau of Meteorology.

To obtain the sea-level in AHD (Australian Height Datum), you will need to subtract 0.993 metres from the values given in the dataset.

Usage

```
Thevenard
```

Format

This data frame contains the following columns:

DateTime A POSIXct object where times are in UTC.

SeaLevel The sea-level in metres above Tide Gauge Zero. Any missing or erroneous data points are set to NA.

Source

Australian Bureau of Meteorology.

See Also

`ftide`, `harmonics`

Index