Package ‘TideCurves’

June 28, 2021

Type Package
Title Analysis and Prediction of Tides
Version 0.0.5
Maintainer Moritz Mueller-Navarra <muellernavarra@gmail.com>
Description Tidal analysis of evenly spaced observed time series (time step 1 to 60 min) with or without shorter gaps using the harmonic representation of inequalities.

The analysis should preferably cover an observation period of at least 19 years.

For shorter periods low frequency constituents are not taken into account, in accordance with the Rayleigh-Criterion.

The main objective of this package is to synthesize or predict a tidal time series.

License GPL-3
LazyData TRUE
Imports chron (>= 2.3-56), data.table (>= 1.14.0), fields (>= 11.6)
Depends R (>= 4.0.0)
RoxygenNote 7.1.1
Suggests testthat (>= 3.0.1)
Config/testthat/edition 3
NeedsCompilation no
Author Moritz Mueller-Navarra [aut, cre],
Repository CRAN
Date/Publication 2021-06-28 16:50:02 UTC

R topics documented:

BuildDesign .......................................................... 2
BuildTC .............................................................. 2
Funcs ................................................................. 3
ResCurve ............................................................ 4
BuildTC

BuildsTC

Description

Builds the xdesign.matrix by calling Funcs. For internal use.

Usage

BuildDesign(tdiffa, numma, numme)

Arguments

tdiffa The difference in days as double which stems from the analysis period.
numma The transit number (start).
numme The transit number (end).

Value

Returns a matrix

BuildTC

Builds a TideCurve model

Description

Builds a TideCurve model of class "tidecurve".

Usage

BuildTC(
    dataInput = NULL,
    otz = 1,
    astime,
    asdate,
    aedate,
    aetime,
    km = -1,
    mindt = 30,
    keep_data = FALSE
)
Args 

dataInput A data frame with the columns observation_date, observation_time and height. See attached data for correct formats.

otz The time zone of the observations

astime A string indicating the time you want the analysis to start with. Format: "hh:mm:ss"

asdate A string indicating the date you want the analysis to start with. Format: "yyyy/mm/dd"

aedate A string indicating the date you want the analysis to end with. Format: "yyyy/mm/dd"

aetime A string indicating the time you want the analysis to end with. Format: "hh:mm:ss"

km The number of nodes between two consecutive mean moon transits. Shall be less or equal to: round(1440 [min] / time step [min]) Example: Time step 5 min: Use km = 288 or even smaller. Leave on default (km = -1) and supply mindt, when unsure.

mindt Observation time step in [min]. Default is 30.

keep_data Indicating whether you want to keep the data for computing residuals later. Default is FALSE which keeps the model footprint small.

Value 
A model of class tidecurve, which is a list.

References 

doi: 10.5194/os1513632019

Examples

## Not run: BuildTC(dataInput = tideObservation, asdate = "2015/12/06",
astime = "00:00:00", aedate = "2015/12/31",
aetime = "23:30:00")
## End(Not run)

---

## Funcs

Returns predictor vector for design matrix

### Description

Returns predictor vector for design matrix from 39 astronomical angular velocities.

### Usage

Funcs(tdiff, xi)
Arguments

tdiff  Length of input time series.
xi      Transit index

Value

A list with the selected angular velocities, their ranks and the predictor vector (Values between -1, 1).

ResCurve

Computes the residuum between the observed data and the synthesis

Description

This function computes the residuum of the computed lunar and solar synthesis and the observed data

Usage

ResCurve(tcData, obsData)

Arguments

tcData  The results from TideCurve or BuildTC + SynTC. Warning: The synthesis period must overlap with the analysis period. Must be a data.table object. Please see examples.
obsData The observation data with the columns observation_date, observation_time and height. See attached data for correct formats.

Value

A list with two data.tables with the joined data input and the computed difference between the observed data and the synthesis (res)

Examples

## Not run:
#Using TideCurve
tc <- TideCurve(dataInput = tideObservation, asdate = "2015/12/06",
                astime = "00:00:00", aedate = "2015/12/31",
                aetime = "23:30:00", ssdate = "2015/12/17",
                sstime = "00:00:00", sedate = "2015/12/31",
                setime = "23:30:00")
res_tc <- ResCurve(tc, tideObservation)

#Using BuildTC and SynTC
your_model <- BuildTC(dataInput = tideObservation, asdate = "2015/12/06",
                       astime = "00:00:00", aedate = "2016/12/31",
                       aetime = "23:30:00", ssdate = "2015/12/17",
                       sstime = "00:00:00", sedate = "2015/12/31",
                       setime = "23:30:00")
res_model <- ResCurve(your_model, tideObservation)
aetime = "23:30:00", keep_data = TRUE)

syn_tc <- SynTC(tmodel = your_model, ssdate = "2015/12/17", sstime = "00:00:00",
                sedate = "2015/12/31", setime = "23:30:00")

syn_tc$data_matrix <- your_model$data_matrix

res_tc_n <- ResCurve(syn_tc, tideObservation)

## End(Not run)

---

**SynTC** | **Synthesizes a tide curve**

---

**Description**

Synthesizes a tide curve; model built with BuildTC().

**Usage**

```
SynTC(tmodel = NULL, ssdate, sstime, sedate, setime, solar_syn = TRUE)
```

**Arguments**

- `tmodel` The model you built with BuildTC(). Please see examples.
- `ssdate` Synthesis start date. This indicates the date you want your tide curve to start with.
- `sstime` Synthesis start time. The starting time for your tide table.
- `sedate` Synthesis end date.
- `setime` Synthesis end time.
- `solar_syn` Compute a solar synthesis? Default is TRUE.

**Value**

Returns a list with two elements, which are of class data.table and data.frame.

- `synthesis.lunar` The lunar synthesis data as a data.table object in UTC.
- `tide.curve` The solar tide curve as a data.table or NULL object (time zone of the observations).

**References**

doi: 10.5194/os1513632019
Examples

```r
## Not run: SynTC(tmodel = your_model, ssdate = "2015/12/17", sstime = "00:00:00",
sedate = "2015/12/31", setime = "23:30:00")
## End(Not run)
```

TideCurve

*Computes tide curves*

Description

Takes a data frame as input with three columns (see example dataset) and returns a tide curve. Internally the analysis is carried out in lunar days. One mean lunar day lasts 1.0350501 mean solar days. Therefore the analysis time period should start one lunar day after the first observation and end one lunar day before the last observation.

Usage

```r
TideCurve(
  dataInput,
  otz = 1,
  km = -1,
  mindt = 30,
  asdate,
  astime,
  aedate,
  aetime,
  ssdate,
  sstime,
  sedate,
  setime
)
```

Arguments

- **dataInput**: A data frame with the columns `observation_date`, `observation_time` and `height`. See attached data for correct formats.
- **otz**: The time zone of the observations
- **km**: The number of nodes between two consecutive mean moon transits. Shall be less or equal to: `round(1440 [min] / time step [min])`. Example: Time step 5 min: Use km = 288 or even smaller. Leave on default (km = -1) and supply `mindt`, when unsure.
- **mindt**: Observation time step in [min]. Default is 30.
- **asdate**: A string indication the date you want the analysis to start with. Format: "yyyy/mm/dd".
- **astime**: A string indicating the time you want the analysis to start with. Format: "hh:mm:ss"
TideCurve

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aedate</td>
<td>A string indication the date you want the analysis to end with. Format: &quot;yyyy/mm/dd&quot;.</td>
</tr>
<tr>
<td>aetime</td>
<td>A string indicating the time you want the analysis to end with. Format: &quot;hh:mm:ss&quot;</td>
</tr>
<tr>
<td>ssdate</td>
<td>Synthesis start date. This indicates the date you want your tide curve to start with. Format: See above</td>
</tr>
<tr>
<td>sstime</td>
<td>Synthesis start time. The starting time for your tide table. Format: See above</td>
</tr>
<tr>
<td>sedate</td>
<td>Synthesis end date. Format: See above</td>
</tr>
<tr>
<td>setime</td>
<td>Synthesis end time. Format: See above</td>
</tr>
</tbody>
</table>

**Value**

Returns a list with elements of the analysis, fitting and the tidal curve for given data

- `synthesis.lunar` The lunar synthesis data as a data.table object in UTC
- `data.matrix` The data needed for analysis
- `tide.curve` The solar tide curve as a data.table object (provided time zone)
- `lm.coeff` Coefficients for the km fitted linear models used in the synthesis as a list of 1-row matrices
- `diff.analyse` Time in days spanning the analysis

**References**

Godin, Gabriel (1972) The Analysis of Tides. Toronto, 264pp

doi: 10.5194/os1513632019


**Examples**

```r
## Not run: TideCurve(dataInput = tideObservation, asdate = "2015/12/06", astime = "00:00:00", aedate = "2015/12/31", aetime = "23:30:00", ssdate = "2015/12/17", sstime = "00:00:00", sedate = "2015/12/31", setime = "23:30:00")
## End(Not run)
```
Description

A sample dataset containing observation date, time and height of high and low water

Usage

tideObservation

Format

A data frame with 10267 rows and 3 variables

observation_date date of observation, character value in "yyyy/mm/dd" format
observation_time time of observation, character value in "hh:mm:ss" format
height observed value, numeric value
Index

* datasets
  tideObservation, 8

BuildDesign, 2
BuildTC, 2

Funcs, 3

ResCurve, 4

SynTC, 5

TideCurve, 6

tideObservation, 8