Package: AHGestimation (via r-universe)

December 5, 2024

Title An R package for Computing Robust, Mass Preserving Hydraulic Geometries and Rating Curves

Description Compute mass preserving 'At a station Hydraulic Geometry' (AHG) fits from river measurements.

Version 0.3.1

Maintainer Mike Johnson <mike.johnson@noaa.gov>

BugReports https://github.com/mikejohnson51/AHGestimation/issues

URL https://github.com/mikejohnson51/AHGestimation

Depends R(>= 4.2.0)

Imports DescTools, dplyr, geodist, mco, pbapply, stats, sf, utils

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests distill, DT, scatterplot3d, ggplot2, ggrepel, kableExtra, knitr, patchwork, rmarkdown, testthat (>= 3.0.0), tidyr

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev make libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev zlib1g-dev

Repository https://mikejohnson51.r-universe.dev

RemoteUrl https://github.com/mikejohnson51/AHGestimation

RemoteRef HEAD

RemoteSha f1783f027477a5eae615abd5fd14f02bc2fc0559

Contents

`ahg_estimate` *Properly estimate AHG values*

Description

Properly estimate AHG values

Usage

```
ahg_estimate(
  df,
  allowance = 0.05,
  gen = 192,
  pop = 200,
  cprob = 0.4,
  mprob = 0.4,
  times = 1,
  scale = 1.5,
  full_fitting = FALSE,
  verbose = FALSE
)
```

Arguments

`df`	hydraulic data.frame with columns named (Q, V, TW, Y). Q and at least one other are required.
`allowance`	allowed deviation from continuity
`gen`	Number of generations to breed.
`pop`	Size of population
`cprob`	Crossover probability
`mprob`	Mutation probability
`times`	how many times (seeds) should nsga2 be run
`scale`	should a scale factor be applied to data pre NSGA-2 fitting
`full_fitting`	should all fits be returned?
`verbose`	should messages be emitted?

Value

list

See Also

Other AHG: `best_optimal()`, `calc_nsga()`, `compute_ahg()`, `min_max()`, `mismash()`

`best_optimal` *Report best optimal*

Description

Report best optimal

Usage

```
best_optimal(best, check, verbose = TRUE)
```

Arguments

`best`	best performing method (character string)
`check`	values to check against
`verbose`	should messages be emitted

Value

vector

See Also

Other AHG: `ahg_estimate()`, `calc_nsga()`, `compute_ahg()`, `min_max()`, `mismash()`

`calc_nsga` *Calculate NSGA2 AHG*

Description

Calculate NSGA2 AHG

Usage

```
calc_nsga(
  df,
  allowance = 0.05,
  r,
  scale = 2,
  gen = 96,
  pop = 500,
  cprob = 0.8,
  mprob = 0.05,
  times = 1
)
```

Arguments

`df`	hydraulic data.frame
`allowance`	allowable deviation from continuity
`r`	fit list
`scale`	should a scale factor be applied to data pre NSGA-2 fitting
`gen`	Number of generations to breed.
`pop`	Size of population
`cprob`	Crossover probability
`mprob`	Mutation probability
`times`	how many times (seeds) should nsga2 be run

Value

data.frame

See Also

Other AHG: `ahg_estimate()`, `best_optimal()`, `compute_ahg()`, `min_max()`, `mismash()`

`compute_ahg` *Approximate AHG relationships*

Description

Approximate AHG relationships using both OLS and NLS methods

Usage

```
compute_ahg(Q, P, type = "relation")
```

Arguments

`Q`	a stream flow time series
`P`	a corresponding time series of a second hydraulic variable
`type`	relationship being tested

Value

data.frame

See Also

Other AHG: `ahg_estimate()`, `best_optimal()`, `calc_nsga()`, `min_max()`, `mismash()`

`compute_channel_slope` *Calculate the slope of 3D linestring*

Description

Given a sf object with 'XYZ' coordinates, return a vector of numeric values representing the average slope of each linestring in the sf data frame input.

The default calculates the slope using ‘slope_weighted()‘. You can also use ‘slope_mean()‘ or any other function that takes the same inputs as these functions.

Usage

```
compute_channel_slope(path, fun = slope_weighted, directed = FALSE)
```

Arguments

`path`	an XYZ LINESTRING representing the path of travel
`fun`	The slope function to calculate per element, ‘slope_weighted‘ is the default.
`directed`	Should the value be directed? ‘FALSE‘ by default. If ‘TRUE‘ the result will be negative when it represents a downslope (when the end point is lower than the start point).

Value

A vector of slopes associated with each linear element The value is a proportion representing the change in elevation for a given change in horizontal distance.

See Also

Other hydraulics: `compute_hydraulic_params()`, `compute_n()`, `cross_section()`, `extract_thalweg()`, `slope_matrix()`

`compute_hydraulic_params` *Approximate channel coeffiecent*

Description

Approximate the hydraulic values from AHG fit

Usage

```
compute_hydraulic_params(fit)
```

Arguments

`fit`	output of ahg_estimate

Value

numeric

See Also

Other hydraulics: `compute_channel_slope()`, `compute_n()`, `cross_section()`, `extract_thalweg()`, `slope_matrix()`

`compute_n` *Approximate Roughness*

Description

Approximate median roughness using Manning Equation

Usage

```
compute_n(df, S = 0.02)
```

Arguments

`df`	a data.frame with at least Y and V.
`S`	reach scale longitudinal slope (m/m). Default mean of the nhdplusV2

Value

numeric

See Also

Other hydraulics: `compute_channel_slope()`, `compute_hydraulic_params()`, `cross_section()`, `extract_thalweg()`, `slope_matrix()`

`cross_section` *Approximate channel shape*

Description

Get a list of points from x axis of a cross section and max depth and produce depth values for those points based on channel shape

Usage

```
cross_section(r, TW = 30, Ymax = 2, n = 30)
```

Arguments

`r`	The corresponding Dingman's r coefficient
`TW`	width of the channel at bankfull
`Ymax`	maximum depth of the channel at bankfull
`n`	the number of points to construct in the XS

Value

depth values every 1m along the cross section

See Also

Other hydraulics: `compute_channel_slope()`, `compute_hydraulic_params()`, `compute_n()`, `extract_thalweg()`, `slope_matrix()`

`date_filter` *Implements filtering by date*

Description

Data is filtered when it is beyond a specified year threshold (e.g. 5 years old). The relative date is based on the newest observation in the data set. Optionally, the maximum flow (Q) record can be retained.

Usage

```
date_filter(df, years, keep_max = FALSE)
```

Arguments

`df`	a data.frame with at least a date and Q field.
`years`	the number of allowed history
`keep_max`	Should the largest flow record be kept, even if older then ”years”

Value

data.frame

See Also

Other filters: `mad_filter()`, `nls_filter()`, `qva_filter()`, `significance_check()`

`extract_thalweg` *Extract Thalweg From a data.frame of cross sections, a classified thalweg can be extracted as the connected LINESTRING*

Description

Extract Thalweg From a data.frame of cross sections, a classified thalweg can be extracted as the connected LINESTRING

Usage

```
extract_thalweg(xs, crs = 5070)
```

Arguments

`xs`	a data.frame containing cross sectional data. Required columns are hf_id, cs_id, X, Y, Z
`crs`	the CRS of the XY coordinates

Value

XYZ LINESTRING object

See Also

Other hydraulics: `compute_channel_slope()`, `compute_hydraulic_params()`, `compute_n()`, `cross_section()`, `slope_matrix()`

`mad_filter` *Implements filtering by median absolute deviation*

Description

An iterative outlier detection procedure is run based on to the linear regression residuals. Values of log-transformed TW, V, and Y residuals falling outside a specified median absolute deviation (MAD) envelope are excluded. Regression coefficients were recalculated and the outlier detection procedure was reapplied until no outliers are detected. This method was identified in HyG

Usage

```
mad_filter(df, envelope = 3)
```

Arguments

`df`	a data.frame with at least a Q and one other AHG field (Y. TW, V).
`envelope`	MAD envelope

Value

data.frame

See Also

Other filters: `date_filter()`, `nls_filter()`, `qva_filter()`, `significance_check()`

`min_max` *Find thresholds for coefficient and exponent limits.*

Description

Find thresholds for coefficient and exponent limits.

Usage

```
min_max(df, scale = 2)
```

Arguments

`df`	hydraulic data.frame
`scale`	Scale by set factor. This limits the exponent at coefficients to the range of (1/s) * nls; s * nls

Value

list

See Also

Other AHG: `ahg_estimate()`, `best_optimal()`, `calc_nsga()`, `compute_ahg()`, `mismash()`

`mismash` *Compute all combos!*

Description

Compute all combos!

Usage

```
mismash(v, V, TW, Y, Q, r, allowance)
```

Arguments

`v`	values
`V`	Velocity time series
`TW`	Top width time series
`Y`	Depth time series
`Q`	Discharge time series
`r`	rrr TODO
`allowance`	Allowable deviation from continuity

Value

list

See Also

Other AHG: `ahg_estimate()`, `best_optimal()`, `calc_nsga()`, `compute_ahg()`, `min_max()`

`nls_filter` *Implements NLS filtering*

Description

An NLS fit provides the best relation by relation fit. For each provided relationship, an NLS fit is computed and used to estimate the predicted V,TW,Y for a given Q. If the actual value is outside the specified allowance it is removed.

Usage

```
nls_filter(df, allowance = 0.5)
```

Arguments

`df`	a data.frame with at least a Q and one other AHG field (Y. TW, V).
`allowance`	how much deviation from observed should be allowed (default = .5)

Value

data.frame

See Also

Other filters: `date_filter()`, `mad_filter()`, `qva_filter()`, `significance_check()`

`nrmse` *Normalized Root Mean Square Error*

Description

Normalized root mean square error (NRMSE) between sim and obs, with treatment of missing values

Usage

```
nrmse(sim, obs)
```

Arguments

`sim`	numeric vector simulated values
`obs`	numeric vector observed values

Value

numeric

See Also

Other evaluation: `pbias()`

`nwis` *Sample gage data Manual measurements made at NWIS site 01096500 Q_cms is a mandatory argument and at least one of TW_m, V_ms, or Y_m.*

Description

Sample gage data Manual measurements made at NWIS site 01096500 Q_cms is a mandatory argument and at least one of TW_m, V_ms, or Y_m.

Usage

```
nwis
```

Format

A data frame with 245 rows and 6 columns:

siteID NWIS ID

date date of measurement

Q_cms Steamflow (cubic meters per second)

Y_m Depth (meters)

V_ms Velocity (meters per second)

TW_m Top width (meters)

`pbias` *Percent Bias*

Description

Percent Bias between sim and obs, with treatment of missing values.

Usage

```
pbias(sim, obs)
```

Arguments

`sim`	numeric vector simulated values
`obs`	numeric vector observed values

Value

numeric

See Also

Other evaluation: `nrmse()`

`qva_filter` *Implements filtering by continuity*

Description

The function tests if the measured Q is outside of the expected range based on the product of measured velocity, top-width, and depth (e.g. Q vA)

Usage

```
qva_filter(df, allowance = 0.05)
```

Arguments

`df`	a data.frame with a Q, Y, TW, V and field.
`allowance`	how much deviation from equality should be allowed (default = .05)

Value

data.frame

See Also

Other filters: `date_filter()`, `mad_filter()`, `nls_filter()`, `significance_check()`

`significance_check` *Implements significance check*

Description

The relationship between all supplied log transformed variables are computed. If the p-value of any of these is less then the supplied p-value an error message is emitted.

Usage

```
significance_check(df, pvalue = 0.05)
```

Arguments

`df`	a data.frame with at least a Q and one other AHG field (Y. TW, V).
`pvalue`	Significant p-value (default = .05)

Value

data.frame

See Also

Other filters: `date_filter()`, `mad_filter()`, `nls_filter()`, `qva_filter()`

`slope_matrix` *Calculate the gradient of line segments from a 3D matrix of coordinates*

Description

Calculate the gradient of line segments from a 3D matrix of coordinates

Usage

```
slope_matrix(mat, lonlat = TRUE)

slope_weighted(mat, lonlat = TRUE, directed = FALSE)

slope_mean(mat, lonlat = TRUE, directed = FALSE)
```

Arguments

`mat` Matrix containing coordinates and elevations. The matrix should have three columns: X, Y, and Z. In data with geographic coordinates, Z values are assumed to be in meters. In data with projected coordinates, Z values are assumed to have the same units as the X and Y coordinates.

`lonlat` Are the elements provided in longitude/latitude coordinates? By default, value is from the CRS of the routes (‘sf::st_is_longlat(...)‘).

Value

A vector of slopes associated with each LINE element The output value is a proportion representing the change in elevation for a given change in horizontal distance.

See Also

Other hydraulics: `compute_channel_slope()`, `compute_hydraulic_params()`, `compute_n()`, `cross_section()`, `extract_thalweg()`

Index