loadcell

February 20, 2019

Title Tools for processing load cell data for DMR Vertical Line project.

Version 0.0.0.9000

Description Tools for processing load cell data for DMR Vertical Line project.

Depends R (>= 3.4)

License DMR/affiliate use only.

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

R topics documented:

`adjust_load` *Load to tension*

Description

Adjusts load values from CSV data based on a correction multipler (the Keliher conversion factor.)

Usage

```
adjust_load(data, kfactor)
```

Arguments

`data`	An object of class ‘LoadCellData‘.
`kfactor`	The Keliher conversion factor to apply to the load data.

Value

An object of class ‘LoadCellData.‘

Author(s)

Bill DeVoe, <William.DeVoe@maine.gov>

export_loadcell *Export load cell data*

Description

For an input object of class ‘LoadCellHauls‘ or ‘LoadCellPeaks‘, flattens the object data into CSVs of data. Optionally, a prefix can be given to be added to the filename. The following CSVs are created:

prefix-lc_data.csv The load cell data, with UID and haul numbers added.

prefix-lc_haul.csv The haul attributes, including the settings used for peak analysis. UID is included to relate hauls to load cell data.

prefix-lc_peaks.csv The actual peak values for each haul, only if peak analysis was successful for the haul. Includes UID and haul number.

Usage

```
export_loadcell(data, prefix)
```

Arguments

data	An object of class ‘LoadCellHauls‘ or ‘LoadCellPeaks‘
prefix	*Optional* A prefix to append to the filenames of the output CSVs.

Value

Boolean True if successful.

Author(s)

Bill DeVoe, <William.DeVoe@maine.gov>

`load_csvs` *Load CSVs*

Description

For an input directory, loads CSVs created by the DMR load cell software into an output class 'LoadCellData'.

Usage

```
load_csvs(dir = getwd())
```

Arguments

`dir`	Directory to scan for load cell CSVs. If no directory is provided, the current working directory will be used.

Value

A class of type 'LoadCellData'; class items are named with the CSV filename. Each class item represents a nested list containing the following attributes:

data A dataframe of the load cell data contained in the CSV.

sn Serial number of the load cell USB adapter.

traps The number of traps per trawl, if entered in the interface.

start_dt A POSIXlt object representing the start datetime of the CSV.

end_dt A POSIXlt object representing the end datetime of the CSV.

seconds The number of elapsed seconds in the CSV.

max_load An integer representing the maximum load within the CSV.

min_load An integer representing the minimum load within the CSV; useful for determining a "zero" value for the weight of the block with no line.

kfactor The Keliher conversion factor applied to the data.

Author(s)

Bill DeVoe, `<William.DeVoe@maine.gov>`

`parse_hauls` *Parse Hauls*

Description

For an input object of class ‘LoadCellData‘, parses load cell file data into hauls by splitting apart load cell data based on minimum load, time between hauls, and minimum haul length. Return an object of class ‘LoadCellHauls‘.

Usage

```
parse_hauls(data, split, min_load, min_time, min_gap, pass)
```

Arguments

`data`	An object of class ‘LoadCellData‘ representing data loaded from load cell CSVs.
`split`	Boolean indicating if CSV files should be split into separate hauls; if a fisherman recorded separate CSVs for each haul, this option would be False and the hauls will be numbered sequentially per day. If True, hauls will be parsed from the CSV data according to options set in the remaining function arguments.
`min_load`	Numeric - If split is True, load cell data below this threshold will be excluded to create gaps between hauls. If this argument is omitted, the minimum value for each CSV will be used.
`min_time`	*Optional* Numeric - If split is True, hauls resulting from excluding load values below ‘min_load‘ must be this many seconds in duration to be considered a haul. This is useful for excluding junk data produced by a load cell being jostled in rough seas in between actual hauls.
`min_gap`	*Optional* Numeric - If split is True, a haul must begin this many seconds after the previous haul to be considered a separate haul.
`pass`	*Optional* Numeric; if the CSV is less than this many seconds in length, it will not be split and will be assumed to represent a single haul.

Value

An object of class ‘LoadCellHauls‘. Each item in the class is represented by a unique name constructed by concaternation of the load cell serial number, date, and haul number, ie "SN-2018-03-24-1". Each item has attributes as follows:

haul The haul number of the data.

data A dataframe containing the load cell data for the haul.

sn Serial number of the load cell adapter.

traps The number of traps per trawl, if entered in the interface.

start_dt A POSIXlt object representing the start datetime of the haul.

end_dt A POSIXlt object representing the end datetime of the haul.

seconds The number of elapsed seconds in the haul.

max_load An integer representing the maximum load within the haul.

kfactor The Keliher conversion factor applied to the data.

Author(s)

Bill DeVoe, `<William.DeVoe@maine.gov>`

`parse_peaks` *Parse peaks*

Description

Searches for peaks and valleys in load data from an object of class 'LoadCellHauls'.

Usage

```
parse_peaks(data, span = 0.05, peakdist = 10, peakheight = 200)
```

Arguments

`data`	Object of class 'LoadCellHauls'
`span`	Loess smoothing factor; defaults to .05
`peakdist`	The minimum distance between peaks in index values
`peakheight`	The minimum height of peaks in LBF

Value

An object of class 'LoadCellPeaks', containing all the attributes of object class 'LoadCellHauls', plus the following attributes:

span The span used to apply Loess smoothing.

peakdist The peak distance parameter used to apply Loess smoothing.

peakheight The peak height parameter used to apply Loess smoothing.

peak_analysis Boolean indicating if peak analysis was successful; if False, the remaining attributes will not be present.

smoothed_peaks A dataframe containing the Index and Load for peaks located in the smoothed data for the top trapcount peaks.

smoothed_valleys A dataframe containing the Index and Load for valleys located in the smoothed data corresponding to peaks located.

smoothed A dataframe containing the Index and Load for each load after Loess smoothing was applied.

actual_peaks A dataframe containing the Index and Load values for actual peaks corresponding to the smoothed peaks located within the load cell data.

Author(s)

Bill DeVoe, `<William.DeVoe@maine.gov>`

plot_hauls *Plot Hauls*

Description

For an input object of class ‘LoadCellPeaks‘, creates a plot for each haul containing the load cell data, smoothed data, and peak analysis.

Usage

```
plot_hauls(data, fileout)
```

Arguments

`data`	An object of class ‘LoadCellPeaks‘.‘
`fileout`	*Optional* - Filename to save plots into. A PDF with this name will be created in the current working directory. If the file exists, it will be overwritten.

Value

Boolean True if successful. If a fileout is provided, a PDF will be created containing the plots. If the fileout argument is omitted, plots will be output to the console/plot viewer.

Author(s)

Bill DeVoe, <William.DeVoe@maine.gov>

Index