Package: kwb.rain (via r-universe)

August 25, 2024

Type Package

Title Functions to Handle and Plot Rain Data at KWB

Version 0.2.0

Description Functions to handle and plot rain data at KWB.

License MIT + file LICENSE

Imports kwb.datetime, kwb.event, kwb.misc, kwb.read, kwb.utils, lattice

Suggests knitr, rmarkdown

VignetteBuilder knitr

Remotes github::kwb-r/kwb.datetime, github::kwb-r/kwb.event, github::kwb-r/kwb.misc, github::kwb-r/kwb.read, github::kwb-r/kwb.utils

Encoding UTF-8

LazyData TRUE

RoxygenNote 7.1.2

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

RemoteUrl https://github.com/KWB-R/kwb.rain

RemoteRef HEAD

RemoteSha d52974ff3919878d8b962c49ff819a942e207eec

Contents

`aggregateByDay` *aggregate rain data by day*

Description

aggregate rain data by day

Usage

```
aggregateByDay(x, FUN, ..., timeColumn = 1, excludeColumns = 1:2)
```

Arguments

`x`	data frame containing at least one date and time column
`FUN`	aggregation function
`...`	arguments passed to FUN
`timeColumn`	number of the date and time column
`excludeColumns`	vector of column numbers to be excluded before aggregation

`checkForDuplicates` *Stop if there are Duplicates and show them*

Description

Stop if there are Duplicates and show them

Usage

```
checkForDuplicates(x, keys)
```

Arguments

`x`	data frame
`keys`	columns that are expected to be key columns and that should not contain duplicated value combinations

`compactForGauges` *Compact Signal Data*

Description

Compact Signal Data

Usage

```
compactForGauges(
  signals,
  gauges,
  validation,
  level = 2,
  stars = TRUE,
  dbg = TRUE
)
```

Arguments

`signals`	data frame as returned by `kwb.read:::.toLongFormat(signals, type = "signal")`
`gauges`	character vector of rain gauge names
`validation`	data frame
`level`	level of compression
`stars`	if `TRUE` (default) stars are used to indicate differences
`dbg`	if `TRUE` (default) debug messages are shown

`correctRainDiffs` *Correct Rain Height Differences*

Description

Add one day to the data in the "Date" column if the time in column "To" is before the time in column "From"

Usage

```
correctRainDiffs(
  rainDiffs,
  userCorrections,
  gauges = names(rainDiffs)[-(1:2)],
  dbg = TRUE
)
```

Arguments

`rainDiffs` rain height differences as provided by `kwb.read:::.toDifferences`

`userCorrections`
user correction data as returned by `readUserCorrection`

`gauges` names of rain gauges (as they appear as column names in `rainDiffs`) to be considered

`dbg` if `TRUE` (default) debug messages are shown

`corrToLongFormat` *rain correction data from 'wide' format to 'long' format*

Description

rain correction data from 'wide' format to 'long' format

Usage

```
corrToLongFormat(corrData)
```

Arguments

`corrData` correction data as returned by `kwb.read::readBwbRainCorrection`

`defaultArgumentAssignment` *Output Template for Argument Assignments*

Description

Output Template for Argument Assignments

Usage

```
defaultArgumentAssignment(x, type = c("R", "csv")[1])
```

Arguments

`x` vector of character

`type` one of `"R", "csv"`

`getCorrectionCases` *get the cases of required rain data correction*

Description

get a data frame with each row representing a gauge and a day at which rain data needs to be corrected

Usage

```
getCorrectionCases(corrData, rainData, ...)
```

Arguments

`corrData`	correction data as returned by `kwb.read::readBwbRainCorrection`
`rainData`	rain data as returned by `kwb.read::readBwbRainData`
`...`	additional arguments passed to the final `merge`, such as `all.x` (keep correction data even for days for which no rain data is available), `all.y` (keep rain data even for days for which no correction data is available)

`getDailyCumulativeRain` *cumulate rain data within each day*

Description

cumulate rain data within each day

Usage

```
getDailyCumulativeRain(rd, to.long = TRUE)
```

Arguments

`rd`	data frame containing rain data as returned by `kwb.read::readBwbRainData`
`to.long`	if `TRUE` (default) the result data frame is converted to 'long' format.

getNeighbours *Get the Names of Neighbouring Rain Gauges*

Description

Get the Names of Neighbouring Rain Gauges

Usage

```
getNeighbours(signals, gaugeInfo, n = 2, gaugePos = NULL)
```

Arguments

signals	data frame as returned by kwb.read:::.toLongFormat(signals, type = "signal")
gaugeInfo	data frame as returned by readAllBwbSignals in attribute gaugeInfo
n	number of neighbours
gaugePos	data frame containing the gauge positions as required by getGaugeDistances

plotCumulativeRain *plot cumulative rain*

Description

plot cumulative rain

Usage

```
plotCumulativeRain(rd.long, to.pdf = TRUE)
```

Arguments

rd.long	data frame in 'long' format as returned by getDailyCumulativeRain
to.pdf	If TRUE (default) the plot goes into a temporary PDF file, otherwise to the current plot device

plotDailyRainHeightTable *Print Tables of Daily Rain Heights (as Plots!)*

Description

Print Tables of Daily Rain Heights (as Plots!)

Usage

```
plotDailyRainHeightTable(
  rd,
  landscape = FALSE,
  cex = ifelse(landscape, 0.8, 0.6),
  ppp = ifelse(landscape, 1, 2),
  to.pdf = TRUE
)
```

Arguments

rd	data frame with time stamps in first column and rain heights per time interval for different rain gauges in all the other columns
landscape	passed to kwb.utils::preparePdf
cex	passed to hsPrintToPlot
ppp	number of plots per page
to.pdf	if TRUE the output goes into a temporary PDF file

plotRainEventOverview *Plot all rain events contained in rain data*

Description

Plot all rain events contained in rain data

Usage

```
plotRainEventOverview(rd, timeColumn = "tEnd_BWB", to.pdf = TRUE)
```

Arguments

rd	data frame with time stamps in first column and rain heights per time interval for different rain gauges in all the other columns
timeColumn	name of the column containing the timestamps. Default: "tEnd_BWB"
to.pdf	if TRUE the output goes into a temporary PDF file

`rainToLongFormat` *rain data from 'wide' format to 'long' format*

Description

rain data from 'wide' format to 'long' format

Usage

```
rainToLongFormat(rainData)
```

Arguments

`rainData`	rain data as returned by `kwb.read::readBwbRainData`

`readUserCorrection` *Read User Correction Data from CSV File*

Description

Read User Correction Data from CSV File

Usage

```
readUserCorrection(
  file,
  sep = ";",
  country = "de",
  date.format = "%d.%m.%Y",
  dbg = TRUE
)
```

Arguments

`file`	full path to CSV file
`sep`	column separator
`country`	country code ("de" or "en")
`date.format`	date format string
`dbg`	if `TRUE` (default) debug messages are shown

reformat_rain_data *Reformat Rain Data*

Description

Reformat rain data as returned by correctRainDiffs to the format required by plotRainEventOverview.

Usage

```
reformat_rain_data(rain)
```

Arguments

rain	data frame with begin and end timestamps of five minute time intervals in the first two (character) columns, rain heights of five minute intervals in the remaining columns. Such a data frame is returned by plotRainEventOverview

toGaugeSignals *Convert raw Signals and Validation Data to Valid Signals*

Description

Convert raw Signals and Validation Data to Valid Signals

Usage

```
toGaugeSignals(signals, validation, neighbours, digits = 3, ...)
```

Arguments

signals	data frame as returned by kwb.read:::.toLongFormat(signals, type = "signal")
validation	data frame as returned by toValidationTable
neighbours	vector of names of neighbour gauges
digits	number of digits to which the signals are rounded
...	arguments passed to compactForGauges

toValidationTable *Create Validation Table from Signal and Correction Data*

Description

Create Validation Table from Signal and Correction Data

Usage

```
toValidationTable(signals, corrections)
```

Arguments

`signals`	data frame as returned by `kwb.read:::.toLongFormat(signals, type = "signal")`
`corrections`	data frame as returned by `kwb.read:::.toLongFormat(corrections, type = "correction")`

writeDiffFiles *Write Files containing the "Diff" between raw and valid rain signal*

Description

Write Files containing the "Diff" between raw and valid rain signal

Usage

```
writeDiffFiles(
  gaugeSignals,
  dir.out,
  gauges = names(gaugeSignals),
  years = NULL,
  suffix = NULL,
  sep = ";",
  dec = ",",
  dbg = TRUE
)
```

Arguments

`gaugeSignals`	list of data frames with each element representing one rain gauge
`dir.out`	full path to output directory
`gauges`	vector of character containing the names of the gauges for which files are to be written
`years`	vector of integer representing the years for which diff files are to be generated

`suffix`	to be used in the names of the created files
`sep`	column separator in created CSV files
`dec`	decimal character in created CSV files
`dbg`	if `TRUE` (default) debug messages are shown

Index