Package 'hydroToolkit'

October 13, 2022

Type Package

Title Hydrological Tools for Handling Hydro-Meteorological Data from Argentina and Chile

Version 0.1.0

Date 2020-05-07

Author Ezequiel Toum <etoum@mendoza-conicet.gob.ar>

Maintainer Ezequiel Toum <etoum@mendoza-conicet.gob.ar>

Description Read, plot, manipulate and process hydro-meteorological data from Argentina and Chile.

Depends R (>= 2.10)

License GPL (>= 3)

Imports ggplot2, plotly, lubridate, utils, methods, readxl, reshape2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-16 10:00:02 UTC

R topics documented:

`agg_hydroMet` *Aggregate slot data*

Description

This method provides common functions to aggregate the data inside a slot.

Usage

```
agg_hydroMet(
  obj,
  slot_name,
  col_name,
  fun,
  period,
  out_name = NULL,
  start_month = NULL,
  end_month = NULL,
  allow_NA = NULL
)

## S4 method for signature 'hydroMet_BDHI'
agg_hydroMet(
```

```
  obj,
  slot_name,
  col_name,
  fun,
  period,
  out_name = NULL,
  start_month = NULL,
  end_month = NULL,
  allow_NA = NULL
)

## S4 method for signature 'hydroMet_DGI'
agg_hydroMet(
  obj,
  slot_name,
  col_name,
  fun,
  period,
  out_name = NULL,
  start_month = NULL,
  end_month = NULL,
  allow_NA = NULL
)

## S4 method for signature 'hydroMet_CR2'
agg_hydroMet(
  obj,
  slot_name,
  col_name,
  fun,
  period,
  out_name = NULL,
  start_month = NULL,
  end_month = NULL,
  allow_NA = NULL
)

## S4 method for signature 'hydroMet_IANIGLA'
agg_hydroMet(
  obj,
  slot_name,
  col_name,
  fun,
  period,
  out_name = NULL,
  start_month = NULL,
  end_month = NULL,
  allow_NA = NULL
```

)

Arguments

obj	an hydroMet_XXX class object. This method is not allowed for hydroMet_compact class. This is because this class was thought as *ready to use*, so when building this class you should have already aggregated your data.
slot_name	a single or vector string containing the slot(s) to aggregate.
col_name	a single or vector string with the name of the column to aggregate in slot_name.
fun	a single or vector string containing one of the following functions: 'mean', 'min', 'max' or 'sum'.
period	a single or vector string with the period of aggregation: 'hourly', 'daily', 'monthly', 'annual' or 'climatic'. **NOTE_1**: the 'climatic' option returns the all series annual statistics ('fun'). **NOTE_2**: if the object is of class hydroMet_IANIGLA you must provide a single period value.
out_name	optional. Single or vector string with the output column name of the variable to aggregate.
start_month	optional. Numeric (or numeric vector) value of the first month. It only makes sense if the 'period' is 'annual'. **NOTE**: as an example, in case you have just two slots (out of five) that you want to aggregate annually you must provide a vector of length two. Default value is January. **NOTE***: if the object is of class hydroMet_IANIGLA you must provide a single start_month value.
end_month	optional. Numeric (or numeric vector) value of the last month. It only makes sense if the 'period' is 'annual'. **NOTE**: as an example, in case you have just two slots (out of five) that you want to aggregate annually you must provide a vector of length two. Default value es December. **NOTE***: if the object is of class hydroMet_IANIGLA you must provide a single end_month value.
allow_NA	optional. Numeric (or numeric vector) value with the maximum allowed number of NA_real_ values. By default the function will not tolerate any NA_real_ in an aggregation period (and will return NA_real_ instead).

Value

An hydroMet_XXX class object with the required slot(s) aggregated.

Functions

- agg_hydroMet,hydroMet_BDHI-method: aggregation method for BDHI data
- agg_hydroMet,hydroMet_DGI-method: aggregation method for DGI data
- agg_hydroMet,hydroMet_CR2-method: aggregation method for CR2 data
- agg_hydroMet,hydroMet_IANIGLA-method: aggregation method for IANIGLA data

Examples

```
# Create BDHI hydro-met station
guido <- create_hydroMet(class_name = 'BDHI')

# List with meteorological variables (slots in BDHI's object)
cargar <- list('precip', 'Qmd', 'Qmm')

# Assign as names the files
hydro_files  <- list.files( system.file('extdata', package = "hydroToolkit"), pattern = 'Guido' )
names(cargar) <- hydro_files

# Build the object with the met records
guido <- build_hydroMet(obj = guido, slot_list = cargar,
               path = system.file('extdata', package = "hydroToolkit") )

# Aggregrate precipitation serie
guido <- agg_hydroMet(obj = guido, slot_name = 'precip', col_name = 'precip', fun = 'sum',
       period = 'monthly', out_name = 'P_month', allow_NA = 3)
```

`agg_serie` *Aggregates a data frame to a larger time period*

Description

This is a useful function to easily aggregate your data.

Usage

```
agg_serie(
  df,
  fun,
  period,
  out_name,
  start_month = NULL,
  end_month = NULL,
  allow_NA = NULL
)
```

Arguments

`df`	data frame with class `Date` or `POSIXct` in the first column. The function always aggregates the second column.
`fun`	string containing one of the following functions: '`mean`', '`min`', '`max`' or '`sum`'.
`period`	string with the period of aggregation: '`hourly`', '`daily`', '`monthly`', '`annual`' or '`climatic`'. **NOTE**: the 'climatic' option returns the all series annual statistics ('fun').

out_name	string with the output column name of the variable to aggregate.
start_month	optional. Numeric value of the first month. It only makes sense if the 'period' is 'annual'.
end_month	optional. Numeric value of the last month. It only makes sense if the 'period' is 'annual'.
allow_NA	optional. Numeric value with the maximum allowed number of NA_real_ values. By default the function will not tolerate any NA_real_ in an aggregation period (and will return NA_real_ instead).

Value

A data frame with to columns: the date and the aggregated variable.

Examples

```
# Path to file
dgi_path  <- system.file('extdata', package = "hydroToolkit")

toscas <- read_DGI(file = 'Toscas.xlsx', sheet = 'tmean', path = dgi_path)

# Monthly mean temperature
m_toscas <- agg_serie(df = toscas, fun = 'mean', period = 'monthly', out_name = 'T_month')
```

---

build_hydroMet *Automatically load native data files*

Description

This method is the recommended one for loading your data-sets (as provided by the agency).

Usage

```
build_hydroMet(
  obj,
  slot_list,
  path = NULL,
  col_names = NULL,
  start_date = NULL,
  end_date = NULL
)

## S4 method for signature 'hydroMet_BDHI'
build_hydroMet(obj, slot_list, path = NULL)

## S4 method for signature 'hydroMet_CR2'
build_hydroMet(obj, slot_list, path = NULL)
```

```
## S4 method for signature 'hydroMet_DGI'
build_hydroMet(obj, slot_list, path = NULL)

## S4 method for signature 'hydroMet_IANIGLA'
build_hydroMet(obj, slot_list, path = NULL)

## S4 method for signature 'hydroMet_compact'
build_hydroMet(
  obj,
  slot_list,
  col_names = NULL,
  start_date = NULL,
  end_date = NULL
)
```

Arguments

obj	an hydroMet_XXX class object (see create_hydroMet).
slot_list	a list containing (in each element) a vector string with the slot names. The name of the list elements are the native file names (e.g.: *Qmd_Guido_BDHI.txt*). **NOTE**: when the obj argument is of class hydroMet_compact, slot_list allows to build from multiple objects. So, in this case you have to provide a list of list: the top list contains as names the objects names (as you read them from *Global Environment*); then every object (top level) contains another list with slot names as names and the column(s) number(s) to extract as numeric value. E.g.: list(bdhi_obj = list(Qmd = 2, Qmm = c(2, 5)), cr2_obj = list(precip = 4)).
path	string with the files directory. If not provided, the method will use the current working directory. **NOTE**: this argument is harmless for an object of class hydroMet_compact.
col_names	it just make sense if 'obj' argument is of hydroMet_compact class. String vector with the names of the column output. Default value (NULL) will return expressive column names.
start_date	it just make sense if 'obj' argument is of hydroMet_compact class. String or POSIXct with the starting date to extract. You can use start_date without end_date. In this case you will subset your data from start_date till the end.
end_date	it just make sense if 'obj' argument is of hydroMet_compact class. String or POSIXct with the last date to extract. You can use end_date without start_date. In this case you will subset your data from the beginning till end_date.

Value

An S4 object of class hydroMet_XXX with the data loaded in each slot.

Functions

- build_hydroMet,hydroMet_BDHI-method: build up method for BDHI class

- build_hydroMet,hydroMet_CR2-method: build up method for CR2 class
- build_hydroMet,hydroMet_DGI-method: build up method for DGI class
- build_hydroMet,hydroMet_IANIGLA-method: build up method for IANIGLA class
- build_hydroMet,hydroMet_compact-method: build up method for compact class

Examples

```
# Path to file
dgi_path  <- system.file('extdata', package = "hydroToolkit")
file_name <- list.files(path = dgi_path, pattern = 'Toscas')

# Read Toscas
var_nom <- list(slotNames(x = 'hydroMet_DGI')[2:7])
names(var_nom) <- file_name

# Load Toscas meteo station data
toscas_dgi <- create_hydroMet(class_name = 'DGI')
toscas_dgi <- build_hydroMet(obj = toscas_dgi, slot_list = var_nom,
                 path = dgi_path)
```

---

create_hydroMet *Creates an hydroMet class or subclass.*

Description

This function is the constructor of hydroMet class and its subclasses.

Usage

```
create_hydroMet(class_name = "hydroMet")
```

Arguments

class_name	string with the name of the class. Valid arguments are: hydroMet, BDHI, CR2, DGI, IANIGLA or compact.

Value

an S4 object of class hydroMet

Examples

```
# Create class 'hydroMet'
met_station <- create_hydroMet(class_name = 'hydroMet')

# Subclass 'BDHI'
bdhi_station <- create_hydroMet(class_name = 'BDHI')
```

```
# Subclass 'DGI'
dgi_station <- create_hydroMet(class_name = 'DGI')

# Subclass 'CR2'
cr2_station <- create_hydroMet(class_name = 'CR2')

# Subclass 'IANIGLA'
ianigla_station <- create_hydroMet(class_name = 'IANIGLA')
```

`fill_serie` *Find non-reported dates and fill them with* `NA_real_`

Description

This function complete non-reported dates and assign `NA_real_` as their value.

Usage

```
fill_serie(df, colName, timeStep)
```

Arguments

`df`	data frame with date and numeric vector as first and second column respectively.
`colName`	output colname of the numeric variable, e.g.: 'Qmd(m3/s)'.
`timeStep`	character with a valid time step: 'day', 'month', '4h', 'day/3', 'hour'.

Value

A data frame with missing time steps filled with NA's.

Examples

```
# Create a data frame
dates   <- seq.Date(from = as.Date('1990-01-01'), to = as.Date('1990-12-01'), by = 'm')
met_var <- runif(n = 12, 0, 10)

met_serie <- data.frame(dates, met_var)

# Fill serie
met_fill  <- fill_serie(df = met_serie, colName = 'Temp', timeStep = 'day')
```

`fill_value` *Fill a time interval in a data frame with a specific numeric value*

Description

Assign specific values to a time interval.

Usage

```
fill_value(df, col, value, from, to)
```

Arguments

`df`	data frame with the first column being the date and the others numeric variables.
`col`	numeric vector with column(s) number(s) to be filled.
`value`	numeric or `NA_real_`. This numeric vector contains the elements to be fill in.
`from`	character, Date or POSIXct with the first date to be filled.
`to`	character, Date or POSIXct with the last date to be filled.

Value

A data frame filled with the 'value' in the specified time period.

Examples

```
# Create a data frame
dates   <- seq.Date(from = as.Date('1990-01-01'), to = as.Date('1990-12-01'), by = 'm')
met_var <- runif(n = 12, 0, 10)

met_serie <- data.frame(dates, met_var)

# Fill serie
met_fill <- fill_serie(df = met_serie, colName = 'Temp', timeStep = 'day')

# Now fill value
met_fill <- fill_value(df = met_fill, col = 2, value = 10, from = '1990-02-01', to = '1990-02-15')
```

get_hydroMet *Get the slot(s) content(s)*

Description

Extract the slots that you want from an hydroMet or hydroMet_XXX class.

Usage

```
get_hydroMet(obj, name = NA_character_)

## S4 method for signature 'hydroMet'
get_hydroMet(obj, name = NA_character_)

## S4 method for signature 'hydroMet_BDHI'
get_hydroMet(obj, name = NA_character_)

## S4 method for signature 'hydroMet_DGI'
get_hydroMet(obj, name = NA_character_)

## S4 method for signature 'hydroMet_IANIGLA'
get_hydroMet(obj, name = NA_character_)

## S4 method for signature 'hydroMet_CR2'
get_hydroMet(obj, name = NA_character_)

## S4 method for signature 'hydroMet_compact'
get_hydroMet(obj, name = NA_character_)
```

Arguments

obj	an hydroMet or hydroMet_XXX class object.
name	a valid single string or vector string with the required slot name(s).

Value

A list with the slot's data.

Functions

- get_hydroMet,hydroMet-method: get method for generic hydroMet object
- get_hydroMet,hydroMet_BDHI-method: get method for BDHI class
- get_hydroMet,hydroMet_DGI-method: get method for DGI class
- get_hydroMet,hydroMet_IANIGLA-method: get method for IANIGLA class
- get_hydroMet,hydroMet_CR2-method: get method for CR2 class
- get_hydroMet,hydroMet_compact-method: get method for compact class

Examples

```
# Create an IANIGLA object
cuevas <- create_hydroMet(class_name = 'IANIGLA')

# Extract one of its slots
tair <- get_hydroMet(obj = cuevas, name = 'tair')
```

`hydroMet-class` `hydroMet` *superclass object*

Description

A suitable object for store basic information about an hydro-meteorological station.

Value

A basic hydroMet class object.

Slots

`id` numeric. This is the ID assigned by the agency.

`agency` character. The name of the agency (or institution) that provides the data of the station.

`station` character. The name of the (hydro)-meteorological station.

`lat` numeric. Latitude of the station.

`long` numeric. Longitude of the station

`alt` numeric. Altitude of the station.

`country` character. Country where the station is located. Argentina is set as default value.

`province` character. Name of the province where the station is located. Mendoza is set as default value.

`river` character. Basin river's name.

`active` logical. It indicates whether or not the station is currently operated. Default value is `TRUE`.

hydroMet_BDHI-class — hydroMet *subclass for BDHI (Base de Datos Hidrologica Integrada) data*

Description

An suitable object for store hydro-meteorological data from BDHI.

Value

A hydroMet_BDHI class object.

Slots

`Qmd` data.frame from read_BDHI containing daily mean river discharge [m3/s].

`Qmm` data.frame from read_BDHI containing monthly mean river discharge [m3/s].

`precip` data.frame from read_BDHI containing daily liquid precipitation [mm].

`tdb` data.frame from read_BDHI containing subdaily dry bulb temperature [ºC].

`tmax` data.frame from read_BDHI containing daily maximum air temperature [ºC].

`tmin` data.frame from read_BDHI containing daily minimum air temperature [ºC].

`swe` data.frame from read_BDHI containing daily snow water equivalent [mm].

`hr` data.frame from read_BDHI containing subdaily relative humidity [%].

`wspd` data.frame from read_BDHI containing subdaily wind speed [km/hr].

`wdir` data.frame from read_BDHI containing subdaily wind direction [º].

`evap` data.frame from read_BDHI containing daily pan-evaporation [mm].

`anem` data.frame from read_BDHI containing daily wind speed above the evap tank [km/hr].

`patm` data.frame from read_BDHI containing subdaily atmospheric pressure [mbar].

hydroMet_compact-class — hydroMet *subclass for compact data*

Description

This subclass is useful for storing in a single data frame ready to use hydro-meteorological series or many variables of the same kind (e.g. lets say precipitacion series).

Value

A hydroMet_compact class object.

Slots

`compact` data.frame with Date as first column (class 'Date' or 'POSIXct'). All other columns are the numeric hydro-meteorological variables (double). This subclass was though to join in a single table ready to use data (e.g. in modelling). You can also use it to put together variables of the same kind (e.g. precipitation records) to make some regional analysis.

`hydroMet_CR2-class` `hydroMet` *subclass for CR2 (Explorador Climático) data*

Description

A suitable object for store hydro-meteorological data from CR2.

Value

A hydroMet_CR2 class object.

Slots

`precip` data.frame from read_CR2 containing daily precipitation [mm].

`tmean` data.frame from read_CR2 containing daily mean air temperature [ºC].

`tmax` data.frame from read_CR2 containing daily maximum air temperature [ºC].

`tmin` data.frame from read_CR2 containing daily minimum air temperature [ºC].

`hydroMet_DGI-class` `hydroMet` *subclass for DGI (Departamento General de Irrigación) data*

Description

A suitable object for store hydro-meteorological data from DGI.

Value

A hydroMet_DGI class object.

Slots

`hsnow` data.frame from read_DGI containing daily snow height [m].

`swe` data.frame from read_DGI containing daily snow water equivalent [mm].

`tmean` data.frame from read_DGI containing daily mean air temperature [ºC].

`tmax` data.frame from read_DGI containing daily max. air temperature [ºC].

`tmin` data.frame from read_DGI containing daily min. air temperature [ºC].

`hr` data.frame from read_DGI containing daily mean relative humidity [%].

`patm` data.frame from read_DGI containing daily mean atmospheric pressure [hPa].

hydroMet_IANIGLA-class

hydroMet *subclass for IANIGLA (Instituto Argentino de Nivología, Glaciología y Ciencias Ambientales) data*

Description

A suitable object for store hydro-meteorological data provided by IANIGLA.

Value

A hydroMet_IANIGLA class object.

Slots

`date` time serie of dates (class `POSIXct` or `Date`).

`tair` numeric matrix with air temperature.

`hr` numeric matrix with relative humidity.

`patm` numeric matrix with atmospheric pressure.

`precip` numeric matrix with precipitacion.

`wspd` numeric matrix with wind speed.

`wdir` numeric matrix with wind direction.

`kin` numeric matrix with incoming short-wave radiation.

`hsnow` numeric matrix with snow height.

`tsoil` numeric matrix with soil temperature.

`hwat` numeric matrix with stream water level.

hydro_year_DGI *Hydrological year classification*

Description

This function allows you to get the hydrological year. The criteria is consistent with the one of Departamento General de Irrigacion (Mendoza - Argentina).

Usage

```
hydro_year_DGI(df)
```

Arguments

`df`	a data frame with total annual volumes discharges created with agg_serie function.

Value

A data frame containing the hydrological classification for each year.

Examples

```
# Create BDHI hydro-met station
guido <- create_hydroMet(class_name = 'BDHI')

# List with meteorological variables (slots in BDHI's object)
cargar <- list('precip', 'Qmd', 'Qmm')

# Now assign as names the files
hydro_files  <- list.files( system.file('extdata', package = "hydroToolkit"), pattern = 'Guido' )
names(cargar) <- hydro_files

# Build the object with the met records
guido <- build_hydroMet(obj = guido, slot_list = cargar,
               path = system.file('extdata', package = "hydroToolkit") )

# Now get mean monthly discharge
Qmm <- get_hydroMet(obj = guido, name = 'Qmm')[[1]]

# Get the monthly water volume
Qmm_vol <- Qmm_to_Dm(df = Qmm)

# Aggregate data frame to get total annual discharges
AD  <- agg_serie(df = Qmm_vol, fun = 'sum', period = 'annual', out_name = 'Ann_vol',
         start_month = 7, end_month = 6, allow_NA = 2)

# Get hydrological year classification
AD_class <- hydro_year_DGI(df = AD)
```

---

interpolate *Interpolation*

Description

This functions applies interpolation to fill in missing (or non-recorded) values.

Usage

```
interpolate(df, miss_table, threshold, method = "linear")
```

Arguments

df	data frame with two columns: 'Date' or 'POSIXct' class in the first column and a numeric variable in the second one.

miss_table	data frame with three columns: first and last date of interpolation (first and second column respectively). The last and third column, is a numeric with the number of steps to interpolate. See report_miss_data.
threshold	numeric variable with the maximum number of dates in which to apply the interpolation.
method	string with the interpolation method. In this version only 'linear' method is allowed.

Value

A data frame with date and the interpolated numeric variable.

Examples

```
# Create BDHI hydro-met station
guido <- create_hydroMet(class_name = 'BDHI')

# List with meteorological variables (slots in BDHI's object)
cargar <- list('precip', 'Qmd', 'Qmm')

# Now assign as names the files
hydro_files  <- list.files( system.file('extdata', package = "hydroToolkit"), pattern = 'Guido' )
names(cargar) <- hydro_files

# Build the object with the met records
guido <- build_hydroMet(obj = guido, slot_list = cargar,
               path = system.file('extdata', package = "hydroToolkit") )

# Get mean daily discharge and report miss data
Qmd  <- get_hydroMet(obj = guido, name = 'Qmd')[[1]]
miss <- report_miss_data(df = Qmd)

# Now interpolate miss values
Qmd_fill <- interpolate(df = Qmd, miss_table = miss, threshold = 5, method = "linear")
```

---

modify_hydroMet *Modify data inside a specific slot*

Description

Apply a pre-defined (e.g.: movAvg, fill_value or Qmm_to_Dm) or user defined function to an existing series inside a slot.

Usage

```
modify_hydroMet(
  obj,
  name = NA_character_,
  colName = NA_character_,
  colNum = 2,
  FUN = NULL,
  ...
)

## S4 method for signature 'hydroMet_BDHI'
modify_hydroMet(
  obj,
  name = NA_character_,
  colName = NA_character_,
  colNum = 2,
  FUN = NULL,
  ...
)

## S4 method for signature 'hydroMet_CR2'
modify_hydroMet(
  obj,
  name = NA_character_,
  colName = NA_character_,
  colNum = 2,
  FUN = NULL,
  ...
)

## S4 method for signature 'hydroMet_DGI'
modify_hydroMet(
  obj,
  name = NA_character_,
  colName = NA_character_,
  colNum = 2,
  FUN = NULL,
  ...
)

## S4 method for signature 'hydroMet_IANIGLA'
modify_hydroMet(
  obj,
  name = NA_character_,
  colName = NA_character_,
  colNum = 1,
  FUN = NULL,
  ...
```

)

Arguments

obj	hydroMet_XXX subclass object. See hydroMet_BDHI, hydroMet_DGI, hydroMet_IANIGLA or hydroMet_CR2.
name	string with the slot name of the data frame.
colName	string with the new column name (from FUN).
colNum	numeric value with the data frame column where to apply FUN. It must be > 1 (except in 'IANIGLA' subclass).
FUN	the function name.
...	FUN arguments to pass.

Value

The same hydroMet subclass provided in obj with an extra column.

Functions

- modify_hydroMet,hydroMet_BDHI-method: modify method for BDHI class
- modify_hydroMet,hydroMet_CR2-method: modify method for CR2 class
- modify_hydroMet,hydroMet_DGI-method: modify method for DGI class
- modify_hydroMet,hydroMet_IANIGLA-method: modify method for IANIGLA class

Examples

```
# Create BDHI hydro-met station
guido <- create_hydroMet(class_name = 'BDHI')

# List with meteorological variables (slots in BDHI's object)
cargar <- list('precip', 'Qmd', 'Qmm')

# Now assign as names the files
hydro_files  <- list.files( system.file('extdata', package = "hydroToolkit"), pattern = 'Guido' )
names(cargar) <- hydro_files

# Build the object with the met records
guido <- build_hydroMet(obj = guido, slot_list = cargar,
               path = system.file('extdata', package = "hydroToolkit") )
```

movAvg *Moving average windows*

Description

Smooth a numeric serie with a moving average windows

Usage

```
movAvg(df, k, pos)
```

Arguments

df	data frame with the serie that you want to smooth. By default, t he function uses column 2.
k	numeric value with windows size., e.g.: 5
pos	string with the position of the window: • 'izq': left aligned. The output value is on the left, so the function weights the (k - 1) values at the right side. • 'der': right aligned. The output value is on the right, so the function weights the (k - 1) values at the left side. • 'cen': center. The output value is in the middle of the window.

Value

data frame with the smooth serie.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Apply function
cuevas <- read_IANIGLA(file = 'Cuevas.csv', path = full_path)

# Get air temperature
cuevas_tair <- cuevas[ , 1:2]

# Create a moving average serie of Tair
Tair_mov <- movAvg(df = cuevas_tair, k = 10, pos = 'izq')
```

plot_hydroMet *Methods to easily use* ggplot2 *or* plotly *(interactive)*

Description

This method allows you to make plots (using simple and expressive arguments) of the variables contained inside an hydroMet_XXX object. The plot outputs can be static (ggplot2) or interactive (plotly).

Usage

```
plot_hydroMet(
  obj,
  slot_name,
  col_number,
  interactive = FALSE,
  line_type = NULL,
  line_color = "dodgerblue",
  x_lab = "Date",
  y_lab = "y",
  title_lab = NULL,
  legend_lab = NULL,
  double_yaxis = NULL,
  list_extra = NULL,
  from = NULL,
  to = NULL,
  scatter = NULL
)

## S4 method for signature 'hydroMet_BDHI'
plot_hydroMet(
  obj,
  slot_name,
  col_number,
  interactive = FALSE,
  line_type = NULL,
  line_color = "dodgerblue",
  x_lab = "Date",
  y_lab = "y",
  title_lab = NULL,
  legend_lab = NULL,
  double_yaxis = NULL,
  list_extra = NULL,
  from = NULL,
  to = NULL
)
```

```
## S4 method for signature 'hydroMet_CR2'
plot_hydroMet(
  obj,
  slot_name,
  col_number,
  interactive = FALSE,
  line_type = NULL,
  line_color = "dodgerblue",
  x_lab = "Date",
  y_lab = "y",
  title_lab = NULL,
  legend_lab = NULL,
  double_yaxis = NULL,
  list_extra = NULL,
  from = NULL,
  to = NULL
)

## S4 method for signature 'hydroMet_DGI'
plot_hydroMet(
  obj,
  slot_name,
  col_number,
  interactive = FALSE,
  line_type = NULL,
  line_color = "dodgerblue",
  x_lab = "Date",
  y_lab = "y",
  title_lab = NULL,
  legend_lab = NULL,
  double_yaxis = NULL,
  list_extra = NULL,
  from = NULL,
  to = NULL
)

## S4 method for signature 'hydroMet_IANIGLA'
plot_hydroMet(
  obj,
  slot_name,
  col_number,
  interactive = FALSE,
  line_type = NULL,
  line_color = "dodgerblue",
  x_lab = "Date",
  y_lab = "y",
  title_lab = NULL,
  legend_lab = NULL,
```

```
  double_yaxis = NULL,
  list_extra = NULL,
  from = NULL,
  to = NULL
)

## S4 method for signature 'hydroMet_compact'
plot_hydroMet(
  obj,
  slot_name,
  col_number,
  interactive = FALSE,
  line_type = NULL,
  line_color = "dodgerblue",
  x_lab = "x",
  y_lab = "y",
  title_lab = NULL,
  legend_lab = NULL,
  double_yaxis = NULL,
  list_extra = NULL,
  from = NULL,
  to = NULL,
  scatter = NULL
)
```

Arguments

obj	a valid hydroMet_XXX object.
slot_name	string(s) with the name of the slot(s) to use in plotting.
col_number	numeric (vector) with the column's variable to plot. In case you decide to merge slots you must provide a list in which each element contains the column numbers of the variable to plot.
interactive	logical. Default value, FALSE, will return a ggplot2 class object. Otherwise you will get a plotly one.
line_type	string with line dash type (ggplot2) or mode in plotly case. ggplot2: 'solid' (default value), 'twodash', 'longdash', 'dotted', 'dotdash', 'dashed' or 'blank'. plotly: 'lines' (default value), 'lines+markers' or 'markers'.
line_color	string with a valid color. See 'colors()' or Rcolor document.
x_lab	string with x axis label.
y_lab	string with y axis label. In case you use double_yaxis argument you must supply both c('ylab', 'y2lab').
title_lab	string with the title of the plot. Default is a plot without title.
legend_lab	string with plot label(s) name(s). **NOTE**: ggplot2 double_yaxis does not support legend_lab in this package version, so giving values to this argument will be harmfulness.

double_yaxis	numeric vector with either 1 (= main axis - left) or 2 (= secondary axis - right) indicating whether the variable should be plotted in either left or right axis. **NOTE**: in this package version ggplot2 supports just one line plot for each 'y' axis.
list_extra	list with the ggplot2 argument to pass. This argument was design to allow the user to modify ggplot2 arguments (you can find nice examples in ggplot2 - Essentials) **NOTE**: in this package version this argument doesn't make sense for plotly (except for scatter plot in hydroMet_compact class).
from	string (or POSIXct - valid only in 'BDHI' and 'IANIGLA') with the starting Date. You can use 'from' without 'to'. In this case you will subset your data 'from' till the end.
to	string (or POSIXct - valid only in 'BDHI' and 'IANIGLA') with the ending Date. You can use 'to' without 'from'. In this case you will subset your data from the beginning till 'to'.
scatter	numeric vector of length two with the column number to plot as scatter. The first variable (column number) will be the 'x' variable and the second one the 'y' variable. This argument will work just for class hydroMet_compact.

Value

A ggplot2 or plotly objects to analyze your data.

Functions

- plot_hydroMet,hydroMet_BDHI-method: plot method for BDHI class
- plot_hydroMet,hydroMet_CR2-method: plot method for CR2 class
- plot_hydroMet,hydroMet_DGI-method: plot method for DGI class
- plot_hydroMet,hydroMet_IANIGLA-method: plot method for IANIGLA class
- plot_hydroMet,hydroMet_compact-method: plot method for compact class

Examples

```
# Path to file
dgi_path  <- system.file('extdata', package = "hydroToolkit")
file_name <- list.files(path = dgi_path, pattern = 'Toscas')

# Read Toscas
var_nom <- list(slotNames(x = 'hydroMet_DGI')[2:7])
names(var_nom) <- file_name

# Load Toscas meteo station data
toscas_dgi <- create_hydroMet(class_name = 'DGI')
toscas_dgi <- build_hydroMet(obj = toscas_dgi, slot_list = var_nom, path = dgi_path)

# Plot mean air temperature
plot_hydroMet(obj = toscas_dgi, col_number = 2, slot_name = 'tmean',
 legend_lab = 'Tmean(ºC)' )
```

```
# Now let's plot an interactive graph
plot_hydroMet(obj = toscas_dgi, col_number = 2, slot_name = 'tmean',
 interactive = TRUE, y_lab = 'Tmean(ºC)' )
```

---

precip_cumsum *Cumulative sum of precipitation series*

Description

Returns a data frame with two columns: the date and the cumulative sum of the chosen col_number. This function can deal with NA_real_.

Usage

```
precip_cumsum(df, col_number = 2, out_name = NULL)
```

Arguments

df	data frame with Date (or POSIXct) in the first column and numeric variables on the others.
col_number	numeric. The column number of the series where to apply the cumulative sum.
out_name	optional. String value with the column output name. Default is 'cumsum_' plus the original name.

Value

A data frame with two columns: date and the cumulative sum of the series.

Examples

```
# Load daily precipitation data-set from BDHI
load( paste0(system.file('extdata', package = "hydroToolkit"), '/bdhi_p.rda') )

# Get compact slot
p_bdhi <- get_hydroMet(obj = bdhi_p, name = 'compact')[[1]]

# Apply cumulative precipitation function
p_cum <- precip_cumsum(df = p_bdhi, col_number = 2, out_name = 'cum_guido')
```

precip_hydroMet *Make homogeneity test or fill gaps in a series*

Description

This method can do both: test homogeneity in precipitation series or fill data gaps using regional analysis.

Usage

```
precip_hydroMet(
  obj,
  col_target = 2,
  fill = FALSE,
  method = "spearman",
  min_value = 0.2
)

## S4 method for signature 'hydroMet_compact'
precip_hydroMet(
  obj,
  col_target = 2,
  fill = FALSE,
  method = "spearman",
  min_value = 0.2
)
```

Arguments

obj	an hydroMet_compact class object.
col_target	numeric. The column number of the target series (either to test homogeneity or to fill gaps) in compact slot.
fill	logical. By default value (FALSE) you will make an homogeneity test to your target series.
method	string (default is spearman - possible values are: spearman, pearson or kendall). When creating the regional (or master series) the method uses a weighted mean. The weighted values are the correlations coefficients.
min_value	numeric. Series with a correlation value less than min_value are thrown away.

Value

If fill = FALSE the method will return a list with three elements: a data frame with all necessary values to correct your target serie, a plot with p-values and the correlation matrix. When fill = TRUE the list will contain: the data frame with the target series gaps filled and the correlation matrix.

Functions

- precip_hydroMet,hydroMet_compact-method: homogeneity test applied to precipitation data stored in compact class.

Examples

```
# Load daily precipitation data-set from BDHI
load( paste0(system.file('extdata', package = "hydroToolkit"), '/bdhi_p.rda') )

# Fill gaps in Tupungato station
relleno <- precip_hydroMet(obj = bdhi_p, col_target = 5, fill = TRUE)
```

Qmm_to_Dm *River discharge [m3/s] to volume [hm3]*

Description

Converts mean monthly river discharge [m3/s] to total volume discharge [hm3].

Usage

```
Qmm_to_Dm(df)
```

Arguments

df	data frame with class Date in the first column. By default the function converts the second column only. If you have daily or hourly data see agg_serie.

Value

A data frame with two columns: Date and total volume discharge.

Examples

```
# Create BDHI hydro-met station
guido <- create_hydroMet(class_name = 'BDHI')

# List with meteorological variables (slots in BDHI's object)
cargar <- list('precip', 'Qmd', 'Qmm')

# Now assign as names the files
hydro_files  <- list.files( system.file('extdata', package = "hydroToolkit"), pattern = 'Guido' )
names(cargar) <- hydro_files

# Build the object with the met records
guido <- build_hydroMet(obj = guido, slot_list = cargar,
               path = system.file('extdata', package = "hydroToolkit") )
```

```
# Now get mean monthly discharge
Qmm <- get_hydroMet(obj = guido, name = 'Qmm')[[1]]

# Get the monthly water volume
Qmm_vol <- Qmm_to_Dm(df = Qmm)
```

`read_BDHI` *Reads data from Base de Datos Hidrológica Integrada (BDHI) - Argentina*

Description

Reads files downloaded from the Base de Datos Hidrológica Integrada (BDHI) as a data frame.

Usage

```
read_BDHI(file, colName, timeStep, is.Wdir = FALSE)
```

Arguments

`file`	string with the name (including extension) of the file.
`colName`	string with variable name. E.g.: Qmd(m3/s)
`timeStep`	string with time step: 'month', 'day', 'day/3', '4h' or 'hour'. • 'day': data recorded once a day • 'month': data recorded monthly • '4h': applies to atmospheric pressure time series only • 'day/3': applies to wind related variables, relative humidity, and dry bulb temperature' • 'hour': in case you have to deal with hourly data.
`is.Wdir`	a logical value indicating if the variable is wind direction. Default value is set to FALSE.

Value

A data frame with two columns: date and variable. Gaps between dates are filled with `NA_real_` and duplicated rows are eliminated automatically.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Apply function
guido_Qmd <- read_BDHI(file = paste0(full_path, '/Qmd_Mendoza_Guido'),
                colName = 'Q(m3/s)', timeStep = 'day')
```

read_CR2 *Reads data from Explorador Climático de Chile*

Description

Reads data downloaded from Explorador Climatico de Chile (CR2) as a data frame.

Usage

```
read_CR2(file, colName, path = NULL)
```

Arguments

file	string with the file name (include extension). The only accepted format is '.csv'.
colName	string with the name of the variable.
path	string with the files directory. If not provided, the function will use the current working directory.

Value

A two column data frame with date and variable. Gaps between dates are filled with NA_real_ and duplicated rows are eliminated automatically.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Apply function
yeso_tmed <- read_CR2(file = 'Tmed_Yeso_Embalse.csv',
                colName = 'T(ºC)', path = full_path)
```

read_DGI *Reads data from Departamento General de Irrigación (Mendoza - Argentina)*

Description

Reads the Departamento General de Irrigacion(Mendoza - Argentina) excel sheet.

Usage

```
read_DGI(file, sheet = NULL, colName = NULL, range = NULL, path = NULL)
```

Arguments

file	string with the file name ('xlsx' excel files).
sheet	sheet to read. Either a string (the name of a sheet), or an integer (the position of the sheet). Default value is sheet one.
colName	string with the name of the second column (as default first column is Date). If ignored first row excel names are used.
range	string providing cell range to read. E.g.: 'A1:B75'.
path	string with the files directory. If not provided, the function will use the current working directory.

Value

A data frame with two columns: date and variable. Gaps between dates are filled with NA_real_ and duplicated rows are eliminated automatically.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Apply function
toscas_hr <- read_DGI(file = 'Toscas.xlsx', sheet = 'hr',
                colName = 'RH(%)', path = full_path)
```

---

read_IANIGLA *Reads data provided by IANIGLA*

Description

Reads the data provided by IANIGLA (Instituto Argentino de Nivologia, Glaciologia y Ciencias Ambientales).

Usage

```
read_IANIGLA(file, all = FALSE, path = NULL)
```

Arguments

file	string with the name of the '.csv' file downloaded from the meteo-stations web page.
all	logical value indicating whether the returned data frame contain all the original columns or just the date and data.
path	string with the files directory. If not provided, the function will use the current working directory.

Value

A data frame containing the hourly data measured by the automatic weather stations. Gaps between dates are filled with NA_real_ and duplicated rows are eliminated automatically.

Note

In this package version we only provide functionality for a specific data-set generated in the institute.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Apply function
cuevas <- read_IANIGLA(file = 'Cuevas.csv', path = full_path)
```

`report_hydroMet` *Object summaries*

Description

This method returns a list with two elements: the first one is a `data frame` with miss data (see also report_miss_data) and the second one is also a `data frame` with the `mean`, `sd`, `max` and `min` values.

Usage

```
report_hydroMet(
  obj,
  slot_name,
  col_name,
  start_date = NULL,
  end_date = NULL,
  Lang = "spanish"
)

## S4 method for signature 'hydroMet_BDHI'
report_hydroMet(
  obj,
  slot_name,
  col_name,
  start_date = NULL,
  end_date = NULL,
  Lang = "spanish"
)
```

```
## S4 method for signature 'hydroMet_CR2'
report_hydroMet(
  obj,
  slot_name,
  col_name,
  start_date = NULL,
  end_date = NULL,
  Lang = "spanish"
)

## S4 method for signature 'hydroMet_DGI'
report_hydroMet(
  obj,
  slot_name,
  col_name,
  start_date = NULL,
  end_date = NULL,
  Lang = "spanish"
)

## S4 method for signature 'hydroMet_IANIGLA'
report_hydroMet(
  obj,
  slot_name,
  col_name,
  start_date = NULL,
  end_date = NULL,
  Lang = "spanish"
)
```

Arguments

obj	an hydroMet_XXX object.
slot_name	a single or vector string containing the slot(s) to report.
col_name	a single or vector string with the name of the column to report in slot_name.
start_date	optional (default is the first Date). Single string or POSIXct with the starting Date to report.
end_date	optional (default is the last Date). Single string or POSIXct with the last Date to report.
Lang	optional (default value is spanish). Single string with the language to report results: spanish or english.

Value

A list containing two data frames: the first one with miss data and the second with the mean, sd, max and min values of the series.

Functions

- report_hydroMet,hydroMet_BDHI-method: report method for BDHI class
- report_hydroMet,hydroMet_CR2-method: report method for CR2 class
- report_hydroMet,hydroMet_DGI-method: report method for DGI class
- report_hydroMet,hydroMet_IANIGLA-method: report method for IANIGLA class

Examples

```
# Create IANIGLA class
cuevas <- create_hydroMet(class_name = 'IANIGLA')

# List with meteorological variables (slots in BDHI's object)
cargar <- list( slotNames(x = 'hydroMet_IANIGLA')[2:11] )

# Assign as names the files
hydro_files  <- list.files( system.file('extdata', package = "hydroToolkit"), pattern = 'Cuevas' )
names(cargar) <- hydro_files

# Build met-station
cuevas <- build_hydroMet(obj = cuevas, slot_list = cargar,
           path = system.file('extdata', package = "hydroToolkit") )

# Get report
report_hydroMet(obj = cuevas, slot_name = 'kin', col_name = 'kin_1')
```

---

report_miss_data *Report* NA_real_ *values*

Description

Creates a data frame with reported dates and number of times-step of missing or not recorded data.

Usage

```
report_miss_data(df, Lang = "spanish")
```

Arguments

df	data frame with hydro-meteo data. First column is date and the second the numeric vector to be reported.
Lang	string with output column name language: 'spanish' (default) or 'english'.

Value

A data frame with three columns: start-date, end-date and number of missing time steps.

Examples

```
# Create BDHI hydro-met station
guido <- create_hydroMet(class_name = 'BDHI')

# List with meteorological variables (slots in BDHI's object)
cargar <- list('precip', 'Qmd', 'Qmm')

# Now assign as names the files
hydro_files  <- list.files( system.file('extdata', package = "hydroToolkit"), pattern = 'Guido' )
names(cargar) <- hydro_files

# Build the object with the met records
guido <- build_hydroMet(obj = guido, slot_list = cargar,
               path = system.file('extdata', package = "hydroToolkit") )

# Get mean daily discharge and report miss data
Qmd  <- get_hydroMet(obj = guido, name = 'Qmd')[[1]]
miss <- report_miss_data(df = Qmd)
```

`rm_spikes` *Remove spikes*

Description

Removes spikes, and sets their value to `NA_real_`.

Usage

```
rm_spikes(df, tolerance)
```

Arguments

`df`	data frame with date and numeric variable in the first and second column respectively (from `read_XXX` functions).
`tolerance`	numeric with maximum tolerance between a number and its successor.

Value

The same data frame but without peaks.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Read IANIGLA file
cuevas <- read_IANIGLA(file = 'Cuevas.csv', path = full_path)
```

```
# Remove spikes from air temperature series
tair_rm_spikes <- rm_spikes(df = cuevas, tolerance = 10)
```

`set_hydroMet` *Set the data of an* `hydroMet` *object or its subclasses*

Description

With this method you can set (or change) an specific slot value.

Usage

```
set_hydroMet(
  obj = NULL,
  id = NULL,
  agency = NULL,
  station = NULL,
  lat = NULL,
  long = NULL,
  alt = NULL,
  country = NULL,
  province = NULL,
  river = NULL,
  active = NULL,
  ...
)

## S4 method for signature 'hydroMet'
set_hydroMet(
  obj = NULL,
  id = NULL,
  agency = NULL,
  station = NULL,
  lat = NULL,
  long = NULL,
  alt = NULL,
  country = NULL,
  province = NULL,
  river = NULL,
  active = NULL
)

## S4 method for signature 'hydroMet_BDHI'
set_hydroMet(
  obj = NULL,
  id = NULL,
```

```
  agency = NULL,
  station = NULL,
  lat = NULL,
  long = NULL,
  alt = NULL,
  country = NULL,
  province = NULL,
  river = NULL,
  active = NULL,
  Qmd = NULL,
  Qmm = NULL,
  precip = NULL,
  tdb = NULL,
  tmax = NULL,
  tmin = NULL,
  swe = NULL,
  hr = NULL,
  wspd = NULL,
  wdir = NULL,
  evap = NULL,
  anem = NULL,
  patm = NULL
)

## S4 method for signature 'hydroMet_DGI'
set_hydroMet(
  obj = NULL,
  id = NULL,
  agency = NULL,
  station = NULL,
  lat = NULL,
  long = NULL,
  alt = NULL,
  country = NULL,
  province = NULL,
  river = NULL,
  active = NULL,
  swe = NULL,
  tmean = NULL,
  tmax = NULL,
  tmin = NULL,
  hr = NULL,
  patm = NULL,
  hsnow = NULL
)

## S4 method for signature 'hydroMet_IANIGLA'
set_hydroMet(
```

```
  obj = NULL,
  id = NULL,
  agency = NULL,
  station = NULL,
  lat = NULL,
  long = NULL,
  alt = NULL,
  country = NULL,
  province = NULL,
  river = NULL,
  active = NULL,
  date = NULL,
  tair = NULL,
  hr = NULL,
  patm = NULL,
  precip = NULL,
  wspd = NULL,
  wdir = NULL,
  kin = NULL,
  hsnow = NULL,
  tsoil = NULL,
  hwat = NULL
)

## S4 method for signature 'hydroMet_CR2'
set_hydroMet(
  obj = NULL,
  id = NULL,
  agency = NULL,
  station = NULL,
  lat = NULL,
  long = NULL,
  alt = NULL,
  country = NULL,
  province = NULL,
  river = NULL,
  active = NULL,
  precip = NULL,
  tmean = NULL,
  tmax = NULL,
  tmin = NULL
)

## S4 method for signature 'hydroMet_compact'
set_hydroMet(
  obj = NULL,
  id = NULL,
  agency = NULL,
```

```
  station = NULL,
  lat = NULL,
  long = NULL,
  alt = NULL,
  country = NULL,
  province = NULL,
  river = NULL,
  active = NULL,
  compact = NULL
)
```

Arguments

obj	an hydroMet or hydroMet_XXX class object.
id	numeric. This is the ID assigned by the agency.
agency	character. The name of the agency (or institution) that provides the data of the station.
station	character. The name of the (hydro)-meteorological station.
lat	numeric. Latitude of the station.
long	numeric. Longitude of the station
alt	numeric. Altitute of the station.
country	character. Country where the station is located. Argentina is set as default value.
province	character. Name of the province where the station is located. Mendoza is set as default value.
river	character. Basin river's name.
active	logical. It indicates whether or not the station is currently operated. Default value is TRUE.
...	arguments to be passed to methods. They rely on the slots of the obj subclass.
Qmd	daily mean river discharge.
Qmm	monthly mean river discharge.
precip	precipitation.
tdb	dry bulb temperature.
tmax	daily maximum air temperature.
tmin	daily minimum air temperature.
swe	snow water equivalent.
hr	relative humidity.
wspd	wind speed.
wdir	wind direction.
evap	evaporation.
anem	wind speed above the pan-evaporation.
patm	atmospheric pressure.

tmean	daily mean air temperature.
hsnow	snow height.
date	time serie with dates.
tair	air temperature.
kin	incoming shortwave radiation.
tsoil	soil temperature.
hwat	stream water level.
compact	data frame with Date as first column. All other columns are hydro-meteorological variables.

Value

The hydroMet object with the slots setted.

Functions

- set_hydroMet,hydroMet-method: set method for generic object
- set_hydroMet,hydroMet_BDHI-method: set method for BDHI object
- set_hydroMet,hydroMet_DGI-method: set method for DGI object
- set_hydroMet,hydroMet_IANIGLA-method: set method for IANIGLA object
- set_hydroMet,hydroMet_CR2-method: set method for CR2 object
- set_hydroMet,hydroMet_compact-method: set method for compact object

Examples

```
# Create BDHI hydro-met station
guido <- create_hydroMet(class_name = 'BDHI')

# Assign altitude
guido <- set_hydroMet(obj = guido, alt = 2480)
```

---

set_threshold *Set a threshold*

Description

Set tolerable extreme values (maximum or minimum). Records greater or equal than ('>=') or lesser or equal than ('<=') 'threshold' argument are set to NA_real_.

Usage

```
set_threshold(x, threshold, case = ">=")
```

Arguments

x	numeric vector or data frame with a numeric series in the second column.
threshold	numeric value with threshold.
case	string with either '>=' (greater or equal than) or '<=' (lesser or equal than) symbol.

Value

Numeric vector or data frame with values greater (or lesser) or equal than 'threshold' set as NA_real_.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Read IANIGLA file
cuevas <- read_IANIGLA(file = 'Cuevas.csv', path = full_path)

# Set threshold from air temperature series
tair_thres <- set_threshold(x = cuevas, threshold = 40)
```

---

subset_hydroMet *Subset your data*

Description

This method allows you to easily cut the data stored in an hydroMet_XXX class object by dates.

Usage

```
subset_hydroMet(obj, slot_name, from = NULL, to = NULL)

## S4 method for signature 'hydroMet_BDHI'
subset_hydroMet(obj, slot_name, from = NULL, to = NULL)

## S4 method for signature 'hydroMet_DGI'
subset_hydroMet(obj, slot_name, from = NULL, to = NULL)

## S4 method for signature 'hydroMet_CR2'
subset_hydroMet(obj, slot_name, from = NULL, to = NULL)

## S4 method for signature 'hydroMet_IANIGLA'
subset_hydroMet(obj, slot_name, from = NULL, to = NULL)

## S4 method for signature 'hydroMet_compact'
subset_hydroMet(obj, slot_name, from = NULL, to = NULL)
```

Arguments

obj	an hydroMet_XXX class object.
slot_name	string vector with the slot(s) name(s) to subset. **NOTE**: in case you want to subset a hydroMet_IANIGLA object is recommended to consider all the slots with data.
from	string (or POSIXct - valid only in 'BDHI' and 'IANIGLA') with the starting Date. You can use from without to. In this case you will subset your data 'from' till the end.
to	string (or POSIXct - valid only in 'BDHI' and 'IANIGLA') with the ending Date. You can use to without from. In this case you will subset your data from the beginning till 'to'.

Value

The same hydroMet_XXX class provided in obj but subsetted.

Functions

- subset_hydroMet,hydroMet_BDHI-method: subset method for BDHI data
- subset_hydroMet,hydroMet_DGI-method: subset method for DGI data
- subset_hydroMet,hydroMet_CR2-method: subset method for CR2 data
- subset_hydroMet,hydroMet_IANIGLA-method: subset method for IANIGLA data
- subset_hydroMet,hydroMet_compact-method: subset method for compact data

Examples

```
# Create BDHI hydro-met station
guido <- create_hydroMet(class_name = 'BDHI')

# List with meteorological variables (slots in BDHI's object)
cargar <- list('precip', 'Qmd', 'Qmm')

# Assign as names the files
hydro_files  <- list.files( system.file('extdata', package = "hydroToolkit"), pattern = 'Guido' )
names(cargar) <- hydro_files

# Build the object with the met records
guido <- build_hydroMet(obj = guido, slot_list = cargar,
               path = system.file('extdata', package = "hydroToolkit") )

# Subset daily mean discharge
guido <- subset_hydroMet(obj = guido, slot_name = 'Qmd', from = '2005-01-01',
            to = '2010-12-31')
```

swe_to_melt *Snow water equivalent to melt*

Description

Converts a snow water equivalent series (from snow pillow) into a melt series.

Usage

```
swe_to_melt(df)
```

Arguments

df	data frame with 'swe' serie in the second column. See 'read_XXX' functions.

Value

Data frame containing the numeric vector with melted snow.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Read swe sheet
toscas_swe <- read_DGI(file = 'Toscas.xlsx', sheet = 'swe',
                colName = 'swe(mm)', path = full_path)

# swe to melt
toscas_melt <- swe_to_melt(df = toscas_swe)
```

swe_to_precip *Snow water equivalent to snowfall*

Description

Converts a snow water equivalent series (from snow pillow) to a snowfall series.

Usage

```
swe_to_precip(df)
```

Arguments

df	data frame with 'swe' series in the second column. See 'read_XXX' functions.

Value

Data frame containing the numeric vector with inferred snowfall.

Examples

```
# Relative path to raw data
full_path <- system.file('extdata', package = "hydroToolkit")

# Read swe sheet
toscas_swe <- read_DGI(file = 'Toscas.xlsx', sheet = 'swe',
                colName = 'swe(mm)', path = full_path)

# swe to snowfall
toscas_snfall <- swe_to_precip(df = toscas_swe)
```

Index