Package 'CSLSfluxes'

April 5, 2021

Type Package

Title CSLS Chemistry Flux Analysis

Version 1.0.0

Author Carolyn Voter

Maintainer Carolyn Voter <cvoter@wisc.edu>

Description Analyze CSLS MODFLOW scenarios for lake water and solute budgets.

License MIT + file LICENSE

Imports CSLSdata,
CSLSevap,
dplyr,
lubridate,
magrittr,
NISTunits,
reshape2,
rlang,
stats,
zoo

Suggests CSLSscenarios,
DT,
extrafont,
ggplot2,
knitr,
rmarkdown,
stringr,
patchwork,
purrr,
raster

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

Depends R (>= 2.10)

R topics documented:

`add_pcpn_isotopes` *Add precipitation isotopes to dataset*

Description

Adds precipitation stable isotope measurements from Maribeth Kniffin (May 2016 - April 2017) and incorporates them in the current timeseries, but only for months without CSLS precipitation measurements.

Usage

```
add_pcpn_isotopes(tracer, start_date, end_date)
```

Arguments

`tracer`	data frame with water chemistry information for stable isotope.
`start_date`	start date of analysis period, POSIXct
`end_date`	end date of analysis period, POSIXct

Value

tracer, same data frame as provided, but with additional precipitation measurements during the analysis period for months with missing precipitation measurements.

`calculate_Cevap` *Evaporation stable isotope*

Description

Calculates the stable isotope composition of evaporation based on equation 5 in Krabbenhoft et al. (1990). Note that alpha* is equivalent to alpha^-1.

Usage

```
calculate_Cevap(atmp, ltmp, RH, Cpcpn, Clake, parameter)
```

Arguments

`atmp`	air temperature (C)
`ltmp`	lake surface temperature (C)
`RH`	relative humidity (percent)
`Cpcpn`	stable isotope composition of precipitation
`Clake`	stable isotope composition of the lake
`parameter`	name of stable isotope, "d18O" or "d2H"

Value

Cevap - the isotope composition of evaporation

References

Krabbenhoft, D. P., C. J. Bowser, M. P. Anderson, and J. W. Valley. (1990). Estimating Groundwater Exchange with Lakes: 1. The Stable Isotope Mass Balance Method. Water Resources Research, 26(10):2445-2453. https://doi.org/10.1029/WR026i010p02445

`calculate_C_GWin` *Calculate lake concentration of parameter*

Description

Calculates the concentration of a parameter in the lake given an initial lake concentration, monthly fluxes of precipitation, groundwater, and lake volume, and median concentrations of groundwater and precipitation.

Usage

```
calculate_C_GWin(
  C_vals,
  fluxes = CSLSdata::MODFLOW,
  scenario = "cal",
  sim = 0,
  start_date = as_datetime("2017-10-01"),
  end_date = as_datetime("2018-09-30"),
  lakes = c("Pleasant", "Long", "Plainfield")
)
```

Arguments

C_vals	data frame with columns for "lake", "C_pcpn", "C1_lake", "C2_lake", and "C_GWout"
fluxes	data frame with lake, date, level_m, GWin_m3, and P_m3 from MODFLOW simulations to use for mass balance calculations.
scenario	which MODFLOW scenario is being evaluated here, defaults to "cal".
sim	which MODFLOW simulation is being evaluated here, defaults to 0
start_date	start date for evaluation, defaults to 2017-10-01
end_date	end date for evaluation, defaults to 2018-09-30.
lakes	name of lakes to include in analysis, defaults to c("Pleasant", "Long", "Plainfield")

Value

C_GWin, a data frame with the following columns:

lake	name of lakes, i.e., "Pleasant", "Long", and "Plainfield"
C_GWin	calculated concentration in groundwater inflow

---

calculate_C_lake *Calculate lake concentration of parameter*

Description

Calculates the concentration of a parameter in the lake given an initial lake concentration, monthly fluxes of precipitation, groundwater, and lake volume, and median concentrations of groundwater and precipitation.

Usage

```
calculate_C_lake(
  C_vals,
  fluxes = CSLSdata::MODFLOW,
  scenario = "no_irr",
  sim = 1,
  start_date = as_datetime("1985-10-01"),
  end_date = as_datetime("2018-09-30"),
  lakes = c("Pleasant", "Long", "Plainfield"),
  dt = "year"
)
```

Arguments

C_vals	data frame with columns for "lake", "C0_lake", "C_pcpn" and "C_GWin".
fluxes	data frame with lake, date, level_m, GWin_m3, and P_m3 from MODFLOW simulations to use for mass balance calculations.
scenario	which MODFLOW scenario is being evaluated here (e.g., "cal", "no_irr", "cur_irr")
sim	which MODFLOW simulation is being evaluated here (a number)
start_date	start date for evaluation, defaults to 1985-10-01

end_date	end date for evaluation, defaults to 2018-09-30.
lakes	a list of lakes being evaluated. Defaults to c("Pleasant", "Long", "Plainfield")
dt	time step of calculations. Defaults to "year", can also be "month", or "day".

Value

output, a data frame with the following columns:

scenario	name of MODFLOW scenario evaluated (e.g., "cal", "no_irr")
sim	number of MODFLOW simulation evaluated
lake	name of lake, i.e., "Pleasant", "Long", and "Plainfield"
date	date of observation (POSIX) used for weather inputs in MODFLOW
level_m	lake level (m) from MODFLOW
vol_m3	lake volume (m3) from MODFLOW
P_m3	precipitation (m3) from MODFLOW
C_pcpn	median precipitation concentration (mg/L) from CSLSdata observations
M_P	calculated mass of parameter in precipitation (g)
GWin_m3	groundwater inflow (m3) from MODFLOW
C_GWin	median groundwater inflow concentration (mg/L) from CSLSdata observations
M_GWin	calculated mass of parameter in groundwater inflow (g)
GWout_m3	groundwater outflow (m3) from MODFLOW
M_GWin	calculated mass of parameter in groundwater outflow (g)
C0_lake	initial lake concentration (mg/L) provided to function
M_lake	calculated mass of parameter in lake (g)
C_lake	calculated concentration of parameter in lake (mg/L)

---

calculate_ice_thickness

Calculate ice thickness

Description

Calculates the thickness of ice on a lake. Assumes a linear increase from a static ice-on date to a static ice-off date with a static date of max thickness and a static max thickness.

Usage

```
calculate_ice_thickness(
  date,
  ice_on = 320,
  ice_off = 105,
  ice_max = 60,
  ice_max_depth = NISTunits::NISTinchTOmeter(24) * 1000
)
```

Arguments

date	date to calculate ice thickness (POSIX)
ice_on	day of year when ice starts forming. Defaults to day 320 (Nov 16).
ice_off	day of year when ice is all gone. Defaults to day 105 (Apr 15)
ice_max	day of year when ice thickness is at a maxiumum. Defaults to day 60 (Mar 1).
ice_max_depth	max thickness of ice (m). Defaults to 24 inches.

Value

ice_depth, a numeric value of ice thickness (m).

---

calculate_water_balance_tracer

Calculate water balance

Description

Calculates the water balance using desired tracer, e.g., stable isotopes or conservative anion/cation.

Usage

```
calculate_water_balance_tracer(
  param = "d18O",
  start_date = as_datetime("2018-10-01"),
  end_date = as_datetime("2019-09-30"),
  dt = "annual",
  no_ice = FALSE,
  C_evap = 0,
  C_ice = 0,
  mean_lake = TRUE
)
```

Arguments

param	name of water chemistry parameter to grab concentration information for. Must match the "description" field of CSLSdata::water_chem for the desired parameter. Defaults to "d18O". Options include "d18O", "d2H", "CALCIUM TOTAL RECOVERABLE", "MAGNESIUM TOTAL RECOVERABLE", "CHLORIDE", and "SODIUM TOTAL RECOVERABLE".
start_date	start date of analysis. Defaults to start of WY2019 ("2018-10-01").
end_date	end date of analysis. Defaults to end of WY2019 ("2019-09-30").
dt	desired time step of inputs (e.g., "day" or "month")
no_ice	logical defaults to TRUE to ignore ice formation.
C_evap	concentraion in evaporation, defaults to zero
C_ice	concentration in ice, defaults to zero.
mean_lake	logical defaults to TRUE to calculate and use mean of min and max lake value (should equate to fall and spring turnover samples) for water budget calculations.

Value

lake_inputs

`Cevap_Catm` *Atmosphere d18O*

Description

Calculates the d18O isotope composition of the atmosphere based on Equation 18 and the definition for espilon+ in the explanation for Equation 3 in Gibson et al. (2016). Alternatively, can instead calcuate this value based on Equation 1.10 and the definition for epsilon in the explanation for Equation 1.4 in Mook (2000).

Usage

```
Cevap_Catm(Cpcpn, alpha, method = "Mook_corrected", k = 1)
```

Arguments

`Cpcpn`	isotopic composition of precipitation
`alpha`	equilibrium isotope fractionation factor (-) at the temperature of the air-water interface in LV form (i.e., alpha > 1).
`method`	defaults to "Gibson" to use those equations, can also be "Mook".
`k`	weighted factor which reflects seasonality, ranging from 0.5 for highly seasonal climates to 1 for non-seasonal climates. Defaults to 0.8

Value

Catm - the d18O isotope composition of the atmosphere

References

Gibson, J.J., S.J. Birks, and Y. Yi. 2016. Stable isotope mass balance of lakes: a contemporary perspective. Quaternary Science Reviews, 131:316-328. https://doi.org/10.1016/j.quascirev.2015.04.013

Mook, W.G. (ed.) 2000. Environmental Isotopes in the Hydrologic Cycle: Volume III: Surface Water. UNESCO. Paris, France.

Cevap_isotope_frac *Equilibrium Isotope Fractionation Factor*

Description

Calculates the equilibrium isotope fractionation factor at the temperature of the air-water interface based on Eq. 16a in Gibson et al. (2016) or Eq. 1.6 in Mook (2000). Returns this value as the ratio in liquid vs. the ratio in vapor (i.e., LV form, alpha > 1). Gibson et al. refer to this formulation as alpha+ and to the VL form (i.e., alpha < 1) as alpha*. Krabbenhoft et al. (1990) use alpha* (i.e, VL form, alpha < 1) in their equations.

Usage

```
Cevap_isotope_frac(ltmp, method = "Mook")
```

Arguments

ltmp	lake surface temperature (K)
method	equation to use, defaults to "Gibson" but can also be "Mook".

Value

alpha (-), the equilibrium isotope fractionation factor in LV form (i.e., alpha > 1)

References

Gibson, J.J., S.J. Birks, and Y. Yi. 2016. Stable isotope mass balance of lakes: a contemporary perspective. Quaternary Science Reviews, 131:316-328. https://doi.org/10.1016/j.quascirev.2015.04.013

Mook, W.G. (ed.) 2000. Environmental Isotopes in the Hydrologic Cycle: Volume III: Surface Water. UNESCO. Paris, France.

Krabbenhoft, D. P., C. J. Bowser, M. P. Anderson, and J. W. Valley. (1990). Estimating Groundwater Exchange with Lakes: 1. The Stable Isotope Mass Balance Method. Water Resources Research, 26(10):2445-2453. https://doi.org/10.1029/WR026i010p02445

Cevap_kinetic_frac *Kinetic Fractionation Factor*

Description

Calculates the kinetic fractionation factor based on equation 6 in Krabeenhoft et al. (1990).

Usage

```
Cevap_kinetic_frac(h, K = 14.3)
```

Arguments

h	relative humidity normalized to the temperature of the surface water (-)
K	constant determinded by wind tunnel experiments for different isotopes, defaults to K(18O) = 14.3.

Value

delta_epsilon - the kinetic fractionation factor (-)

References

Krabbenhoft, D. P., C. J. Bowser, M. P. Anderson, and J. W. Valley. (1990). Estimating Groundwater Exchange with Lakes: 1. The Stable Isotope Mass Balance Method. Water Resources Research, 26(10):2445-2453. https://doi.org/10.1029/WR026i010p02445

Cevap_normalized_humidity *Normalized Relative Humidity*

Description

Calculates the relative humidity normalized to the temperature of the surface water. Based on Equation 1.8 of Mook (2000).

Usage

```
Cevap_normalized_humidity(RH, es_a, es_l)
```

Arguments

RH	relative humidity (percent)
es_a	saturation vapor pressure for the air (kPa)
es_l	saturation vapor pressure for the lake (kPa)

Value

h - the relative humidity normalized to the temperature of the surface water (-)

References

Mook, W.G. (ed.) 2000. Environmental Isotopes in the Hydrologic Cycle: Volume III: Surface Water. UNESCO. Paris, France.

Cevap_sat_vapor_press *Saturation Vapor Pressure*

Description

Calculates the saturation vapor pressure based on temperature (of air or water) based on equations 11 and 12 of Allen et al. (1998).

Usage

```
Cevap_sat_vapor_press(tmp)
```

Arguments

tmp	temperature of air or water (degrees C)

Value

es - saturation vapor pressure (kPa)

References

Allen, R. G., Pereira, L. S., Raes, D., & Smith, M. (1998). Crop evapotranspiration: Guidelines for computing crop water requirements. Rome: FAO. Retrieved from http://www.fao.org/docrep/X0490E/x0490e00.htm.

Cevap_total_frac *Total Fractionation Factor*

Description

Calculates the total fractionation factor based on the definition of epsilon for equation 5 in Krabbenhoft et al. (1990). Note that while here we calculate calculates alpha in LV form (i.e, alpha+ > 1), the equation in Krabbenhoft et al. (1990) assumes alpha is in V/L form (i.e., alpha* < 1).

Usage

```
Cevap_total_frac(alpha, delta_epsilon)
```

Arguments

alpha	equilibrium isotope fractionation factor (-) at the temperature of the air-water interface in LV form (i.e., alpha > 1).
delta_epsilon	kinetic fractionation factor (-)

Value

epsilon - the total fractionation factor (-)

References

Krabbenhoft, D. P., C. J. Bowser, M. P. Anderson, and J. W. Valley. (1990). Estimating Groundwater Exchange with Lakes: 1. The Stable Isotope Mass Balance Method. Water Resources Research, 26(10):2445-2453. https://doi.org/10.1029/WR026i010p02445

filter_ltmp *Filter to lake surface or bottom temperature*

Description

Filters large dataset of water chemistry parameters to only lake surface or lake bottom temperature measurements.

Usage

```
filter_ltmp(chem_df = CSLSdata::water_chem, depth = "surface", use_HOBO = TRUE)
```

Arguments

chem_df	data frame with water chemistry information for all sites. Defaults to CSLSdata::water_chem.
depth	indicates which depth to use. "surface" pulls shallowest records, "bottom" pulls deepest records. Defaults to "shallow".
use_HOBO	logical defaults to TRUE to use HOBO data for temperature measurements. if false, uses field profile data.

Value

lst, same data frame as chem_df, but subset to only lake surface temperature measurements.

filter_parameter *Filter SWIMS by parameter*

Description

Subsets SWIMS dataset for a CSLS lake to only measurements for a parameter of interest.

Usage

```
filter_parameter(
  water_chem = CSLSdata::water_chem,
  param,
  plotting_name = "",
  numeric = TRUE,
  no_blanks = TRUE,
  no_dups = TRUE,
  no_age = FALSE,
  no_bad_well = FALSE,
  no_bad_sample = TRUE,
  no_lod = FALSE,
  no_comment = FALSE,
  note_lake_bottom = FALSE
)
```

Arguments

water_chem	data frame with water_chem information for sites of interest
param	parameter to subset by, must exactly match DNR description (e.g., "ALUMINUM,TOTAL RECOVERABLE"). Can be a vector of multiple descriptors.
plotting_name	- name of parameter to use for plotting, e.g., "Total Recoverable Alumninum", no units. Defaults to "".
numeric	logical defaults to TRUE to indicate results should be numeric
no_blanks	logical defaults to TRUE to exclude blank samples
no_dups	logical defaults to TRUE to exclude duplicate samples
no_age	logical defaults to FALSE to include samples that were analyzed past the holding date.
no_bad_well	logical defaults to FALSE to include samples from wells known to behave weirdly.
no_bad_sample	logical defaults to TRUE to exclude bad measurements from QC.
no_lod	logical defaults to FALSE to include samples with results below the limit of detection.
no_comment	logical defaults to FALSE to include samples with other comments from the lab
note_lake_bottom	logical defaults to FALSE to leave all lake samples as "lake". If true, notes samples > 5m deep as "lake_bottom".

Value

df, same data frame, with only the results for the parameter of interest. Also includes the given plotting name ("name").

---

get_C_lake *Calculate lake concentration*

Description

Calculates the concentration of a parameter in the lake across all samples (default) or between given dates.

Usage

```
get_C_lake(
  param = "MAGNESIUM TOTAL RECOVERABLE",
  water_chem = CSLSdata::water_chem,
  start_date = NULL,
  end_date = NULL,
  summary_type = "median",
  value_name = "C_lake"
)
```

Arguments

param	description of parameter to evaluate, defaults to "MAGNESIUM TOTAL RECOVERABLE"
water_chem	data frame with water chemistry to evaluate in "filter_parameter", defaults toCSLSdata::water_chem.
start_date	start date for evaluation, defaults to NULL to start with first sample.
end_date	end date for evaluation, defaults to NULL to end with last sample.
summary_type	Either "median", "min", or "max". Defaults to "median".
value_name	name of column with resulting values. Defaults to "C_lake".

Value

C_pcpn, a data frame with the following columns:

lake	name of lakes, i.e., "Pleasant", "Long", and "Plainfield"
C_lake	summary value of parameter in lake to use

---

get_C_pcpn *Calculate precipitation concentration*

Description

Calculates the median concentration of a parameter in precipitation across all samples (default) or between given dates.

Usage

```
get_C_pcpn(
  param = "MAGNESIUM TOTAL RECOVERABLE",
  water_chem = CSLSdata::water_chem,
  start_date = NULL,
  end_date = NULL,
  lakes = c("Pleasant", "Long", "Plainfield")
)
```

Arguments

param	description of parameter to evaluate, defaults to "MAGNESIUM TOTAL RECOVERABLE"
water_chem	data frame with water chemistry to evaluate in "filter_parameter", defaults toCSLSdata::water_chem.
start_date	start date for evaluation, defaults to NULL to start with first sample.
end_date	end date for evaluation, defaults to NULL to end with last sample.
lakes	lakes to map precipitation values to.

Value

C_pcpn, a data frame with the following columns:

lake	name of lake, i.e., "Pleasant", "Long", and "Plainfield"
C_pcpn	median value of parameter in precipitation to use

interpolate_values *Interpolate values*

Description

Interpolates values to a daily time step given start and end date of desired timeseries. Can summarize at a larger time step (e.g., monthly) if desired using final_dt flag.

Usage

```
interpolate_values(
  df,
  group_vars,
  val_var,
  start_date,
  end_date,
  final_dt = "day"
)
```

Arguments

df	data frame with water chemistry information for tracer/solute. Includes: * **date:** date of measurement * **group1 and/or group2:** grouping for measurements * **value:** value of measurement
group_vars	name of group1 and/or group2 columns, for identifying and renaming them.
val_var	name of value column, for identifying and renaming it
start_date	start date of interpolated timeseries (POSIX). If earlier than earliest date in df, values before earliest date in df will have value equal to earliest date in df.
end_date	end date of interpolated timeseries (POSIX). If later than latest date in df, values after latest date in df will have value equal to latest date in df.
final_dt	unit of time to use for interpolation. Defaults to "day", can also be "month". Used as the `unit` for `lubridate::floor_date`

Value

df, a data frame with interpolated values.

MODFLOW_Mg_metrics *Dataset: MODFLOW Mg budget metrics*

Description

Mg budget metrics calculated using all simulation results from monte carlo MODFLOW scenarios

Usage

```
data(MODFLOW_Mg_metrics)
```

Format

A data frame with the following columns.

lake name of lake, e.g., Pleasant, Long, Plainfield

metric name of hydrologic metric, i.e. solute_budget

variable name of variation on metrics, e.g. median, max, q10

value value of hydrologic metric

scenario MODFLOW scenario (e.g., "cur_irr" or "no_irr")

sim id of MODFLOW simulation

---

process_lake_temp *Process lake temperature data for calcs*

Description

Processes CSLS lake temperature data for water/solute balance calculations. Filters HOBO temperature data from CSLSdata::water_chem for surface and bottom temperatures, and summarizes at a daily or monthly time step for the desired time period.

Usage

```
process_lake_temp(start_date, end_date, dt = "day")
```

Arguments

start_date	start date of analysis period (POSIX).
end_date	end date of analysis period (POSIX).
dt	time step at which to summarize data. Defaults to "day" for daily time step, can also be "month" for monthly time step.

Details

If using with non-CSLS data, will need an alternate way to get weather data (including lake evaporation) into the format outputted by this function.

Value

df, a data frame with the following columns:

lake	name of lake, i.e., "Pleasant", "Long", and "Plainfield"
date	date of observation (POSIX). If monthly time step, monthly summary is assigned to first day of the month
ltmp_bot_C	lake temperature at bottom of lake, deg C. Used for some solute reaction rates.
ltmp_surf_C	lake temperature at surface of lake, deg C. Used for stable isotope in evaporation calculations

process_levels *Process lake elevation, area, volume data for calcs*

Description

Processes CSLS lake level data for water/solute balance calculations. Ensures no gaps in daily lake elevation data from CSLSdata::lake_levels, uses CSLSdata::bathymetry to convert daily elevations to areas and volumes, then summarizes to desired time step (daily or monthly).

Usage

```
process_levels(start_date, end_date, dt = "month")
```

Arguments

start_date	start date of analysis period (POSIX).
end_date	end date of analysis period (POSIX).
dt	time step at which to summarize data. Defaults to "day" for daily time step, can also be "month" for monthly time step.

Details

If using with non-CSLS data, will need an alternate way to get lake level, lake area, and lake volume data into the format outputted by this function.

Value

df, a data frame with the following columns:

lake	name of lake, i.e., "Pleasant", "Long", and "Plainfield"
date	date of observation (POSIX). If monthly time step, monthly summary is assigned to first day of the month
level_m	mean lake elevation (mamsl) for the time step (daily or monthly)
area_m2	mean lake area (m2) for the time step (daily or monthly)
vol_m3	mean lake volume (m3) for the time step (daily or monthly)
dV_m3	change in lake volume (m3) for the time step (daily or monthly)

process_tracer *Process tracer data for water balance calcs*

Description

Processes concentrations of parameters (e.g., d18O, Magnesium, Calcium) for use in water/solute balances. Filters CSLSdata::water_chem to specified parameter, limits to lake, precipitation, and upgradient groundwater sites, and interpolates to a daily timestep.

Usage

```
process_tracer(
  parameter,
  start_date,
  end_date,
  dt = "day",
  use_kniffin = TRUE,
  median_gw = TRUE,
  mean_lake = FALSE
)
```

Arguments

parameter	description of parameter to use in analysis, as in CSLSdata::water_chem$description (e.g., "d18O", "MAGNESIUM TOTAL RECOVERABLE")
start_date	start date of analysis period (POSIX).
end_date	end date of analysis period (POSIX).
dt	time step at which to summarize data. Defaults to "day" for daily time step, can also be "month" for monthly time step.
use_kniffin	logical defaults to TRUE to use kniffin precip stable isotope data (only when stable isotopes selected as the paramter)
median_gw	logical defaults to TRUE to use median groundwater values, rather than dynamic time series of interpolated groundwater values.
mean_lake	logical defaults to FALSE. If TRUE, calculate and use mean of min and max lake value (should equate to fall and spring turnover samples)

Details

If using with non-CSLS data, will need an alternate way to get solute/isotope concentrations into the format outputted by this function.

Value

df, a data frame with the following columns:

lake	name of lake, i.e., "Pleasant", "Long", and "Plainfield"
date	date of observation (POSIX). If monthly time step, monthly means are assigned to first day of the month
C_lake	concentration in the lake (units in per mil, mg/L, or whatever unit parameter has in CSLSdata::water_chem)
C_pcpn	concentration in precipitation (units in per mil, mg/L, or whatever unit parameter has in CSLSdata::water_chem)
C_GWin	concentration in upgradient groundwater (units in per mil, mg/L, or whatever unit parameter has in CSLSdata::water_chem)

process_weather *Process weather data for calcs*

Description

Processes CSLS weather data for water/solute balance calculations. Transforms data from CSLSdata package, calculates lake evaporation using CSLSevap package, and summarizes at a daily or monthly time step for the desired time period.

Usage

```
process_weather(start_date, end_date, dt = "day")
```

Arguments

start_date	start date of analysis period (POSIX).
end_date	end date of analysis period (POSIX).
dt	time step at which to summarize data. Defaults to "day" for daily time step, can also be "month" for monthly time step.

Details

If using with non-CSLS data, will need an alternate way to get weather data (including lake evaporation) into the format outputted by this function.

Value

df, a data frame with the following columns:

lake	name of lake, i.e., "Pleasant", "Long", and "Plainfield"
date	date of observation (POSIX). If monthly time step, monthly summary is assigned to first day of the month
day	day of year of observation (1-366)
P_mm	precipitation (mm)
E_mm	lake evaporation (mm)
atmp_C	mean air temperature (deg C)
RH_pct	mean relative humidity (percent)
irr_factor	irradiance factor, fraction from 0-1 representing relative intensity of solar radiation based on day of year. For use in dynamic lake model for some solute reactions

Index