Package 'efreadr'

May 6, 2017

Type Package

Title Read European Eddy Fluxes CSV Files

Version 0.2.2

Date 2017-05-05

Description The European Eddy Fluxes Database Cluster distributes fluxes of different Green House Gases measured mainly using the eddy covariance technique acquired in sites involved in EU projects but also single sites in Europe, Africa and others continents that decided to share their measurements in the database <http://gaia.agraria.unitus.it>. The package provides two functions to load and row-wise bind CSV files distributed by the database. Currently only L2, L3, and L4 (L=Level), half-hourly and daily (aggregation) files are supported.

License GPL-3

LazyData TRUE

RoxygenNote 6.0.1

Depends R (>= 3.2.0), readr, dplyr, ensurer, magrittr

NeedsCompilation no

Author Marco Bascietto [aut, cre]

Maintainer Marco Bascietto <marco.bascietto@crea.gov.it>

Repository CRAN

Date/Publication 2017-05-06 05:21:16 UTC

R topics documented:

---

efreadr-package   *Read European Fluxes CSV Files*

Description

The European Eddy Fluxes Database Cluster distributes fluxes of different Green House Gases measured mainly using the eddy covariance technique acquired in sites involved in EU projects but also single sites in Europe, Africa and others continents that decided to share their measurements in the database (cit. http://gaia.agraria.unitus.it). The package provides two functions to load and row-wise bind CSV files distributed by the database. Currently only L3 and L4 (L=Level), half-hourly and daily (aggregation) files are supported.

Author(s)

Marco Bascietto <marco.bascietto@crea.gov.it>

References

Source code is hosted at GitHub (<https://github.com/mbask/efreadr>)

---

read_ef_file   *Load a CSV file from the European Flux Database*

Description

File name is parsed to extract year, site identification and aggregation type. The file name must point to a valid European Fluxes file, in CSV format and must resolve to a valid file format name.

Usage

```
read_ef_file(file_name, aggregation = NA_character_, year = NA_integer_,
  fill_value = -9999L)
```

Arguments

file_name	Full path to 1 fluxes file
aggregation	character string, aggregation of the flux file to be imported (default NA_character_). This is important to properly define variable types for each aggregation type. When NA_character_ it is assumed no aggregation (eg L2 files).
year	integer value (default NA_integer). Important to correctly add efreadr_date field
fill_value	a code for a not available (NA) observation in CSV file. All the observations with 'fill_value' values are converted to NAs during import. Default is -9999L.

Details

File name is added as a field in the returned data frame as `pathname`. The `pathname` variable may be used to join a dataframe with file metadata such as year, site code, level, aggregation, as `reaf_ef_files` does.

The fluxes files code not available measures as -9999 (integer variables) or -9999.00 (floating point variables). Default behaviour of `read_ef_file` is to treat them as NAs. All -9999L, and -9999.0, -9999.00, -9999.000 values are converted to `NA` during import of files. Occasionally, -9999.00 (floting point representation of NA) appear in variables that are inherently integer (i.e. sqc variables in daily flux file). This forces a type conversion of the entire variable to `double`.

Value

a data frame as loaded from the file, added with 'pathname' column, and 'efreadr_date' column for half-hourly fluxes files

Note

For semi-hourly L4 aggregation (i.e. "h" aggregation in file name) the last row is reported as month 1, day 1, hour 00:00. A normal date conversion would convert this date to be the very first half-hour in January 1st of the current year whereas it should be the first half-hour of the January 1st of the following year. Therefore a class date field (`efreader_date`) is added to the returned data frame holding the correct date (ie: 1st January of the following year).

Examples

```
file_name <- system.file(package = "efreadr", "examples", "CEIP_EC_L4_d_FABar_2015_v02.txt")
read_ef_file(file_name)
```

---

read_ef_files	*Load all European Fluxes CSV files in one or more directories, bind all observations in a list of dataframes*

Description

European fluxes CSV files are distributed as one or more zip-compressed files from http://gaia.agraria.unitus.it . Once unzipped, all CSV files are to be found in uniquely identifed directories.

Usage

```
read_ef_files(dirs = getwd(), only_level = NULL, only_aggr = NULL, ...)
```

Arguments

`dirs`	a vector of directories where fluxes files are looked for. Defaults to current directory.
`only_level`	levels of fluxes files (defaults to NULL). Allowed levels are (currently) 3 and 4. When NULL, either L3 and L4 files are looked for.
`only_aggr`	aggregations of data (defaults to NULL). Allowed aggregations are (currently) "h" (half-hourly) and "d" (daily). When NULL, either "d" and "h" files are looked for.
`...`	additional arguments to be passed to `read_ef_file`, specifically `fill_value`

Details

All CSV files in that or those directories will be loaded and returned as a single row-wise bound data frame. The function assumes the file name regular expression pattern is like `^[A-Z]{4}_EC_Ln_a_[A-Z]{2}[a-ZA-Z]{3}_20` where n is level `[1-4]` and a is aggregation period `[hdwm]` (optionally given as function arguments)

Value

a data frame of 3 variables: `level`, `aggr`, and `fluxes`. `fluxes` is a dataframe that binds the rows of all fluxes files imported for each level/aggregation combination found. Additional columns to `fluxes` include metadata parsed from the file names: For levels 3 and 4: `project`, `level`, `aggr`, `country_id`, `site_id`, `year`, `version`, `pathname`, `dirname` For level 2: `project`, `level`, `type`, `country_id`, `site_id`, `year`, `version`, `time_res`, `pathname`, `dirname`

Note

Files in the same directory may belong to different aggregation/level combinations. The returned dataframe will keep aggregation/level combinations in separate rows.

For semi-hourly L4 aggregation (i.e. "h" aggregation in file name) the last row is reported as month 1, day 1, hour 00:00. A normal date conversion would convert this date to be the very first half-hour of the current year whereas it should be the very first half-hour of the following year. Therefore a class date field ('efreader_date') is added to the returned data frame holding the correct date (ie: January 1st of the following year).

Examples

```
dir_name <- system.file(package = "efreadr", "examples")
read_ef_files(dir_name)
```

Index