Wikiprint Book

Title: udg/ecoms/RPackage/homogeneization

Subject: TracMeteo - udg/ecoms/RPackage/homogeneization

Version: 24

Date: 05/26/2022 01:06:13 PM

Table of Contents

Vocabulary definition 3
Dictionary 3

The different nature of the different climate products, models and variables, and the idiosyncratic naming and storage conventions often applied by the various modelling centres, makes necessary a previous homogeneization across datasets in order to implement a truly user-friendly toolbox for data access. The `ecomsUDG.Raccess` package achieves this aim by defining a common *vocabulary* to all climate datasets. The particular variables of each dataset are translated -and transformed if necessary- into the common vocabulary by means of a *dictionary*. Both features are described below:

Vocabulary definition

In order to set a common framework with a precise definition of the variables, the `ecomsUDG.Raccess` package is based on the use of a vocabulary. In essence, the vocabulary is a table containing the standard names of a number of variables commonly used in impact studies and downscaling applications, subject to permanent revision or addition of new standard variables. The naming conventions and the units are based on the ?NetCDF Climate and Forecast Metadata Convention and to the ?SPECS naming convention.

Note that not all the variables listed in the vocabulary are necessarily available at the ECOMS-UDG, although all variables available must be included in the dictionary, ensuring in all cases the retrieval of homogenized data by default. The vocabulary contains the following variable information:

- `identifier`: this is the standard name that the `loadSeasonalForecast` function will recognize automatically.
- `standard_name`: standard name of the variable as defined by the CF convention.
- `units`: units in which the standard variable is returned.

The vocabulary has been included as a built-in dataset of the `ecomsUDG.Raccess` package in order to provide the user with a reference of standard variables.

```
> library(ecomsUDG.Raccess)
Loading required package: rJava
> data(vocabulary)
> print(vocabulary)
identifier                           standard_name           units
1         tas                     2-meter temperature degrees Celsius
2      tasmax                 maximum 2-m temperature degrees Celsius
3      tasmin                 minimum 2-m temperature degrees Celsius
4          tp              total precipitation amount              mm
5         psl               air pressure at sea level              Pa
6        rsds surface downwelling shortwave radiation           W.m-2
7        rlds  surface downwelling longwave radiation           W.m-2
8         uas              eastward near-surface wind           m.s-1
9         vas             northward near-surface wind           m.s-1
10       tdps            2-meter dewpoint temperature degrees Celsius
11       snld                              snow depth              mm
12       huss               2-meter specific humidity         kg.kg-1
13       hurs               2-meter relative humidity               %
14         ps                    surface air pressure              Pa
```

Dictionary

The dictionary is a table whose aim is twofold:

1. On the one hand, the dictionary is intended for the translation of generic variables, as idiosyncratically defined in each particular dataset, to the standard variables defined in the vocabulary with their corresponding nomenclature and units. This is achieved by providing a correspondence between the name of the variable as encoded in the dataset (this is the variable name returned by the `datasetInventory` function) and the corresponding name of the standard variable as defined in the vocabulary (i.e., the `identifier`), and by applying the corresponding transformation to the native variable in order to match the standard units by means of an `offset` and a `scale` factor. In some particular cases (e.g. the precipitation provided by the System4 model outputs), the variables are also deaccumulated.
2. The dictionary also provides additional metadata often not explicitly declared in the datasets, regarding the *time* aggregation of the dataset (often referred to as the *cell method*). This includes the fields `time_step`, which is merely informative, and describes the time interval between two consecutive values, and the `lower_time_bound` and `upper_time_bound`, which are the values that should be summed to each verification time to unequivocally delimit the time span encompassed by each value.

In essence, the dictionary is a comma-sepparated text file (csv), that by default is identified with the same name than the argument `dataset` of the `loadSeasonalForecast` function. The dictionaries for the currently available datasets are included in the `ecomsUDG.Raccess` package, within the *dictionaries* folder. The dictionaries are read internally by `loadSeasonalForecast` to undertake the conversions needed for returning the standard variables, so by default, the user does not need to worry about it. In case an interested user wants to inspect a particular dictionary, he/she can proceed as follows:

```
> ip <- installed.packages()
> # Path to the installed library
> libPath <- ip[grep("ecomsUDG.Raccess", ip[ ,1]), 2]
> # Path to the dictionaries folder
> dicPath <- paste(libPath, "/ecomsUDG.Raccess/dictionaries", sep = "")
> list.files(dicPath)
[1] "CFSv2_seasonal_16.csv"   "System4_annual_15.csv"   "System4_seasonal_15.csv"
[4] "System4_seasonal_51.csv"
> dic <- read.csv(list.files(dicPath, full = TRUE)[grep("CFSv2", list.files(dicPath))])
> str(dic.cfs)
'data.frame':      5 obs. of  9 variables:
$ identifier      : Factor w/ 5 levels "psl","tas","tasmax",..: 3 4 2 5 1
$ short_name      : Factor w/ 5 levels "Maximum_temperature_height_above_ground" ...
$ time_step       : Factor w/ 1 level "6h": 1 1 1 1 1
$ lower_time_bound: int  0 0 0 0 0
$ upper_time_bound: int  6 6 6 6 0
$ aggr_fun        : Factor w/ 5 levels "max","mean","min",..: 1 3 2 5 4
$ offset          : num  -273 -273 -273 0 0
$ scale           : int  1 1 1 21600 1
$ deaccum         : int  0 0 0 0 0
```

The latest version of the dictionaries can be checked-out in the development version of the package at the ?GitHub repository.

The columns of the dictionary are next described:

- `identifier`: this is the name of the standard variable, as defined in the vocabulary
- `short_name`: this is the name with which the original variable has been coded in the dataset
- `time_step`: the time interval between consecutive times in the time dimension axis (in hours). The standard nomenclature number of hours followed by the letter *h* (i.e.: 12h, 6h, 24h ...) is used in the default dictionaries included in the R package.
- `lower_time_bound`: lower time bound of the variable. This should be in hours. This is the number of hours to substract from the verification times to define the lower time boundary of the verification period. See the example below.
- `upper_time_bound`: upper time bound of the variable. This should be in hours. This is the number of hours to add to the verification times to define the upper time boundary of the verification period. See the example below.
- `aggr_fun`: time aggregation function. Type of aggregation function applied to the variable between the lower and upper time bound. For instance, typically its value is "none" for instantaneous variables, "mean" for mean daily temperatures or "sum" for daily precipitation values. See the example below.
- `offset`: constant summed to the original variable for units conversion (e.g.: offset = -273.15 for conversion from Kelvin to Celsius)
- `scale`: scale factor applied to the original variable for units conversion (e.g.: scale = 0.001 for conversion from m to mm)
- `deaccum`. This is a logical flag (0 = FALSE, 1= TRUE), which indicates if the variable should be de-accumulated at each time step. Typically applied to precipitation in some forecast datasets.

Example: Suppose we have daily temperature data, each record associated to a date (e.g. 19780420 for 20th April 1978). The lower time bound in this case is 0, and the upper time bound 24, indicating that the verification period of the value starts 20 April 1978 at 00:00 and ends 20 April 1978 23:59:59 (i.e., the range `[19780420, 19780421)`. Because it is mean temperature, the aggregation function for this time interval would be "mean". If the same record has a time associated at 12:00 rather than 00:00 (i.e. 20 April 1978 at 12:00), then the lower and the upper time bounds would be 12.