

Wikiprint Book

Title: udg/ecoms/RPackage/homogeneization

Subject: TracMeteo - udg/ecoms/RPackage/homogeneization

Version: 24

Date: 05/26/2022 01:46:42 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 homogenization across datasets in order to implement a truly user-friendly toolbox for data access. The `meteor` 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 next described.

Vocabulary definition

In order to set a common framework with a precise definition of the variables, the `meteor` 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 standard name [?table](#) provided in the frame of the SPECS project, although in case of conflict, and in order to maximize the inter-operability of the vocabulary, the nomenclature is also compliant with the [?NetCDF Climate and Forecast Metadata Convention](#).

- **Identifier:** this is the standard name that the loading functions require as argument when we set the `standard.vars` argument to `TRUE` (see the [defined functions?](#)).
- **Standard_name:** standard name of the variable as defined by the CF convention.
- **Units:** units in which the standard variable is returned

```
"identifier","standard_name","units"
"ta","temperature","degrees Celsius"
"tas","2-meter temperature","degrees Celsius"
"tasmax","maximum 2-m temperature","degrees Celsius"
"tasmin","minimum 2-m temperature","degrees Celsius"
"pr","Precipitation amount","mm"
"zg","geopotential_height","m"
"plev","air_pressure","Pa"
"psl","air_pressure_at_sea_level","Pa"
"ps","surface_air_pressure","Pa"
"hus","specific_humidity","kg kg-1"
"hur","relative_humidity","1"
"ua","eastward_wind","m s-1"
"va","northward_wind","m s-1"
```

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 (`short_name`) and the corresponding name of the standard variable as defined in the vocabulary (`identifier`), and by applying the corresponding transformation to the native variable in order to match the standard units by means of a *scale factor* and an *offset*. 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.

The dictionary is a comma-separated text file (csv), that by default is identified with the same name than the dataset, and the extension `.dic`, and stored in the same directory than the dataset, although its name and location can be other if adequately specified in the loading functions by the argument `dictionary`. The dictionary must be created *"by hand"* by the user, because it requires some *a priori* knowledge about the characteristics of the data stored in the dataset, that can be partly obtained using the function [dataInventory?](#). 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)
- **lower_time_bound:** lower time bound of the variable. This should be in hours.
- **upper_time_bound:** upper time bound of the variable. This should be in hours. For instance, if a variable has identical lower and upper time bounds, it means that it is instantaneous.
- **aggr_fun:** time aggregation function. Type of aggregation function applied to the variable between the lower and upper time bound.

- `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.

In the following example, we show the characteristics of the dictionary constructed for the 15 members seasonal forecast of the ECMWF's System4 model:

```

identifier,short_name,time_step,lower_time_bound,upper_time_bound,aggr_fun,offset,scale,deaccum
tasmax,Maximum_temperature_at_2_metres_since_last_24_hours_surface,24h,0,24,max,-273.15,1,0
tasmin,Minimum_temperature_at_2_metres_since_last_24_hours_surface,24h,0,24,min,-273.15,1,0
tas,Mean_temperature_at_2_metres_since_last_24_hours_surface,24h,0,24,mean,-273.15,1,0
pr>Total_precipitation_surface,24h,0,24,sum,0,1000,1
psl,Mean_sea_level_pressure_surface,6h,0,0,none,0,1,0

```

Note that column names matter (not so their relative order), because the data load functions will perform the conversion of the variables to the standard format by finding the corresponding values by the name of the columns. The [?meteoR](#) package includes some dictionaries, and specific examples are given in the [Examples section?](#)