

Regional-Continental domain selections

In this example we will load data for Europe for the variable surface (2m) minimum temperature (`var = tasmin`), for the first two members (`members = 1:2`) of the CFSv2 hindcast (`dataset = CFSv2_seasonal_16`), considering the wintertime (DJF, `season = c(12,1,2)`) for the 10-year period 2001-2010 (`years = 2001:2010`), according to the forecast of previous September (`leadMonth = 3`). The original variable is stored as 6-hourly data for this particular dataset. Instead of loading the whole 6-hourly time series, or filtering by a particular time as in the [?previous example](#) we will retrieve the daily mean values, by setting the argument `time = "DD"`, that internally computes the daily mean from the 6-hourly instantaneous values.

```
> ex2 <- loadECOMS(dataset = "CFSv2_seasonal_16", var = "tasmin", members = 1:2, lonLim = c(-15,35), latLim = c(32, 75), s
[2014-06-17 12:47:49] Defining homogeneization parameters for variable "tasmin"
NOTE: daily mean will be calculated from the 6-h instantaneous model output
[2014-06-17 12:47:49] Defining geo-location parameters
[2014-06-17 12:47:49] Defining initialization time parameters
[2014-06-17 12:47:54] Retrieving data subset ...
[2014-06-17 12:54:33] Done
> print(object.size(ex2), units = "Mb")
35 Mb
```

In this case, the data are stored in a 4D-array, with the dimensions indicated by the `dimensions` attribute:

```
> str(ex2$Data)
num [1:902, 1:54, 1:47, 1:2] 17.4 16.4 17.4 18.7 18.4 ...
- attr(*, "dimensions")= chr [1:4] "time" "lon" "lat" "member"
```

This is an example on how to plot the members selected as spatial means for the 10-year period. Note that this example uses the library `fields`, not attached on load of the `ecomUDG.Raccess` package:

```
> library(fields) # Install if not available to reproduce the example
> member1 <- apply(ex2$Data[,,,1], FUN = mean, MARGIN = c(2,3))
> member2 <- apply(ex2$Data[,,,2], FUN = mean, MARGIN = c(2,3))
> x <- ex2$xyCoords$x
> y <- ex2$xyCoords$y
> par(mfrow = c(1,2))
> image.plot(x,y,member1, asp = 1, main = "Member 1")
> world(add = TRUE)
> image.plot(x,y,member2, asp = 1, main = "Member 2")
> world(add = TRUE)
```

