

Wikiprint Book

Title: Forecast skill visualization: a worked example using visualizeR

Subject: TracMeteo - udg/ecomms/RPackage/examples/visualization

Version: 9

Date: 08/16/2022 07:08:48 AM

Table of Contents

Forecast skill visualization: a worked example using visualizeR	3
Package loading/install	3
Data loading from the ECOMS-UDG	3
Tercile plot	5
Bubble plots	6

Forecast skill visualization: a worked example using visualizeR

NOTE: In this example we use the same forecast and observations datasets than in the [?previous example](#) on forecast verification.

Package loading/install

We first load (and install if necessary) the required libraries. `loader.ECOMS` and `visualizeR`, used for data loading and visualization respectively (see the installation instructions for [?loader.ECOMS](#) and [?visualizeR](#) packages). In addition, `downscaleR` will be used for data manipulation (regridding).

```
library(loader.ECOMS)
library(visualizeR)
library(downscaleR)
```

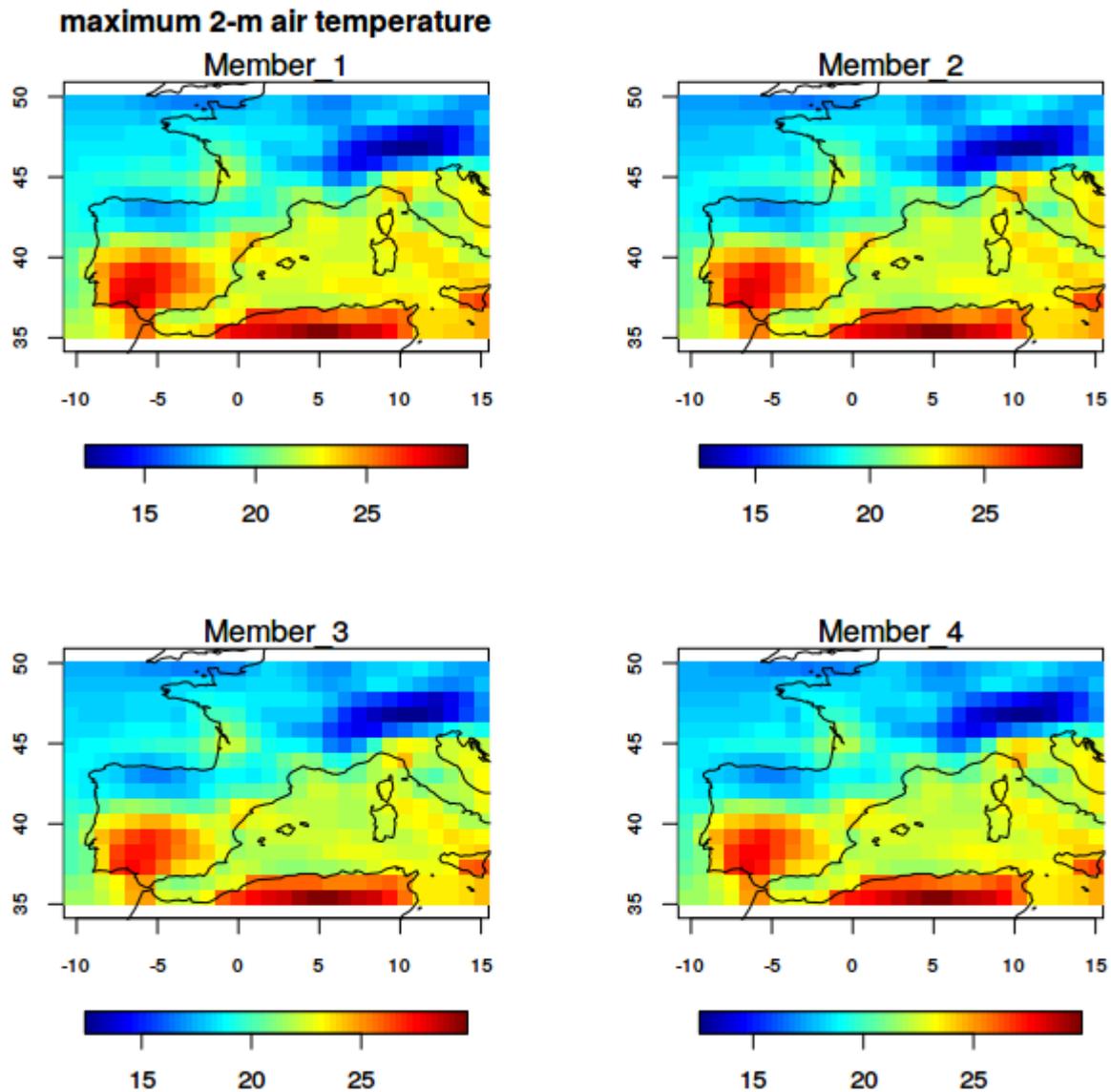
Data loading from the ECOMS-UDG

We load the predictions:

```
tx.forecast <- loadECOMS(dataset = "CFSv2_seasonal",
                        var = "tasmax",
                        members = 1:4,
                        lonLim = c(-10, 15),
                        latLim = c(35, 50),
                        season = 6:8,
                        years = 1991:2000,
                        leadMonth = 2)

## [2016-05-12 12:56:18] Defining homogeneization parameters for variable "tasmax"
## [2016-05-12 12:56:18] Opening dataset...
## [2016-05-12 12:56:48] The dataset was successfully opened
## [2016-05-12 12:56:48] Defining geo-location parameters
## [2016-05-12 12:56:49] Defining initialization time parameters
## [2016-05-12 12:56:51] Retrieving data subset ...
## [2016-05-12 13:07:02] Done

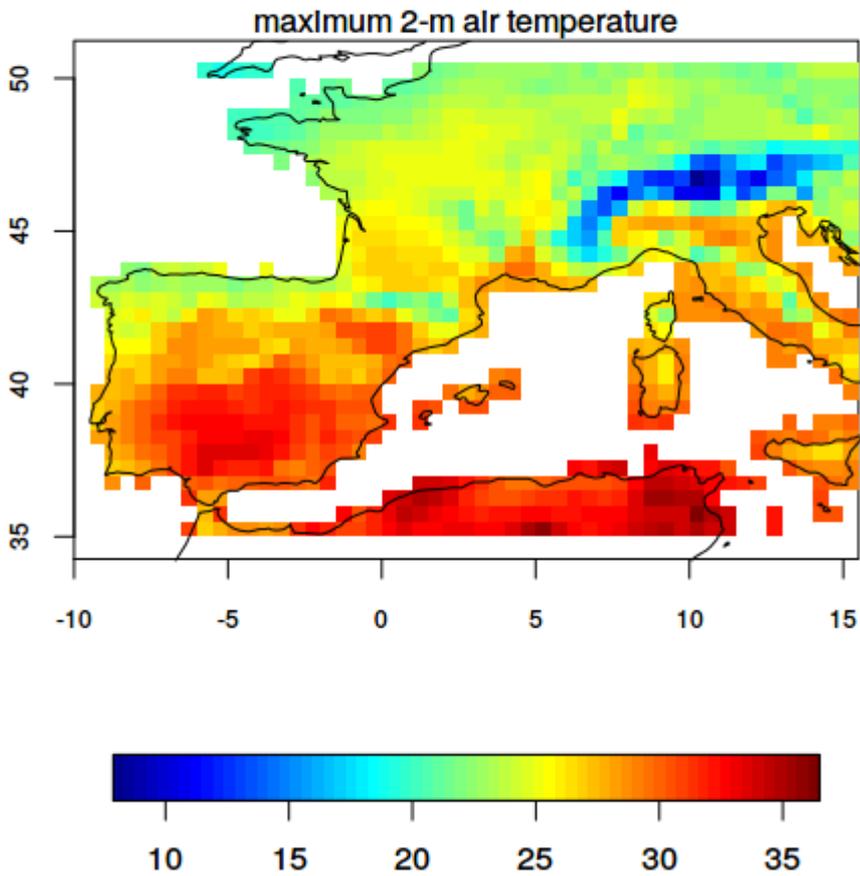
plotMeanGrid(tx.forecast, multi.member = TRUE)
```



And then we load the verifying observations (WFDEI dataset):

```
tx.obs <- loadECOMS(dataset = "WFDEI",
  var = "tasmax",
  lonLim = c(-10 ,15),
  latLim = c(35, 50),
  season = 6:8,
  years = 1991:2000)
## [2016-05-12 14:03:40] Defining homogeneization parameters for variable "tasmax"
## [2016-05-12 14:03:40] Opening dataset...
## [2016-05-12 14:03:42] The dataset was successfully opened
## [2016-05-12 14:03:42] Defining geo-location parameters
## [2016-05-12 14:03:42] Defining time selection parameters
## [2016-05-12 14:03:42] Retrieving data subset ...
## [2016-05-12 14:03:52] Done

plotMeanGrid(tx.obs)
```



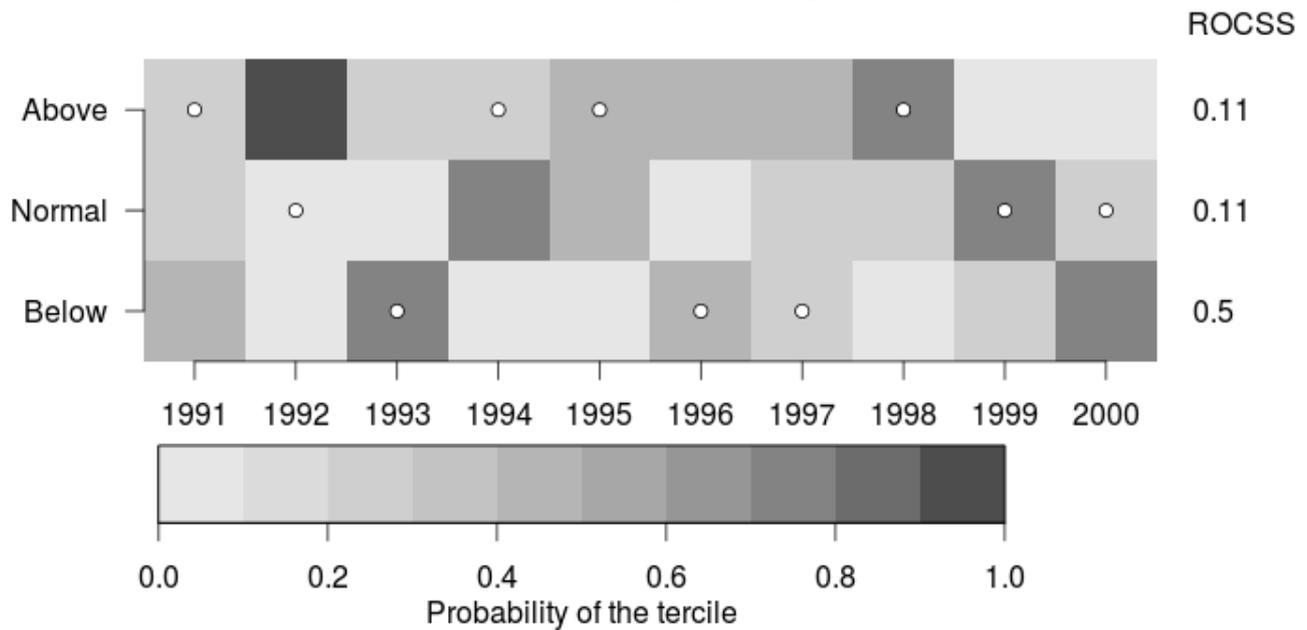
We next show different forecast visualization plots:

Tercile plot

Tercile plots are very useful in order to obtain a quick overview of the overall skill of the predictions over a region of interest. First of all, we select a target year for which the predictions are to be analysed:

```
year.target <- 1995
tercilePlots4(tx.forecast, tx.obs, year.target, detrend = TRUE, color.pal = "bw")
## Warning messages:
## 1: In spatialMean(mm.obj) :
##   The results presented are the spatial mean of the input field
## 2: In spatialMean(obs) :
##   The results presented are the spatial mean of the input field
```

maximum 2-m air temperature, jun to ago, 1995



For each member, the daily predictions are averaged to obtain a single seasonal forecast (this yields a first warning, as in this example). For rectangular spatial domains (i.e., for grids), the spatial average is first computed (with a warning) to obtain a unique series for the whole domain, as in this example. The corresponding terciles for each ensemble member are then computed for the analysis period. Thus, data is converted to a series of tercile categories by considering values above, between or below the terciles of the whole period. The probability of a member to fall into its respective climatological tercile is represented by the colorbar (different color palettes are available through the argument `color.pal`). For instance, probabilities below 1/3 are very low, indicating that a minority of the members falls in the tercile. Conversely, probabilities above 2/3 indicate a high level of member agreement (more than 66% of members falling in the same tercile). The observed terciles (the events that actually occurred) are represented by the white circles.

Finally, the ROC Skill Score (ROCSS) is indicated in the secondary (right) Y axis. For each tercile, it provides a quantitative measure of the forecast skill, and it is commonly used to evaluate the performance of probabilistic systems. The value of this score ranges from 1 (perfect forecast system) to -1 (perfectly bad forecast system). A value zero indicates no skill compared with a random prediction.

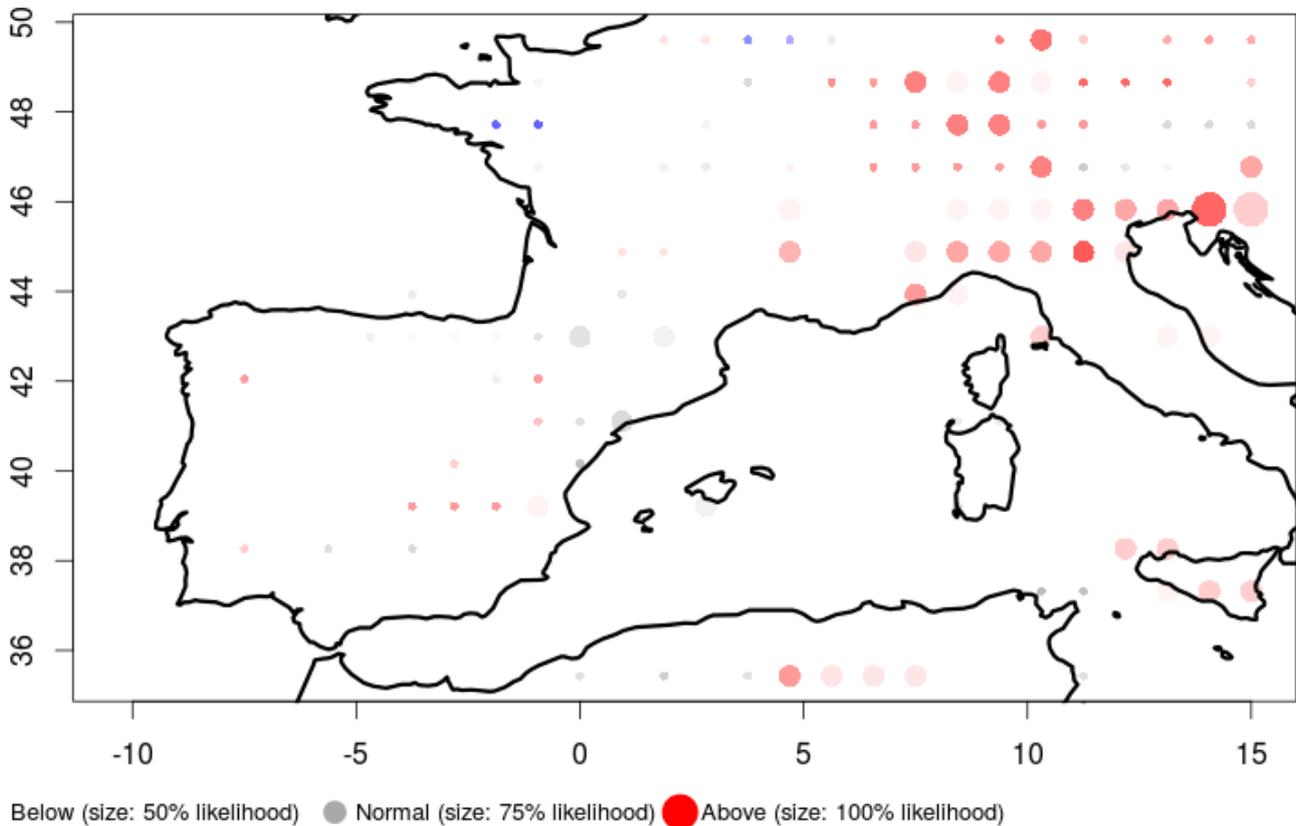
Bubble plots

While the tercile plot provides an areal overview for rectangular spatial domains, to focus on particular regions bubble plots are very useful. In order to compare the predictions against the observations for every grid point these need to be in the same reference grid. We use the interpolation capabilities of `downscaler` to this aim:

```
obsintp <- interpGrid(tx.obs,
  new.coordinates = getGrid(tx.forecast),
  method = "nearest")
## [2016-05-13 16:06:45] Calculating nearest neighbors...
## [2016-05-13 16:06:45] Performing nearest interpolation... may take a while
## [2016-05-13 16:06:45] Done
## Warning message:
## In interpGrid(tx.obs, new.coordinates = getGrid(tx.forecast), method = "nearest") :
##   The new longitudes are outside the data extent
```

```
bubblePlotS4(tx.forecast,
  obsintp,
  year.target,
  detrend = FALSE,
  size.as.probability=TRUE,
  score = TRUE)
```

maximum 2-m air temperature, jun to ago, 1995

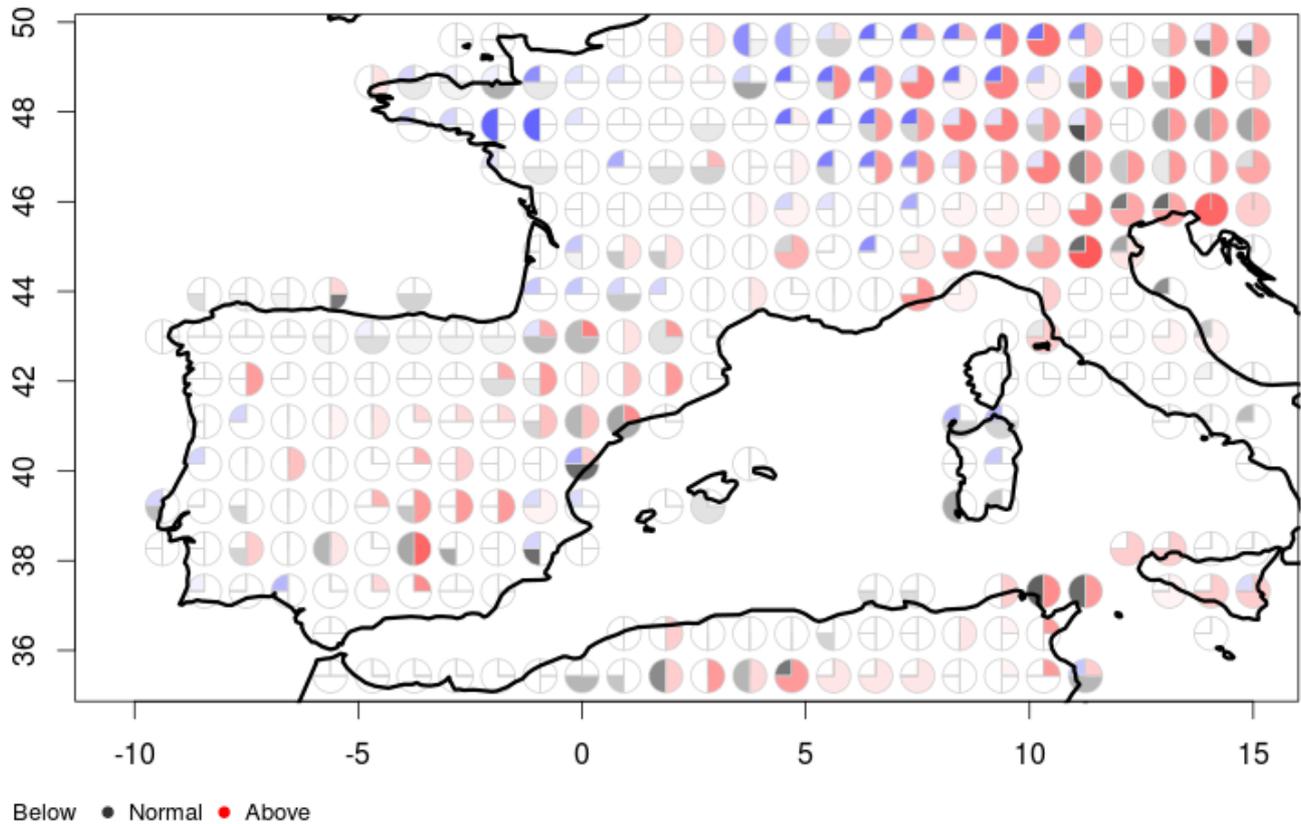


The bubble plot represents the most likely tercile in colors, the probability of that tercile with the size of the bubble (optional) and the skill of the forecast system for that tercile as transparency of the bubble (optional). Currently, the skill score used is the ROCSS. The color palette for the terciles (blue, grey, red) can be reversed (e.g for precipitation) using the `color.reverse=TRUE` option.

Pie charts instead of bubbles can be drawn indicating the predicted likelihood of each tercile, as in this example (using the `piechart=TRUE` option).

```
bubblePlotS4(tx.forecast,
             obsintp,
             year.target,
             detrend = FALSE,
             piechart = TRUE,
             score = TRUE)
```

maximum 2-m air temperature, jun to ago, 1995



Using the bubble plot allows for instance to test the effect of data detrending, as we suggested in the previous example that may be the source of some -artificial- skill in the north-eastern sector of the study area. We control this through the logical argument `detrend`:

```
bubblePlotS4(tx.forecast,
             obsintp,
             year.target,
             detrend = TRUE,
             piechart = TRUE,
             score = TRUE
             )
```

maximum 2-m air temperature, jun to ago, 1995

