

visualizeR

*M. D. Frías (friasmd@unican.es), J. Fernández. Santander Meteorology Group
(http://meteo.unican.es). Universidad de Cantabria. Spain*

27/09/2016

Contents

1 About visualizeR	1
1.1 Requirements and installation	1
1.2 Sample data	2
1.3 Visualization functions	3
1.3.1 Tercile plot	3
1.3.2 Tercile bar plot	4
1.3.3 Box plots on climatology fan chart	5
1.3.4 Bubble plot	8

1 About visualizeR

`visualizeR` is an R package implementing a set of advanced visualization tools for forecast verification. It is fully integrated from the R climate data structures generated by the loading functions of the `loadeR` bundle (including the `loadeR.ECOMS` interface to the **ECOMS-UDG**), thus providing seamless integration with all steps of forecast data analysis, from data loading to post-processing, downscaling and bias correction and visualization.

This document shows some sample R functions to visualize seasonal forecast information with different levels of complexity.

1.1 Requirements and installation

For data loading, get the latest version of `loadeR` and `loadeR.ECOMS` using the following code. This has to be done **just once** in this order:

```
devtools::install_github(c("SantanderMetGroup/loadeR.java",  
                           "SantanderMetGroup/loadeR",  
                           "SantanderMetGroup/loadeR.ECOMS"))  
library(loadeR.ECOMS)
```

We also use `transformeR` functions for some climate data transformations (e.g. interpolate data to a common grid, etc).

```
devtools::install_github("SantanderMetGroup/transformeR")
```

```
library(transformeR)
```

```
## transformeR version 0.0-1 (2016-09-23) is loaded
```

The recommended installation procedure for visualizeR is to use the `install_github` command from the `devtools` R package.

```
devtools::install_github("SantanderMetGroup/visualizeR@devel")
```

```
library(visualizeR)
```

```
## Loading required package: sm
```

```
## Package 'sm', version 2.2-5.4: type help(sm) for summary information
```

```
## visualizeR version 0.1-0 (2016-09-27) is loaded
```

1.2 Sample data

We should log into the **ECOMS-UDG** to download the data:

```
loginECOMS_UDG("username", "password")
```

We can load some sample data from the **ECOMS-UDG**. E.g. for surface temperature (**tas**):

```
var <- "tas"
year.ini <- 1983
year.end <- 2008
season <- c(12,1,2)
lead.month <- 1
members <- 1:15
```

The plots shown in this document are focused on Spain:

```
lonlim <- c(-10,5)
latlim <- c(35,45)
```

but you could choose any other region by setting appropriate lat-lon boundaries. For example, for Peru:

```
lonlim <- c(-83,-66)
latlim <- c(-20,0)
```

We are ready to load the predictions and observations.

```
prd <- loadECOMS(dataset = "CFSv2_seasonal", var=var,
                 lonLim=lonlim, latLim=latlim, season=season, years=year.ini:year.end,
                 leadMonth=lead.month, members=members, time="DD", aggr.d="mean"
                 )
```

For the observation the loading process is faster. Then, we could use the `loadECOMS` function

```
obs <- loadECOMS(dataset = "WFDEI", var=var,
                lonLim=lonlim, latLim=latlim, season=season, years=year.ini:year.end
                )
```

1.3 Visualization functions

1.3.1 Tercile plot

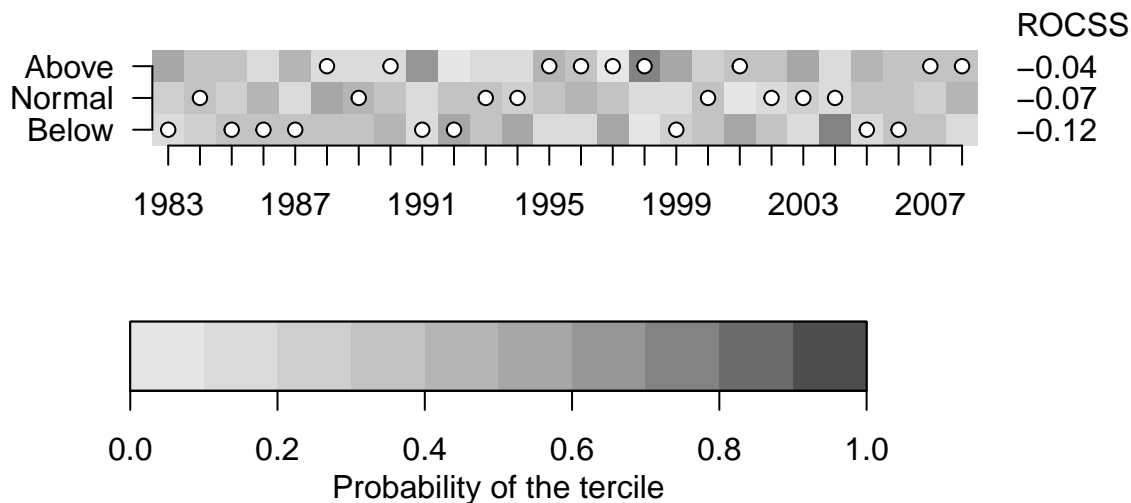
Tercile plots are intended to show the performance of a forecast system along a time period. This plot depicts the predicted probability of upper, middle and lower terciles. The shading of each square represents probability (darker shade=greater probability) and the dots show the corresponding observed tercile for each of the years. Skill is represented by the ROCSS for each tercile (values are shown on the right side). The next version of this function will also include an additional column with the forecast for the next season (without the dot representing the observed outcome).

```
tercilePlot(prd, obs)
```

```
## Warning in spatialMean(mm.obj): The results presented are
## the spatial mean of the input field
```

```
## Warning in spatialMean(obs): The results presented are the
## spatial mean of the input field
```

2-meter air temperature, dic to feb



This plot can be used to allow users to “get a sense” of what a particular ROCCS value means in terms of past performance of the probabilistic forecasts.

1.3.2 Tercile bar plot

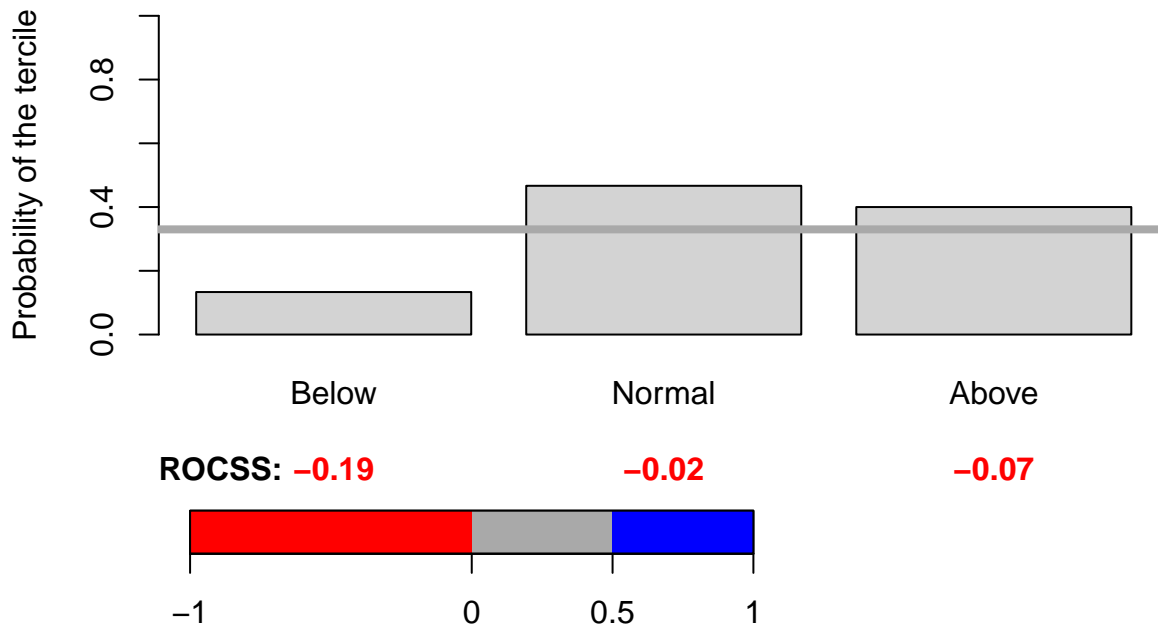
This simple bar graph depicts the predicted likelihood of upper, middle and lower tercile relative to climatology (represented by a grey line). The ROCSS is represented numerically for each tercile and it is colour coded to indicate whether there is No Skill (Red), Some Skill (Grey), or Good Skill (Blue); so as provide a salient warning to users where scores are negative. This plot is based on the currently used by MeteoSwiss in their online seasonal temperature and precipitation forecasts (<http://www.meteoswiss.admin.ch/home/climate/future/seasonal-outlook.html>).

```
year.target <- 2008
tercileBarplot(prd, obs, year.target)
```

```
## Warning in spatialMean(mm.obj): The results presented are
## the spatial mean of the input field
```

```
## Warning in spatialMean(obs): The results presented are the
## spatial mean of the input field
```

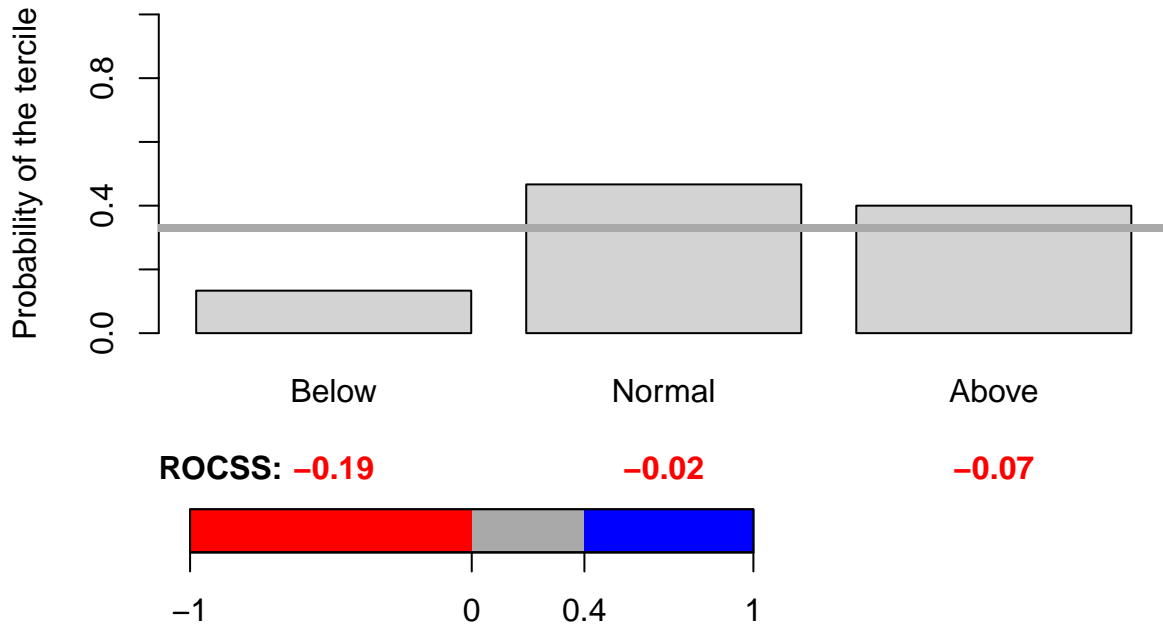
2-meter air temperature, dic to feb, 2008



Depending on the area, the positive score threshold to be remarked can be different. It is possible to indicate a positive score threshold to highlight in a different colour those values of the score above a particular value. 0.5 is the default value.

```
tercileBarplot(prd, obs, year.target, score.threshold=0.4)
```

2-meter air temperature, dic to feb, 2008

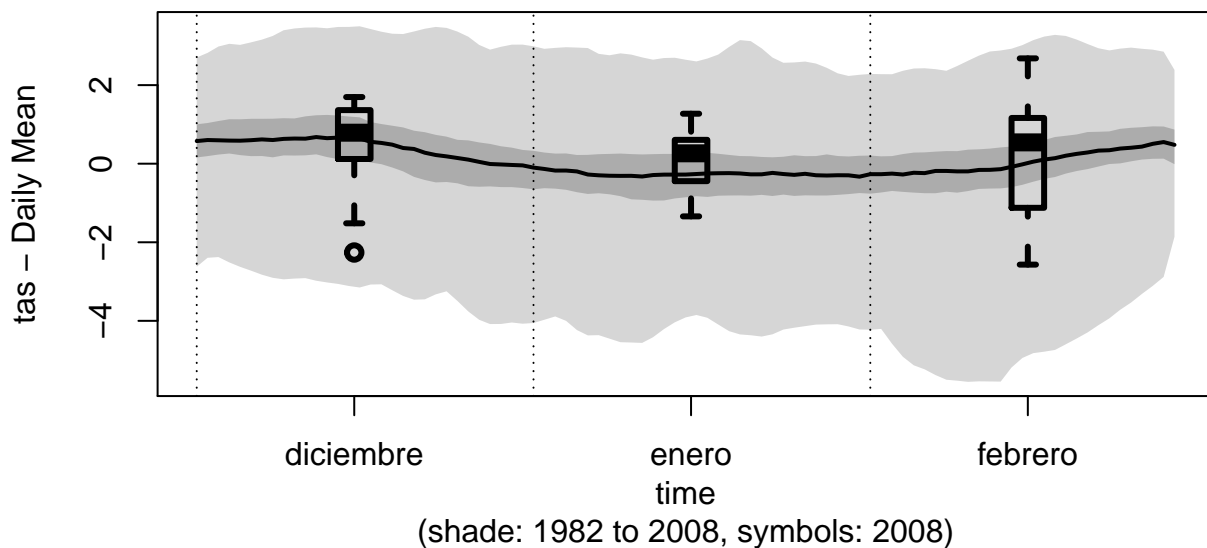


1.3.3 Box plots on climatology fan chart

In this plot the background represents the climatology for the forecast period. The shaded areas show the central tercile (dark shade) and the maximum and minimum (light shade). To avoid overinterpretation of daily peaks, the daily data has been smoothed by means of a (centered) moving average of 31 days. Therefore, at the location of the boxplots, the background shows the monthly mean forecast (the terciles and extremes being computed over members and years).

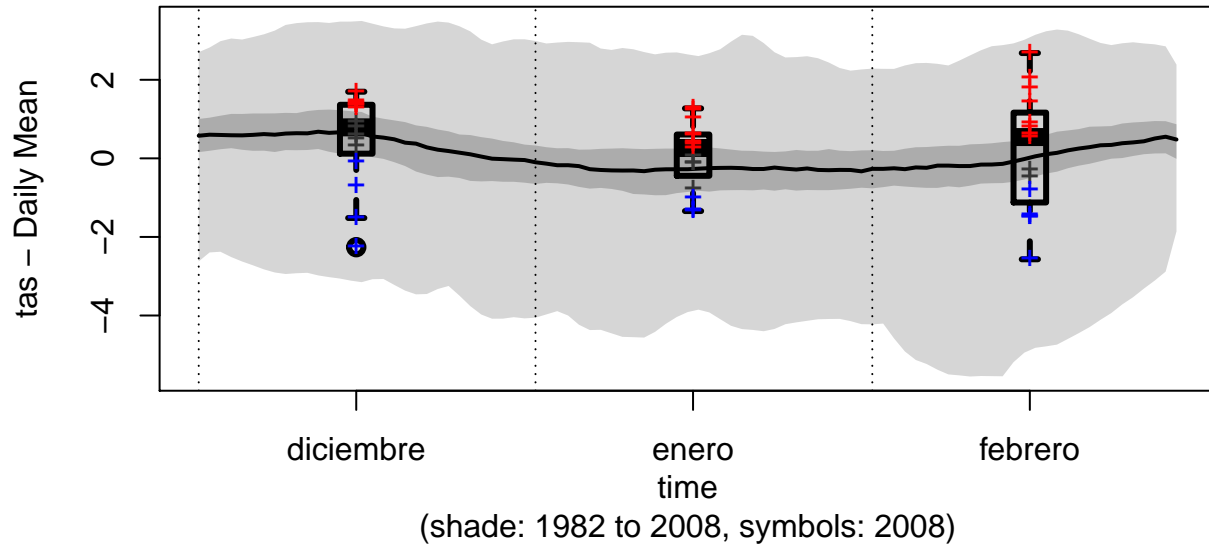
The boxplots show the spread of the monthly mean forecast (for the different members).

```
spreadPlot(prd, year.target)
```

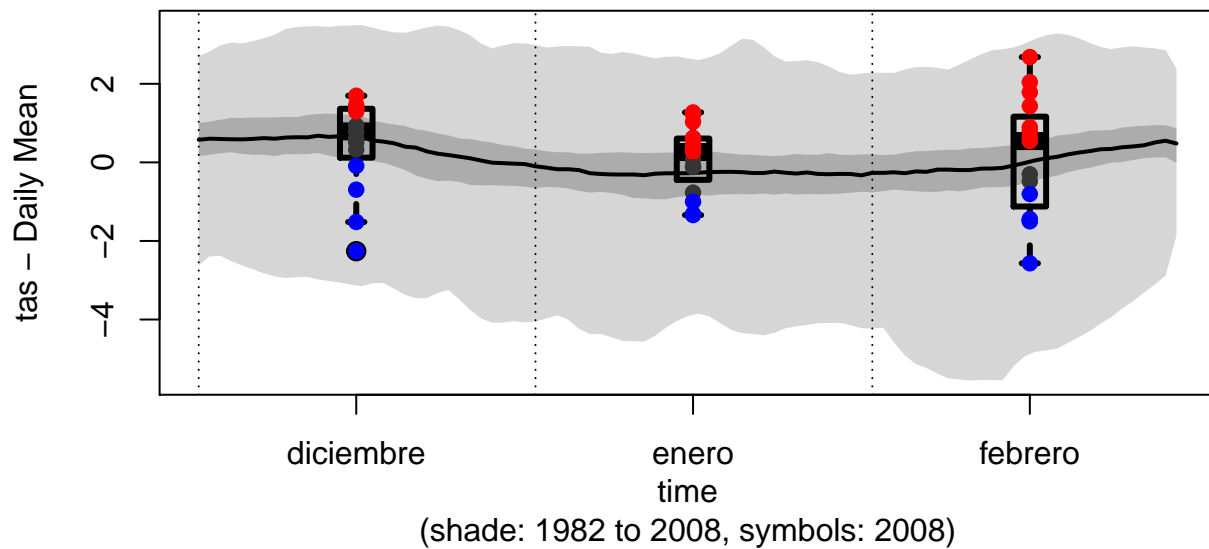


You could also add to the boxplot the values of the ensemble members using crosses or any other symbol (pch option). Single symbol represents the mean for each ensemble member for each time period. They are colour coded to indicate whether they fall into the upper, middle or lower tercile.

```
spreadPlot(prd, year.target, add.points=T)
```

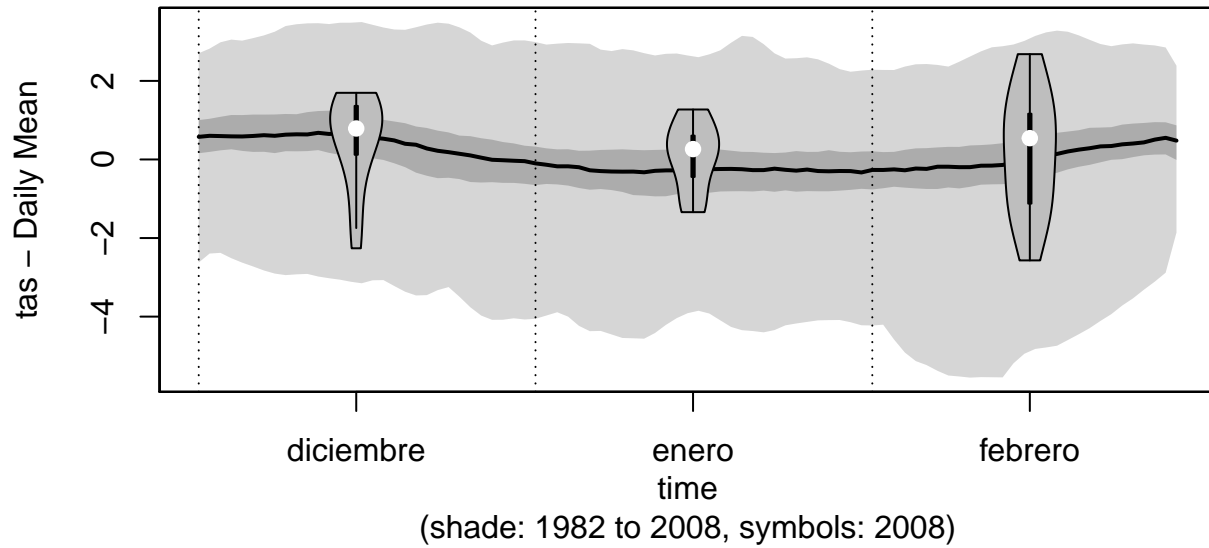


```
spreadPlot(prd, year.target, add.points=T, pch=19)
```

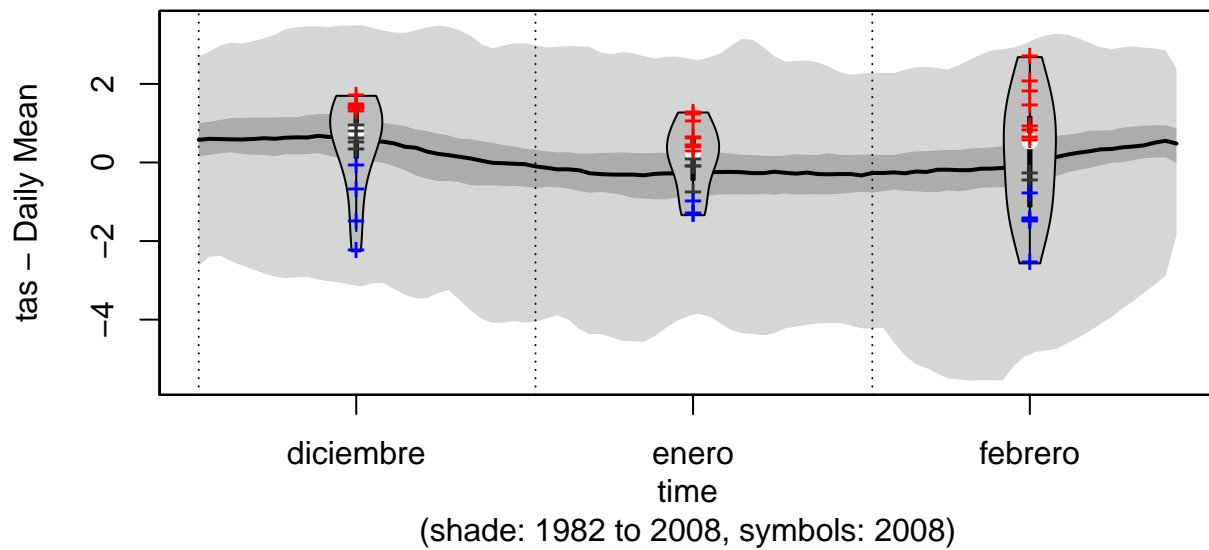


The boxplots can be replaced by *violin plots*, to unveil multimodalities in the data.

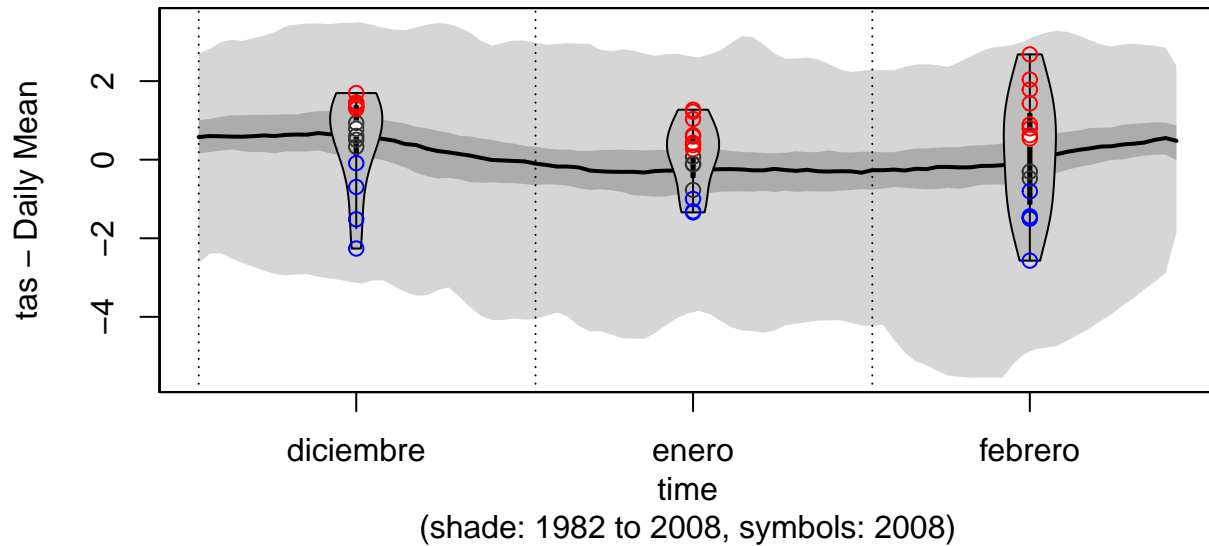
```
spreadPlot(prd, year.target, violin=T)
```



```
spreadPlot(prd, year.target, violin=T, add.points=T)
```

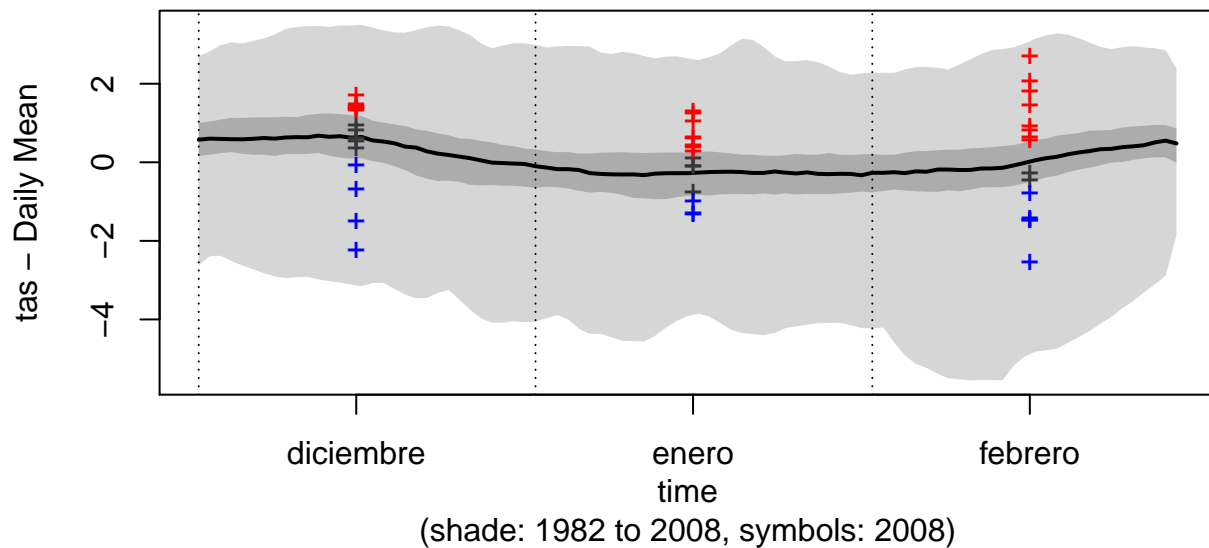


```
spreadPlot(prd, year.target, violin=T, add.points=T, pch=21)
```



It is also possible to plot only the values of the ensemble members.

```
spreadPlot(prd, year.target, boxplot=F, violin=F, add.points=T)
```



1.3.4 Bubble plot

The bubble plot is a map that represents the most likely tercile in colours, 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. Only positive scores are shown (the negative ones –the system is worse than the climatology– are not plotted). Pie charts in stead of bubbles can be drawn indicating the predicted likelihood of each tercile. This is based on visualisations developed by Slingsby et al. (2009).

```
# Interpolation to model grid
obs <- interpGrid(obs, new.coordinates = getGrid(prd), method = "nearest")
```

```
## Warning in interpGrid(obs, new.coordinates = getGrid(prd),
```



```
## method = "nearest"): The new longitudes are outside the data
## extent

## [2016-09-27 12:34:54] Calculating nearest neighbors...

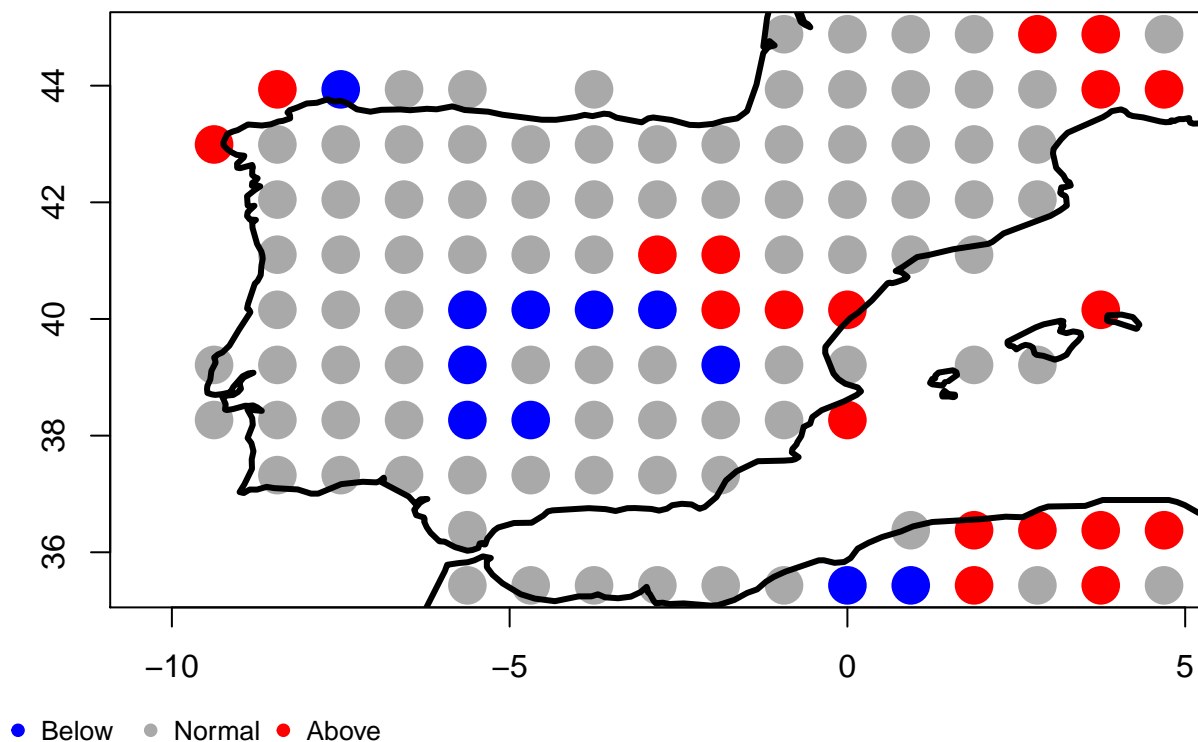
## [2016-09-27 12:34:54] Performing nearest interpolation... may take a while

## [2016-09-27 12:34:54] Done
```

The bubble plot can be invoked with different levels of complexity:

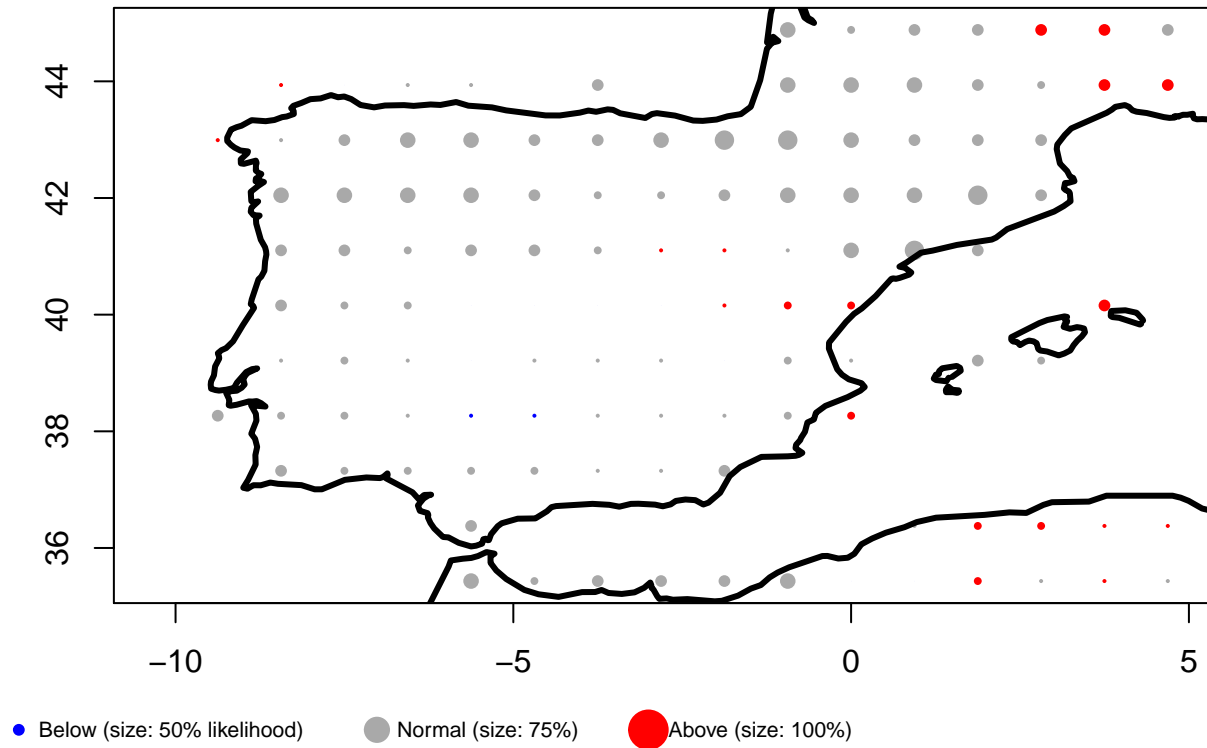
```
bubblePlot(prd, obs, year.target, size.as.probability=F, score=F)
```

2-meter air temperature, dic to feb, 2008



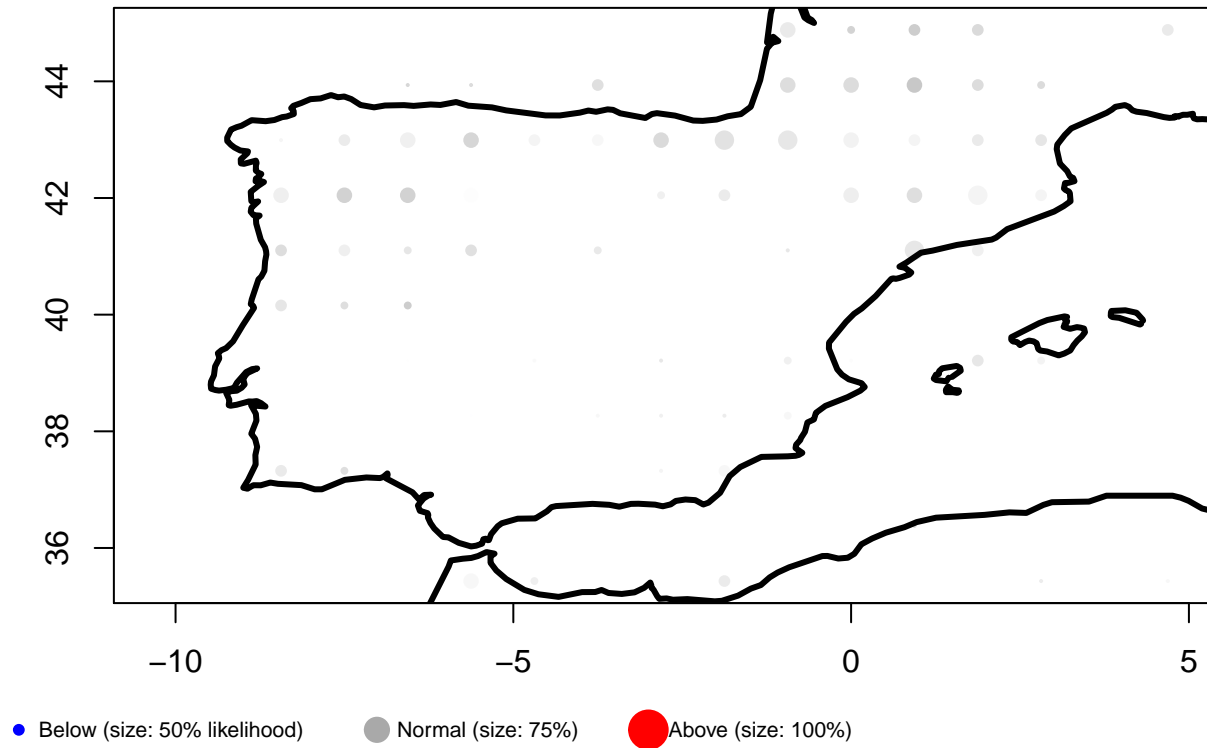
```
bubblePlot(prd, obs, year.target, size.as.probability=T, score=F)
```

2-meter air temperature, dic to feb, 2008



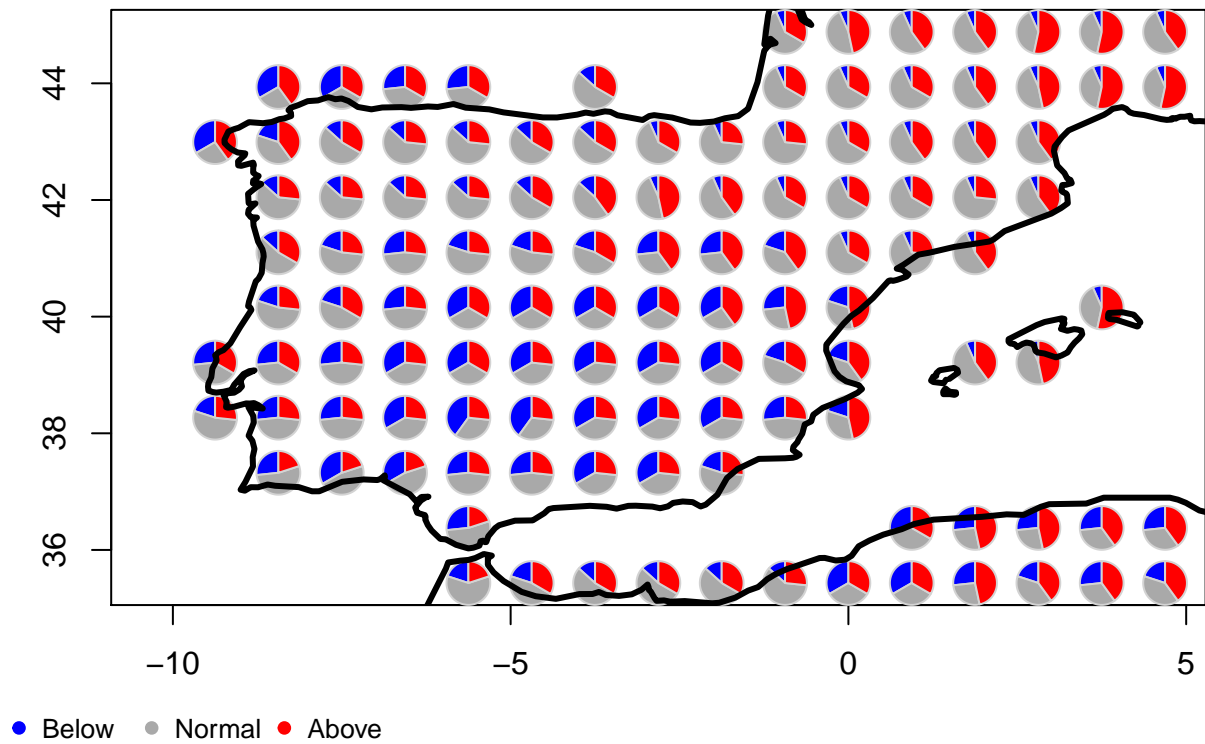
```
bubblePlot(prd, obs, year.target, size.as.probability=T, score=T)
```

2-meter air temperature, dic to feb, 2008



```
bubblePlot(prd, obs, year.target, pie=T, score=F)
```

2-meter air temperature, dic to feb, 2008



```
bubblePlot(prd, obs, year.target, pie=T, score=T)
```

2-meter air temperature, dic to feb, 2008

