visualizeR: communication and visualization of uncertainty in seasonal climate prediction

Companion examples to the paper by Frias *et al.* 2018 in Environmental Modelling & Software DOI:10.1016/j.envsoft.2017.09.008

Santander Meteorology Group

07 de Sep 2017

Abstract

This notebook is a companion document to the paper indicated in the title, illustrating the main characteristics and functionalities of package `visualizeR`, an R package for implementing a set of advanced visualization tools for the communication of probabilistic forecasts together with different aspects of forecast quality. `visualizeR` is a core package of the climate4R framework.**CONTENTS**

Contents

Package installation

```
install.packages('devtools')

devtools::install_github(c("SantanderMetGroup/transformeR",
                           "SantanderMetGroup/visualizeR"))

library(visualizeR)

## Loading required package: sm

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

## Loading required package: transformeR

## transformeR version 1.3.3 (2018-04-24) is loaded
```

```
## visualizeR version 1.2.0 (2017-12-20) is loaded
##
## Attaching package: 'visualizeR'
## The following objects are masked from 'package:transformeR':
##
##     clim2sgdf, map.lines, map.stippling
```

Example data

In this part we will use the built-in datasets included in the package (see `data(package = "visualizeR")` for an overview). Seasonal hindcast and operational predictions are taken from the CFSv2 seasonal forecasting system produced by NCEP (Saha *et al.* 2013). Data from the NCEP-NCAR reanalysis 1 (Kalnay *et al.* 1996) are used as reference for verification. All datasets contain global data for near-surface air temperature for boreal winter (DJF). The hindcast dataset spans the temporal period 1983-2010.

```
data(tas.cfs)
data(tas.cfs.operative.2016)
data(tas.ncep)
```

In this step, the resolution of all data is downgraded to a 5 degree resolution regular grid, to improve the visualiation at a global scale, and for the sake of brevity in teh calculations.

```
# Adjusting data spatial resolution to 5º lat-lon resolution
newgrid <- getGrid(tas.cfs)
attr(newgrid, "resX") <- 5
attr(newgrid, "resY") <- 5
lower.res <- function(x, newgrid) {
      interpGrid(x, new.coordinates = newgrid, method = "bilinear", bilin.method = "fields")
}
obs <- lower.res(tas.ncep, newgrid)
```

```
## Warning in interpGrid(x, new.coordinates = newgrid, method = "bilinear", :
## The new latitudes are outside the data extent
```

```
hindcast <- lower.res(tas.cfs, newgrid)
forecast <- lower.res(tas.cfs.operative.2016, newgrid)
```

Bubble plots

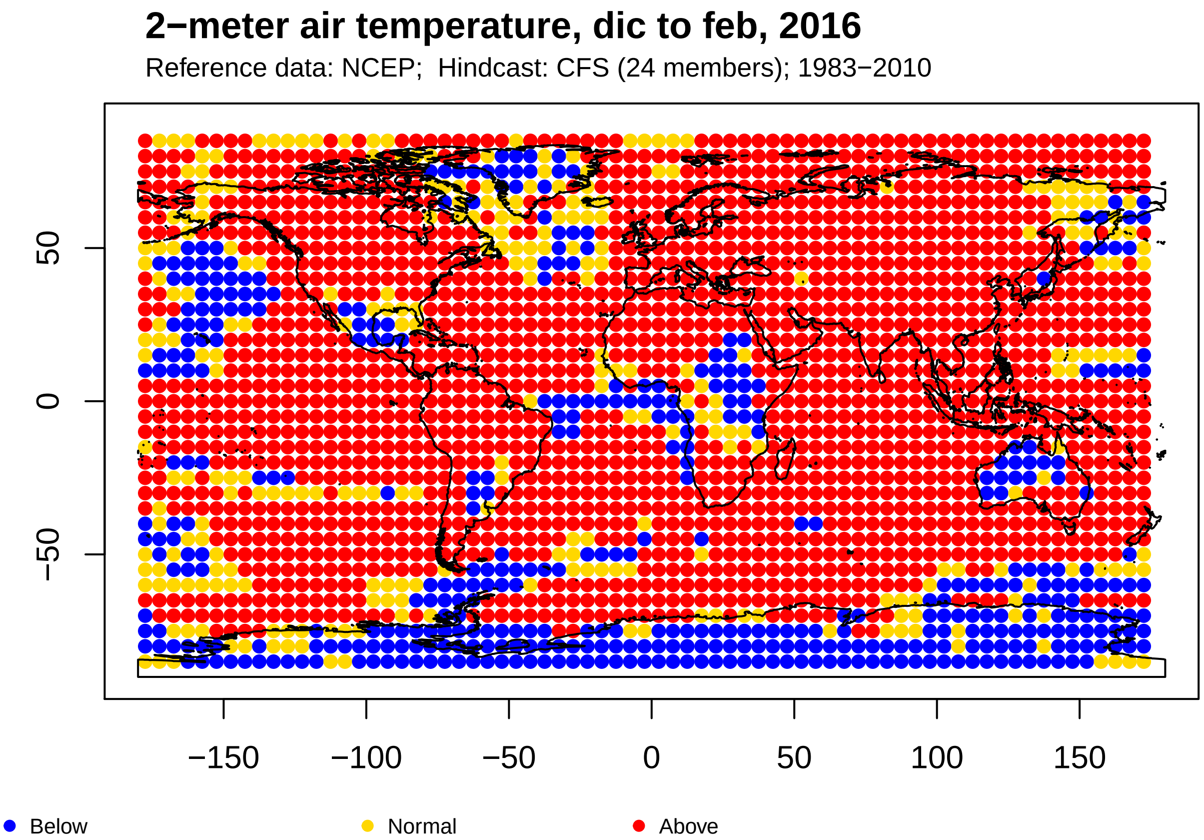
For convenience, a text string is generated based on the metadata information to use as subtitle in the different plots:

```
subtitle <- sprintf("Reference data: NCEP;  Hindcast: CFS (%d members); %d-%d",
                    length(hindcast$Members),
                    getYearsAsINDEX(hindcast)[1],
                    tail(getYearsAsINDEX(hindcast),1)
)
```

In its most basic setup, the only information provided is the most likely tercile, indicated by the color of the bubble:

```
# Only colour of the bubble is plotted indicating the most likely tercile
bubblePlot(hindcast, obs, forecast = forecast,
           bubble.size = 1.5,
```

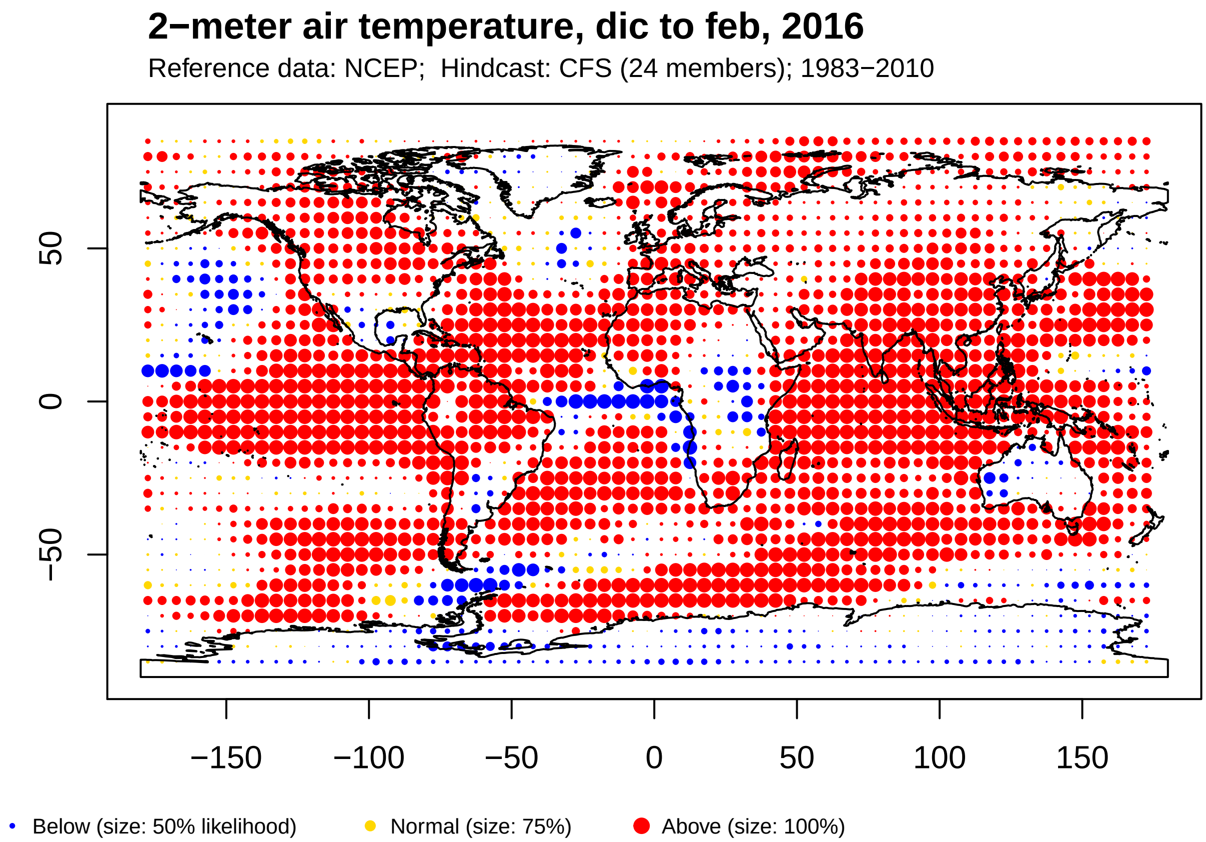
```
            subtitle = subtitle,
            size.as.probability = FALSE,
            score = FALSE
)
```



Note that by default, red indicates the upper tercile and blue the lower (yellow is reserved for the mid tercile). This default behaviour can be reversed for precipitation for a more intuitive interpretation of the plot, or other colors can be alternatively chosen for each tercile (via the argument `t.color`).

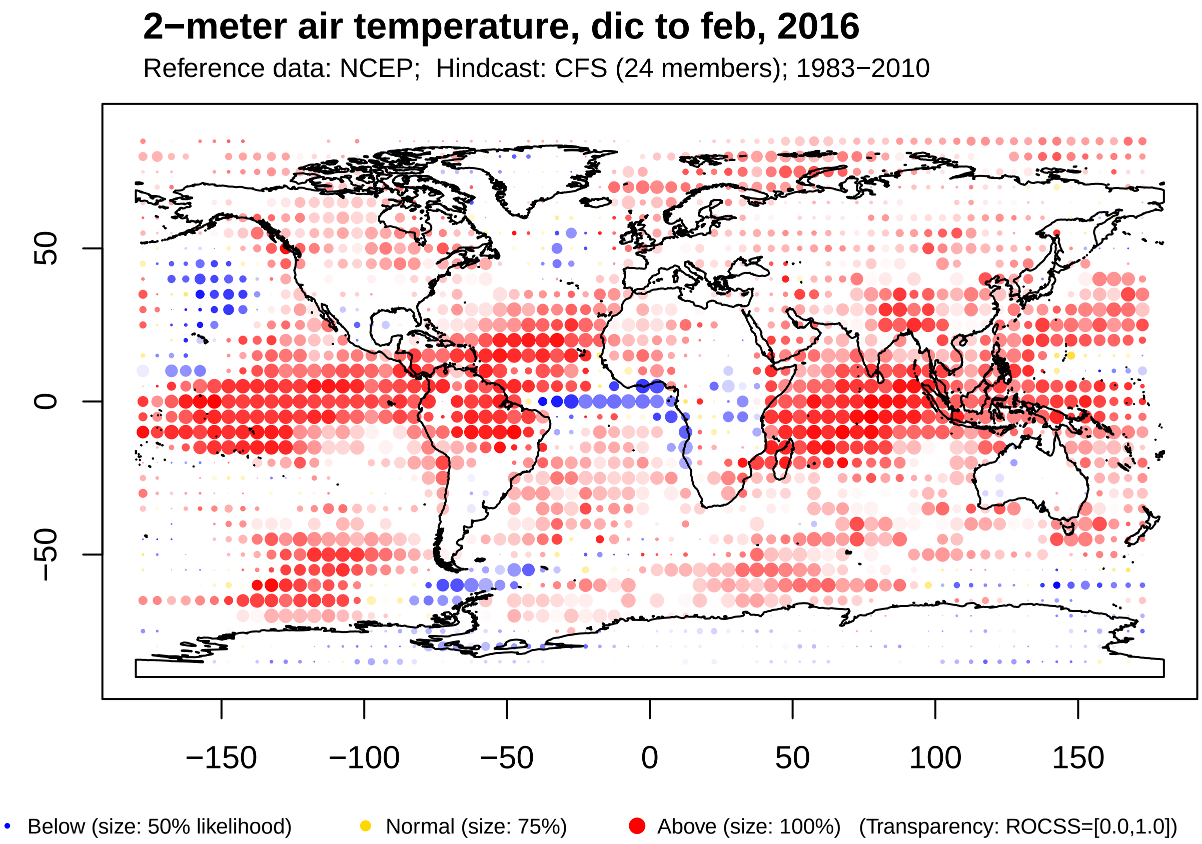
An additional information that can be added to the plot is the probability of the most likely tercile. This is represented by the size of the bubble through the argument `size.as.probability`:

```
bubblePlot(hindcast, obs, forecast = forecast,
            bubble.size = 1.5,
            subtitle = subtitle,
            size.as.probability = TRUE,
            score = FALSE
)
```



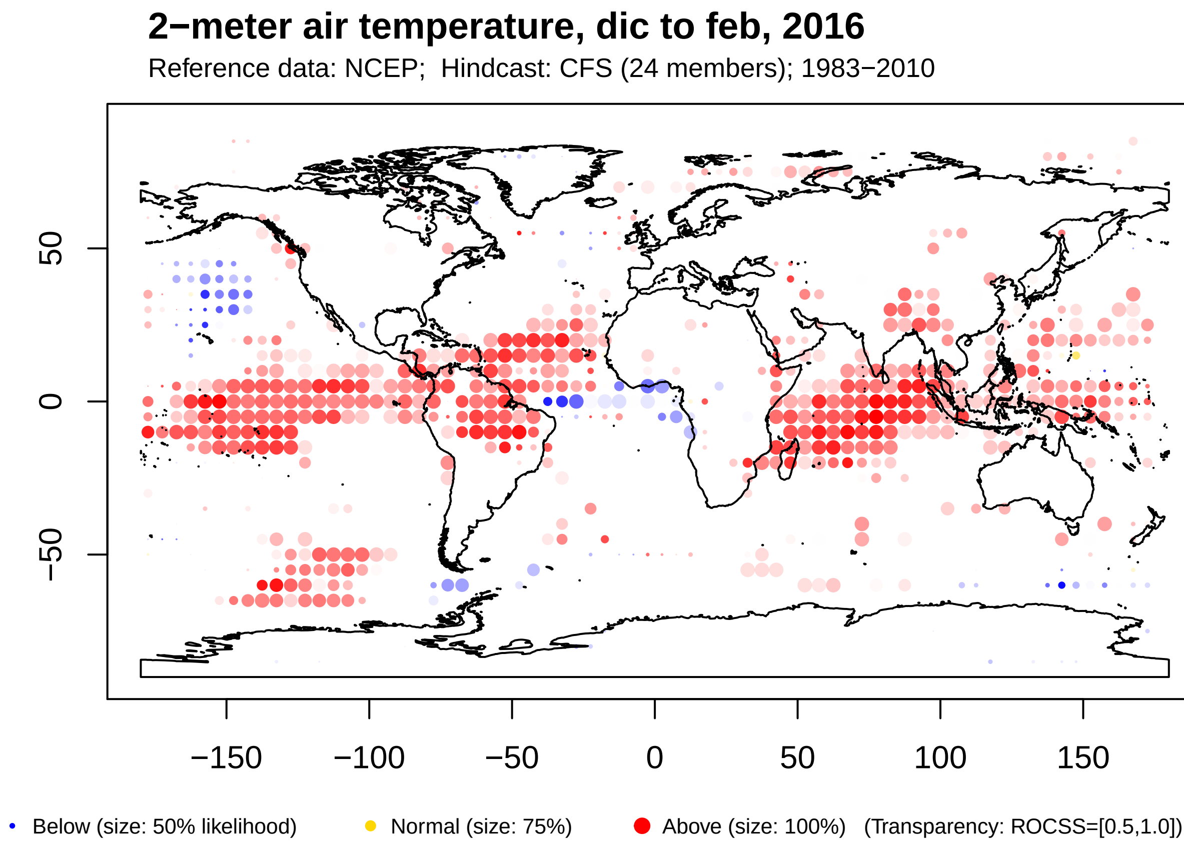
Until now, the forecast information is displayed, but no information regarding the 'quality' of the forecasting system is provided. A usual quality measure of accuracy is the ROC skill score (ROCSS). ROCSS can be also indicated in the plot, using to this aim different levels of transparency proportional to the ROCSS value in each grid point. This is automatically done by setting the argument `score` to `TRUE`:

```
bubblePlot(hindcast, obs, forecast = forecast,
           bubble.size = 1.5,
           subtitle = subtitle,
           size.as.probability = TRUE,
           score = TRUE
)
```



Note that many of the grid points correpsond to areas where the forecasting system exhibits few or no skill at all. It is often desirable to mask the areas where the forecast can not be trusted and focus on those where the user can have more confidence. This can also help to communicate the forecast to users with low level of expertise, so they don't focus their attention in unreliable areas. A ROCSS `score.range` can be set to this aim:

```
bubblePlot(hindcast, obs, forecast = forecast,
           bubble.size = 1.5,
           subtitle = subtitle,
           size.as.probability = TRUE,
           score = TRUE,
           score.range = c(0.5, 1)
)
```

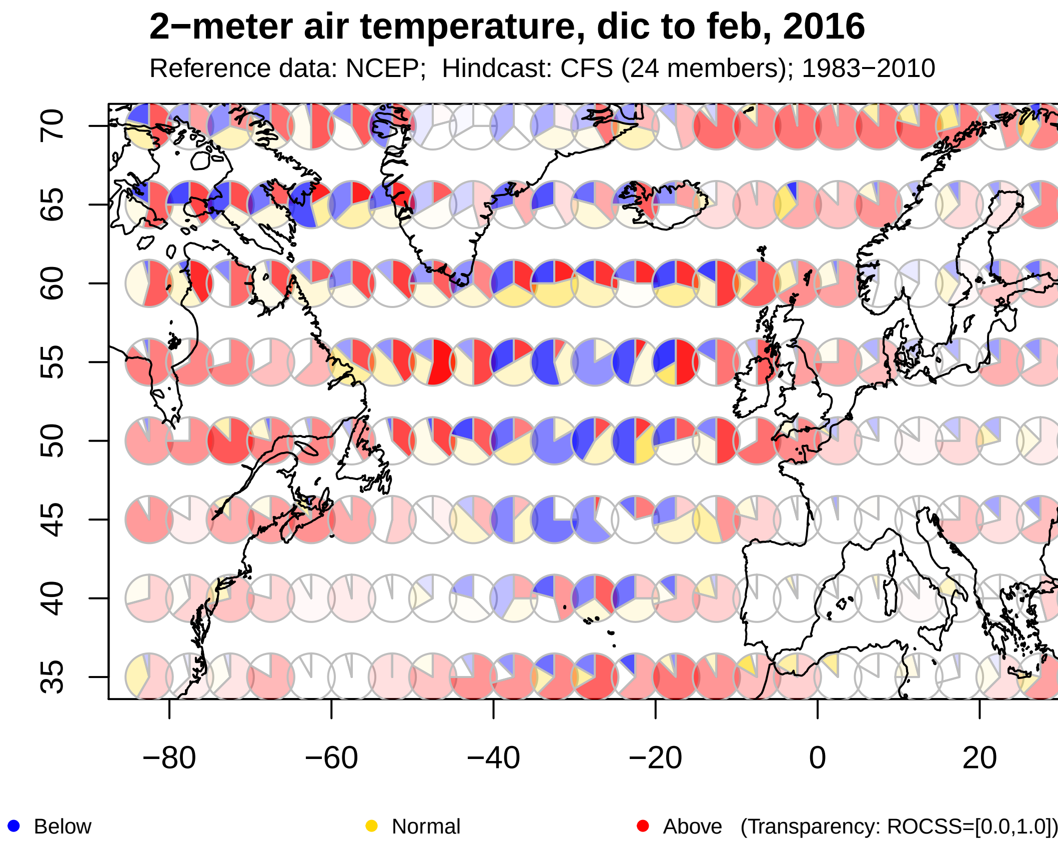


Finally, it is also possible to display the information for all terciles simultaneously. The probabilies of each tercile are in this case represented with a tercile plot. In order to avoid a congested map with hundreds of tiny pie charts, this visualization is better suited for regional domains. Next, we subset the global datasets for the North Atlantic domain:

```
# Cropping the North Atlantic region
crop.natl <- function(x) subsetGrid(x, lonLim = c(-80, 42), latLim = c(35, 72))
hindcast.natl <- crop.natl(hindcast)
forecast.natl <- crop.natl(forecast)
obs.natl <- crop.natl(obs)
```

The bubble plots with 3-piece pie charts are generated with the option `piechart = TRUE`.

```
bubblePlot(hindcast.natl, obs.natl, forecast = forecast.natl,
           bubble.size = 1.5,
           subtitle = subtitle,
           piechart = TRUE,
           score = TRUE
)
```



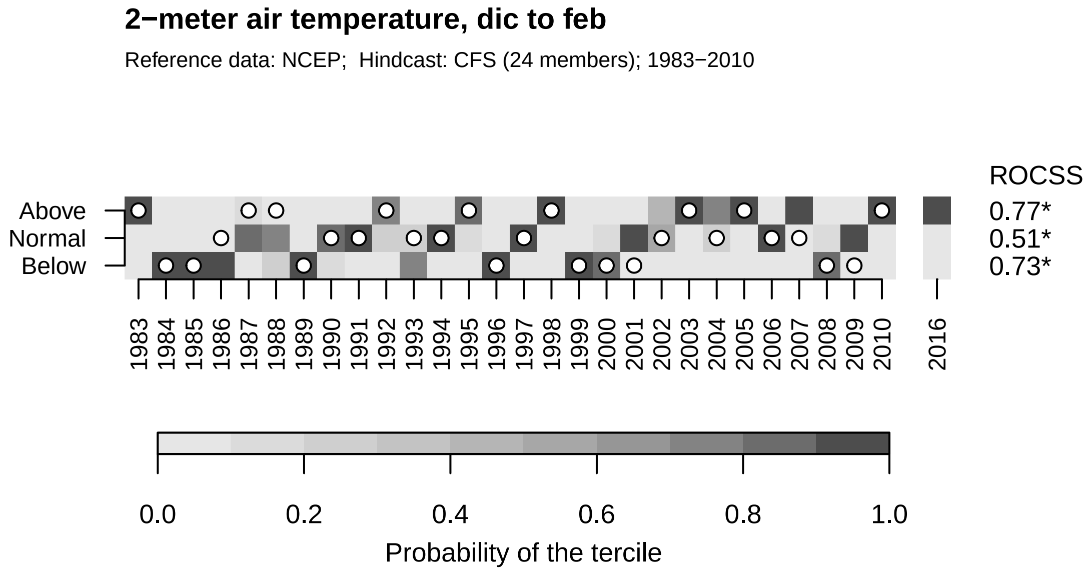
Tercile plots

To reproduce Figure 3 in Frías et al 2018, we firt crop the data considering the Niño 3.4 Region:

```
crop.nino <- function(x) subsetGrid(x, lonLim = c(-170, -120), latLim = c(-5, 5))
hindcast.nino <- crop.nino(hindcast)
obs.nino <- crop.nino(obs)
forecast.nino <- crop.nino(forecast)
```

The tercile plot is next produced:

```
tercilePlot(hindcast.nino, obs.nino, forecast = forecast.nino, subtitle = subtitle)
```



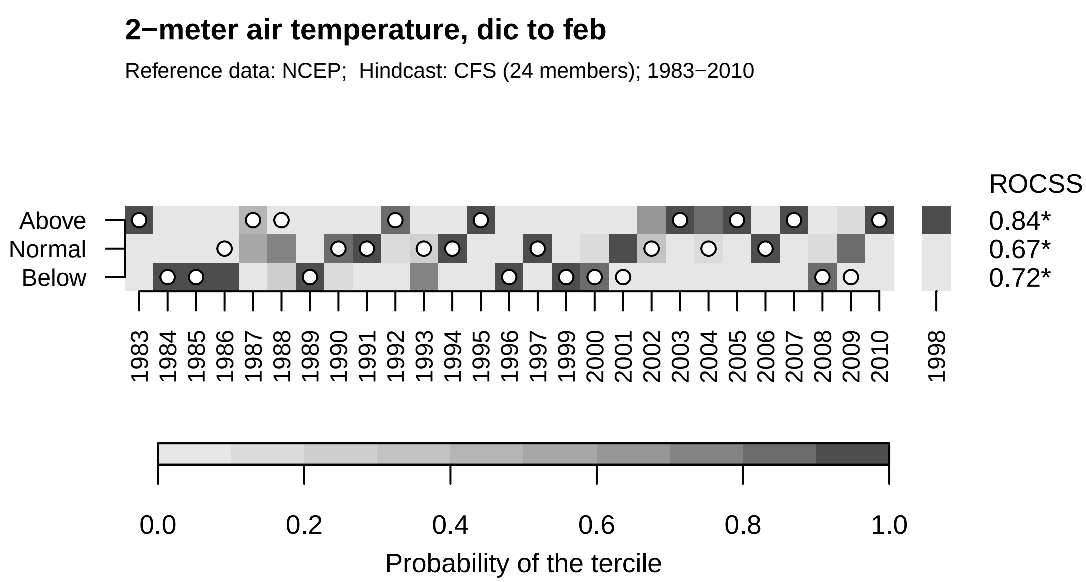
It is also possible to use tercile plots considering the forecast of a specific year of the forecast. For instance, in this case the `forecast` argument is set to `NULL` (the default), and we select year 1998 of the hindcast as the forecast (`year.target = 1998`):

```
tercilePlot(hindcast.nino, obs.nino, forecast = NULL, year.target = 1998, subtitle = subtitle)
```

```
## Warning in spatialMean(hindcast): 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
```

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



Tercile bar plots

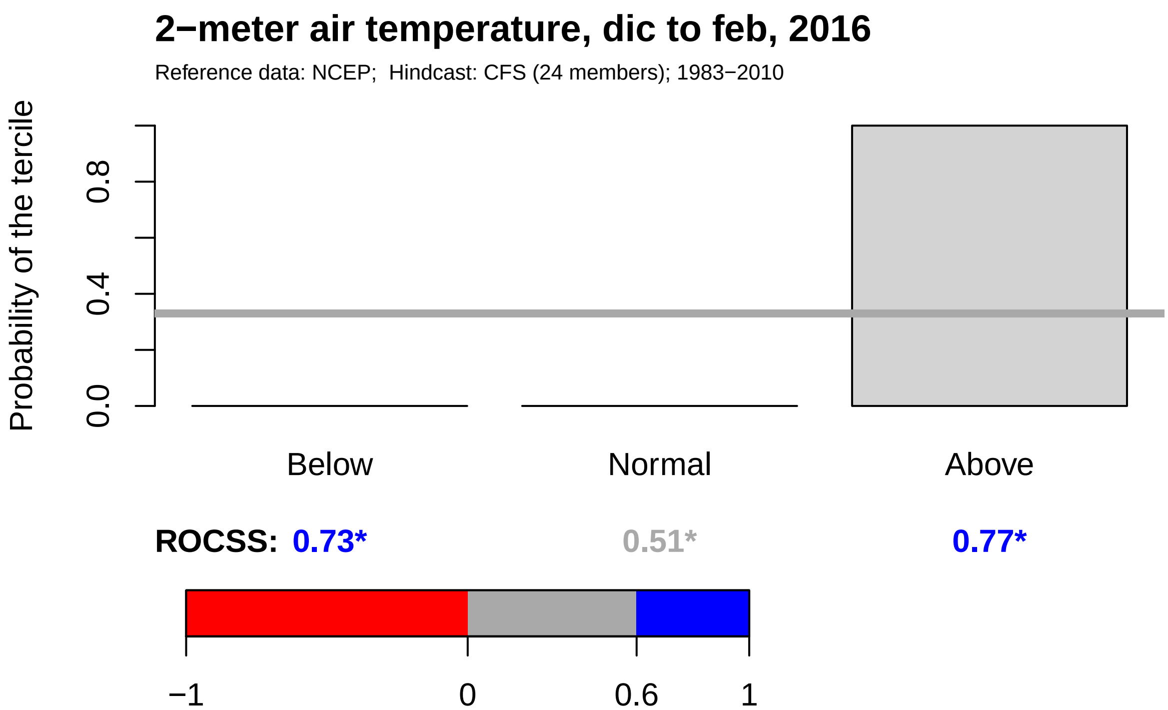
This code reproduces Figure 4 in Frías et al. 2018

```
# Plot for winter 2016 (forecast data)
tercileBarplot(hindcast.nino, obs.nino, forecast = forecast.nino, score.threshold = 0.6,
               subtitle = subtitle)

## Warning in spatialMean(hindcast): 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

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

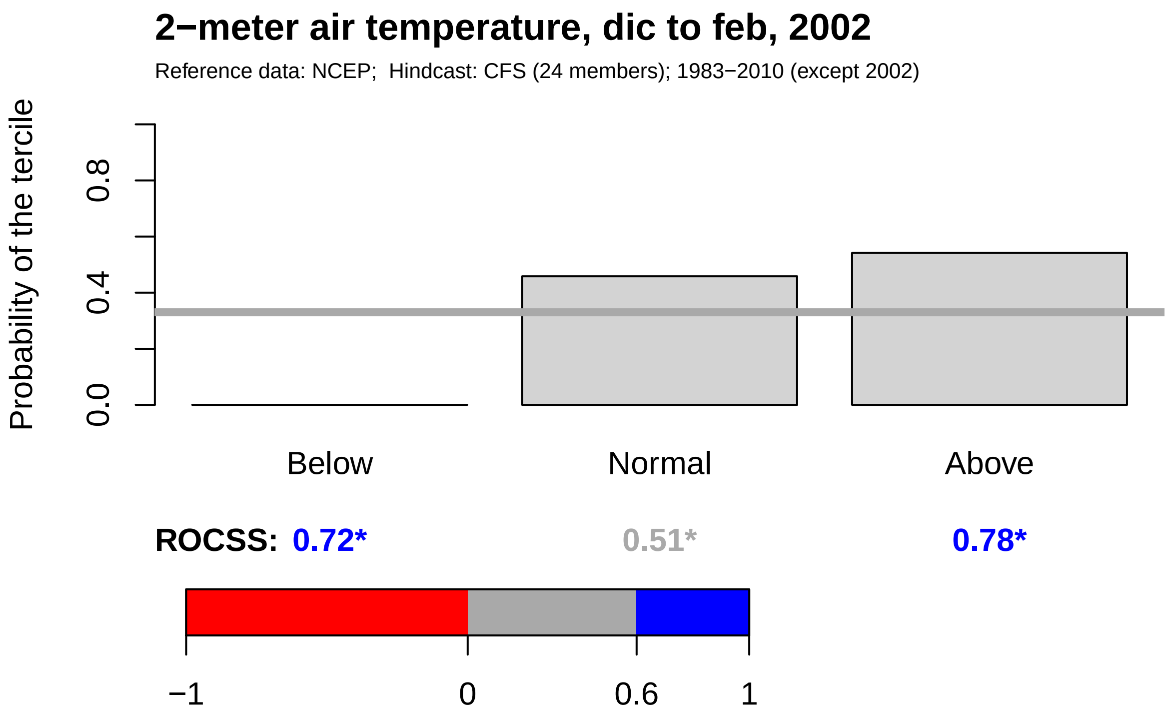



```
# Plot for winter 2002 (selected from the hindcast)
year.target <- 2002
subtitle_year.target <- sprintf("Reference data: NCEP;  Hindcast: CFS (%d members); %d-%d (except %d)",
                                length(hindcast$Members), getYearsAsINDEX(hindcast)[1],
                                tail(getYearsAsINDEX(hindcast),1), year.target)
tercileBarplot(hindcast.nino, obs.nino, year.target = year.target, score.threshold = 0.6,
               subtitle = subtitle_year.target)

## Warning in spatialMean(hindcast): 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

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



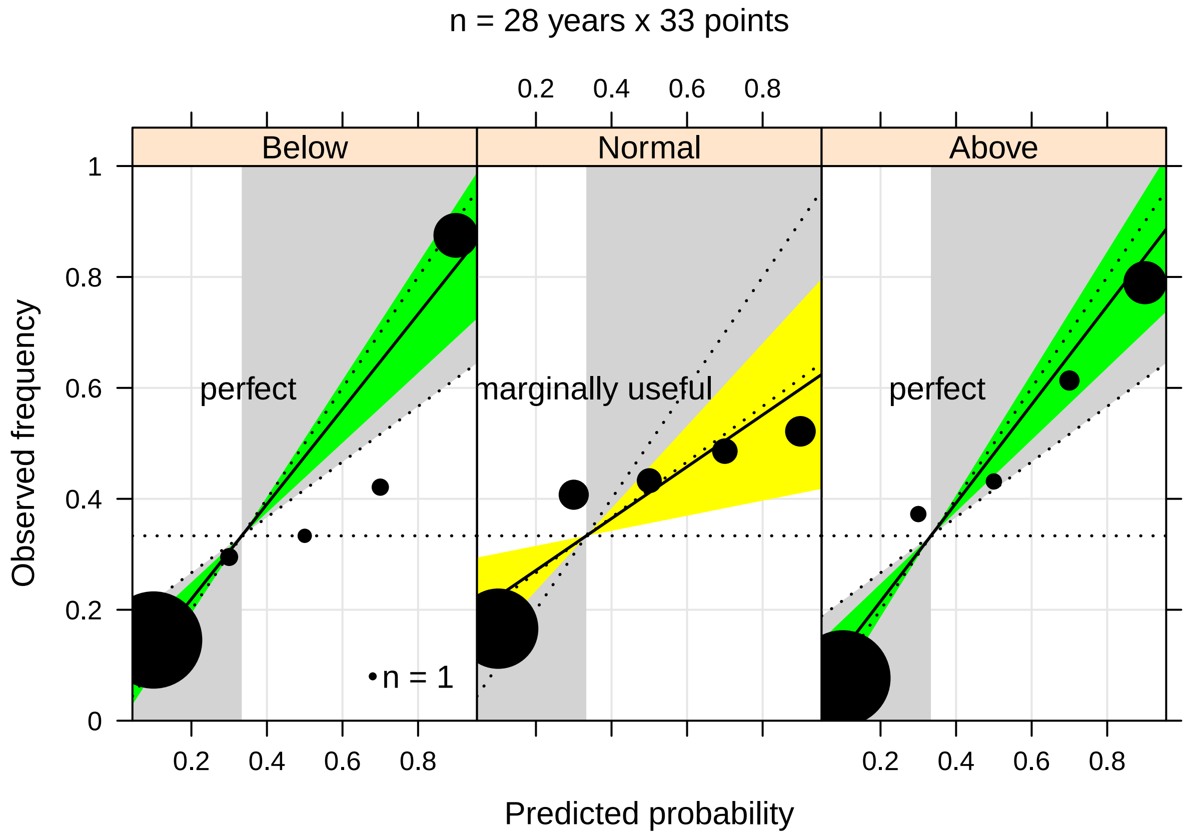
Reliability categories

Reliability assessment in `visualizeR` is implemented in the function `reliabilityCategories`. It computes reliability categories for probabilistic forecasts following the implementation described in Weisheimer *et al.* 2014, and the modifications later proposed by Manzanas *et al.* 2017, including the bootstrapping procedure to provide confidence intervals for the slope of the reliability line.

The code below reproduces Figure 5 in Frías *et al.* 2018, representing the classical reliability diagrams:

```
rl.nino <- reliabilityCategories(hindcast = hindcast.nino,
                                  obs = obs.nino,
                                  n.events = 3, labels = c("Below", "Normal", "Above"), n.bins = 5,
                                  n.boot = 1000, conf.level = 0.9,
                                  cex0 = 0.5, cex.scale = 20
)
```

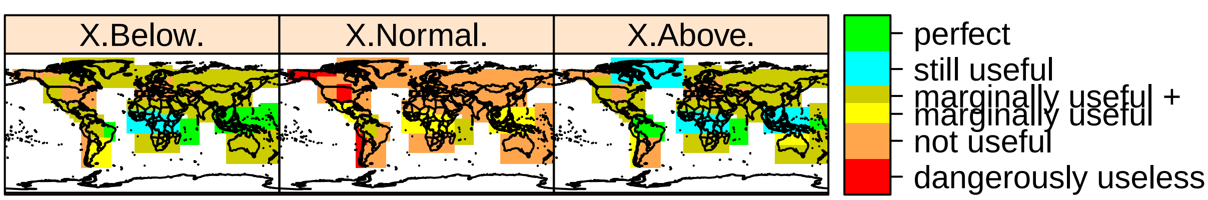
```
## [2018-04-25 16:56:00] Calculating categories for region 1 out of 1
## ...[2018-04-25 16:56:01] Computing bootstrapping...
## ...[2018-04-25 16:56:43] Done.
## [2018-04-25 16:56:44] Done.
```



It is also possible to depict spatially the reliability, through aggregation by user-defined regions. In this example (Fig. 6 in Frías *et al*), a vector layer with the polygons defining the IPCC AR5 regions is used to summarize the results by regions:

```
rl.map <- reliabilityCategories(hindcast, obs,
                                 n.events = 3, labels = c("Below", "Normal", "Above"), n.bins = 5,
                                 n.boot = 1000, conf.level = 0.9,
                                 regions = AR5regions)
```

```
## Warning: 'plotClimatology' is deprecated and will eventually be removed from transformeR.
## Use 'spatialPlot' from package visualizeR instead.
```



Spread plots

This type of visualization was initially meant for daily data. As a result, the built-in datasets in `visualizeR` (monthly data) can not be used. Instead, the necessary data to reproduce the spread plots in Frías *et al.* 2018 can be retrieved from the ECOMS User Data Gateway (Cofiño *et al.* 2018). The code to download the data from the ECOMS User Data Gateway (ECOMS-UDG) is shown in the next subsection.

NOTE: that data loading from the ECOMS-UDG may take up to a few hours depending on several factors (network traffic, temporal and spatial resolution, etc). Daily data for this section can be also retrieved from the remote URLs indicated below. If this is the case, you can skip the *UDG data access* section and go directly to the *Spread plots* subsection.

CFSv2 hindcast data can be loaded from:

```
load(url("http://meteo.unican.es/work/visualizeR/data/tas.cfs.dly.rda"), verbose = TRUE)
```

```
## Loading objects:
##   tas.cfs.dly
```

And the CFSV2 operative forecast data from:

```
load(url("http://meteo.unican.es/work/visualizeR/data/tas.cfs.operative.dly.2016.rda"), verbose = TRUE)
```

```
## Loading objects:
##   tas.cfs.operative.dly.2016
```

UDG data access

Accessing the UDG requires previous registration via TAP. See Cofiño *et al.* 2018 for further details, or visit the corresponding paper notebook.

The version of the packages used to reproduce the results of this manuscript are next installed (note that updated versions are currently available that may work as well).

```
devtools::install_github(c("SantanderMetGroup/loadeR.java@v1.1-0",
                           "SantanderMetGroup/loadeR@v1.1.0",
                           "SantanderMetGroup/loadeR.ECOMS@v1.3.1"))
```

```
library(loadeR.ECOMS)
# UDG Authentification
loginUDG(username = "jdoe", password = "******") # Note username and password are provided after UDG re
# Load reanalysis
tas.ncep.dly <- loadECOMS(dataset = "NCEP_reanalysis1",
                          var = "tas",
                          years = 1983:2010,
                          season = c(12, 1, 2),
                          time = "DD",
                          aggr.d = "mean",
                          lonLim = c(-170, -120),
                          latLim = c(-5, 5))
# Load hindcast
tas.cfs.dly <- loadECOMS(dataset = "CFSv2_seasonal",
                         var = "tas",
                         years = 1983:2010, season = c(12, 1, 2), time = "DD", aggr.d = "mean",
                         lonLim = c(-170, -120), latLim = c(-5, 5),
                         leadMonth = 1, members = 1:24     # note 'leadMonth' and 'members'
)
# Load operational predictions for winter 2016
tas.cfs.operative.dly.2016 <- loadECOMS(dataset = "CFSv2_seasonal_operative",
                                        var = "tas",
                                        years = 2016, season = c(12, 1, 2), time = "DD", aggr.d = "mean
                                        lonLim = c(-170, -120), latLim = c(-5, 5),
```

```
                                          leadMonth = 1, members = 1:24      # note 'leadMonth' and 'membe
)
```

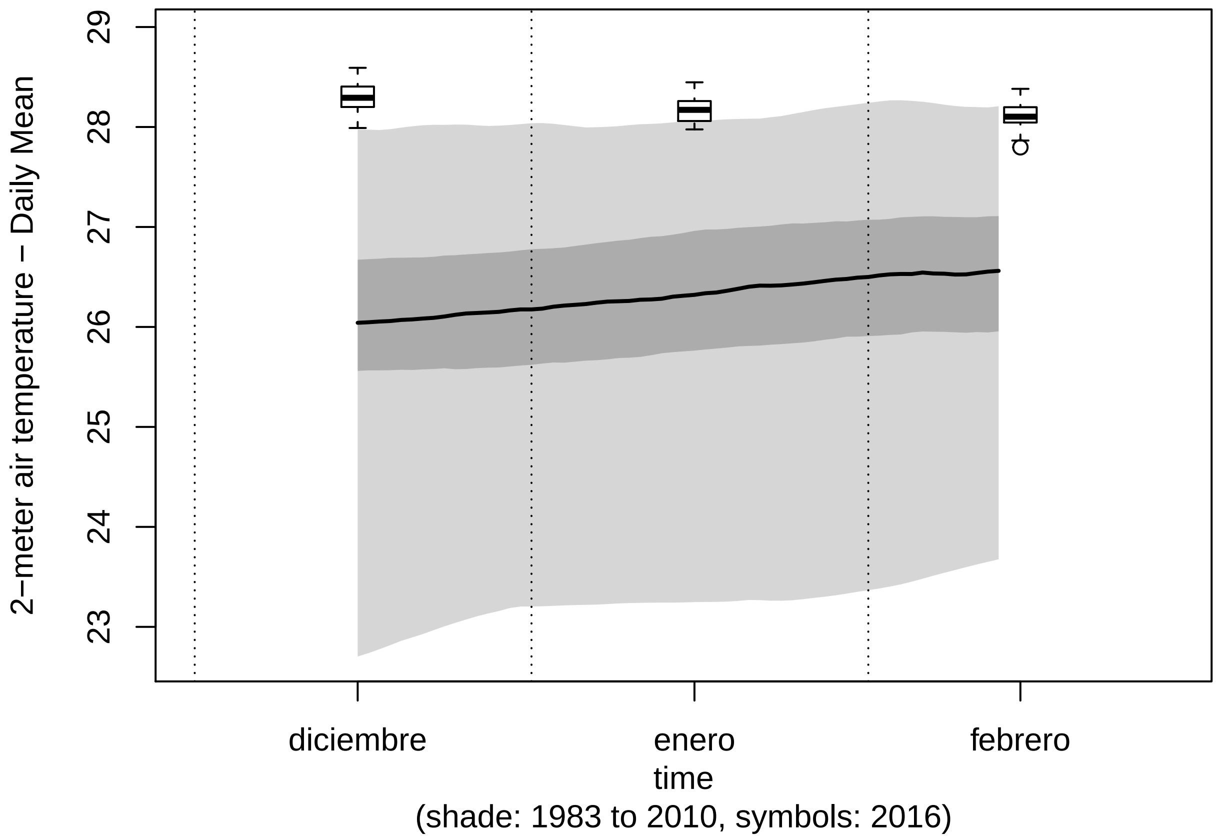
Spread plots

These are the resulting spread plots depicted in Fig. 7 in Frías *et al.*:

```
# Plot for winter 2016 (forecast data)
spreadPlot(tas.cfs.dly, forecast = tas.cfs.operative.dly.2016, boxplot = TRUE)
```

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

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

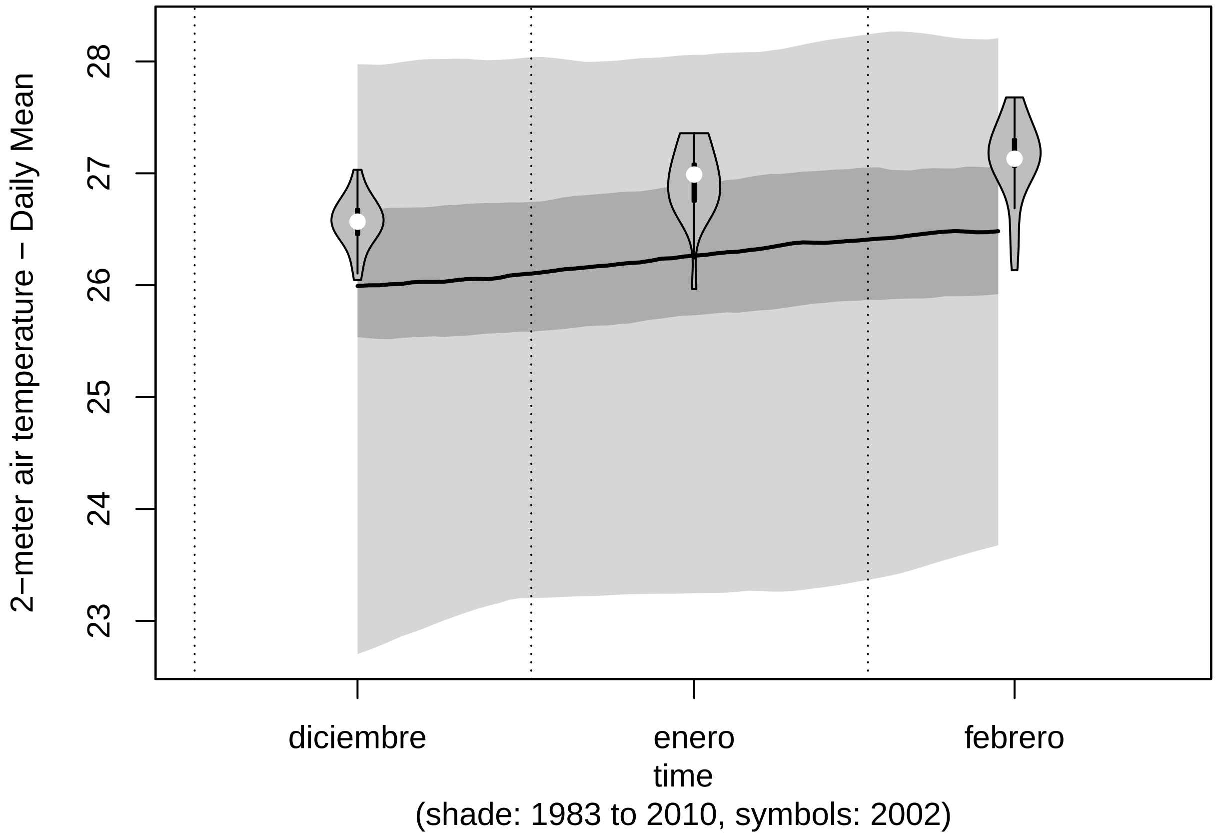


As in the case of tercile plots, it is also possible to extract a year from the hindcast and use it as forecast with the `year.target` argument:

```
# Plot for winter 2002 (year selected from the hindcast)
spreadPlot(tas.cfs.dly, year.target = 2002, violin = TRUE)
```

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

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



References

- Cofiño, A.S., Bedia, J., Iturbide, M., Vega, M., Herrera, S., Fernández, J., Frías, M.D., Manzanas, R., Gutiérrez, J.M., 2018. The ECOMS User Data Gateway: Towards seasonal forecast data provision and research reproducibility in the era of Climate Services. Climate Services 9, 33–43. https://doi.org/10.1016/j.cliser.2017.07.001
- Frías, M.D., Iturbide, M., Manzanas, R., Bedia, J., Fernández, J., Herrera, S., Cofiño, A.S., Gutiérrez, J.M., 2018. An R package to visualize and communicate uncertainty in seasonal climate prediction. Environmental Modelling & Software 99, 101–110. https://doi.org/10.1016/j.envsoft.2017.09.008
- Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., Zhu, Y., Leetmaa, A., Reynolds, R., Chelliah, M., Ebisuzaki, W., Higgins, W., Janowiak, J., Mo, K.C., Ropelewski, C., Wang, J., Jenne, R., Joseph, D., 1996. The NCEP/NCAR 40-Year Reanalysis Project. Bulletin of the American Meteorological Society 77, 437–471. https://doi.org/10.1175/1520-0477(1996)077<0437:TNYRP>2.0.CO;2
- Manzanas, R., Lucero, A., Weisheimer, A., Gutiérrez, J.M., 2017. Can bias correction and statistical downscaling methods improve the skill of seasonal precipitation forecasts? Climate Dynamics, pp 1-16, doi:10.1007/s00382-017-3668-z
- Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y.-T., Chuang, H., Iredell, M., Ek, M., Meng, J., Yang, R., Peña Mendez, M., van den Dool, H., Zhang, Q., Wang, W., Chen, M., Becker, E., 2013. The NCEP Climate Forecast System Version 2. J Clim 130925135638001. https://doi.org/10.1175/JCLI-D-12-00823.1
- Weisheimer, A., Palmer, T.N., 2014. On the reliability of seasonal climate forecasts. Journal of The Royal Society Interface 11, 20131162. doi:10.1098/rsif.2013.1162

Session information

```r
print(sessionInfo(package = c("visualizeR", "transformeR")))
```

```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.4 LTS
##
## Matrix products: default
## BLAS: /usr/lib/openblas-base/libblas.so.3
## LAPACK: /usr/lib/libopenblasp-r0.2.18.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=es_ES.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=es_ES.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## character(0)
##
## other attached packages:
## [1] visualizeR_1.2.0  transformeR_1.3.3
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.16           plyr_1.8.4
##  [3] compiler_3.4.4         RColorBrewer_1.1-2
##  [5] methods_3.4.4          bitops_1.0-6
##  [7] utils_3.4.4            tools_3.4.4
##  [9] grDevices_3.4.4        boot_1.3-20
## [11] digest_0.6.12          dotCall64_0.9-5.2
## [13] vioplot_0.2            evaluate_0.10.1
## [15] lattice_0.20-35        Matrix_1.2-14
## [17] yaml_2.1.18            parallel_3.4.4
## [19] spam_2.1-3             akima_0.6-2
## [21] padr_0.4.0             stringr_1.3.0
## [23] knitr_1.20             raster_2.6-7
## [25] mapplots_1.5           graphics_3.4.4
## [27] datasets_3.4.4         stats_3.4.4
## [29] fields_9.6             maps_3.3.0
## [31] rprojroot_1.3-2        grid_3.4.4
## [33] data.table_1.10.4-3    base_3.4.4
## [35] dtw_1.18-1             sm_2.2-5.4
## [37] SpecsVerification_0.5-2 rmarkdown_1.9
## [39] sp_1.2-7               latticeExtra_0.6-28
## [41] magrittr_1.5           codetools_0.2-15
## [43] scales_0.5.0           backports_1.1.2
## [45] htmltools_0.3.6        CircStats_0.2-4
## [47] MASS_7.3-49            abind_1.4-5
## [49] colorspace_1.3-2       stringi_1.1.7
## [51] proxy_0.4-19           munsell_0.4.3
## [53] RCurl_1.95-4.10        verification_1.42
```

```
## [55] RcppEigen_0.3.3.4.0
```