Bias-corrected future SPEI projections with climate4R

A worked example using Euro-CORDEX projections

J. Bedia

2018-07-26

Abstract

This is a worked example on how to obtain bias-corrected future drought index projections using the climate4R framework for climate data access and analysis. The example is based on RCM data from EURO-CORDEX (Jacob *et al.* 2014), using the CRU-TS4 gridded observations (Harris and Jones 2017) as reference. The example describes the main steps to obtain the future SPEI projections, as used in related paper by Turco *et al.* (2018). This notebook is also available in the following link http://www.meteo.unican.es/work/climate4r/drought4R/drought4R_notebook.html

Contents

1 Introduction and required R packages

1.1 The *climate4R* framework

climate4R (Iturbide et al., submitted) is a bundle of R packages for transparent climate data access, post-processing (including bias correction and downscaling) and visualization. climate4R builds on two main data structures (grid and station, including metadata) to deal with gridded and point data from observations, reanalysis, seasonal forecasts and climate projections. Thus, it considers ensemble members as a basic dimension of the data structures. climate4R exploits NetCDF-Java and allows accessing local and remote (OPeNDAP) data sources, including the User Data Gateway (UDG), a THREDDS-based service for a variety of widely used datasets (e.g. reanalysis, CMIP5, CORDEX).

This provides a unique comprehensive framework for end-to-end sectoral applications, favouring the reproducibility of scientific outcomes (see e.g. Cofiño et al. 2018).

Compatibility with some external packages has been achieved by appropriate two-way bridging functions, enhancing climate4R with sector-specific functionalities: model validation, extreme climate indices, etc. In this notebook we illustrate the extension of climate4R to calculate drought indices. In particular, here we use `drought4R`, a wrapping package tailoring the SPI, SPEI and evapotranspiration routines for monthly data implemented in the R package `SPEI` (Beguería and Serrano 2017) to the needs of climate4R users handling large climate model datasets (multi-decadal projections, seasonal predictions…) and gridded observations or reanalysis.

1.2 Package availability

All the required packages are publicly available through the Santander MetGroup GitHub repository. Further details and access points can be obtained in the climate4R site

climate4R is formed by four main packages `loadeR` (Cofiño et al. 2018), `transformeR` (Iturbide et al., submitted), `downscaleR` (Bedia et al. 2018) and `visualizeR` (Frías et al. 2018), performing the tasks of data loading, manipulation and transformation, bias correction and downscaling and data visualization respectively. These are next loaded:

Next, installation procedure is shown. The use of the `install_github` function from package `devtools` (Wickham et al. 2018) is recommended. Note that the installation paths point to the stable versions used to produce this notebook. The full reproducibility of the results can't be guaranteed with other versions.

```
devtools::install_github("SantanderMetGroup/loadeR.java",
                         "SantanderMetGroup/loadeR@v1.4.5",
                         "SantanderMetGroup/transformeR@v1.4.3",
                         "SantanderMetGroup/downscaleR@v3.0.1",
                         "SantanderMetGroup/visualizeR@v1.2.1")
```

In addition, two “add-on” packages seamlessly integrated in climate4R are used for SPEI calculation (`drought4R`) and unit conversion (`convertR`). The latter relies on the Unidata's UDUNITS software libraries for handling physical quantity units. Specific installation instructions are available in their respective GitHub sites in case any problems arise:

```
devtools::install_github("SantanderMetGroup/drought4R@v0.1.0",
                         "SantanderMetGroup/convertR@v0.1.2")
```

Finally, the pipe operator (`%>%`) in package `magrittr` (Bache and Wickham 2014) will be used throughout the examples to concatenate command calls in a convenient way (the package is available in CRAN and can be installed using the R base function `install.packages`):

```
library(magrittr)
```

2 Data access

2.1 Baseline observations (CRU 4.0, period 1971-2000)

The CRU TS gridded observations (Harris and Jones 2017) can be downloaded from the data access site. For brevity, the process to prepare these raw data to be accessible using `loadeR` is skipped (the interested reader has worked examples in other sources, see e.g. Iturbide et al. (in press), or the examples in the loadeR's wiki.

Thus, the data have been already prepared in a convenient format to be used in this example, and are available on-line. The following function can be used for a straightforward download into the current R session:

```
my_readRDS <- function(file.url) {
    tmpfile <- tempfile()
    utils::download.file(url = file.url, destfile = tmpfile)
    readRDS(tmpfile)
}
```

We next load the three observational datasets used:

```
tasmin.cru <- my_readRDS("http://www.meteo.unican.es/work/UDG/drought4R/CRU_tasmin.rds")
tasmax.cru <- my_readRDS("http://www.meteo.unican.es/work/UDG/drought4R/CRU_tasmax.rds")
pr.cru <- my_readRDS("http://www.meteo.unican.es/work/UDG/drought4R/CRU_tpr.rds")
```

The reference grid of the CRU data is retained in order to interpolate the RCM data afterwards:

```
library(transformeR)
```

```
## transformeR version 1.4.4 (2018-07-19) is loaded

## WARNING: Your current version of transformeR (v1.4.4) is ahead of the master branch version (1.4.3

## Development version may have an unexpected behaviour
```

```
ref.grid <- getGrid(pr.cru)
```

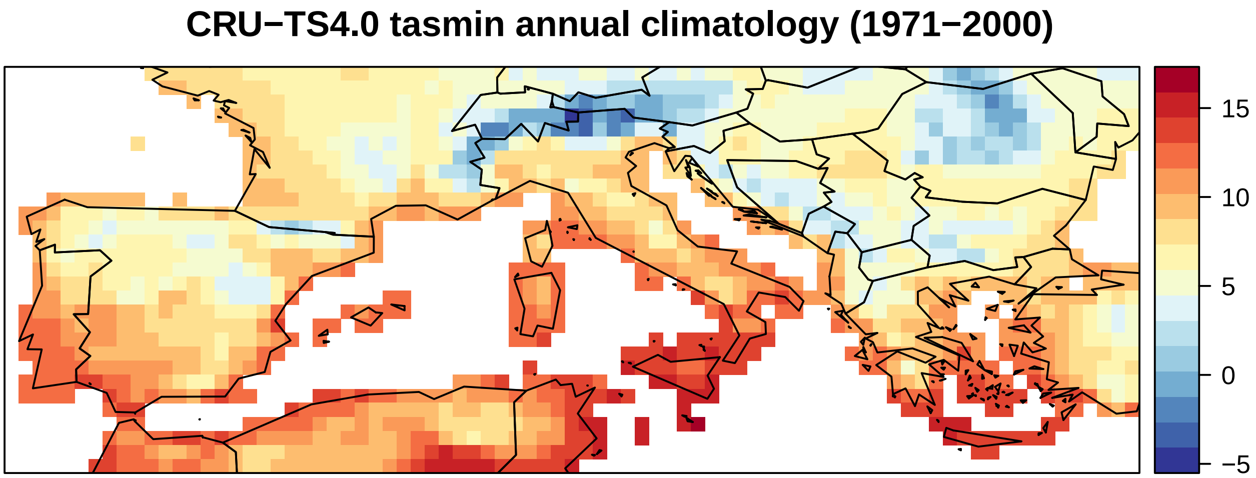
Next, the climatology of one of the reference observations just loaded is shown:

```
library(visualizeR)
```

```
## Loading required package: sm

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

## visualizeR version 1.2.1 (2018-07-19) is loaded

##
## Attaching package: 'visualizeR'

## The following objects are masked from 'package:transformeR':
##
##     clim2sgdf, map.lines, map.stippling
```

```
climatology(tasmin.cru) %>% spatialPlot(backdrop.theme = "countries",
                                        rev.colors = TRUE,
                                        main = "CRU-TS4.0 tasmin annual climatology (1971-2000)"
```

```
## [2018-07-26 11:05:36] - Computing climatology...

## [2018-07-26 11:05:36] - Done.
```



2.2 Accessing Euro-CORDEX data from the User Data Gateway (and bias correction)

In this section we illustrate the use of climate4R as the data access layer of the UDG. The added value of using the `loadeR` package tools for data access directly pointing to the UDG server are here exemplified.

2.2.1 Historical experiment data (1971-2000)

The UDG service requires (free) registration to accept the data policies of the different data providers (http://www.meteo.unican.es/udg-wiki). Prior to data access, authentication with valid UDG credentials is required for the current R session in order to access the UDG. Once a valid user name and password have been issued, the authentication can be done in one step within the R session using the `loginUDG` function from `loadeR`:

```
library(loadeR)
```

```
## Loading required package: rJava

## Loading required package: loadeR.java

## Java version 1.8x amd64 by Oracle Corporation detected

## NetCDF Java Library v4.6.0-SNAPSHOT (23 Apr 2015) loaded and ready

## loadeR version 1.4.5 (2018-07-13) is loaded
```

```
loginUDG(username = "jdoe", password = "****")
```

To get a quick overview of the available datasets, the function `UDG.datasets` prints a summary. For all datasets included in the UDG and listed by the function, the name of the target dataset can be used as a valid entry for the argument dataset in `loadGridData`, instead of the full URL. Next, data from the CORDEX historical scenario available at UDG are loaded. In this example, for simplicity and faster calculations, the 0.44 regular degree grid will be used (note that the 0.11º simulations are also available at UDG).

```
models <- UDG.datasets()
```

There are many available datasets. Here, for illustration we will use one of the GCM/RCM couplings of Euro-CORDEX used by Turco et al. 2018, in particular, the EC-Earth/RACMO22. However, for brevity and minimal computational cost, the 0.44 degree horizontal resolution will be used, instead of the 11 degree used in the paper. Note that the latter would be accessed just by replacing the "EUR44"" prefix by "EUR11". Using pattern matching we identify the dataset:

```
grep("EUR44.*EC-EARTH.*RACMO22", models$name, value = TRUE)
```

```
## [1] "CORDEX-EUR44_EC-EARTH_r1i1p1_historical_RACMO22E_v1"
## [2] "CORDEX-EUR44_EC-EARTH_r1i1p1_rcp45_RACMO22E_v1"
## [3] "CORDEX-EUR44_EC-EARTH_r1i1p1_rcp85_RACMO22E_v1"
```

The following call to the function `loadGridData` retrieves the historical simulation considering all the months of the year (`season = 1:12`, is omitted because it is the default), minimum temperature (`var = "tasmin"`), considering a Euro-Mediterranean spatial domain (`lonLim`and `latLim`) for the period 1971-2000 (argument `years`). Furthermore, monthly mean data is requested, via the corresponding temporal aggregation arguments. More details in `help("loadGridData", package = "loadeR")`

```
lonLim <- c(-10, 30)
latLim <- c(34, 48)
```

```
tasmin.hist <- loadGridData(dataset = "CORDEX-EUR44_EC-EARTH_r1i1p1_historical_RACMO22E_v1",
                            var = "tasmin",
                            lonLim = lonLim,
                            latLim = latLim,
                            years = 1971:2000,
                            time = "DD",
                            aggr.d = "mean",
                            aggr.m = "mean")
```

Next, we ensure that the data are in degress Celsius (model data are originally stored in Kelvin in most cases). To this aim, the function `udConvertGrid` from package `convertR` is used:

```
library(convertR)
tasmin.hist <- udConvertGrid(tasmin.hist, new.units = "degC")
```

Finally, the data are regridded to the reference regular grid from the CRU (the original one is a rotated grid). This is done using the `interpGrid` function from package `transformeR`:

```
tasmin.hist <- interpGrid(tasmin.hist, new.coordinates = ref.grid)
```

For brevity, in the following the above steps (data loading + unit conversion + regridding) will be concatenated using the `%>%` operator in the same call.

The function `biasCorrection` of the package `downscaleR` allows applying a number of standard bias correction techniques within the climate4R framework. In particular, when dealing with monthly data, the common bias correction technique is the (additive and/or multiplicative) local scaling method. It is next applied to the historical data, using as reference the CRU observations:

```
library(downscaleR)
```

```
## Loading required package: deepnet
## Loading required package: glmnet
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-16
## downscaleR version 3.0.2 (2018-07-12) is loaded
```

```
tasmin.hist.corr <- biasCorrection(tasmin.hist,
                                   y = tasmin.cru,
                                   method = "scaling",
                                   scaling.type = "additive") %>% redim(drop = TRUE)
```

The same steps are next undertaken with maximum temperature (`var = "tasmax"`):

```
tasmax.hist <- loadGridData(dataset = "CORDEX-EUR44_EC-EARTH_r1i1p1_historical_RACMO22E_v1",
                            var = "tasmax",
                            dictionary = dic,
                            lonLim = lonLim,
                            latLim = latLim,
                            years = 1971:2000,
                            time = "DD",
                            aggr.d = "mean",
                            aggr.m = "mean") %>% udConvertGrid(new.units = "degC") %>% interpGr
tasmax.hist.corr <- biasCorrection(tasmax.hist,
                                   y = tasmax.cru,
                                   method = "scaling",
                                   scaling.type = "additive") %>% redim(drop = TRUE)
```

and precipitation (`var = "pr"`). Note that the monthly aggregation function is now set to `aggr.m = "sum"` (i.e., total accumulated monthly precipitation), as opposite to `"mean"` monthly temperature:

```
pr.hist <- loadGridData(dataset = "CORDEX-EUR44_EC-EARTH_r1i1p1_historical_RACMO22E_v1",
                        var = "pr",
                        lonLim = lonLim,
                        latLim = latLim,
                        years = 1971:2000,
                        time = "DD",
                        aggr.d = "sum",
                        aggr.m = "sum") %>% udConvertGrid(new.units = "degC") %>% interpGrid(ne
pr.hist.corr <- biasCorrection(pr.hist,
                               y = pr.cru,
                               method = "scaling",
                               scaling.type = "multiplicative") %>% redim(drop = TRUE)
```

2.2.2 RCP 8.5 experiment data

In a similar vein, the future (`years = 2010:2100`) data from the RCP 8.5 are loaded and transformed. Note that the value of the argument `dataset` is changed to point to the RCP 8.5 experiment dataset (`dataset = "CORDEX-EUR44_EC-EARTH_r1i1p1_rcp85_RACMO22E_v1"`)

```
rcp85 <- "CORDEX-EUR44_EC-EARTH_r1i1p1_rcp85_RACMO22E_v1"
tasmin.85 <- loadGridData(dataset = rcp85,
                          var = "tasmin",
                          lonLim = lonLim,
                          latLim = latLim,
                          years = 2010:2100,
                          time = "DD",
                          aggr.d = "mean",
                          aggr.m = "mean") %>% udConvertGrid(new.units = "degC") %>% interpGrid
```

Note that for the future projection correction, the calibration is undertaken considering the parameters estimated from the training period (1971-2000), using the historical simulation (`x = tasmin.hist`), and then applied to the new future slice of RCP 8.5 (passed via the `newdata` argument):

```
tasmin.85.corr <- biasCorrection(y = tasmin.cru,
                                 x = tasmin.hist,
                                 newdata = tasmin.85,
                                 method = "scaling",
                                 scaling.type = "additive") %>% redim(drop = TRUE)
```

The same is done with maximum temperature:

```
tasmax.85 <- loadGridData(dataset = rcp85,
                          var = "tasmax",
                          lonLim = lonLim,
                          latLim = latLim,
                          years = 2010:2100,
                          time = "DD",
                          aggr.d = "mean",
                          aggr.m = "mean") %>% udConvertGrid(new.units = "degC") %>% interpGrid
tasmax.85.corr <- biasCorrection(y = tasmax.cru,
                                 x = tasmax.hist,
                                 newdata = tasmax.85,
                                 method = "scaling",
                                 scaling.type = "additive") %>% redim(drop = TRUE)
```

and precipitation:

```
pr.85 <- loadGridData(dataset = rcp85,
                      var = "pr",
                      dictionary = dic,
                      lonLim = lonLim,
                      latLim = latLim,
                      years = 2010:2100,
                      time = "DD",
                      aggr.d = "sum",
                      aggr.m = "sum") %>% interpGrid(new.coordinates = ref.grid)
pr.85.corr <- biasCorrection(y = pr.cru, x = pr.hist, newdata = pr.85, method = "scaling",
                             scaling.type = "multiplicative") %>% redim(drop = TRUE)
```

3 Calculatig future SPEI projections

Once both the historical and the RCP 8.5 projections have been bias-corrected using the CRU observations as reference, these can be joined along time into a single object, so a continuous time series for SPEI can be computed (however, note that there is a gap between the end of the reference period in 2001 and the start of transient period in 2010. This will be transparently handled by the involver climate4R functions involved). This is achieved by the function `bindGrid` from package `transformeR`:

```
tx <- bindGrid(tasmax.hist.corr, tasmax.85.corr, dimension = "time") %>% redim(drop = TRUE)
tn <- bindGrid(tasmin.hist.corr, tasmin.85.corr, dimension = "time") %>% redim(drop = TRUE)
pr <- bindGrid(pr.hist.corr, pr.85.corr, dimension = "time") %>% redim(drop = TRUE)
```

Potential Evapotranspiration need to be calculated prior to computing SPEI. The function `petGrid` of the `drought4R` package is a wrapper of the `hargreaves` function from package `SPEI`, implementing the Hargreaves method for estimation of Potential Evapo-transpiration (see `help(hargreaves, package = "SPEI")` for further details):

```
library(drought4R)
pet.har.85 <- petGrid(tasmin = tn, tasmax = tx, pr = pr, method = "hargreaves")
```

Finally, `speiGrid` is used, a wrapper to the `spei` function from package `SPEI`. Note that the index is computed considering the historical reference period (1971-2000), although the SPEI time series is calculated afterwards over the transient period.

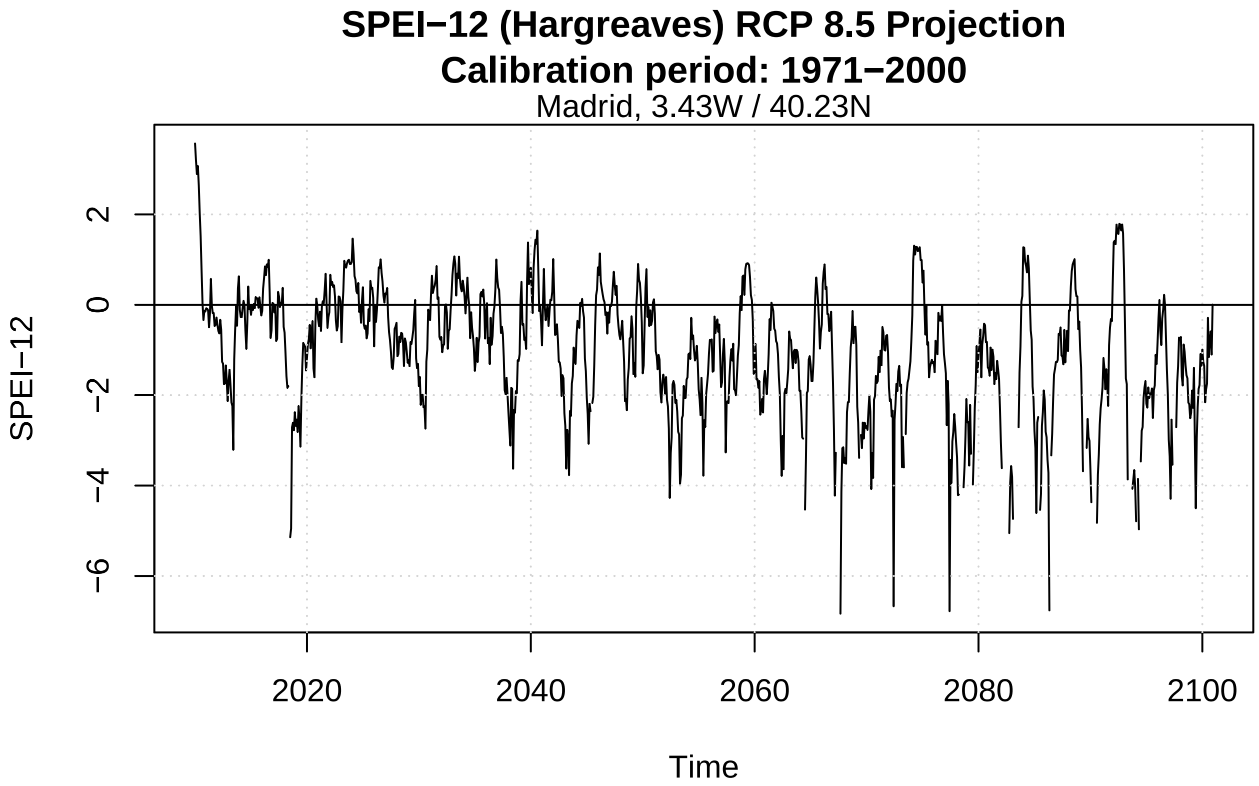
This is done by passing the function the argument `ref.start` and `ref.end`, providing the temporal extent for calibrating the index:

```
spei.rcp85 <- speiGrid(et0.grid = pet.har.85,
                       pr.grid = pr,
                       scale = 12,
                       ref.start = c(1971, 1),
                       ref.end = c(2000, 12),
                       na.rm = TRUE)
```

In the next plot, a time series for the nearest grid point to Madrid (Spain) shows the RCP 8.5 SPEI projections obtained. Note the use of `subsetGrid` to remove the historical period before plotting:

```
m <- subsetGrid(spei.rcp85, lonLim = -3.43, latLim = 40.23, years = 2010:2100)
```

```
plot(ts(m$Data, start = c(2010,1), frequency = 12),
     main = "SPEI-12 (Hargreaves) RCP 8.5 Projection\nCalibration period: 1971-2000",
     ylab = "SPEI-12")
grid()
abline(h = 0)
mtext("Madrid, 3.43W / 40.23N")
```



4 References

- Bache, S.M. and Wickham, H. (2014). magrittr: A Forward-Pipe Operator for R. R package version 1.5. https://CRAN.R-project.org/package=magrittr
- Bedia, J., Gutiérrez, J.M., Herrera, S., Iturbide, M., Manzanas, R. and Medina, J.B. (2018). downscaleR: An R package for bias correction and statistical downscaling. R package version 3.0.2. https://github.com/SantanderMetGroup/downscaleR/wiki
- Beguería, S. and Vicente-Serrano, S.M., 2017. SPEI: Calculation of the Standardised

Precipitation-Evapotranspiration Index. R package version 1.7. https://CRAN.R-project.org/package=SPEI

- Cofiño A, Bedia J, Iturbide M, Vega M, Herrera S, Fernandez J, Frias M, Manzanas R and Gutierrez J, 2018. The ECOMS User Data Gateway: Towards seasonal forecast data provision and research reproducibility in the era of Climate Services. Climate Services. http://doi.org/10.1016/j.cliser.2017.07.001
- Frias, 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
- Harris, I.C., Jones, P.D. (2017): CRU TS4.01: Climatic Research Unit (CRU) Time-Series (TS) version 4.01 of high-resolution gridded data of month-by-month variation in climate (Jan. 1901- Dec. 2016). University of East Anglia Climatic Research Unit; Centre for Environmental Data Analysis, 04 December 2017. doi:10.5285/58a8802721c94c66ae45c3baa4d814d0. http://dx.doi.org/10.5285/58a8802721c94c66ae45c3baa4d814d0
- Iturbide, M., Bedia, J., Herrera, S., Bano-Medina, J., Fernández, J., Frías, M., Manzanas, R., San-Martín, D., Cimadevilla, E., Cofiño, A., Gutiérrez, J., submitted. climate4R: An R-based framework for Climate Data Access, Post-processing and Bias Correction. Submitted to Environmental Modelling and Software.
- Jacob, D. et al., 2014. EURO-CORDEX: new high-resolution climate change projections for European impact research. doi:10.1007/s10113-013-0499-2.
- Turco, M., Cánovas, J.J.R., Bedia, J., Jerez, S., Montávez, J.P., Llasat, M.C. and Provenzale, A., 2018. Exacerbated fires in Mediterranean Europe due to anthropogenic warming projected with non-stationary climate-fire models. Nature Communications, in press

5 Session information

```
sessionInfo() %>% print()
```

```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.5 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=es_ES.UTF-8
##  [9] LC_ADDRESS=es_ES.UTF-8        LC_TELEPHONE=es_ES.UTF-8
## [11] LC_MEASUREMENT=es_ES.UTF-8    LC_IDENTIFICATION=es_ES.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] drought4R_0.1.0   downscaleR_3.0.2  glmnet_2.0-16
##  [4] foreach_1.4.4     Matrix_1.2-14     deepnet_0.2
##  [7] loadeR_1.4.5      loadeR.java_1.1.1 rJava_0.9-8
## [10] visualizeR_1.2.1  sm_2.2-5.4        transformeR_1.4.4
## [13] magrittr_1.5
##
## loaded via a namespace (and not attached):
##  [1] httr_1.3.1              maps_3.3.0
##  [3] SpecsVerification_0.5-2 dotCall64_0.9-5.2
##  [5] sp_1.3-1                latticeExtra_0.6-28
##  [7] yaml_2.1.19             backports_1.1.2
##  [9] lattice_0.20-35         RcppEigen_0.3.3.4.0
## [11] digest_0.6.12           RColorBrewer_1.1-2
## [13] lmomco_2.2.9            colorspace_1.3-2
## [15] htmltools_0.3.6         plyr_1.8.4
## [17] devtools_1.13.6         raster_2.6-7
## [19] padr_0.4.0              scales_0.5.0
## [21] CircStats_0.2-4         git2r_0.19.0
## [23] dtw_1.18-1              tibble_1.3.4
## [25] proxy_0.4-19            ggplot2_2.2.1
## [27] withr_2.1.0             Lmoments_1.2-3
## [29] verification_1.42       pbapply_1.3-3
## [31] lazyeval_0.2.1          SPEI_1.7
## [33] easyVerification_0.4.4  memoise_1.1.0
## [35] evaluate_0.10.1         MASS_7.3-50
## [37] tools_3.4.4             data.table_1.10.4-3
## [39] stringr_1.3.0           munsell_0.4.3
## [41] akima_0.6-2             compiler_3.4.4
## [43] mapplots_1.5            evd_2.3-3
## [45] gridGraphics_0.3-0      rlang_0.1.4
## [47] grid_3.4.4              RCurl_1.95-4.10
## [49] iterators_1.0.9         goftest_1.1-1
## [51] spam_2.1-3              bitops_1.0-6
## [53] rmarkdown_1.10          boot_1.3-20
## [55] vioplot_0.2             gtable_0.2.0
## [57] codetools_0.2-15        abind_1.4-5
## [59] curl_3.0                R6_2.2.2
## [61] gridExtra_2.3           knitr_1.20
## [63] rprojroot_1.3-2         stringi_1.1.7
## [65] parallel_3.4.4          Rcpp_0.12.17
## [67] fields_9.6
```