Full code for Example 2 of the paper 'climate4R: An Ecosystem of R packages for Climate Data Access, Post-processing and Bias Correction'

M. Iturbide, J. Bedia, S. Herrera, J. Baño, J. Fernández, M. D. Frías, R. Manzanas, D. San Martín, E. Cimadevilla, A.S. Cofiño, J. M. Gutiérrez

2018-04-11

Contents

1 Introduction

This worked example contains the full code that reproduces the 2nd example of the paper "climate4R: An Ecosystem of R packages for Climate Data Access, Post-processing and Bias Correction" (Sec. 6 of the manuscript). The main section is divided in additional subsections to help with the understanding of the different code chunks. All operations hereinafter are performed with the core packages of climate4R, excepting (1) package installation and (2) the creation of color palettes, for which packages `devtools` and `RColorBrewer` are used respectively:

(1) Package installation:

```
library(devtools)
install_github(c("SantanderMetGroup/loadeR",
                 "SantanderMetGroup/loadeR.java",
                 "SantanderMetGroup/transformeR",
                 "SantanderMetGroup/visualizeR",
                 "SantanderMetGroup/downscaleR",
                 "SantanderMetGroup/climate4R.climdex")
```

```
library(loadeR)
library(transformeR)
library(visualizeR)
library(downscaleR)
library(climate4R.climdex)
```

(2) Brewer palettes:

```
library(RColorBrewer)
colstx <- rev(brewer.pal(n = 9, "Spectral"))
colsindex <- rev(brewer.pal(n = 9, "RdYlBu"))
colsdelta <- brewer.pal(n = 9, "Reds")
colsbias <- brewer.pal(n = 9, "PiYG")
colssd <- brewer.pal(n = 9, "Blues")
```

NOTE: see also 2018_climate4R_example1.pdf for a better understanding of this document.

---

2 Example 2: CORDEX Ensembles via the User Data Gateway

2.1 Loading via the User Data Gateway

We define the domain of the Iberian Peninsula with the following bounding coordinates:

```
lon <- c(-10, 5)
lat <- c(36, 44)
```

The UDG service requires (free) registration to accept the data policies of the different data providers. Once a valid user name and password have been issued, the authentication must be done within the R session before data loading with function `loginUDG`:

```
loginUDG(username = "userUDG", password = "pswrdUDG")
```

If the data is to be loaded from the UDG, we can use function `UDG.datasets` to print the inventory of the available public and harmonized **UDG** datasets, where the name, type and url are specified (see Table 1 in the manuscript). Additionally, we can use the "name" (`UDG.datasets()["name"]`) of the desired dataset instead of passing the complete url to `loadGridData`. If this is the case, we do not need to create a dictionary, since the data is harmonized by default. Nevertheless, we still can use the complete url (`UDG.datasets()["url"]`) to access the data in its original form.

For example, if we are interested in loading observations from the **E-OBS** dataset at 0.25 degrees resolution we can filter the names returned by `UDG.datasets` passing an appropriate pattern to function `grepl`:

```
models <- UDG.datasets()
eobs <- models$name[grepl("E-OBS.*0.25", models$name)]
```

Object `eobs` contains the name of the dataset and is passed to `loadGridData` for loading maximum temperature using the standard name "tasmax" (and without using the dictionary argument, see 2018_climate4R_example1.pdf):

```
TX <- loadGridData(eobs,
                           var = "tasmax",
                           season = 1:12,
                           lonLim = lon,
                           latLim = lat,
                           years = 1971:2000)
```

To load historical CORDEX data for the Iberian Peninsula we also filtered the names in `UDG.datasets()` with an appropriate pattern:

```
models <- UDG.datasets()
ensemble.h <- models$name[grepl("EUROCORDEX44.*historical", models$name)][1:6]
```

Unlike the first example of the paper (see 2018_climate4R_example1.pdf), here we considered 6 regional climate models (RCMs) from EURO-CORDEX, thus, object `ensemble.h` contains 6 names. Everything can be loaded in a single step by combining function `loadGridData` with `lapply`:

```
TXh.list <- lapply(ensemble.h, function(x)
                               loadGridData(dataset = x,
                               var = "tasmax",
                               season = 1:12,
                               lonLim = lon,
                               latLim = lat,
                               years = 1971:2000))
```

We repeat the operation for the RCP8.5 scenario:

```
ensemble.f <- as.character(models$name[grepl("EUROCORDEX44.*rcp85", models$name)])[1:6]
TXf.list <- lapply(ensemble.f, function(x)
                           loadGridData(dataset = x,
                           var = "tasmax",
                           season = 1:12,
                           lonLim = lon,
                           latLim = lat,
                           years = 2071:2100))
```

As a result, we obtain the following harmonized grids:

- `TX`: a single grid for E-OBS and reference period 1971-2000
- `TXh.list`: a list of 6 grids for the reference period 1971-2000 (historical scenario)
- `TXf.list`: a list of 6 grids for the future period 2071-2100 (RCP8.5 scenario).

2.2 Working with multi-model ensembles

climate4R functionalities allow working with model ensembles through the `member` dimension. Thus, we can aggregate the list of `grid`s created before (`TXh.list` and `TXf.list`) along the `member` dimension to obtain a single `grid`. However, we need to check temporal and spatial consistency among the different models. In this case, a temporal inconsistency exists in our ensemble, since two of the models contain less calendar days. We can check this easily with function `getShape`:

```
lapply(TXh.list, function(x) getShape(x))
```

```
## [[1]]
##  time   lat   lon
## 10958    26    32
##
## [[2]]
##  time   lat   lon
## 10958    26    32
##
## [[3]]
##  time   lat   lon
## 10958    26    32
##
## [[4]]
##  time   lat   lon
## 10958    26    32
##
## [[5]]
```

```
##  time   lat   lon
## 10950    26    32
##
## [[6]]
##  time   lat   lon
## 10748    26    32
```

Function `intersectGrid.time` performs time subsetting of a collection of grids, to the dates they have in common:

```
# Temporal intersection
TXh.list <- do.call("intersectGrid.time", list(TXh.list, which.return = 1:6))
TXf.list <- do.call("intersectGrid.time", list(TXf.list, which.return = 1:6))
```

If there is not spatial consistency, we can use `interpGrid` to interpolate all `grid`s to the same spatial structure. Despite this not being the case, here we perform interpolation of CORDEX data to the E-OBS spatial grid in order to apply a land-sea mask (by means of function `gridArithmetics`) and produce comparable map figures.

```
# Interpolation
TXh.list <- lapply(TXh.list, function(x) interpGrid(x, getGrid(TX)))
TXf.list <- lapply(TXf.list, function(x) interpGrid(x, getGrid(TX)))
```

```
# Create mask
m <- TX$Data[1,,]*0
mask.hist <- array(dim = c(getShape(TXh.list[[1]])["time"], dim(m)))
for (i in 1:dim(mask.hist)[1]) mask.hist[i,,] <- m
mask.rcp <- array(dim = c(getShape(TXf.list[[1]])["time"], dim(m)))
for (i in 1:dim(mask.rcp)[1]) mask.rcp[i,,] <- m

# Apply mask
TXh.list <- lapply(TXh.list, function(x)
  gridArithmetics(x, mask.hist, operator = "+"))
TXf.list <- lapply(TXf.list, function(x)
  gridArithmetics(x, mask.rcp, operator = "+"))
```

Finally, we crate the multi-member grids for the historical and RCP8.5 scenarios of CORDEX (objects `TXh.ens` and `TXf.ens`):

```
# Create a multimember grid
TXh.ens <- do.call("bindGrid.member", TXh.list)
TXf.ens <- do.call("bindGrid.member", TXf.list)
```

Note that function `spatialPlot` recognizes a multi-member grid and displays a map for each member. For instance, next we plot the ensemble of the RCP8.5 scenario to generate Figure 1 (Fig. 8(above) in the manuscript)):

```
spatialPlot(climatology(TXf.ens), at = seq(5, 33, 1), backdrop.theme = "countries",
            col.regions = colorRampPalette(colstx), layout = c(3, 2), as.table = TRUE)
```

In order to calculate the bias of each model with respect to the observations and generate Figure 2 (Fig. 8(below) in the manuscript) we apply function `aggregateGrid`, `gridArithmetics`, `bindGridMember` and `spatialPlot` as follows:

```
# Create a multimember grid
TXh.list.ann <- lapply(TXh.list, function(x)
  aggregateGrid(x, aggr.y = list(FUN = "mean", na.rm = TRUE)))
TX.ann <- aggregateGrid(TX, aggr.y = list(FUN = "mean", na.rm = TRUE))
TXh.list.bias <- lapply(TXh.list.ann, function(x)
```

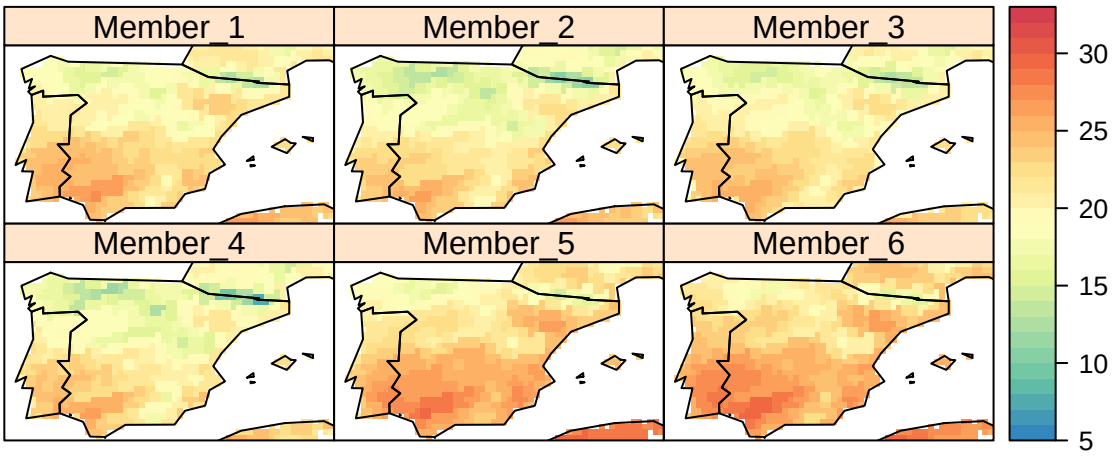

Figure 1: Maximum temperature (ºC) in Iberia for an ensemble of 6 CORDEX RCMs under the RCP8.5 scenario and for future period 2071-2100. Fig. 8(above) in the manuscript.

```
  gridArithmetics(x, TX.ann, operator = "-"))
```

```
TXh.bias.ens <- do.call("bindGrid.member", TXh.list.bias)
```

```
spatialPlot(climatology(TXh.bias.ens), at = seq(-10, 10, 1), backdrop.theme = "countries",
            col.regions = colorRampPalette(colsbias), layout = c(3, 2), as.table = TRUE)
```

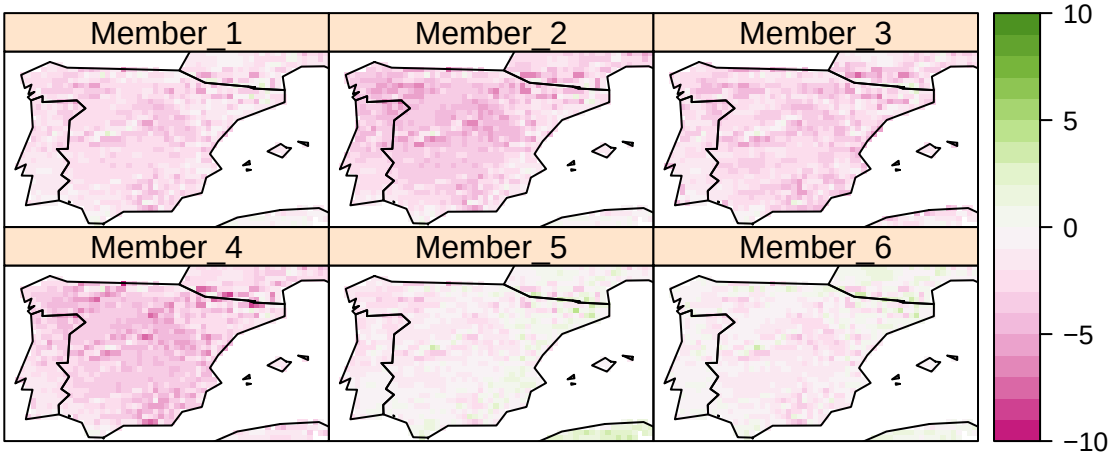

Figure 2: Bias of the maximum temperature (ºC) in Iberia for an ensemble of 6 CORDEX RCMs w.r.t. E-OBS in the historical period 1971-2000. Fig. 8(below) in the manuscript.

We can use function `aggregateGrid` to for example calculate the multi-member mean and deviation of the ensemble:

```
TXf.ens.mean <- aggregateGrid(TXf.ens, aggr.mem = list(FUN = mean, na.rm = TRUE))
TXf.ens.sd <- aggregateGrid(TXf.ens, aggr.mem = list(FUN = sd, na.rm = TRUE))
```

Next we generate the corresponding Figures 3 and 4 (Not shown in the manuscript).

```
spatialPlot(climatology(TXf.ens.mean), at = seq(5, 33, 1),
            col.regions = colorRampPalette(colstx), backdrop.theme = "countries")
```

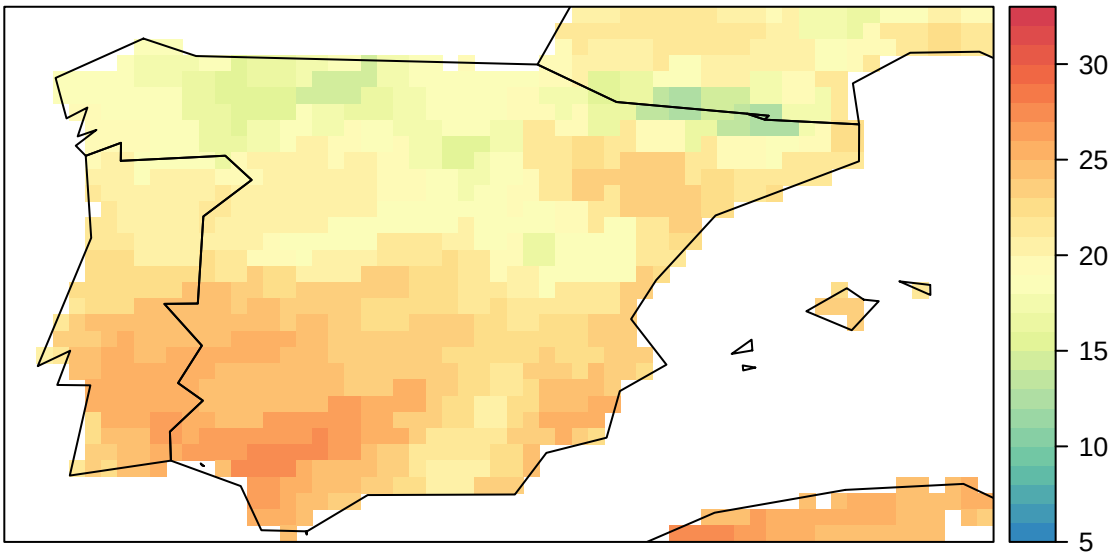

Figure 3: Ensemble mean of maximum temperature (ºC) in Iberia for 6 CORDEX RCMs under the RCP8.5 scenario and for future period 2071-2100. Not shown in the manuscript.

```
spatialPlot(climatology(TXf.ens.sd), at = seq(0, 6, .5),
            col.regions = colorRampPalette(colssd), backdrop.theme = "countries")
```

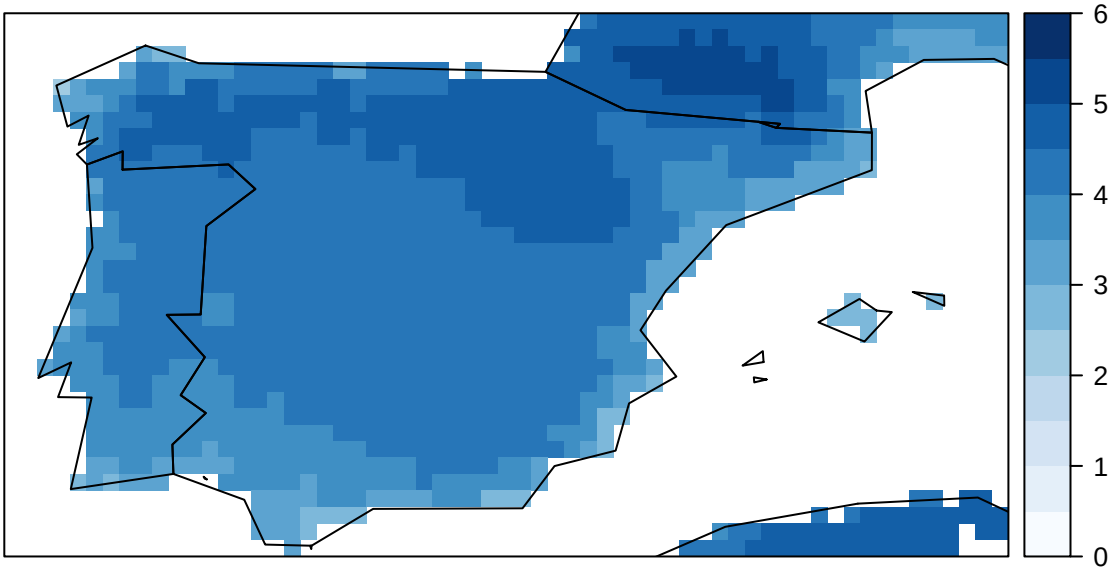

Figure 4: Ensemble standard deviation (sd) of maximum temperature (ºC) in Iberia for 6 CORDEX RCMs under the RCP8.5 scenario and for future period 2071-2100. Not shown in the manuscript.

2.3 ETCCDI index calculation (SU) from raw data

Next the raw SU index (summer days) is calculated for the whole ensemble (object `SUf.ens`) and future period 2071-2100, in a single line with function `climdexGrid`:

```
SUf.ens <- climdexGrid(tx = TXf.ens, index.code = "SU")
```

We calculate the ensemble mean and deviation using function `aggregateGrid`:

```
SUf.ens.mean <- aggregateGrid(SUf.ens, aggr.mem = list(FUN = mean, na.rm = TRUE))
SUf.ens.sd <- aggregateGrid(SUf.ens, aggr.mem = list(FUN = sd, na.rm = TRUE))
```

And finally, we plot the results to generate Figures 5 and 6 (Figs. 6(above, left) and 6(above, right) in the manuscript:

```
spatialPlot(climatology(SUf.ens.mean), backdrop.theme = "countries",
            at = seq(0, 260, 10), col.regions = colorRampPalette(colsindex))
```

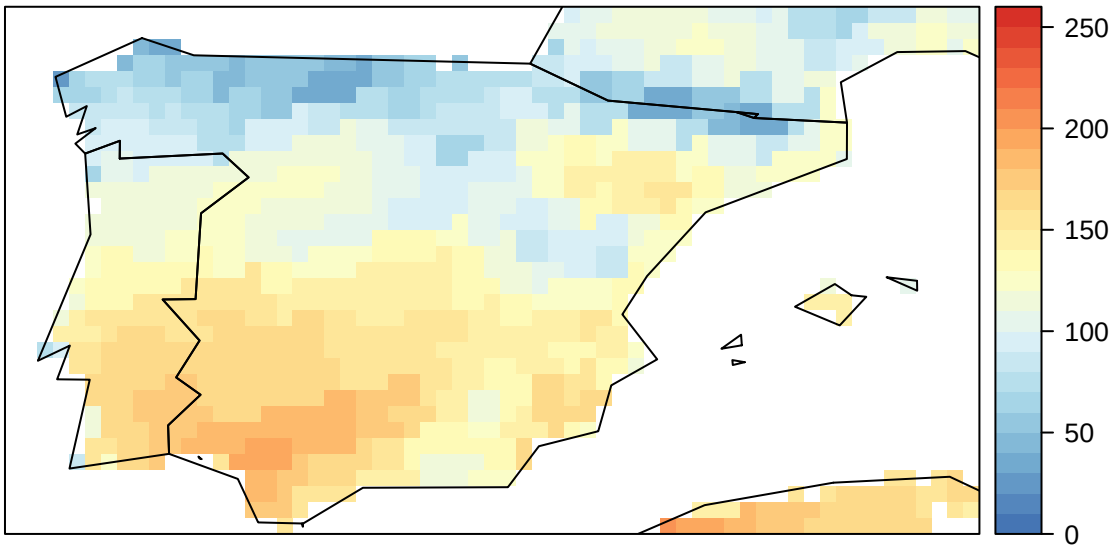

Figure 5: Summer days in Iberia for the future period 2071-2100 computed from the original RCM daily maximum temperature data. The figure shows the ensemble mean. Fig. 6(above, left) in the manuscript.

```
spatialPlot(climatology(SUf.ens.sd), set.max = 50, backdrop.theme = "countries",
            at = seq(0, 50, 5), col.regions = colorRampPalette(colssd))
```

2.4 Bias correction of the maximum temperature

Here we apply Empirical Quantile Mapping (EQM) for correcting the bias of each ensemble member (by default argument `join.members = FALSE`) by considering a moving correction window of 30 days to correct each 7-day time interval:

```
TXf.ens.bc <- biasCorrection(TX,
                             TXh.ens,
                             TXf.ens,
                             window = c(30, 7),
```

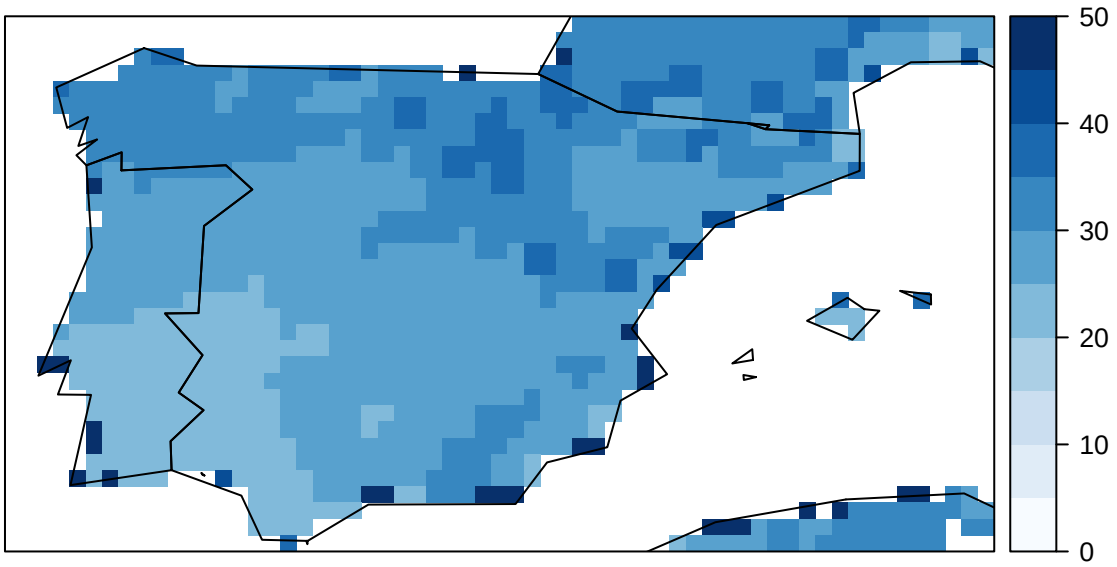

Figure 6: Standard deviation of summer days in Iberia for the future period 2071-2100 computed from the original RCM daily maximum temperature data. The figure shows the spread of the ensemble. Fig. 6(above, right) in the manuscript.

```
                                    extrapolation = "constant",
                                    method = "eqm")
```

As done before, we calculate the ensemble mean and deviation of the maximum temperature with function `aggregateGrid`:

```
TXf.ens.bc.mean <- aggregateGrid(TXf.ens.bc, aggr.mem = list(FUN = mean, na.rm = TRUE))
TXf.ens.bc.sd <- aggregateGrid(TXf.ens.bc, aggr.mem = list(FUN = sd, na.rm = TRUE))
```

We plot the results to generate Figures 7 and 8 (Not shown in the manuscript):

```
spatialPlot(climatology(TXf.ens.bc.mean), backdrop.theme = "countries",
            at = seq(5, 33, 1), col.regions = colorRampPalette(colstx))
```

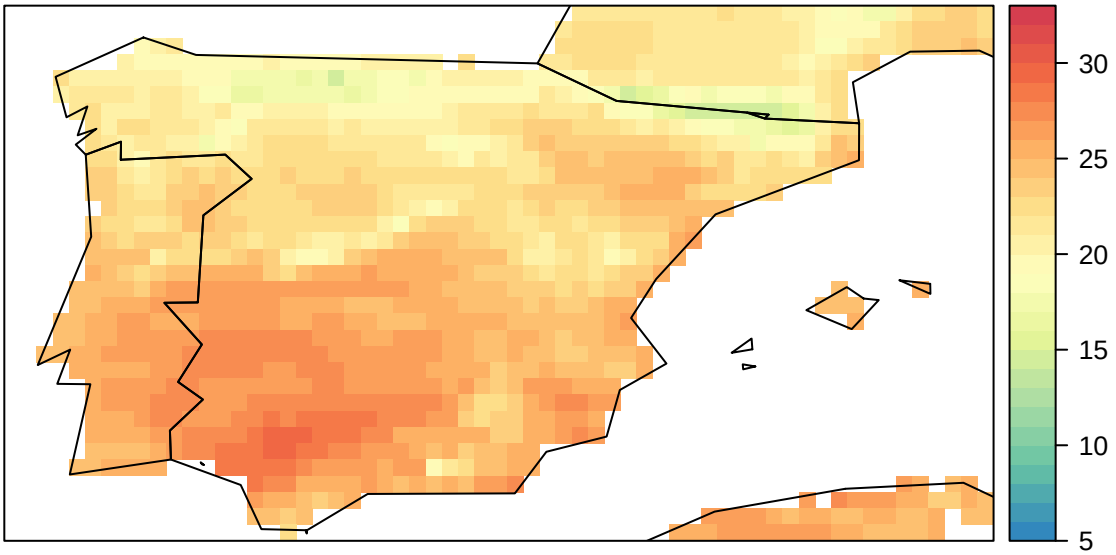

Figure 7: Ensemble mean of bias corrected maximum temperature (ºC) in Iberia for 6 CORDEX RCMs under the RCP8.5 scenario and for future period 2071-2100. Not shown in the manuscript.

```
spatialPlot(climatology(TXf.ens.bc.sd), backdrop.theme = "countries",
            at = seq(0, 6, .5), col.regions = colorRampPalette(colssd))
```

2.5 ETCCDI index calculation (SU) from bias corrected data

Next we apply function `climdexGrid` over the bias corrected maximum temperature (object `TXf.ens.bc`) to obtain the corrected SU index for the future period.

```
SUf.ens.bc <- climdexGrid(tx = TXf.ens.bc, index.code = "SU")
```

Again, the ensemble mean and deviation is calculated:

```
SUf.ens.bc.mean <- aggregateGrid(SUf.ens.bc, aggr.mem = list(FUN = mean, na.rm = TRUE))
SUf.ens.bc.sd <- aggregateGrid(SUf.ens.bc, aggr.mem = list(FUN = sd, na.rm = TRUE))
```

By plotting these results Figures 9 and 10 are generated (Figs. 6(below, left) and 6(below, right) in the manuscript):

```
spatialPlot(climatology(SUf.ens.bc.mean), backdrop.theme = "countries",
            at = seq(0, 230, 10), col.regions = colorRampPalette(colsindex))
```

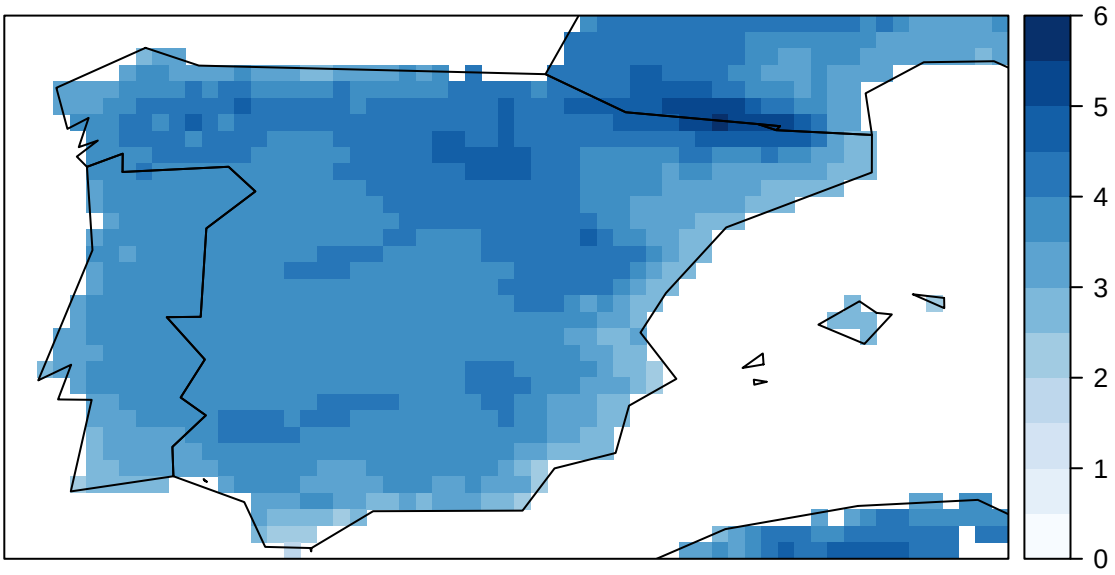

Figure 8: Ensemble standard deviation (sd) of bias corrected maximum temperature (ºC) in Iberia for 6 CORDEX RCMs under the RCP8.5 scenario and for future period 2071-2100. Not shown in the manuscript.

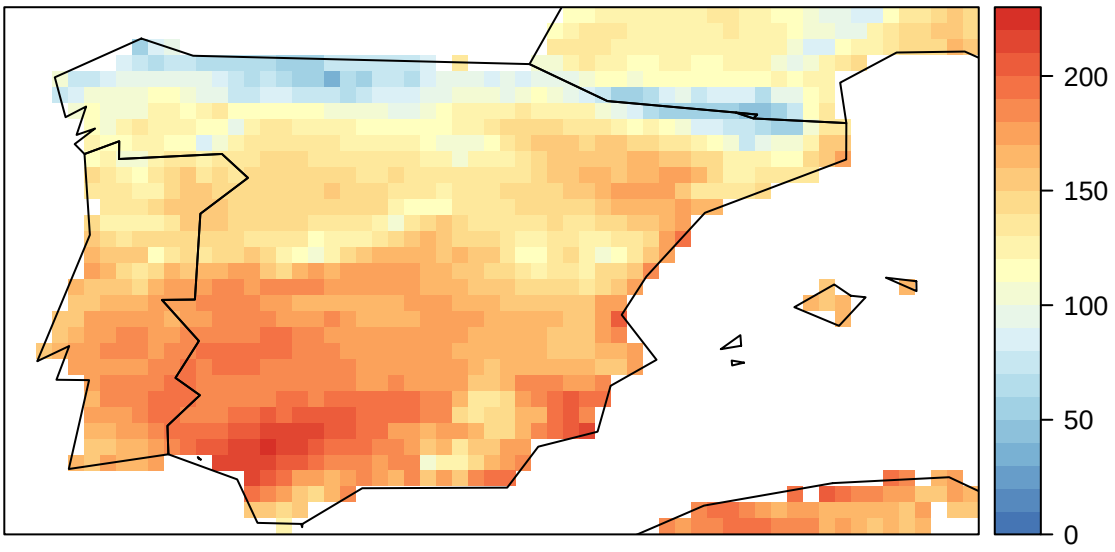

Figure 9: Summer days in Iberia for the future period 2071-2100 computed from the bias corrected RCM daily maximum temperature data. The figure shows the ensemble mean. Fig. 6(below, left) in the manuscript.

```
spatialPlot(climatology(SUf.ens.bc.sd), backdrop.theme = "countries",
            at = seq(0, 50, 5), col.regions = colorRampPalette(colssd))
```

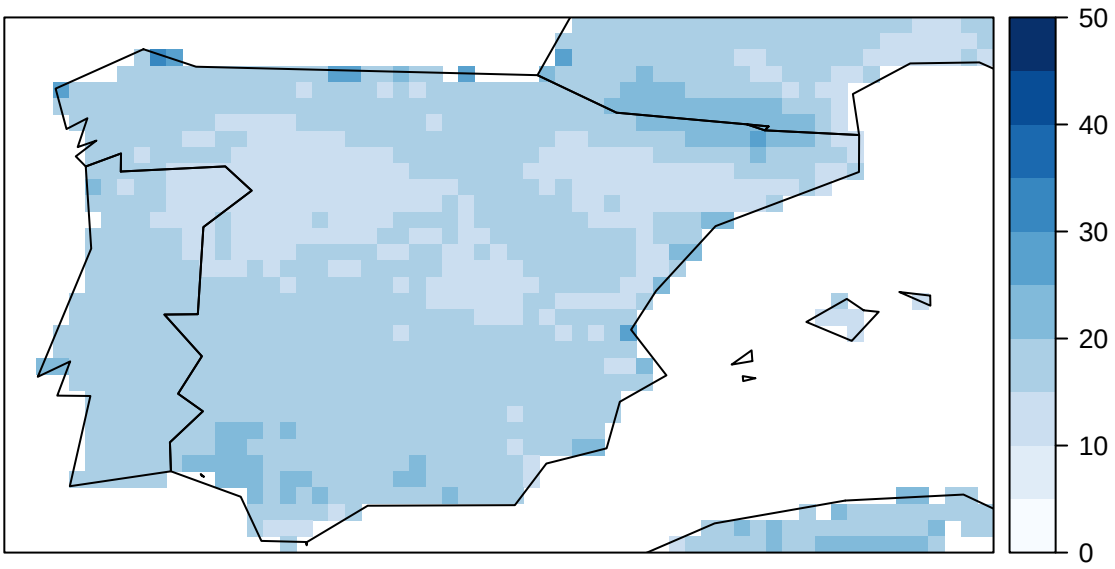

Figure 10: Standard deviation of summer days in Iberia for the future period 2071-2100 computed from bias corrected RCM daily maximum temperature data. The figure shows the spread of the ensemble. Fig. 6(below, right) in the manuscript.

In order to generate Figure 11 (Fig. 7 in the manuscript), we next calculate the SU index from observational data (`SU`) and the historical model ensemble (`SUh.ens`):

```
#A single location
SU <- climdexGrid(tx = TX, index.code = "SU")
SUh.ens <- climdexGrid(tx = TXh.ens, index.code = "SU")
```

We also extract the nearest grid box to Zaragoza for E-OBS (object `SU.Z.eobs`) and CORDEX (object `SU.Z.cdx`). The 1rst memeber of CORDEX is also extracted and everything is listed in object `SU.Z` for plotting:

```
SU.Z.eobs <- subsetGrid(SU, latLim = 41.64, lonLim = -0.89)
SU.Z.cdx <- lapply(list(SUh.ens, SUf.ens, SUf.ens.bc), function(x)
                        subsetGrid(x, latLim = 41.64, lonLim = -0.89))
SU.Z.1m <- lapply(SU.Z.cdx, function(x)
                        subsetGrid(x, members = 1))
SU.Z <- c(list(SU.Z.eobs), SU.Z.cdx, SU.Z.1m)
```

Finally `temporalPlot` is used to generate Figure 7 (Fig. 7 in the manuscript):

```
cols  <- c("black", rep(c("red", "red", "blue"), 2))
temporalPlot(SU.Z,
             cols = cols,
             lty = rep(c(1,3), each = 4),
             lwd = 0.8,
             xyplot.custom =
               list(ylim = c(70, 220), ylab = "",
                                key = list(space = "top",
                                           lines = list(lty = c(rep(1,4), 3),
                                                        col = c(cols[1:4], cols[1]),
```

```
                                    lwd = 0.8),
                text = list(c("E-OBS", "CDX_hist",
                       "CDX_rcp85", "CDX_rcp85_corrected",
                              "1st_member"),
                            cex = .7),
                columns = 3, rows = 3)))
```

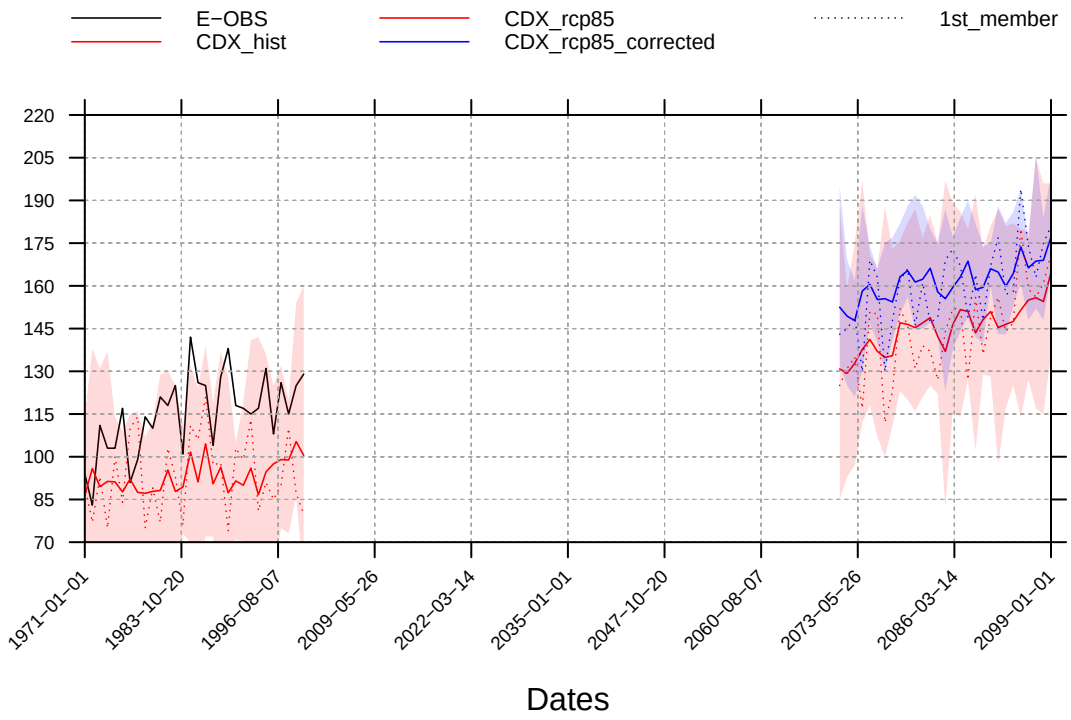

Figure 11: Annual summer days time series for a single gridbox (the one closest to Zaragoza, Spain) computed form (red) the original RCM daily maximum temperature data, and (blue) daily maximum temperature bias corrected data using E-OBS (black). When it comes to CORDEX data, continuous lines correspond to the ensemble mean and the shadowed area to the range (uncertainty). Dashed lines correspond to the 1st member of the ensemble. Fig. 7 in the manuscript.

2.6 Calculation of the ETCCDI index CDD (Consecutive Dry Days)

This section replicates the procedures shown above but for calculating a precipitation based index, i.e. CDD (consecutive dry days). Function `climdexShow` displays an overview of the available ETCDDI indices:

```
climdexShow()[,1:6]
```

2.6.1 Data loading and multi-member ensemble building

In order to calculate the CDD index, we need to load daily precipitation data. Therefore, we repeat the loading operation (function `loadGridData`) shown in the previous section (this time `var = "pr"`) for observational data (E-OBS) and historical and RCP8.5 projection data (6 models from CORDEX).

```
pr <- loadGridData(eobs,
                           var = "pr",
                           season = 1:12,
                           lonLim = lon,
```

```
                          latLim = lat,
                          years = 1971:2000)
```

```
prh <- lapply(ensemble.h, function(x)
                              loadGridData(dataset = x,
                              var = "pr",
                              season = 1:12,
                              lonLim = lon,
                              latLim = lat,
                              years = 1971:2000))
prf <- lapply(ensemble.f, function(x)
                          loadGridData(dataset = x,
                          var = "pr",
                          season = 1:12,
                          lonLim = lon,
                          latLim = lat,
                          years = 2071:2100))
prh.t <- do.call("intersectGrid.time", list(prh, which.return = 1:6))
prf.t <- do.call("intersectGrid.time", list(prf, which.return = 1:6))
prh.ens <- do.call("bindGrid.member", prh.t)
prf.ens <- do.call("bindGrid.member", prf.t)
```

Next the CDD index is calculated (function `climdexGrid`), and regridded (function `interpGrid`) to the spatial structure given by the observational data (`getGrid(pr)`). Then, the ensemble mean (object `CDDf.ens.mean`) and deviation (object `CDDf.ens.sd`) are calculated with `aggregateGrid`.

```
CDDf.ens <- climdexGrid(pr = prf.ens, index.code = "CDD")
CDDf.ens.interp <- interpGrid(CDDf.ens, getGrid(pr))
CDDf.ens.mean <- aggregateGrid(CDDf.ens.interp,
                               aggr.mem = list(FUN = "mean", na.rm = TRUE))
CDDf.ens.sd <- aggregateGrid(CDDf.ens.interp,
                             aggr.mem = list(FUN = "sd", na.rm = TRUE))
```

We also create and apply the land-sea mask:

```
m <- pr$Data[1,,]*0
mask.cdd <- array(dim = c(getShape(CDDf.ens.mean)["time"], dim(m)))
for (i in 1:dim(mask.cdd)[1]) mask.cdd[i,,] <- m
CDDf.ens.mean <- gridArithmetics(CDDf.ens.mean, mask.cdd, operator = "+")
CDDf.ens.sd <- gridArithmetics(CDDf.ens.sd, mask.cdd, operator = "+")
```

Finally, we plot the result with function `spatialPlot` to generate Figures 12 and 13 (not shown in the manuscript).

```
spatialPlot(climatology(CDDf.ens.mean), backdrop.theme = "countries", at = seq(0, 60, 5),
            col.regions = colorRampPalette(colsindex))
```

```
spatialPlot(climatology(CDDf.ens.sd), backdrop.theme = "countries", at = seq(0, 60, 5),
            col.regions = colorRampPalette(colssd))
```

2.6.2 Bias correction of the precipitation

Here, we use again the EQM method for bias correcting precipitation data to subsequently calculate the corrected CDD index (objects `CDDf.ens.bc`, `CDDf.ens.bc.mean`, `CDDf.ens.bc.sd`).

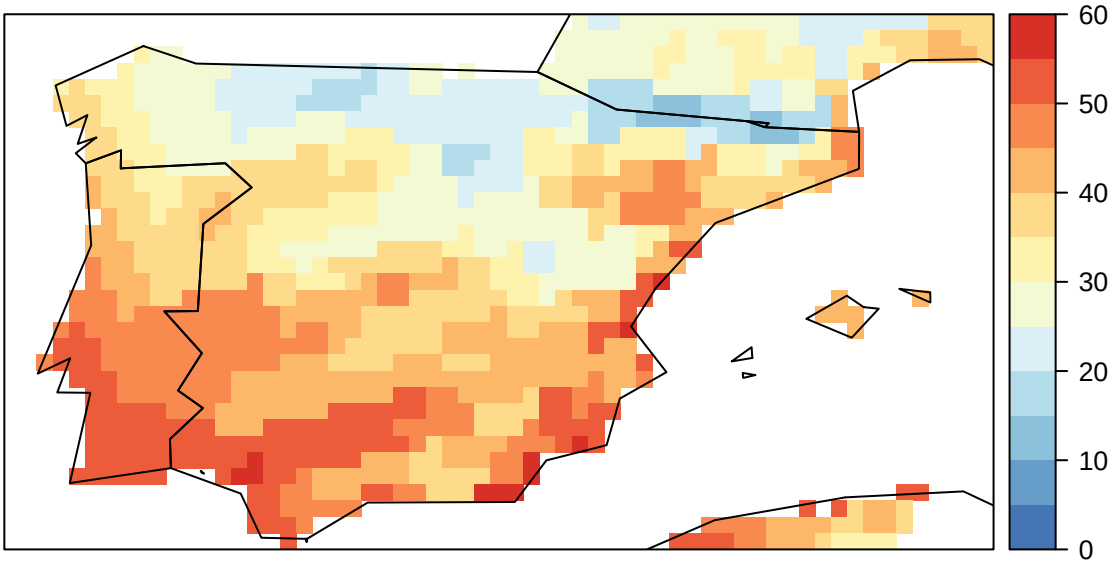

Figure 12: Consecutive dry days (CDD) in Iberia for the future period 2071-2100 computed from raw RCM daily precipitation data. The figure shows the the ensemble mean. Not shown in the manuscript.

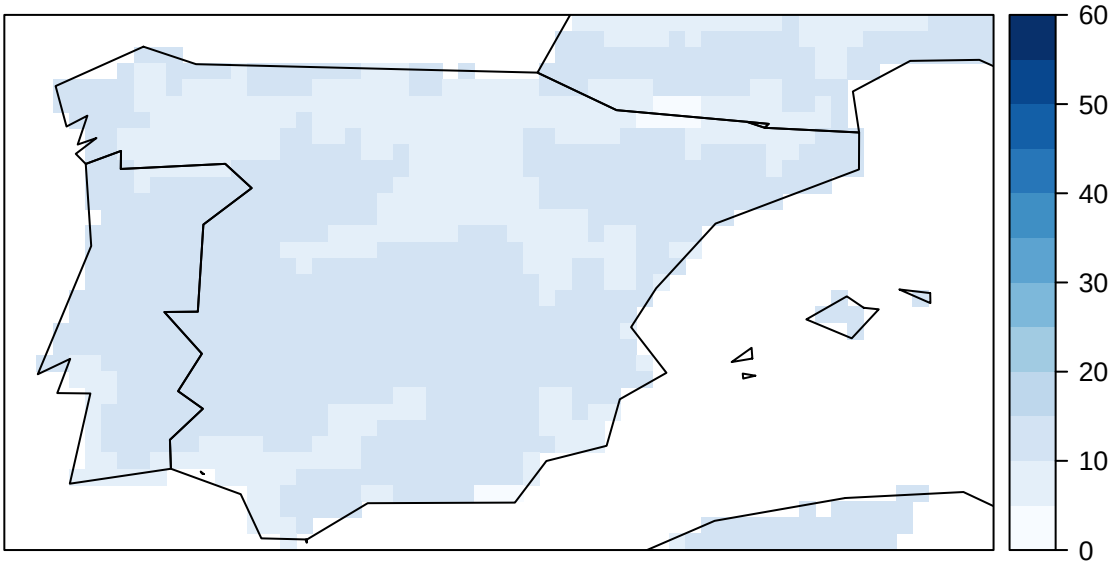

Figure 13: Standard deviation of consecutive dry days (CDD) in Iberia for the future period 2071-2100 computed from raw RCM daily precipitation data. The figure shows the spread of the ensemble. Not shown in the manuscript.

```
prf.ens.bc <- biasCorrection(y = pr,
                                    x = prh.ens,
                                    newdata = prf.ens,
                                    precipitation = TRUE,
                                    window = c(30, 7),
                                    extrapolation = "constant",
                                    method = "eqm",
                                    wet.threshold = 0.1)
CDDf.ens.bc <- climdexGrid(pr = prf.ens.bc, index.code = "CDD")
CDDf.ens.bc.mean <- aggregateGrid(CDDf.ens.bc,
                                  aggr.mem = list(FUN = "mean", na.rm = TRUE))
CDDf.ens.bc.sd <- aggregateGrid(CDDf.ens.bc,
                                aggr.mem = list(FUN = "sd", na.rm = TRUE))
```

This results are plotted to generate Figures 14 and 15 (not shown in the manuscript):

```
spatialPlot(climatology(CDDf.ens.bc.mean), backdrop.theme = "countries",
            at = seq(0, 60, 5), col.regions = colorRampPalette(colsindex))
```

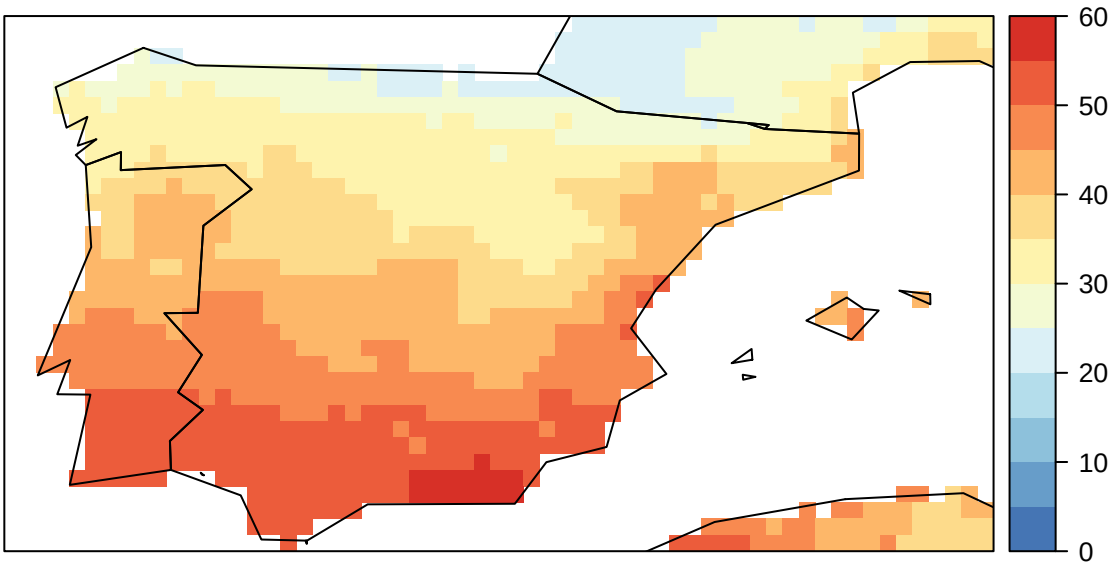

Figure 14: Consecutive dry days (CDD) in Iberia for the future period 2071-2100 computed from bias corrected RCM daily precipitation data. The figure shows the the ensemble mean. Not shown in the manuscript.

```
spatialPlot(climatology(CDDf.ens.bc.sd), backdrop.theme = "countries",
            at = seq(0, 60, 5), col.regions = colorRampPalette(colssd))
```

In this case, unlike the SU index, the uncertainty is not reduced before and after bias correcting the original variable (compare Figs. 13 and 15). Nevertheless the pattern of the index is significantly modified (compare Figs. 12 and 14).

3 Other available material

- 2018_climate4R_example1.pdf contains the full code for **Example 1** of the paper 'climate4R: An Ecosystem of R packages for Climate Data Access, Post-processing and Bias Correction'.

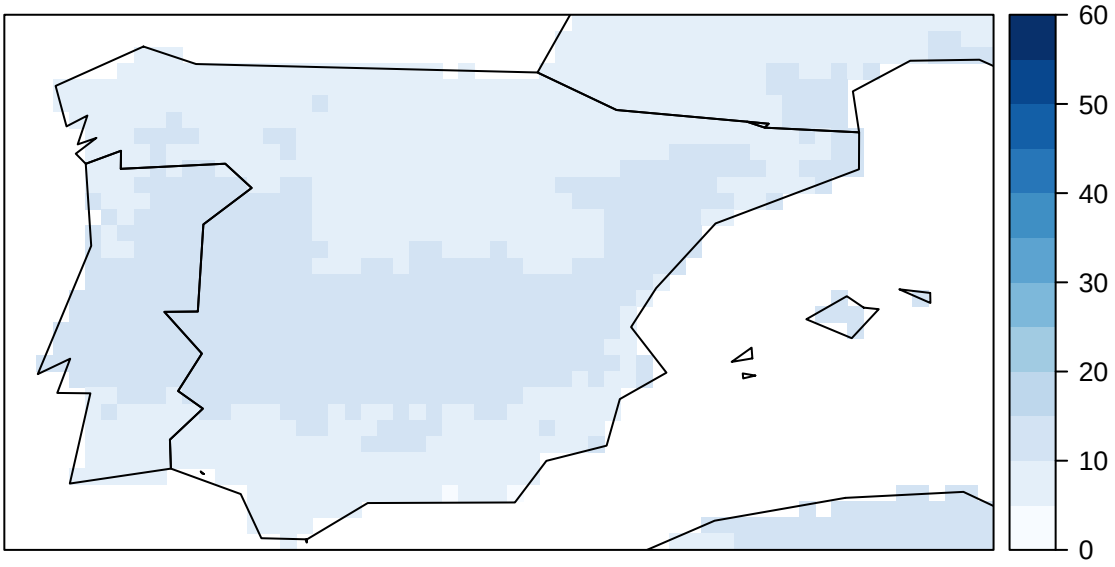

Figure 15: Standard deviation of consecutive dry days (CDD) in Iberia for the future period 2071-2100 computed from bias corrected RCM daily precipitation data. The figure shows the spread of the ensemble. Not shown in the manuscript.

- Find more worked examples on the utilization of climate4R packages in their respective GitHub **wiki**-s at https://github.com/SantanderMetGroup:
  - loadeR: https://github.com/SantanderMetGroup/loadeR/wiki
  - transformeR: https://github.com/SantanderMetGroup/transformeR/wiki
  - downscaleR: https://github.com/SantanderMetGroup/downscaleR/wiki
  - visualizeR: https://github.com/SantanderMetGroup/visualizeR/wiki