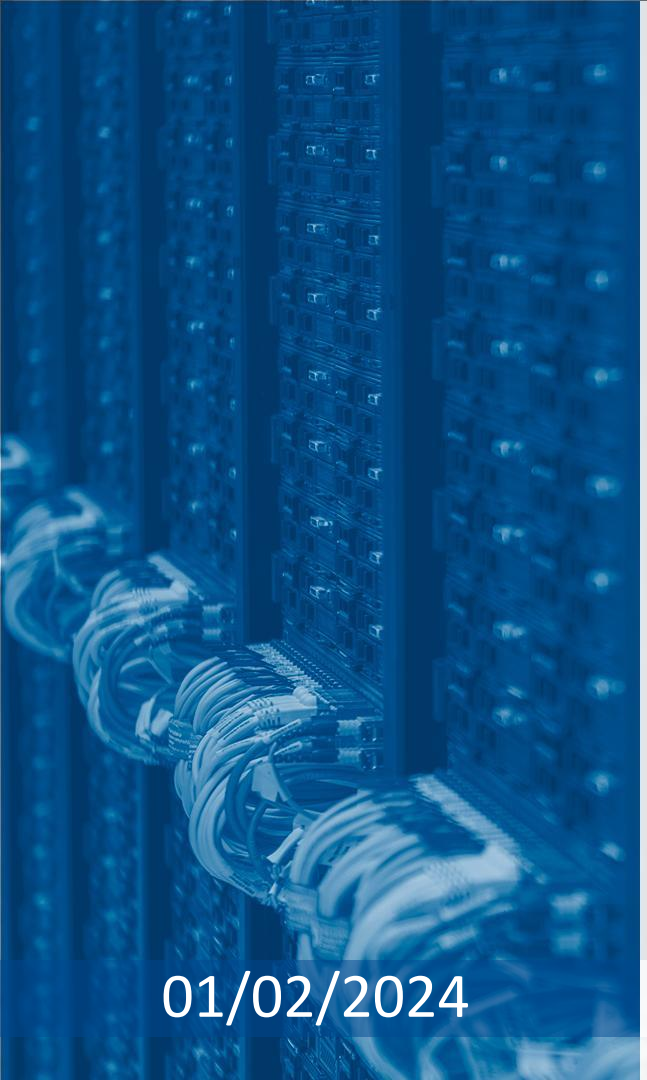


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

R user meeting

Eva Rifà, Victòria Agudetse

contributor: Javier Corvillo

01/02/2024

Agenda

1. Ice-breaker: Subset and CST_Subset
2. News
    - General R
    - s2dv
    - startR
    - ClimProjDiags
    - CSTools
    - CSIndicators
    - esviz
    - SUNSET
3. Presentation: Javier Corvillo
4. Q&A

Ice-breaker


BSC
Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

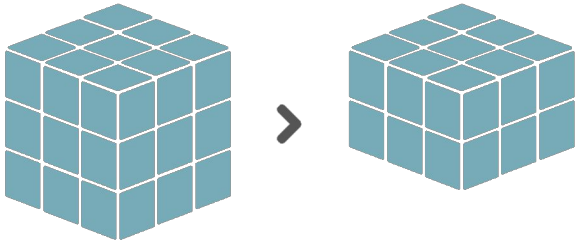
Subset and CST_Subset

★ Subset an array along any dimension

★ CST version subsets also the coordinates and metadata of the ‘s2dv_cube’

```
# Subset the s2dv_cube
exp <- CST_Subset(lonlat_temp$exp,
                  along = c("lat", "lon"),
                  indices = list(1:10, 1:10),
                  drop = 'all')
```

```
> dim(lonlat_temp$exp$data)
dataset  member  sdate  ftime  lat  lon
      1      15      6      3     22     53
```



```
's2dv_cube'
Data        [ 279.99, 280.34, 279.45, 281.99, ... ]
Dimensions  (member = 15, sdate = 6, ftime = 3,
             lat = 10, lon = 10 )
Coordinates
  member : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ...
* sdate : 20001101, 20011101, 20021101, 20031101, ...
  ftime : 1, 2, 3
* lat : 48, 47, 46, 45, 44, 43, 42, 41, 40, 39
* lon : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Attributes
  varName  : tas
  metadata :
     lat
       cdo_grid_name : r360x181
     lon
       cdo_grid_name : r360x181
      ...
```

See example script: https://earth.bsc.es/gitlab/external/cstools/-/blob/master/inst/doc/usecase/ex4_subset.R

General R

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

R Tools next developments

★ Some developments will remain stopped for a while

★ R Tools are created from collaboration between researchers and developers

s2dv

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

CDORemap added tolerance in crop condition

CDORemap returned wrong number of output latitudes when 'lats' are sorted in decreasing order

- For global grid, when crop = T and lat is decreasing, the returned grid was missing points. The function took the original grid boundaries and some points were missing. ⟶ SOLUTION: added development for this case

issues: https://earth.bsc.es/gitlab/es/s2dv/-/issues/113

CDORemap returned error for crop = T and global grid

- The function adjusts the crop window if possible in order to keep lons from 0 to 360 or from -180 to 180 when the extremes of the cropped window are contiguous.
- But a condition was not satisfied due to a slight difference in the significant figures, originated from different R versions (predict function results) ⟶ Added tolerance in the condition

issues: https://earth.bsc.es/gitlab/es/s2dv/-/issues/110

status: in master

startR

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Reminder: Data size limitation of 16 GB in Start()

`startR::Start()` uses the bigmemory package to efficiently create large matrix objects. However, loading netCDF data with `Start()` fails if the total size of the data requested is more than 16 GB. **This is the theoretical size limit of a single R vector, even if more RAM is available**. We do not currently know if bigmemory or other packages can provide a workaround. See the GitLab issue for details.

If you need to load larger datasets, you can:

1. Set the parameter `retrieve = FALSE` and process the data in chunks using Compute(). See the use cases (section 2) and practical guide.
2. Manually split your workflow into smaller pieces and process them separately.

If you have any insight, don't hesitate to comment on the issue!

ClimProjDiags

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

New release ClimProjDiags 0.3.3 (25-01-2024)

NEWS

- Bugfix in Subset() for drop = T, didn't have correct given output dimension therefore returned error.
  - It happened when the subset dimension is > 1 and the output dimension is of length 1

```
arr <- array(1:3, dim = c(dat = 1, var = 1, sdate = 3))
Subset(arr, "sdate", 1:2, drop = T)

Error in dim(subset) <- dim(x)[dim_names] :
 dims [product 3] do not match the length of object [2]
```

status: Installed

MR: https://earth.bsc.es/gitlab/es/ClimProjDiags/-/merge_requests/48

CSTools

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

New release CSTools 5.2.0 (25-01-2024)

NEWS

Development

- New function CST_ChangeDimNames
- CST_SplitDim: added dimension names and split also Dates
- CST_SaveExp: save time bounds and global attributes; improved code

Other

- Updated README
- Added citation file

New function CST_ChangeDimNames

★ Change the name of one or more dimensions for an object of class s2dv_cube. The coordinate names and the dimensions of any attributes are also modified accordingly.

```
# (1) Check original dimensions and coordinates
lonlat_temp$exp$dims
names(lonlat_temp$exp$coords)
dim(lonlat_temp$exp$attrs$Dates)
# (2) Change 'dataset' to 'dat' and 'ftime' to 'time'
exp <- CST_ChangeDimNames(lonlat_temp$exp, original_names = c("dataset", "ftime", "lon", "lat"),
                          new_names = c("dat", "time", "longitude", "latitude"))
# (3) Check new dimensions and coordinates
exp$dims
names(exp$coords)
dim(exp$attrs$Dates)
```

dataset	member	sdate	ftime	lat	lon
1	15	6	3	22	53

dat	member	sdate	time	latitude	longitude
1	15	6	3	22	53

status: Installed

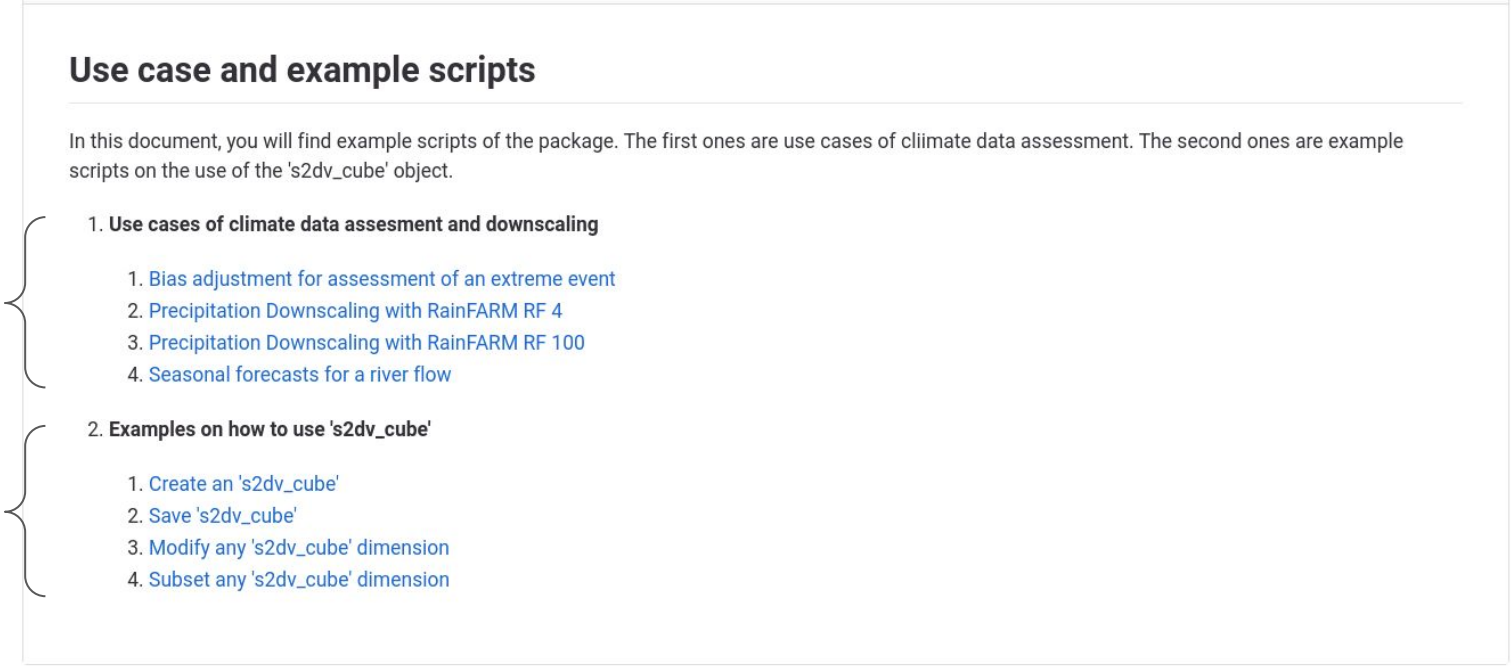
MR: https://earth.bsc.es/gitlab/external/cstools/-/merge_requests/202

New section use cases

★ New section with use cases and example scripts (under /inst/doc): https://earth.bsc.es/gitlab/external/cstools/-/blob/master/inst/doc/usecase.md

Use cases of climate assesment

Examples of ‘s2dv_cube’ class methods



Use case and example scripts

In this document, you will find example scripts of the package. The first ones are use cases of cliimate data assessment. The second ones are example scripts on the use of the 's2dv_cube' object.

1. **Use cases of climate data assesment and downscaling**
    1. Bias adjustment for assessment of an extreme event
    2. Precipitation Downscaling with RainFARM RF 4
    3. Precipitation Downscaling with RainFARM RF 100
    4. Seasonal forecasts for a river flow
2. **Examples on how to use 's2dv_cube'**
    1. Create an 's2dv_cube'
    2. Save 's2dv_cube'
    3. Modify any 's2dv_cube' dimension
    4. Subset any 's2dv_cube' dimension

status: In master and in CRAN

Developments in CST_SplitDim

- ★ Changed hard-coded parts by adding dimension names parameters (ftime_dim and sdate_dim)
- ★ Added also the split for time dimensions in Dates of ‘s2dv_cube’

Is missing still the spatial coordinates development. Currently, the function doesn't update the coordinates and remain the original ones.

```
# (1) Decadal example: We define an array of consecutive days of different years:
dates <- seq(as.Date("01-01-2000", "%d-%m-%Y", tz = 'UTC'),
             as.Date("31-12-2005","%d-%m-%Y", tz = 'UTC'), "day")
dim(dates) <- c(time = 2192)
# (2) Now, we will split the array in a new 'year' dimension:
dates_year <- SplitDim(dates, indices = dates, split_dim = 'time', freq = 'year')
dim(dates_year)
# time year
#  366    6
```

status: In master and in CRAN

issues: https://earth.bsc.es/gitlab/external/cstools/-/issues/145

Bugfix in CST_Subset

The function returned incorrect $coords values when only subsetting along one dimension.
The bug is now fixed and included in the 5.2.0 release

```
# We tried to subset ftime indices 1 to 3, but the coordinates displayed only one index
exp <- CSTools::lonlat_prec_st$exp
exp_subset <- CST_Subset(exp, along = "ftime", indices = 1:3, drop = F)
exp_subset$dims
# dataset     var  member   sdate   ftime     lat     lon
#       1       1       6       3       3       4       4
exp_subset$coords$ftime
# [1] 1
# attr(,"indices")
# [1] TRUE
```

status: Installed

MR: https://earth.bsc.es/gitlab/external/cstools/-/merge_requests/201

CSIndicators

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

New release CSIndicators 1.1.1 (2024-01-24)

NEWS

Fixes

- Corrected error in SelectPeriodOnDates to allow dates to be transposed

Other

- Included CITATION file in the pacakge

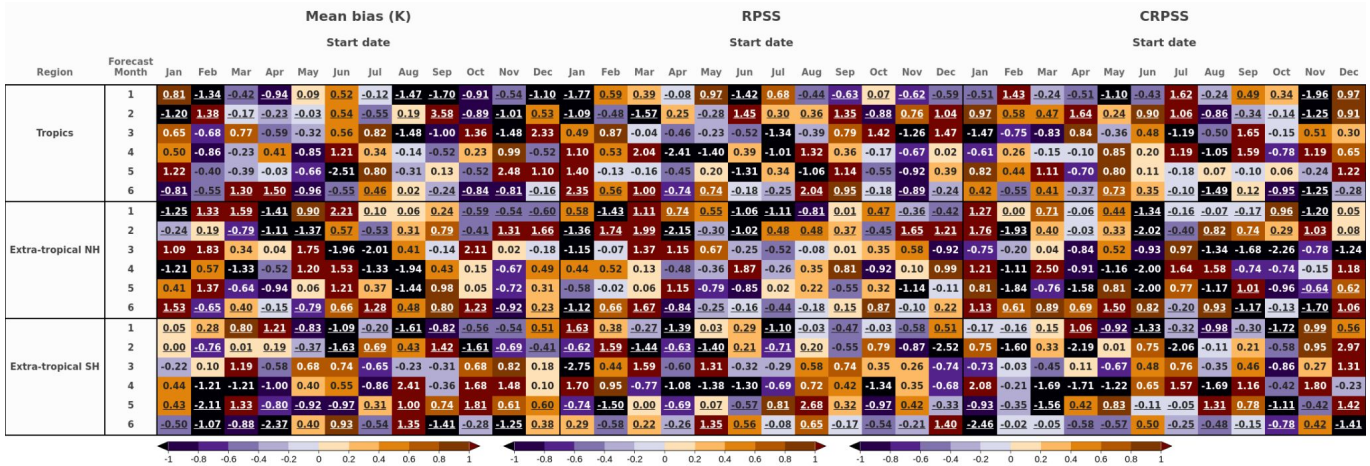
esviz

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Significance in VizScorecards

★ The function VizScorecards is on **esviz** with the added parameter **sign**

```
data <- array(rnorm(1000), dim = c('sdate' = 12, 'metric' = 3, 'region' = 3, 'time' = 6))
sign <- array(c(rep(T, 200), rep(F, 300), T, F, rep(T, 200), rep(F, 162)), dim = dim(data))
row_names <- c('Tropics', 'Extra-tropical NH', 'Extra-tropical SH')
col_names <- c('Mean bias (K)', 'RPSS', 'CRPSS')
VizScorecard(data = data, sign = sign, row_names = row_names, col_names = col_names, row_title = 'Region',
             subcol_names =  month.abb[as.numeric(1:12)], col_title = 'Start date')
```



status: In main (esviz repo)

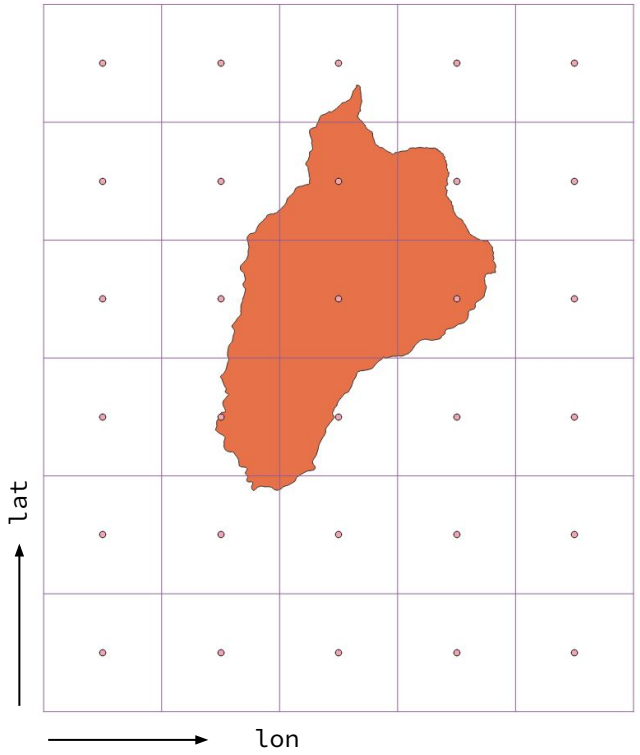
issues: https://earth.bsc.es/gitlab/es/esviz/-/issues/6

ShapeToMask area coverage ratio

★ The mask can return the coverage ratio between the intersection of the shapefile and the grid cell.

```
ref_grid <- list(lon = seq(-75.7, -75.3, 0.1),
                 lat = seq(4.6, 4.1, -0.1))
mask <- ShapeToMask(shp_file = small_region, shp_system = NULL,
                    compute_area_coverage = TRUE, ref_grid = ref_grid)
```

```
> mask   # [lon = 5, lat = 6, region = 1]
, , 1                                          → lat
        [,1]     [,2]     [,3]     [,4]     [,5] [,6]
[1,] 0.000000 0.000000 0.000000 0.000000 0.000000    0
[2,] 0.000000 0.042423 0.336787 0.447728 0.026097    0
[3,] 0.054301 0.791287 0.998896 0.473483 0.010993    0
[4,] 0.000000 0.363182 0.595227 0.000000 0.000000    0
[5,] 0.000000 0.000000 0.000000 0.000000 0.000000    0
↓ lon
```



status: In branch [develop-ShapeToMask_area]

issues: [https://earth.bsc.es/gitlab/es/esviz/-/issues/4#note_253510]

SUNSET

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Sample data and conda environment use case

New use case: SUNSET Conda environment installation + Example of a skill assessment workflow using the sample data from CSTools.
It is designed to be an example that external users can run even without data of their own. The conda environment installs all dependencies.

You can find it on GitLab:
https://earth.bsc.es/gitlab/es/sunset/-/blob/master/use_cases/ex0_1_sample_dataset/ex0_1-handson.md

status: in master

Sample data and conda environment use case

The sample dataset used is `CSTools::lonlat_temp_st` . It consists of hindcast (ECMWF SEAS5) and reference (ERA5) surface temperature data, for the period 2000-2006.

The `Run:Filesystem: 'sample'` option in the recipe will return this dataset. The variable, time period and region are fixed.

Find an example of a recipe in the new use case:
https://earth.bsc.es/gitlab/es/sunset/-/blob/master/use_cases/ex0_1_sample_dataset/ex0_1-recipe.yml

status: in master

Bug in Units(): transformation of monthly precipitation

When transforming monthly precipitation units, the Units() module incorrectly computes the number of February days for some of the years, due to a bug in the line of code that checks if the leap year correction should be applied. The bug only produces a warning in R < 4.2.x, but it becomes an error with higher versions of R.

GitLab issue: https://earth.bsc.es/gitlab/es/sunset/-/issues/99

The data is only incorrect when:

a) Unit transformation is requested for monthly precipitation, and
b) The forecast month is February.

User presentation

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Creating AeDES v2.0.0 for BSC

- Obtaining R0 indices with EnvSuitability
- Forecast calibration with Neural Networks (neuralnet & nn_plus_kcv)
- 2AFC Skill assessment (afc)

AeDES v2.0.0: A next-generation monitoring and forecasting system for environmental suitability of Aedes-borne disease transmission. It attempts to improve on earlier work (AeDES v1.0.0, Muñoz et al. 2020) on two fronts:

- Monitoring: By creating a multi-reference monitoring system
- Forecasting: By calibrating non-linear patterns using non-linear methods

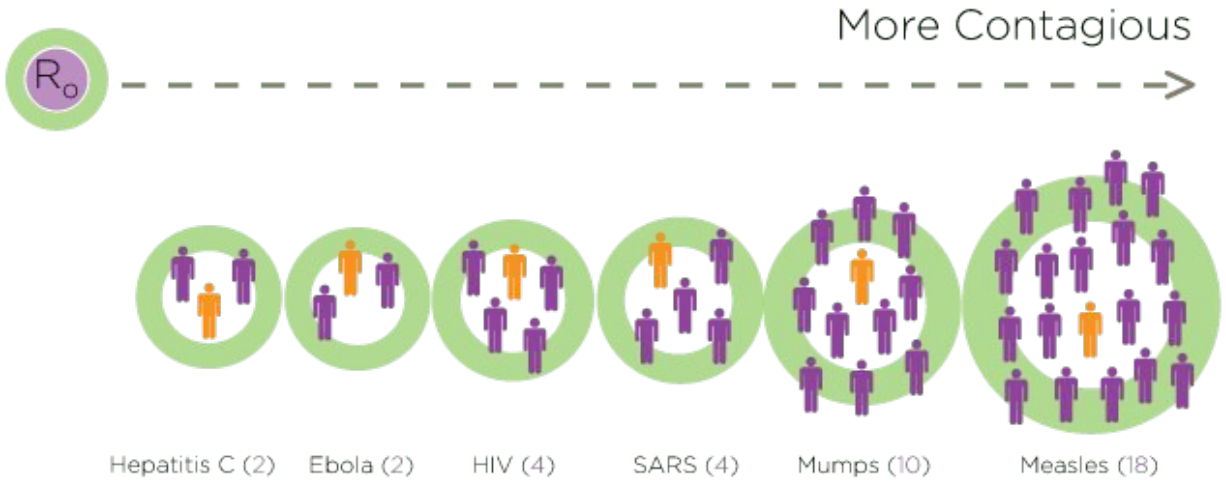
Gitlab Repo: https://earth.bsc.es/gitlab/ess/aedes

Obtaining R0 Indices with EnvSuitability

What is R0?

R0 is a positive index that outlines the environmental suitability for disease proliferation in any area…barring socio-economic conditions and human behavior (R0 is not tried and true!)

R0 values higher than 1 mean the disease is spreading, while R0 values lower than 1 means the disease is waning



Obtaining R0 Indices with EnvSuitability

For Aedes-borne diseases, there are four different empirical indexes:

- Caminade et al., 2015
- Liu-Helmerssohn et al., 2014
- Mordecai et al., 2017
- Wesolowski et al., 2015

All of these indexes are a function of temperature (in ºC) and take biological values like transmission probability, fertility rate…as constants

EnvSuitability:

Input: any s2dv_cube temperature dataset (exp, obs, or both)

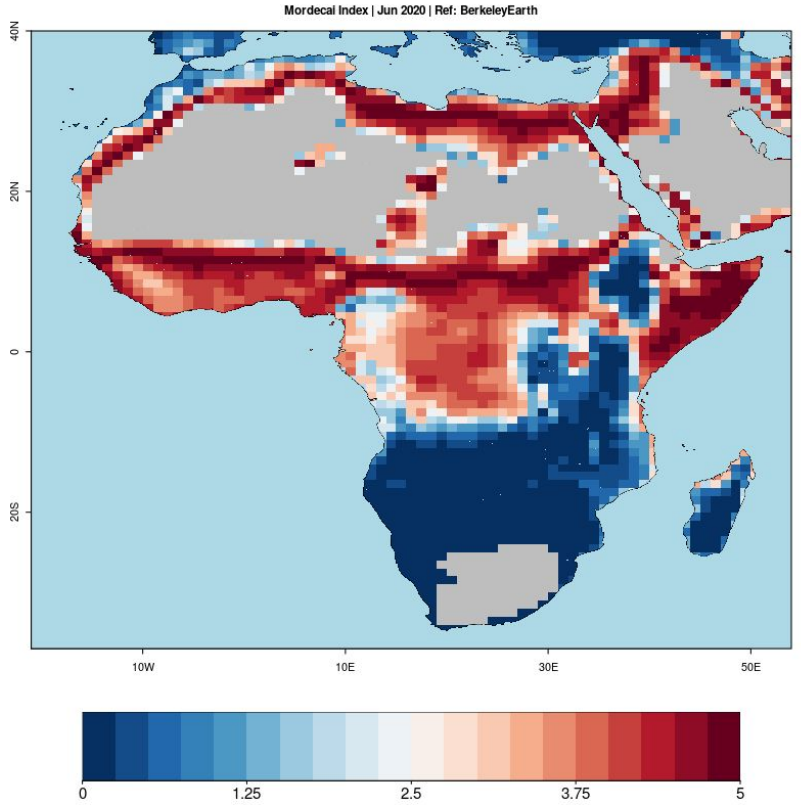
Output: a list of two (index_exp & index_obs) with the requested method in env_method (can be set to all). The list elements are also s2dv_cube objects

```
res <- EnvSuitability(
    exp = exp,
    obs = obs,
    lon_exp = lon_exp,
    lat_exp = lat_exp,
    lon_obs = lon_obs,
    lat_obs = lat_obs,
    grid_remap = grid_remap,
    region = region,
    env_method = env_method,
    exp_env_ok = exp_env_ok,
    obs_env_ok = obs_env_ok,
    rds_save = rds_save,
    filepath = filepath,
    filename = filename,
    ncores = ncores
)
```

Obtaining R0 Indices with EnvSuitability

```
res <- EnvSuitability(
    exp = exp,
    obs = obs,
    lon_exp = lon_exp,
    lat_exp = lat_exp,
    lon_obs = lon_obs,
    lat_obs = lat_obs,
    grid_remap = grid_remap,
    region = region,
    env_method = env_method,
    exp_env_ok = exp_env_ok,
    obs_env_ok = obs_env_ok,
    rds_save = rds_save,
    filepath = filepath,
    filename = filename,
    ncores = ncores
)
```

How does it look?



Forecast calibration with Neural Networks

We need non-linear calibration methods to account for non-linear patterns…

```
.nn_plus_kcv <- function(x, y, cv, hidden, kfold)
```

nn_plus_kcv performs k-fold cross validation given a forecast (y), observational reference (x) and any nn configuration (hidden)

```
nn_cal <- Apply(
    list(normalize(x), normalize(y)),
    target_dims = "sdate",
    fun = .nn_plus_kcv,
    loocv = TRUE,
    hidden = c(sample.int(4:10, 1)),
    ncores = 4
)$output1
```

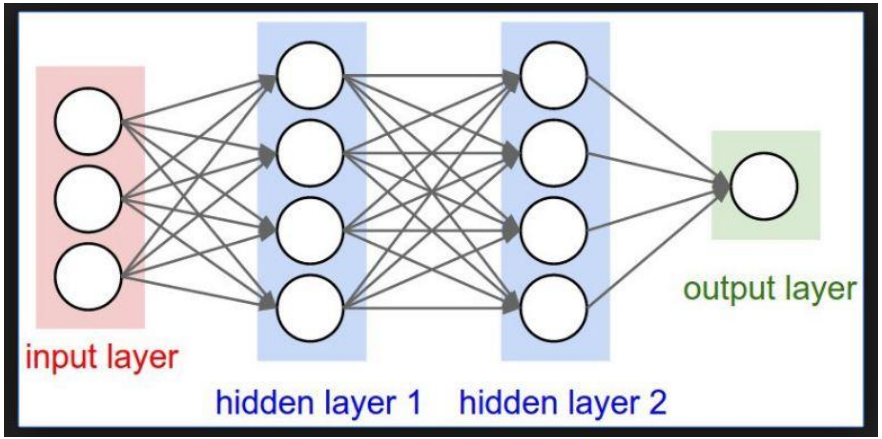
Forecast calibration with Neural Networks

We need non-linear calibration methods to account for non-linear patterns…

```
.nn_plus_kcv <- function(x, y, cv, hidden, kfold)
```

nn_plus_kcv performs k-fold cross validation given a forecast (y), observational reference (x) and any nn configuration (hidden)

```
nn_cal <- Apply(
    list(normalize(x), normalize(y)),
    target_dims = "sdate",
    fun = .nn_plus_kcv,
    loocv = TRUE,
    hidden = c(sample.int(4:10, 1)),
    ncores = 4
)$output1
```



Forecast calibration with Neural Networks

nn_plus_kcv uses the neuralnet R package, which builds simple neural networks using back propagation…just make sure to normalize first!

```
nn <- neuralnet(
  y ~ x,
  data = df,
  hidden = c(2, 1),
  linear.output = FALSE,
  threshold = 0.01
)

y_pred <- nn.results$net.result * abs(diff(range(y))) + min(y)
```

For more info on this package:
https://datascienceplus.com/neuralnet-train-and-test-neural-networks-using-r/

2AFC Skill Assessment

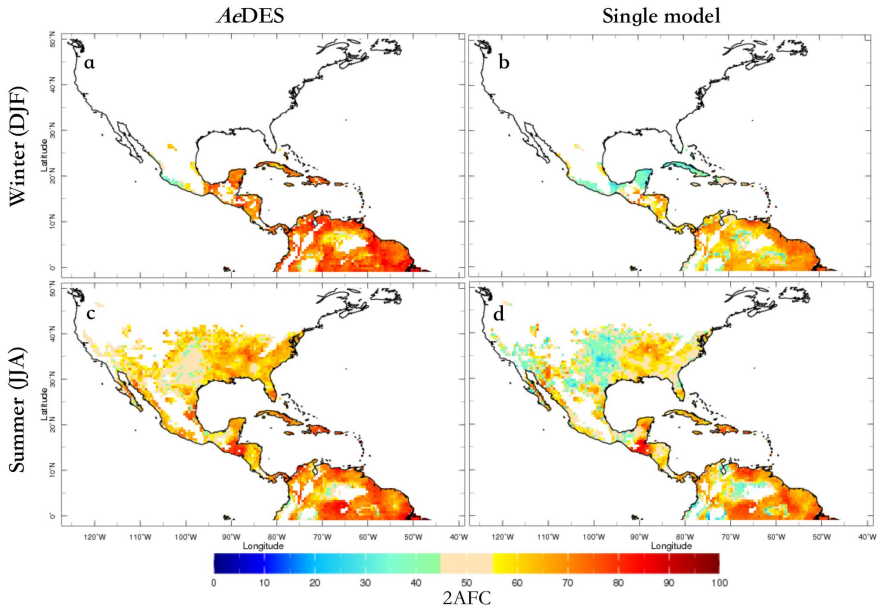
2AFC = Two Alternate Forced Choice (%)

When terciles (above-normal, normal, below-normal conditions) are used, the 2AFC measures how well the system distinguishes between the different categories

2AFC Skill Assessment can easily be done using the afc and MultiApply R packages:

```
afc.cc(obs, exp)

pcr.afc <- Apply(
    list(y_pred, x),
    target_dims = c("sdate"),
    fun = afc.cc
)$output1
```



Thanks for joining

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Next meeting: TBD