



04/05/2023



R user meeting

An-Chi Ho, Eva Rifà, Victòria Agudetse
contributor:

Agenda

1. Ice-breaker: reticulate
2. News
 - General R
 - s2dv
 - startR
 - CSTools
 - CSIndicators
 - Verification Suite
3. Hands-on: How do you find PlotRobinson()?
4. Q&A

Ice-breaker



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

Package reticulate

It is an R package that embeds a Python session within your R session. It calls Python from R being able to translate objects within both packages.

How to use reticulate? Different methods

- a. **Importing Python modules** — The `import()` function can import any Python module and call its functions directly from R.
- b. **Sourcing Python scripts** — The `source_python()` function can source a Python script the same way as `source()` an R script.
- c. **Python REPL** — The `repl_python()` function creates an interactive Python console within R
- d. **Python in R Markdown** — R chunks can access Python objects and vice-versa.

Package reticulate: Example I

Terminal

a. Importing Python modules

```
# Example 1: Call Python matplotlib
module load
matplotlib/3.2.2-foss-2015a-Python-3.7.3
module load Python/3.7.3-foss-2015a
module load R/4.1.2-foss-2015a-bare

library(reticulate)
matplotlib <- import("matplotlib")
# Generate some data
x <- c(1, 2, 3, 4, 5)
y <- c(2, 4, 6, 8, 10)
# Create a scatter plot
plt <- matplotlib$pyplot
plt$scatter(x, y)
plt$savefig(fname = "plot1.png")
```

R session

```
# Example 2: Call Python xarray
module load xarray/0.20.0-foss-2015a-Python-3.7.3
# Other dependencies of xarray
module load netcdf4-python/1.5.3-foss-2015a-Python-3.7.3
module load numpy/1.20.3-foss-2015a-Python-3.7.3

library(reticulate)
xr <- import("xarray")
obs <-
"/esarchive/recon/ecmwf/erainterim/monthly_mean/tas_f6h/"
obs_file <- paste0(obs, "tas_201811.nc")
obs_out <- xr$open_dataset(obs_file)

> obs
<xarray.Dataset>
Dimensions: (time: 1, lat: 256, lon: 512)
```



Package reticulate: Example II

b. Sourcing Python scripts

- Save a python script:

```
import matplotlib.pyplot as plt
import numpy as np

# Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)
fig, ax = plt.subplots()
ax.plot(t, s)
ax.set(xlabel = 'time (s)', ylabel = 'x (m)',
       title = 'Created with R')
ax.grid()
fig.savefig("plot2.png")
```

- Source it from R session:

```
library(reticulate)

source_python("/esarchive/scratch/erifarov/rpa
ckages/cstools-local/dev-xarray/test_reticulat
e_source.py")
```

Package reticulate: Example III

c. Python REPL

Inside R session:

```
library(reticulate)
repl_python()

import matplotlib.pyplot as plt
import numpy as np

# Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)
fig, ax = plt.subplots()
ax.plot(t, s)
ax.set(xlabel = 'time (s)', ylabel = 'x (m)',
       title = 'Created with R')
ax.grid()
fig.savefig("plot2.png")
```

General R



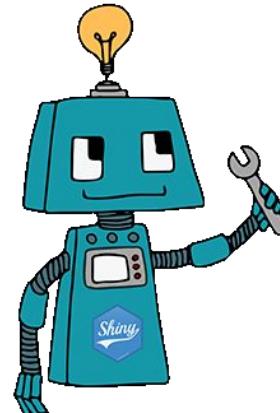
**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

R-community activity

Upcoming R conferences:

- **useR!** - They don't do it this year
- **Spain R community**: November 2023 in Barcelona (UPF)
- **posit::conf(2023)** - from September 17 to September 20 in Chicago and online
- **Shiny in Production** on 12th - 13th October 2023



shapefile function

The function now depends on **sf** only.

Check the issue: <https://earth.bsc.es/gitlab/es/s2dv/-/issues/75>

Progress report:

[Progress on R-spatial evolution, Apr 2023](#)

[Progress on R-spatial evolution, Dec 2022](#)

[R-spatial evolution: retirement of rgdal, rgeos and maptools](#)

Suggestion: Migrate from raster to terra; from sp to sf / stars

Which sf or terra functions matche the retiring functions? Find explanation and examples [here](#). The comparison of sp/rgeos/rgdal and sf functions: [here](#)

In sp >= 1.6.0, you can:

```
> options("sp_evolution_status"=2)
> library(sp)
{sp 1.6.0 is installed in Nord3v2}
```

*0: business as usual,
1: stop if rgdal or rgeos are absent, or
2: use sf instead of rgdal and rgeos*

s2dv



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

RMS p-value

We decided to remove the p-value output since we're not sure if it is correct.

The function returns RMS and confidence interval.

Issue: https://earth.bsc.es/gitlab/es/s2dv/-/issues/79#note_210124

RPS, RPSS, ROCSS input to be probabilities

These three functions took data as inputs, calculated probabilities, then calculated the (skill) scores.

Now, you can directly provide probabilities as input. It's more efficient if you need to calculate several skill scores and they all need probabilities.

Status: In master

```
#' @param exp A named numerical array of either the forecasts with at least time  
#' and member dimensions, or the probabilities with at least time and category  
#' dimensions. The probabilities can be generated by \code{s2dv::GetProbs}.
```

PlotRobinson

Vignette:

https://earth.bsc.es/gitlab/es/s2dv/-/blob/develop-PlotRobinson/vignettes/visualization_projection.md

Problems:

- Projection ID varies from PROJ version. You may get different problems on different machines.
- Polygon doesn't work on WS (memory is not mapped)
- Polygon is too slow (and wrong?) to plot global
- parameter “crop_coastlines” doesn't work if longitude range is > 180 degrees

Status: in branch [develop-PlotRobinson](#)

Issue: <https://earth.bsc.es/gitlab/es/s2dv/-/issues/95>



startR



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

New documentation

Weekly ECV Subseasonal Hindcast Verification ([ex3_1](#))

This is a practical case to compute monthly skill scores for the ECMWF/S2S-ENSForhc subseasonal hindcast using as a reference dataset ERA5.

RainFARM precipitation downscaling ([ex2_5](#))

This example shows how to apply a statistical downscaling function with startR and simultaneously saves the data by chunks in the esarchive format.

FAQ: Utilize chunk number in the function ([How-to-27](#))

CSTools



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Standarize coordinates in s2dv_cube

In the version CSTools 5.0.0 the **s2dv_cube** coordinates are defined as 1D arrays. Some coordinates could have an assigned dimension (lon, lat) and others not (e.g. indices).

Example:

```
$ coords:List of 6
..$ dataset: chr "exp1"
..$ member : int [1:6] 1 2 3 4 5 6
..$ sdate  : chr [1:3] "20101101" "20111101" "20121101"
..$ ftime   : int [1:31] 1 2 3 4 5 6 7 8 9 10 ...
..$ lat     : num [1:4(1d)] 47 46 45 44
..$ lon     : num [1:4(1d)] 6 7 8 9
```

Now, the functions to create the object will assign its dimension to NULL. With this change, it is easier to index and subset coordinates as vectors are simpler than arrays.

Changes:

- Functions: s2dv_cube, as.s2dv_cube, sample data and CST_Subset

Standarize coordinates in s2dv_cube

Examples of this development:

(1) Coordinates dimensions are NULL

```
> str(lonlat_prec$coords)
List of 6
$ coords:List of 6
..$ dataset: chr "exp1"
..$ member : int [1:6] 1 2 3 4 5 6
..$ sdate  : chr [1:3] "20101101" "20111101" "20121101"
..$ ftime   : int [1:31] 1 2 3 4 5 6 7 8 9 10 ...
..$ lat     : num [1:4] 47 46 45 44
..$ lon     : num [1:4] 6 7 8 9

> dim(lonlat_prec$coords$lon)
NULL
```

(2) Variable metadata can have dimensions

```
> str(lonlat_prec$attrs$Variable$metadata)
List of 3
$ prlr:List of 7
..$ use_dictionary      : logi FALSE
..$ units                : chr "m s-1"
$ lon : num [1:4(1d)] 6 7 8 9
$ lat : num [1:4(1d)] 47 46 45 44
> dim(lonlat_prec$attrs$Variable$metadata$lon)
[1] 4
```

Filenames for monthly data in CST_SaveExp

How to save monthly data?

- Time values of monthly mean example:

```
from "/esarchive/exp/ecmwf/system5c3s/monthly_mean/pls_f6h/pls_19930101.nc"
  ○ time: units = "hours since 1993-01-01 00:00:00" ;
    ■ time = 738, 1410, 2154, 2874, 3618, 4338, 5082, 5160 ;
```

- Start retrieval gives:

```
> attributes(hcst)$Variables$common$time
[1] "1993-01-31 18:00:00 UTC" "1994-01-31 18:00:00 UTC"
[3] "1995-01-31 18:00:00 UTC" "1993-02-28 18:00:00 UTC" ...
```

- Using the function CST_SaveExp will take the dates “pls_19930131.nc” and the time values will be:

```
○ time:units = "hours since 1993-01-31 18:00:00" ;
  ■ time = 0, 672 ;
```

Solution? (1) Add a parameter to know the time frequency (2) Add a parameter for filenames

ESS Verification Suite



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

New output structure

Instead of having to call the `save_data()` function, **the output of each module is now saved at the end of the computation.**

A folder for each module is created inside of `/outputs/`, and the netCDF files are saved inside these subfolders.

The user should indicate which outputs they want to save for each module in the ‘Workflow’ section of the recipe, using the parameter ‘`save`’.

status: in master

New output structure

Workflow:

Anomalies:

```
compute: yes
cross_validation: yes
save: 'all' # options: 'all', 'none', 'exp_only', 'fcst_only'
```

Calibration:

```
method: raw
save: 'none' # options: 'all', 'none', 'exp_only', 'fcst_only'
```

Skill:

```
metric: RPS, RPSS, BSS10, BSS90, mean_bias, mean_bias_SS
save: 'all' # options: 'all', 'none'
```

Probabilities:

```
percentiles: [[1/3, 2/3], [1/10, 9/10], [1/4, 2/4, 3/4]]
save: 'all' # options: 'all', 'none', 'bins_only', 'percentiles_only'
```

Choosing plots

New ‘Visualization’ section in the recipe to specify which plots you want:

```
...
```

```
Workflow:
```

```
    Visualization:
```

```
        plots: skill_metrics, forecast_ensemble_mean, most_likely_terciles
```

This parameter is used when calling the `plot_data()` function. **The plots are all saved in the /plots/ subdirectory** inside the output directory.

status: in master

Option to disable unique ID in output folder

A new ‘uniqueID’ parameter in the `prepare_outputs()` function can be set to FALSE to avoid attaching unique ID to the output dir. Not recommended for general use, but it can be useful when testing code.

```
# Remove unique ID:  
> recipe <- prepare_outputs(recipe_file, uniqueID = FALSE)  
[1] "Saving all outputs to:"  
[1] "/esarchive/scratch/vagudets/auto-s2s-outputs/recipe_system7c3s-tas"  
  
# Default behavior:  
> recipe <- prepare_outputs(recipe_file)  
[1] "Saving all outputs to:"  
[1] "/esarchive/scratch/vagudets/auto-s2s-outputs/recipe_system7c3s-tas_20230427121124"
```

To disable the unique ID when splitting a recipe into atomic recipes:

```
Rscript split.R <path_to_recipe>--disable_unique_ID
```

status: in master

Skill module: cross-validation and climatology RPS

New parameter `cross_validation` can be added to the recipe in the Skill module section. The default value is ‘no/FALSE’. If it is set to TRUE, cross-validation will be applied when computing the following metrics:

- RPS(S), FRPS(S), BSS10 and BSS90

Also, when requesting RPS, FRPS or CRPS, **the scores for the climatology will also be returned**, as `rps_clim`, `frps_clim` and `crps_clim`, respectively.

status: in master

Hands-on: PlotRobinson



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

PlotRobinson

Use Nord3v2 since “polygon” option doesn’t work well on workstation.

Function:

<https://earth.bsc.es/gitlab/es/s2dv/-/blob/develop-PlotRobinson/R/PlotRobinson.R>

Vignette:

https://earth.bsc.es/gitlab/es/s2dv/-/blob/develop-PlotRobinson/vignettes/visualization_projection.md

Try the function and share what problem you find or any suggestion. Put them in the minutes:

https://docs.google.com/document/d/1mdLCS7SC5-8NwAB_gAzUsEnJN4yYEYmAlVys0qT8Edk/edit

Q & A



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Thanks for joining

Next meeting: 1st June, 12h