

BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación

# Create R package in BSC-ES

An-Chi Ho

## **Department R Packages Review**



Barcelona Supercomputing Center Centro Nacional de Supercomputación

### **BSC-ES R packages**

- ★ Functions are split on packages depending on their objective
- ★ Functions from different packages (including external packages) can be used to perform analyses or obtain climate service products

	Package name	Short description	Link to CRAN and GitLab
Data loading	easyNCDF	Read/write netCDF files into/from multidimensional R array.	https://CRAN.R-project.org/package=easyNCDF https://earth.bsc.es/gitlab/es/easyNCDF
and manipulation	startR	Data retrieval and processing tools	https://CRAN.R-project.org/package=startR https://earth.bsc.es/gitlab/es/startR
	multiApply	Apply functions to multiple multidimensional arrays or vectors allowing parallel computation	https://CRAN.R-project.org/package=multiApply https://earth.bsc.es/gitlab/ces/multiApply
	s2dv	Functions for Forecast Verification and visualization	https://CRAN.R-project.org/package=s2dv https://earth.bsc.es/gitlab/es/s2dv
Analysis and processing	CSTools	Methods for forecast calibration, statistical and stochastic downscaling, optimal forecast combination and tools to obtain tailored products.	https://CRAN.R-project.org/package=CSTools https://earth.bsc.es/gitlab/external/cstools
	CSIndicators	Sectorial Indicators for Climate Service	https://CRAN.R-project.org/package=CSIndicators https://earth.bsc.es/gitlab/es/csindicators
Climate indicators		Climate extreme indices, evaluation of the agreement between models, weight and combination functions.	https://CRAN.R-project.org/package=ClimProjDiag <u>S</u> https://earth.bsc.es/gitlab/es/ClimProjDiags

## **BSC-ES R packages**

### Important features in our tools:

### LOADING

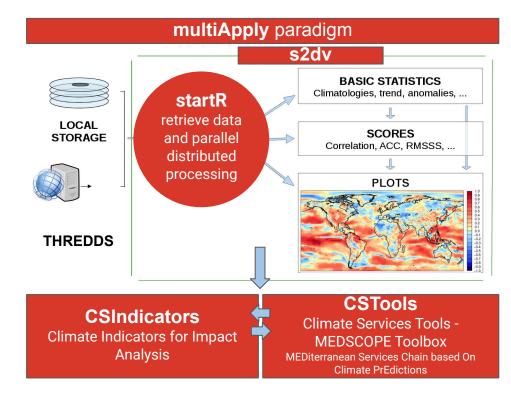
- Input data format: netCDF
- Different datasets to be loaded
  - Data loading flexibility required

### ANALYZING

- Accepted R object type by functions: Named multi-dimensional array mainly
- Different forecast horizons and frequency to be analyzed
  - Function flexibility required
- Multiple-core/node and parallel computation on HPCs
  - use package "multiApply"



New methods and packages for different needs are being developed continuously.



### Function structure: From a developer's view

#### 

(Bias.R from s2dv)

1	#'Compute the Mean Bias
2	#'
3	#'The Mean Bias or Mean Error (Wilks, 2011) is defined as the mean difference
4	#'between the ensemble mean forecast and the observations. It is a deterministic
5	#'metric. Positive values indicate that the forecasts are on average too high
6	#'and negative values indicate that the forecasts are on average too low.
7	#'It also allows to compute the Absolute Mean Bias or bias without temporal
8	#'mean. If there is more than one dataset, the result will be computed for each
9	#'pair of exp and obs data.
10	# pair of exp and obs data. #'
10	" #'@param exp A named numerical array of the forecast with at least time
12	# uparam exp A named numerical array of the forecast with at least time #' dimension.
12	# 0111/PITS1011.
49	#'@export
50	Bias <- function(exp, obs, time dim = 'sdate', memb dim = NULL, dat dim = NULL, na.rm = FALSE,
51	absolute = FALSE, time mean = TRUE, ncores = NULL) {
52	
53	# Check inputs
54	## exp and obs (1)
55	<pre>if (!is.array(exp)   !is.numeric(exp))</pre>
56	star/"Denamatan "ava" mist ka a nimeris erau "1
138	## (Mean) Bias
139	<pre>bias &lt;- Apply(data = list(exp, obs),</pre>
140	<pre>target_dims = c(time_dim, dat_dim),</pre>
141	fun = .Bias,
142	time_dim = time_dim,
143 144	dat_dim = dat_dim,
144	na.rm = na.rm, absolute = absolute,
145	time mean = time mean,
147	ncores = ncores)Soutput1
148	1
149	return(bias)
150	
151	
153	.Bias <- function(exp, obs, time_dim = 'sdate', dat_dim = NULL, na.rm = FALSE,
154	absolute = FALSE, time_mean = TRUE) {
155	<pre># exp and obs: [sdate, (dat)]</pre>
156	

#### A typical in-house function would be like:

**Header**: Documentation following Roxygen2 convention

**Function**: Work with named multi-dimensional array, with multiple cores option

**Sanity check**: First step in the function. Check all the input parameters

**Computation**: Usually short, use multiApply::Apply to allow flexible dimensions and multiple cores.

**Atomic function**: Work with fixed essential dimensions, store the main analytical code. Non-exported (i.e., no documentation needed)

### Function structure: From a developer's view

Barcelona Supercomputing Center

entro Nacional de Supercomputación

**Q CST BiasCorrection.R** [2 15.60 KiB (CST BiasCorrection.R from CSTools) A CST prefix function would be like: #'Bias Correction based on the mean and standard deviation adjustment Same but with a top level CST \* function #'@author Verónica Torralba, \email{veronica.torralba@bsc.es} #'@description This function applies the simple bias adjustment technique #'described in Torralba et al. (2017). The adjusted forecasts have an equivalent #'standard deviation and mean to that of the reference dataset. #'@param exp An object of class \code{s2dv\_cube} as returned by \code{CST\_Load} #'@export CST\_BiasCorrection <- function(exp, obs, exp\_cor = NULL, na.rm = FALSE, memb dim = 'member', sdate dim = 'sdate', 61 **CST Function**: Work with s2dv cube, a 62 dat dim = NULL, ncores = NULL) { wrapper of without CST prefix function # Check 's2dv cube' if (!inherits(exp, 's2dv cube') || !inherits(obs, 's2dv cube')) { 64 stop("Parameter 'exp' and 'obs' must be of the class 's2dv cube'.") **Sanity check**: For s2dv cube object 67 if (!is.null(exp\_cor)) { if (!inherits(exp cor, 's2dv cube')) { stop("Parameter 'exp cor' must be of the class 's2dv cube'.") 3 72 BiasCorrected <- BiasCorrection(exp = exp\$data, obs = obs\$data, exp cor = exp cor\$data, **Call function**: Call without prefix function, 74 memb dim = memb dim, sdate dim = sdate dim, dat dim = dat dim inputs are assigned from the s2dv cube 75 na.rm = na.rm, ncores = ncores)

The rest part is the same as previous slide.

## **Build an R Package in BSC-ES**



Barcelona Supercomputing Center Centro Nacional de Supercomputación

## Start on GitLab: Create a new project

¥	= Menu	D		Create n	ew project	
	-	<ul> <li>&gt; Q Search your projects</li> <li>&gt; Frequently visited</li> <li>\$2dv Earth Sciences</li> <li>\$startR</li> </ul>	+	Create blank project Create a blank project to house your files, plan your work, and collaborate on code, among other things.	+	Create from template Create a project pre-populated with the necessary files to get you started quickly.
	S Activity	startR         Earth Sciences         CSTools         External         requests         Earth Sciences         A         auto-S2S         Earth Sciences		Import project Migrate your data from an external source like GitHub, Bitbucket, or another instance of GitLab.		
		Your projects Starred projects Explore projects Explore topics Create new project	•	tion from GitLa gitlab.com/ee/		



## Start on GitLab: Create a new project

New project > Create blank project

Project name		
My awesome project	s	lug: the project name in URL (auto-filled by "Project name")
Project URL		Project slug
https://earth.bsc.es/gitlab/	Pick a group or namespace 🗸 🗸	my-awesome-project
Want to house several depende		eate a group.
Project description (optional)	Q Search	
Description format	Groups Under which	n group should this package be?
	ces	
	es	
Visibility Level 🕜	focus-africa-bsc	
O ✿ Private Project access must be @	es/ifs	t is part of a group, access will be granted to members of the group.
●          ● Internal         The project can be acces         ■         ■         ■	Users aho nal u:	sers.
O ⊕ Public The project can be access	sed without any authentication.	
Project Configuration	Choose	the suitable visibility level (internal or public)
Initialize repository with a RE	ADME	
Allows you to immediately cl	one this project's repository. Skip this if you	u plan to push up an existing repository.
Enable Static Application Sec		· · · · · · · · · · · · · · · · · · ·
Analyze your source code for	r known security vulnerab	the <b>project name</b> and <b>slug</b> wisely since the change afterward may cause
Create project Cancel	much	n inconvenience. Other details can be further modified without problem.

## Start on GitLab: Create a new project

E exampleR D Project ID: 1226		□ → Star 0 v Fork
1 Commit 🛛 🖗 1 Branch 🛷 0	) Tags 🛛 🖬 61 KB Project Storage	
nain v examp	oler / [+ ~]	Find file Web IDE
		Clone with SSH
Initial commit aho authored just now		git@earth.bsc.es:aho/exampler.git
		Clone with HTTPS
README	Add CHANGELOG	ING 🕀 Enα https://earth.bsc.es/gitlab/aho/ε 🗗
3 Set up CI/CD	e Integrations	Open in your IDE
Name	Last commit	Visual Studio Code (SSH)
* README.md	Initial commit	Visual Studio Code (HTTPS)
READIVIE.IIIU	initial continu	IntelliJ IDEA (SSH)
README.md		IntelliJ IDEA (HTTPS)
exampleR		

Already a pro? Just edit this README.md and make it your own. Want to make it easy? Use the template at the bottom!

Once clicking "Create project" in the previous page, you'll land on the project main page.

Clone the project to your repository (recommended path: under /esarchive/scratch/<*userID*>/) > git clone <HTTPS URL>

Now, you can start building the package both on GitLab and with git.

First level of the package:

	Fundamental information about the package
—— NAMESPACE	Imports and Exports. Automatically generated by roxygen2.
—— R/	Store the .R files, i.e., R functions.
man/	Store the .Rd files. Automatically generated by roxygen2.
—— NEWS.md	News for each release (optional)
tests/	Unit test <mark>(optional)</mark>
vignettes/	To demonstrate the usage of the package (optional)



## Build the R package structure

First level of the package — helper files:

gitignore	Folders and files to ignore when git push to GitLab.( <i>ref</i> )
gitlab-ci.yml	Run GitLab CI/CD pipeline.
	Folders and files to ignore when R package is built (ref)

Too many things to build?  $\rightarrow$  Take the files from the existing R packages and modify them for your need.



т

## Recommended package: roxygen2

- An <u>R package</u> to make documenting the code as easy as possible.
- Automatically generating . Rd files under folder man/ and file NAMESPACE, and will manage the Collate field in DESCRIPTION.

How to run it:

Under the git repo of the package, run devtools::document() in R session.



## **Recommended package: testthat**

- An <u>R package</u> to build unit tests for R functions.
- Under folder test/, folder testthat/ to store the unit test file for each function; file testthat. R to run the tests.

How to run it:

- Under the git repo of the package, run devtools::test() in R session.
- library(testthat) then source the function and unit test file.
- etc.



## **R CMD** build and check

- To build a package from the git project, go one layer above the git folder, and run R CMD build <folder name>. A .tar.gz file will be generated.
- To check if the package is accepted by CRAN, run R CMD check --as-cran
   <.tar.gz file>. You should get 0 error and a couple of acceptable warnings due to our system environment.
- The .tar.gz file is the one to be submitted to CRAN and/or be installed as a package.



## How to start? Some tips...

- 1. Get familiar with one BSC package, understand the function usage and structure.
  - a. How to choose the package?
    - i. Need to use s2dv\_cube (functions with "CST" prefix): CSTools, CSIndicators
    - ii. No need to use s2dv\_cube: s2dv, ClimProjDiags
  - b. Read function documentation: On CRAN and on GitLab
  - c. Read and run the vignettes: On CRAN and/or on GitLab
- 2. Create a GitLab project

Choose one existing package similar to your new one, copy the folder structure and some files. Modify based on them.

3. Create R functions

Based on one function you've been familiar with, mimic its structure (header, sanity check, atomic function, etc.)



## **QUESTIONS?**



Barcelona Supercomputing Center Centro Nacional de Supercomputación