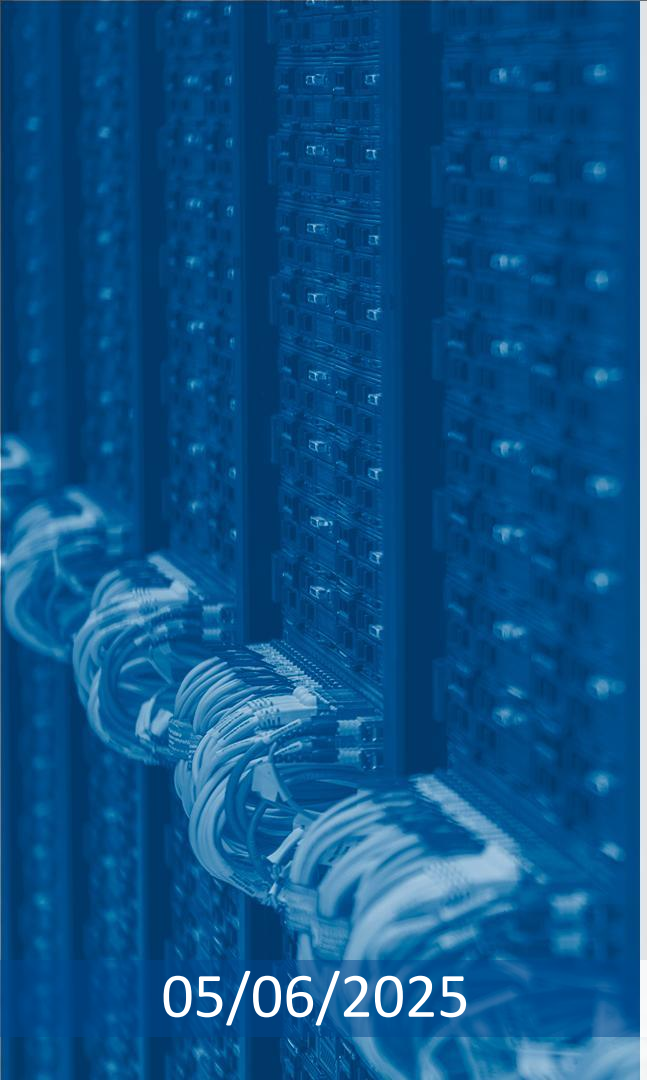


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

R user meeting

05/06/2025

Victòria A., Ariadna B., Theertha K.

Agenda

1. Ice-breaker: package ‘beepr’
2. News
    - General
    - startR
    - multiApply
    - CSTools
    - CSDownscale
    - esviz
    - SUNSET
3. User presentation: BSC-ES Infrastructure (Victòria)
4. Q&A

Ice-breaker: Package ‘beepr’

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Easily Play Notification Sounds on any Platform

Package ‘beepr’ : The main function of this package is beep(). It is intended to play notification sounds on whatever platform you are on, for example, if you are running a long analysis in the background and want to know when it is ready.

There are a few number of different sounds to choose from.

1. "ping"
2. "coin"
3. "fanfare"
4. "complete"
5. "treasure"
6. "ready"
7. "shotgun"
8. "mario"
9. "wilhelm"
10. "facebook"
11. "sword"

```
# install.packages("beepr")

# load package
library("beepr")

# play custom sound
beep(8)

# play a random sound
beep(0)

# play custom sound on error
beep_on_error(1 + "a", sound = "wilhelm")

```

General

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

RStudio and Jupyter Notebook in Hub

Several IDE options now available in the new BSC-ES Hubs:

★ RStudio in the Hub:
https://earth.bsc.es/wiki/doku.php?id=computing:bsceshub#rstudio-server_on_the_hub

★ Jupyter Notebooks with R kernel in the Hub:
https://earth.bsc.es/wiki/doku.php?id=computing:bsceshub#using_jupyter_notebooks

Reminder of the new R modules in the hub:
R/4.4.1-gfbf-2023b (+ **R-bundle-CRAN/2024.06-foss-2023b** for most CRAN Packages; and **R-bundle-Bioconductor/3.19-foss-2023b-R-4.4.1** for Bioconductor packages)

startR

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

New release 2.5.0

New startR version 2.5.0 is installed in the WS, Hub and HPC machines. C**RAN submission is pending due to an issue with a dependency**.

The main features of this new release are:

★ New special setup to use Compute() on Nord4.

★ Improvements to use Compute() with Autosubmit more smoothly, making the experiment creation step more flexible and consistent.

★ Support for conda environments when using Compute().

★ Bugfix in Start() to correctly round latitude and longitude values in netCDF files.

You can find the full list of changes and all details on GitLab.

multiApply

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Apply(): Matching names in input ‘data’ to ‘fun’

Apply() applies a function ‘fun’ to a list of arrays ‘data’. In the current version of the multiApply(), the elements in ‘data’ are expected to have the same order as the arguments in ‘fun’. **List names are not matched to the arguments**:

```
fun <- function(x, y, z) {
  return(x + y - z)
}

# Correct
res <- Apply(data = list(x, y, z),
             fun = fun)$output1
# Also correct
res <- Apply(data = list(x = x, y = y, z = z),
             fun = fun)$output1
# Technically also correct, list names are ignored!
res <- Apply(data = list(x = x, z = y, y = z),
             fun = fun)$output1
# Wrong order, will return wrong result
res <- Apply(data = list(x = x, z = z, y = y),
             fun = fun)$output1
```

Apply(): Matching names in input ‘data’ to ‘fun’

The new fix in Apply() matches the list names in ‘data’ to the arguments of ‘fun’ and reorders data accordingly:

```
# Correct
res <- Apply(data = list(x, y, z),
             fun = fun)$output
# Also correct
res <- Apply(data = list(x = x, y = y, z = z),
             fun = fun)$output1
# Now wrong!
res <- Apply(data = list(x = x, z = y, y = z),
             fun = fun)$output1
# Now correct!
res <- Apply(data = list(x = x, z = z, y = y),
             fun = fun)$output1
```

issue: https://earth.bsc.es/gitlab/ces/multiApply/-/issues/20
status: in branch dev-reorder_named_data

Apply(): Matching names in input ‘data’ to ‘fun’

Functions with the `...` parameter can also be used, by keeping the corresponding elements unnamed:

```
# Function using dots:
# When defining the function, '...' should be the last of the arguments
# that will be passed onto 'data'.
test_fun <- function(x, ...) {
  sum(...) / x
}
# The elements that correspond to '...' should be unnamed:
data <- list(x = array(rnorm(1000), c(5, 10)),
             array(rnorm(1000), c(5, 10)),
             array(rnorm(1000), c(5, 10)))
test <- Apply(data, target_dims = list(1, 1, 1), test_fun)
```

issue: https://earth.bsc.es/gitlab/ces/multiApply/-/issues/20
status: in branch dev-reorder_named_data

easyNCDF

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Minor release easyNCDF v0.1.3

The easyNCDF package provides wrappers for the ncdf4 package to simplify reading from and writing to NetCDF files, offering tools to handle multi-dimensional R arrays.

Minor release updates:

- Documentation fix in `ArrayToNc()` function (takes multi-dimensional R arrays and stores them in a NetCDF file).
- Maintainer change from An-Chi Ho to Ariadna Batalla.

Status: on CRAN

s2dv

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Difference between ProbBins() and GetProbs()

ProbBins() and GetProbs() are two s2dv functions for computing categorical forecast information.

They have a similar use and it can be unclear what the specific differences are:

Difference between ProbBins() and GetProbs()

Feature	ProbBins()	GetProbs()
Input data	Typically anomalies	Forecasts/observations
Ouptut	One-hot assignments to the defined categories. Member dimension.	Probabilities for each defined category (e.g., terciles). No member dimension
Threshold type	Quantile or absolute	Quantile or absolute
Cross-validation	Yes	Yes
Weights	No	Yes
Flexibility	Moderate	High (reference period, bin dim, etc.)

CSTools

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

New functions BindDim() and CST_BindDim()

New function BindDim() is a **wrapper of abind::abind()**, designed to work with arrays with named dimensions in a more user-friendly way:

```
array1 <- array(rnorm(100), dim = c(lon = 10, lat = 10, time = 1))
array2 <- array(rnorm(100), dim = c(lon = 10, lat = 10, time = 1))
# With abind(), code is less readable, dim order must match and names are not preserved:
res <- abind(x = list(array1, array2), along = which(names(dim(array1)) == "time"))
dim(res)
# [1] 10 10  2

# BindDim() returns dimension names and works even if dim order does not match:
array3 <- array(rnorm(100), dim = c(time = 1, lon = 10, lat = 10))
res <- BindDim(x = list(array1, array2, array3),
               along = "time")
dim(res)
#  lon  lat time
#   10   10    3
```

New functions BindDim() and CST_BindDim()

CST_BindDim() serves the same function as BindDim() for s2dv_cubes, binding data and metadata to keep the result consistent. It has the following parameters:

- **x**: Two or more objects of class s2dv_cube to be bound together.
- **along**: Name of the binding dimension.
- **dat_dim**: Name of the dataset dimension in the s2dv_cube. Default value is NULL. Specifying this dimension ensures the dataset metadata is correctly preserved.
- **var_dim**: Name of the dataset dimension in the s2dv_cube. Default value is NULL. Specifying this dimension ensures the dataset metadata is correctly preserved.

MR: https://earth.bsc.es/gitlab/external/cstools/-/merge_requests/215
status: in master

Implemented EvalTrainIndices in QuantileMapping

- EvalTrainIndices() generates Training and Evaluation Indices for different cross-validation methods.
- Implemented EvalTrainIndices() in QuantileMapping() to be able to use more cross-validation methods (through parameter eval.method)

```
#' @param eval.method Character. The cross-validation method. Options include:
#'   -\code{leave-k-out}: Leaves out \code{k} points at a time for evaluation.
#'   -\code{retrospective}: Uses past data for training and a future point for evaluation.
#'   -\code{in-sample}: Uses the entire dataset for both training and evaluation.
#'   -\code{hindcast-vs-forecast}: Uses all years from the hindcast sdate dimension
#'    as training and all years from the forecast sdate dimension will be corrected.
```

issue: https://earth.bsc.es/gitlab/external/cstools/-/issues/161
status: in branch dev-quantilemapping

CSDownscale

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Added examples for functions, in branch - example_update

- Examples generally shows :
  - simple parameter usage
  - downscaling forecast
  - downscaling using Large-scale variables (Analogs())
  - downscaling for daily/monthly data
  - point downscaling

path : https://earth.bsc.es/gitlab/es/csdownscale/-/tree/example_update/inst/examples/example_theertha
status: in branch example_updates

esviz

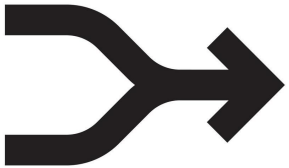
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Parameter name standarization

esviz:

- Contains different **plotting functions** (for maps, time series, scorecards, etc.
- Currently being prepared for its first package release, and is not yet available on CRAN.

Because these functions were originally created in different context, similar parameters sometimes had different names



We've been **standardizing** these inconsistencies to make the future package more user-friendly.

⚠️ These updates are **backward compatible**: older code will still work, but warning messages will appear to notify you of the new names.

Parameter name standarization

Some examples:

- `var` → `data`
- `sizetit` and `title_size` → `title_scale`
- `dots_size` → `dot_size`
- `dots_shape` and `dots_symbol` → `dot_symbol`
- `leg` and `legend` → `drawleg`
- …

Please **open an issue** if you find any problems related to the name changes or if you have any suggestions/observations!

path: https://earth.bsc.es/gitlab/es/esviz/-/merge_requests/31
status: in branch dev-unify_esviz_parameters

SUNSET

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

New full cross-validation workflow: Anomalies

CST is developing a series of workflows to allow climate forecast products to be computed using cross-validation in all steps of the workflow. The first one includes **anomaly computation** and **skill metrics**, with two functions:

Crossval_anomalies.R: The hcst, fcst and obs anomalies are computed in cross-validation. Limits and forecast probabilities are also computed for the categories requested in the recipe.

Crossval_metrics.R: The results of Crossval_anomalies() are used to compute the metrics requested in the Skill section of the recipe.

Example recipe and script on GitLab

MR: https://earth.bsc.es/gitlab/es/sunset/-/merge_requests/204
status: in master

Visualization: Mask extreme probabilities with RPSS

The plots showing the probability of extremes can now be masked with the corresponding RPSS as calculated by Crossval_metrics(). If the different sets of probabilities are named and RPSS is requested, an RPSS will be returned for each of the sets with the names given in the recipe:

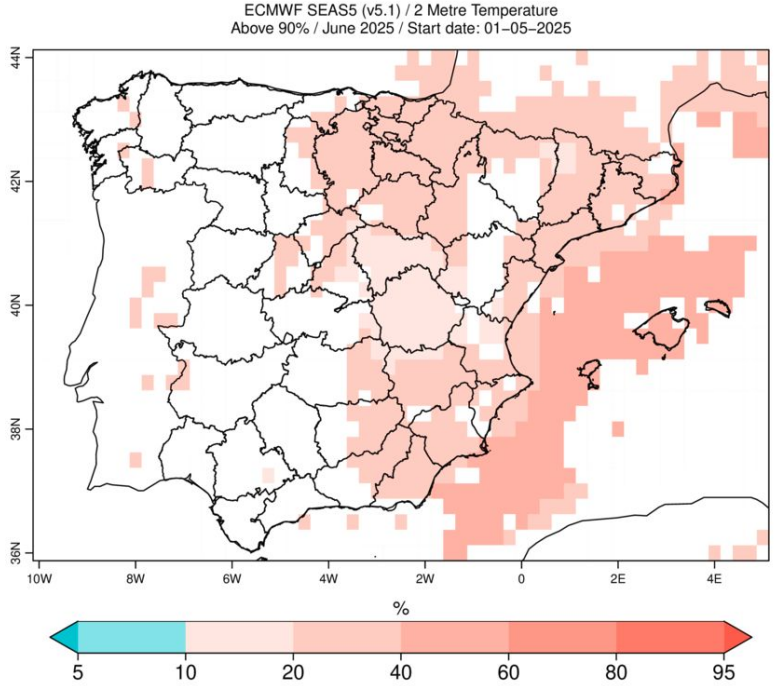
```
Workflow:
  Probabilities:
    percentiles:
      Terciles: [1/3, 2/3]
      P10: [1/10]
      P90: [9/10]
```

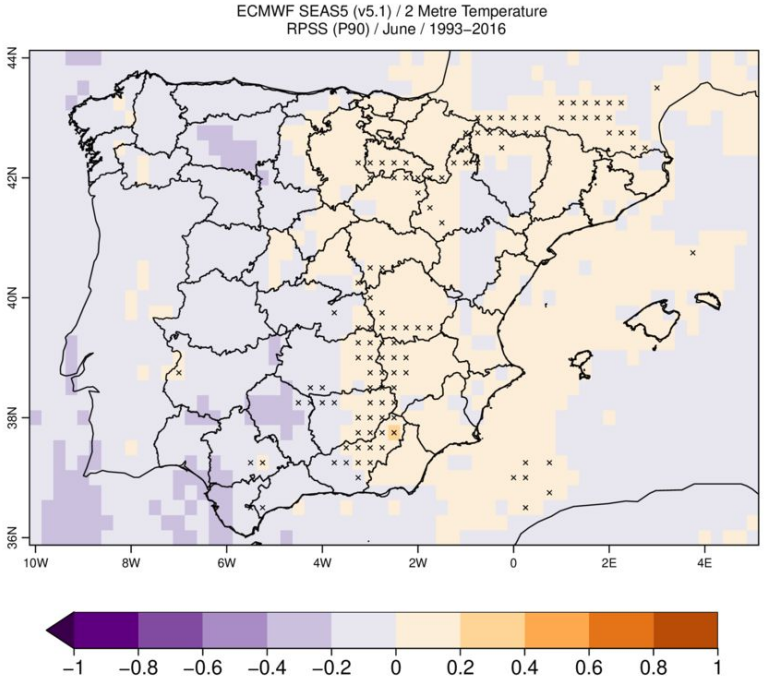
User-defined names for each set of categories

The Visualization module can automatically use these names to apply the correct skill score mask to each map.

MR: https://earth.bsc.es/gitlab/es/sunset/-/merge_requests/204
status: in master

Visualization: Mask extreme probabilities with RPSS





MR: https://earth.bsc.es/gitlab/es/sunset/-/merge_requests/204

status: in master

Loading module improvements

Previously, reference dataset dates were loaded based on the metadata from the model netCDF files. If the model metadata was not correct, this could cause issues, such as a mismatch between model and reference dates.

This has been improved for **seasonal monthly** and **daily** data as well as **subseasonal daily**. SUNSET now computes the time stamps needed based on the information in the recipe.

We have also added new unit tests for the generation of the expected dates.

issue: https://earth.bsc.es/gitlab/es/sunset/-/issues/173
status: in master

What do we know about BSC infrastructure?

Authors: Núria Pérez-Zanón and An-Chi Ho (December 2021)
Updated by Victòria Agudetse (May 2025)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

What do we know about BSC-ES infrastructure?

Aside from the data and software in our personal laptops, we all have access to common BSC infrastructure.

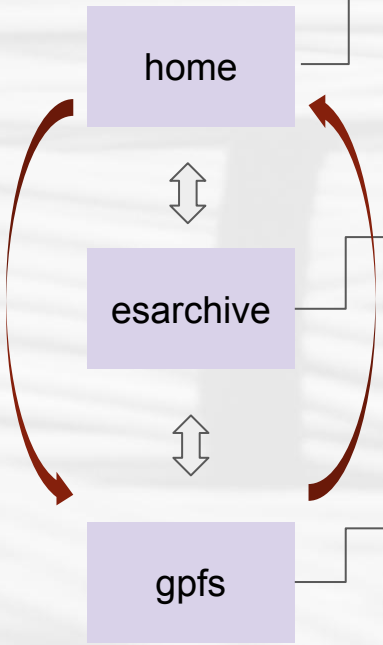
We **access the BSC infrastructure**:

- ★ When we connect to the BSC-ES Hub
- ★ When we use the workstations in the office
- ★ When we connect remotely via ssh to a workstation (bscearthXXX.int.bsc.es)
  - To ssh from windows: https://earth.bsc.es/wiki/doku.php?id=computing:sshwindows
  - To set up passwordless ssh connection: https://earth.bsc.es/wiki/doku.php?id=computing:sshkeyautologon
- ★ When we connect to one of the servers or HPC machines in BSC (MN5, Nord4, etc.)

PARTITIONS

What do we know about BSC-ES infrastructure?

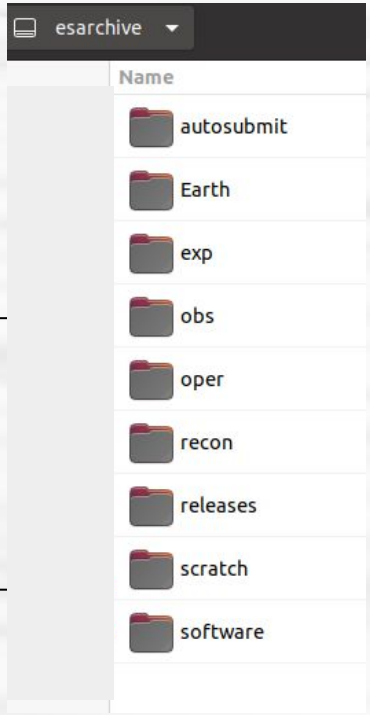
When we connect to the BSC infrastructure, we find several **partitions**. A disk partition, or simply 'partition', is a segment of a hard drive that is separate and independent from other segments. Each partition serves a different purpose and is accessible from different machines.



First entry point, accessible from Hub and WS
/home/Earth/<username>/
Internet connection available
Daily back-up, limited space: 50 GB per user
Tips: https://earth.bsc.es/wiki/doku.php?id=computing:workstations
https://earth.bsc.es/wiki/doku.php?id=computing:sharedstorage

Data storage for BSC-ES department
Personal folder: /esarchive/scratch/<username>/
Internet connection available (from WS and Hub only)
NO back-up, limited total space for all users
Tip: add esarchive partition in a laptop (forbidden in Windows):
https://earth.bsc.es/wiki/doku.php?id=computing:addesarchive

General storage accessible from all HPC machines (Nord4, CTE-AMD, MN5)
Personal folder: /gpfs/scratch/bsc32/bsc032XXX/
No internet connection
Limited total space for all users in ES



PARTITIONS

What do we know about BSC-ES infrastructure?

It is also possible to connect to BSC infrastructure through **servers** (physical machines), which have different uses:

- ★ **bscearth000.int.bsc.es and bscearth001.int.bsc.es**
  - Download data
  - run the automatic package tests (GitLab CI/CD, see e.g.: Pipelines · Earth Sciences / s2dv · GitLab)
- ★ **transfer1.bsc.es**
  - Internal transfer of data, e.g. from esarchive to GPFS and vice versa.
- ★ **bscesshiny01.bsc.es**
  - Shiny server, hosts shiny apps.
- ★ **bscesftp.bsc.es**
  - Share files externally, see: https://earth.bsc.es/wiki/doku.php?id=computing:public_ftp
- ★ **bscesautosubmit01.bsc.es and bscesautosubmit02.bsc.es**
  - Dedicated machines to launch workflows with the Autosubmit workflow manager. To be decommissioned (date TBD) https://earth.bsc.es/wiki/doku.php?id=tools:autosubmit

SERVERS

What do we know about BSC-ES infrastructure?

A **software stack** is the collection of programs and modules (including the operating system, architectural layers, protocols, runtime environments, …) that are installed in a machine.

- ★ The software stack at BSC can be different among different machines and departments
- ★ We have access to:
  - BSC software stack (not managed by CES)
  - BSC-ES software stack (managed by CES)
    - Workstations, Nord4 and CTE-AMD already using it
    - Hubs have a different software stack, more updated
    - In some machines, we should edit the **bashrc** to use it (instructions are always in the wiki: https://earth.bsc.es/wiki/doku.php?id=library:computing)
    - It is built on **modules**, some useful commands are:
      - `module list` # show all loaded modules
      - `module load *` # load the '*' module
      - `module av *` # show all available modules matching '*'
    - other software programs like mendeley can be opened in the WS/Hub: */shared/earth/software/mendeley/latest/bin/mendeleydesktop*
- ★ Open an issue in **the Requests GitLab** to ask for new software or R packages (@smirosla)

SOFTWARE STACK

What do we know about BSC-ES infrastructure?

What information do we need to know for each machine?

- Does it have BSC-ES software?
- is /esarchive/ mounted?
- Internet access?
- How to send jobs?
- Memory per node, cores per node….

Hub
Workstations (WS)
Marenostrum 5
AMD cluster
Nord4

Find the information here: https://earth.bsc.es/wiki/doku.php?id=library:computing

MACHINES

What do we know about BSC-ES infrastructure?

On the HPC machines (Nord4, CTE-AMD, MN5) there are two types of nodes:

- ★ **Login nodes:** The node we are in when we to connect to the machine via ssh. It can be used to execute basic tasks and send jobs. Should not be used for computation.
- ★ **Compute nodes:** For computation and memory-intensive tasks. They are assigned by the scheduler (e.g. slurm) when sending jobs or requesting interactive sessions. The job scheduler assigns resources on compute nodes based on priority.

More information here:
https://www.bsc.es/supportkc/docs/Nord4/slurm
https://www.bsc.es/supportkc/docs/MareNostrum5/slurm
https://www.bsc.es/supportkc/docs/CTE-AMD/slurm

MACHINES

What do we know about BSC-ES infrastructure?

Workstations

- R/4.1.2
- For debugging code (small data) or running startR workflows in remote machines
- Internet connection
- BSC-ES software stack
- /esarchive is mounted
- To be decommissioned

Hub (bsceshub03-07)

- R/4.4.1
- For debugging code (small data), startR workflows or running autosubmit.
- Internet connection
- BSC-ES software stack
- /esarchive is mounted
- Replaces workstations.

Nord4

- R/4.1.2
- For memory-intensive jobs
- Internet access in login node 0
- BSC-ES software stack; **must use nord3_singu_es wrapper**
- /esarchive is mounted

Marenostrum 5

- R/4.3.3 or R/4.2.2 (SUNSET conda)
- For memory-intensive jobs
- Some BSC-ES software stack available, conda environments can be installed
- Internet access in login node 4
- no access to /esarchive

CTE-AMD

- R/4.1.2 or R/4.3.3 (for R-INLA)
- For memory-intensive jobs
- BSC-ES software stack
- no access to /esarchive

MACHINES

What do we know about BSC-ES infrastructure?

RECOMMENDATIONS

Recommendations

★ **Save your scripts in GitLab (intermediate and final versions)**
- In an existing GitLab project
- In a personal project
- Documentation: https://earth.bsc.es/wiki/doku.php?id=library:computing#git
- If you have internet connection, you can source your code directly from GitLab
- Clone repositories under /esarchive/scratch/<username>/
  - You will have internet connection to push your changes
  - The code will be accessible from Hub, Nord4 and shiny server
  - There is no back-up copy of /esarchive (another good reason to use GitLab)

★ **Don't install local versions of R packages**
- If you do, we cannot debug the code and reproduce the errors
- Better to open an issue in Requests to ask for the installation: it's easier to debug and everyone can use it

★ **Infrastructure in the wiki:**
https://earth.bsc.es/wiki/doku.php?id=library:best_practices#network_infrastructure

What do we know about BSC-ES infrastructure?

Q&A: What else do we need to know? What questions do we have?

Thanks for joining

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Next meeting: TBA