



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación



**EXCELENCIA
SEVERO
OCHOA**

R tools user meeting

An-Chi Ho and Núria Pérez-Zanón

contributors: Andrea Manrique-Suñén

07/05/2021

Agenda

1. News

- s2dv
- startR
- CSIndicators
- Tips for sharing code on GitLab and Slack
- Presentations in the 8th BSC Doctoral Symposium
- How to cite in-house packages

2. S2S forecast (Andrea)

3. Q&A

s2dv

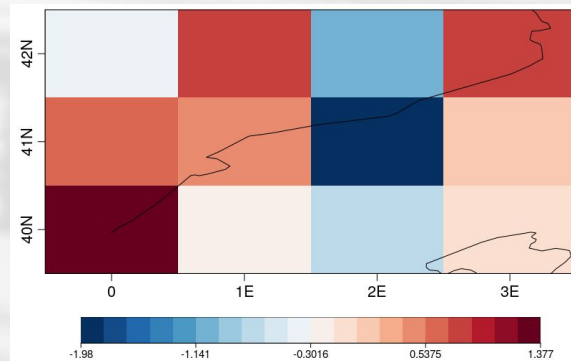
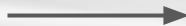
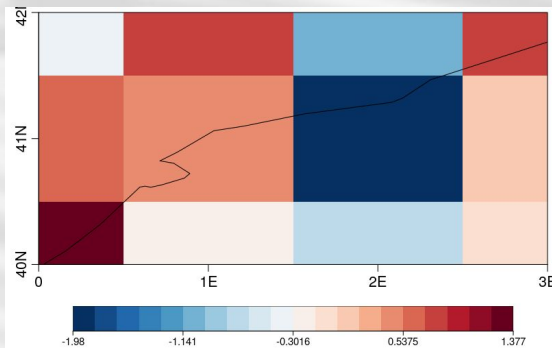
Development and bugfixes

1. PlotEquiMap()

- ***contour_label_scale***: Scale factor for the superimposed labels when drawing contour levels.

This argument didn't work as the description before. Fix it now.

- The border grids were only half plotted before. Fully plotted now.



Development and bugfixes

2. PlotStereoMap()

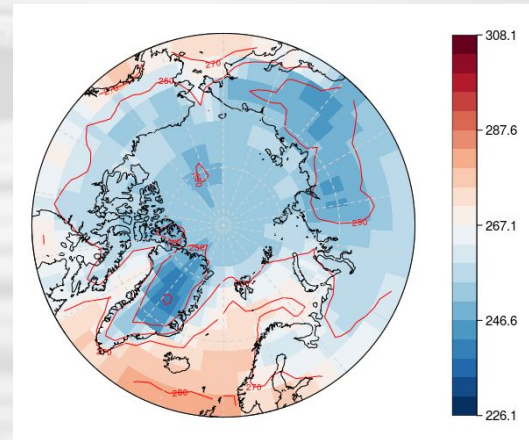
- Add contour function. Check [documentation](#) of the new contour arguments.

3. CDORemap

- it has been fixed to regrid irregular grids in s2dverification package
- it is being tested for the 'native ocean curvilinear grid'
- issue:

<https://earth.bsc.es/gitlab/es/s2dverification/-/issues/259>

- branch: [develop-CDORemap](#)



Development and bugfixes

Reminder: To avoid confusion, don't load `s2dv` and `s2dverification` at the same time in your script.

Recommendation: **load `s2dv` only**, and use `s2dverification::<func_name>` if needed.

```
library(s2dv)
trend <- Trend(data, time_dim = 'time') # Trend() is from s2dv
trend <- s2dverification::Enlarge(trend, 5) # Enlarge() from s2dverification
```

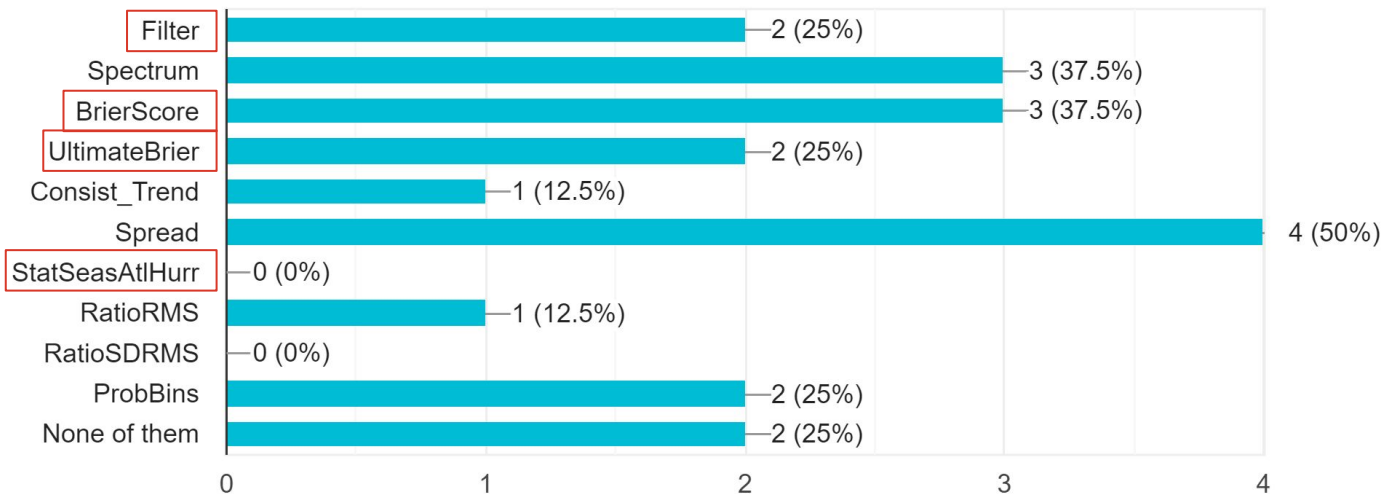
→ It is not encouraged though. If you need some functions in `s2dverification`, please discuss with us.

If you're compelled to load both packages meanwhile, specify the package name for each function, e.g., `s2dv::Trend`, `s2dv::Load`, etc.

Survey results: function usage

The following functions have been transformed to s2dv. Do you use any of them recently, or find them potentially useful for future work?

8 則回應



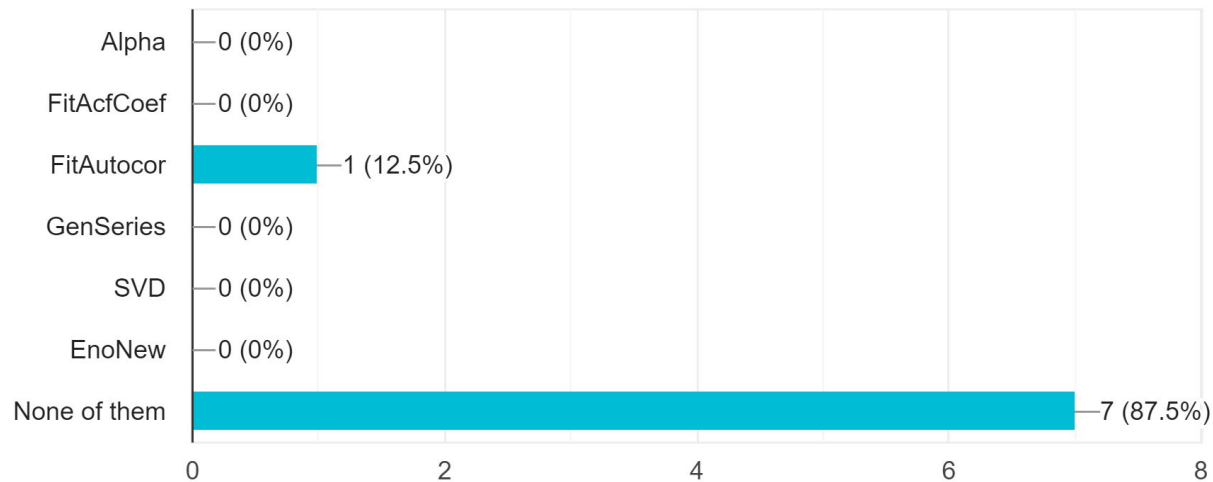
Functions that I have query

Call for user review

Survey results: function usage

The following functions WONT be transformed to s2dv if no one uses or is interested in them. Do you use any of them recently, or find them potentially useful for future work?

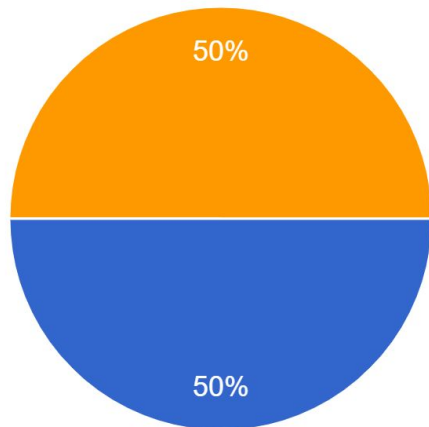
8 則回應



Survey results

Have you started replacing s2dverification with s2dv in your scripts?

8 則回應



- Yes, and it works fine
- Yes, but I found difficulty using them/it doesn't meet my expectation
- Not yet
- I've never used s2dverification

startR

Bugfixes

We've detected two problems regarding reshaping parameters:

- The mixed-dimension problem when using ``merge_across_dims`` + ``split_multiselect_dims`` ([ex1_2](#) is an example). It doesn't happen to all the cases, depending on how the dimension selectors are defined and the structure of the requested files.
- With ``merge_across_dims_narm``, the dimensions are not mixed but it causes problems at the very last part of data.

Bugs have been fixed and we'll have a new release next week.

Users' action: Check your startR scripts if these parameters are used.

Data checking methods:

https://earth.bsc.es/gitlab/es/startR/-/blob/master/inst/doc/data_check.md

CSIndicators

CSIndicators

CSIndicators

<https://CRAN.R-project.org/package=CSIndicators>
<https://earth.bsc.es/gitlab/es/csindicators/>

Sectoral Indicators for Climate Services Based on Sub-Seasonal to Decadal Climate Predictions

Description

Set of generalised tools for the flexible computation of climate related indicators defined by the user. Each method represents a specific mathematical approach which is combined with the possibility to select an arbitrary time period to define the indicator. This enables a wide range of possibilities to tailor the most suitable indicator for each particular climate service application (agriculture, food security, energy, water management...). This package is intended for sub-seasonal, seasonal and decadal climate predictions, but its methods are also applicable to other time-scales, provided the dimensional structure of the input is maintained. Additionally, the outputs of the functions in this package are compatible with CStools.

Functions and documentation

To learn how to use the package see:

- [Agricultural Indicators](#)
- [Wind Energy Indicators](#)

CSIndicators

Function	CST version	Indicators
PeriodMean	CST_PeriodMean	GST, SprTX, DTR
PeriodAccumulation	CST_PeriodAccumulation	SprR, HarR, PRCPTOT
AccumulationExceedingThreshold	CST_AccumulationExceedingThreshold	GDD, R95pTOT, R99pTOT
TotalTimeExceedingThreshold	CST_TotalTimeExceedingThreshold	SU35, SU, FD, ID, TR, R10mm, Rnmm
TotalSpellTimeExceedingThreshold	CST_TotalSpellTimeExceedingThreshold	WSDI, CSDI
WindCapacityFactor	CST_WindCapacityFactor	Wind Capacity Factor
WindPowerDensity	CST_WindPowerDensity	Wind Power Density

- A few parameters:
 - **data** object
 - **dates, start, end, time_dim** → allows temporal subsetting
 - **threshold, spell** → for extremes and waves
 - **na.rm, ncores** → internal use multiApply for parallel computing

CSIndicators

Auxiliar function	CST version
AbsToProbs	CST_AbsToProbs
QThreshold	CST_QThreshold
Threshold	CST_Threshold
MergeRefToExp	CST_MergeRefToExp
SelectPeriodOnData	CST_SelectPeriodOnData
SelectPeriodOnDates	

- Select temporal periods
 - Probabilities and Thresholds
 - Merge forecasts to a reference dataset
-
- Unit testing
 - Continuous Integration

Tips for script sharing

Tips for script sharing

- If possible, minimize the weight of the script.
- Clean the memory space or open another console to run the script again before sharing.
- On GitLab, use ```r` to report a chunk of R code (you'll get a colorful script!); use `<code>` for a single sentence.
- On Slack, use “Code block” for a chunk of code; use “Code” for a single sentence.

The 8th BSC Doctoral Symposium

The 8th BSC Doctoral Symposium



Wednesday 11th

9.20h Second Talk Session: Modelling & HPC

9.20h	startR: A tool for large multi-dimensional data processing	An-Chi Ho	Rosa Badia
-------	--	-----------	------------

Thursday 12th

10.00h Fifth Talk Session: HPC and Modelling for Earth Science

10.00h	High Resolution Decadal Prediction - Impacts on the predictability of the Pacific variability	Aude Carréric	Pablo Ortega
10.20h	Climate Forecast Analysis Tools Framework	Núria Pérez-Zanón	
10.40h	Bias-adjustment method for street-scale air quality models	Jan Mateu Armengol	
11.00h	Exploiting parallelism for CPU and GPU linear solvers on chemistry for atmospheric models	Christian Guzman Ruiz	
11.20h	Super-resolution for downscaling climate data	Carlos Alberto Gómez Gonzalez	

11.40h Break

How to cite in-house packages

How to cite in-house packages

Section in the wiki: [https://earth.bsc.es/wiki/doku.php?id=tools:Rtools&s\[\]=Rtools#how_to_cite](https://earth.bsc.es/wiki/doku.php?id=tools:Rtools&s[]=Rtools#how_to_cite)

1. s2dverification has a publication:

Nicolau Manubens, Louis-Philippe Caron, Alasdair Hunter, Omar Bellprat, Eleftheria Exarchou, Neven S. Fučkar, Javier Garcia-Serrano, François Massonnet, Martin Ménégoz, Valentina Sicardi, Lauriane Batté, Chloé Prodhomme, Verónica Torralba, Nicola Cortesi, Oriol Mula-Valls, Kim Serradell, Virginie Guemas, Francisco J. Doblas-Reyes, An R package for climate forecast verification, *Environmental Modelling & Software*, Volume 103, 2018, Pages 29-42, ISSN 1364-8152, <https://doi.org/10.1016/j.envsoft.2018.01.018>.

2. Specific packages → `citation("PackageName")`

3. Cite R software → `citation()`

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

4. Mention the software used in the **acknowledgement** section

Here is an example from 'How Reliable Are Decadal Climate Predictions of Near-Surface Air Temperature?' (Verfaille et al, 2020):
We acknowledge the use of the s2dverification (Manubens et al. 2018), startR (BSC/CNS and Manubens 2020), SpecsVerification (Siegert 2017), CSTools (Pérez-Zanón et al. 2019), ClimProjDiags (BSC/CNS et al. 2020), and boot (Davison and Hinkley 1997; Canty and Ripley 2020) R (R Core Team 2013) software packages.

S2S Forecast

S2S Systems (S2S Project dataset)

Forecast

Hindcasts

Main challenges with S2S systems:

Heterogeneity in the subseasonal systems

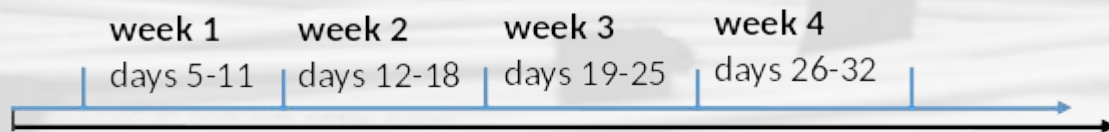
- Initializations (frequency and day)
- Ensembles
 - burst
 - lagged
- Hindcast
 - fixed
 - on the fly

Limited data both in forecast and hindcast

- Sample size for probabilistic skill scores
- Definition of the climatology
- Application of bias adjustment

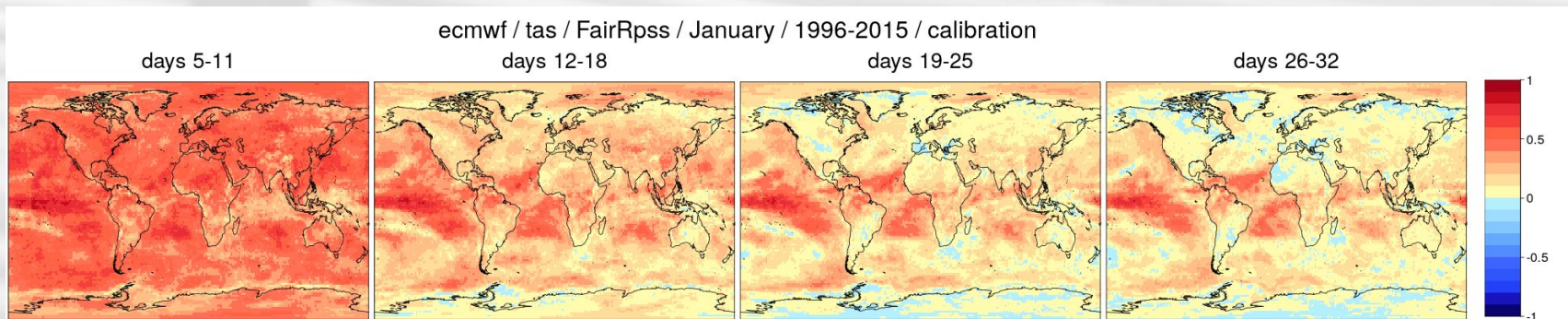
Status on 2020-10-27	Time range	Resolution	Ens. Size	Frequency	Re-forecasts	Rfc length	Rfc frequency	Rfc size
BoM (ammc)	d 0-62	T47L17	3*11	2/week	fixed	1981-2013	6/month	3*11
CMA (babj)	d 0-60	T266L56	4	2/week	on the fly	past 15 years	2/week	4
CNR-ISAC (isac)	d 0-32	0.75x0.56 L54	41	weekly	fixed	1981-2010	every 5 days	5
CNRM (lfpw)	d 0-47	T255L91	25	weekly	fixed	1993-2017	every 7 days	10
ECCC (cwao)	d 0-32	39 km L45	21	weekly	on the fly	1998-2017	weekly	4
ECMWF (ecmf)	d 0-46	Tco639/319 L91	51	2/week	on the fly	past 20 years	2/week	11
HMCR (rums)	d 0-61	1.1x1.4 L28	20	weekly	on the fly	1985-2010	weekly	10
JMA (rjtd)	d 0-33	TI479/TI319L100	50	weekly	fixed*	1981-2010	2/month	13
KMA (rksl)	d 0-60	N216L85	4	daily	on the fly	1991-2016	4/month	3
NCEP (kwbc)	d 0-44	T126L64	16	daily	fixed	1999-2010	daily	4
UKMO (egrr)	d 0-60	N216L85	4	daily	on the fly	1993-2016	4/month	7

S2S verification



Startdate

e.g. Monday



S2S verification

```
# Script to compute EnsCorr, FairRPSS, FairCRPSS (from SpecsVerification)
# to ECMWF subseasonal hindcasts for a given forecast date
# Generalizo a cualquier año
# with window calibration
# climatology using a running window of different sizes- with Compute
# Uses StartR and chunking on 'time' dimension ??
# Author: Andrea Manrique
# Feb 2020

library(multiApply)
library(startR)
library(easyVerification) # for veriApply() function
library(SpecsVerification) # for skill scores
library(s2dverification)

work.dir<-paste0("/esarchive/scratch/amanriqu/subseasonal/sub_chunking_S2S")

cfs.name <- 'ecmwf'
rean.name <- "ERA5"
forecast.year<-2020
yr1.hind <- forecast.year -20 #2000 # first hindcast year
yr2.hind <- forecast.year -1 ##2019 # last hindcast year (usually the forecast year -1)
var_name <- 'tas'
var.name.map <- '2m Temperature'

cal.method <- 'calibration' #'simple_bias' # #

# tas:
# exp:
ecmwf_path <- paste0("/esarchive/exp/ecmwf/s2s-monthly_ensforhc/",
                    "weekly_mean/$var$_f6h/$sdate/$var$_$year$.nc")

# obs:
obs_path <- paste0("/esarchive/recon/ecmwf/era5/weekly_mean/",
                  "$var$_f1h-240x121/$var$_$file_date$.nc")
```

Start dates selection

```
#dates 2020
# 2020-02-01 first thursday of 2020

# Thursdays in 2020
sdates.seq.thu <- format(seq(as.Date(paste(2020,01,02,sep='-')),as.Date(paste
(2020,12,31,sep='-')),by='weeks'),format='%Y%m%d')
# Mondays in 2020
sdates.seq.mon <- format(seq(as.Date(paste(2020,01,02,sep='-'))+4,as.Date(paste
(2020,12,31,sep='-')),by='weeks'),format='%Y%m%d')

# merge monday and thursday start dates
sdates.seq2<-c(sdates.seq.mon,sdates.seq.thu)
ind<-order(as.Date(sdates.seq2,format='%Y%m%d')) # dates in order
sdates.seq <-sdates.seq2[ind]
pos.bis <- which(sdates.seq == paste0(2020,"0229")) # find if there is a startdate at the 29th of
february and return its position in the vector sdates.seq
if(length(pos.bis) != 0) sdates.seq <- sdates.seq[-pos.bis] # if there is a startdate at the 29th of
february, remove it.

# loop here for month
month<-4 # April

# Select dates of that month
sdates.month<-sdates.seq[which(as.integer(substr(sdates.seq,5,6)) == month)]

# numbers of start dates for skill computation
number_of_sd<-length(sdates.month) #9
# Select startdates dates for skill comp:
```

```
> sdates.seq
[1] "20200102" "20200106" "20200109" "20200113" "20200116" "20200120"
[7] "20200123" "20200127" "20200130" "20200203" "20200206" "20200210"
[13] "20200213" "20200217" "20200220" "20200224" "20200227" "20200302"
[19] "20200305" "20200309" "20200312" "20200316" "20200319" "20200323"
[25] "20200326" "20200330" "20200402" "20200406" "20200409" "20200413"
[31] "20200416" "20200420" "20200423" "20200427" "20200430" "20200504"
[37] "20200507" "20200511" "20200514" "20200518" "20200521" "20200525"
[43] "20200528" "20200601" "20200604" "20200608" "20200611" "20200615"
[49] "20200618" "20200622" "20200625" "20200629" "20200702" "20200706"
[55] "20200709" "20200713" "20200716" "20200720" "20200723" "20200727"
[61] "20200730" "20200803" "20200806" "20200810" "20200813" "20200817"
[67] "20200820" "20200824" "20200827" "20200831" "20200903" "20200907"
[73] "20200910" "20200914" "20200917" "20200921" "20200924" "20200928"
[79] "20201001" "20201005" "20201008" "20201012" "20201015" "20201019"
[85] "20201022" "20201026" "20201029" "20201102" "20201105" "20201109"
[91] "20201112" "20201116" "20201119" "20201123" "20201126" "20201130"
[97] "20201203" "20201207" "20201210" "20201214" "20201217" "20201221"
[103] "20201224" "20201228" "20201231"
```

```
> sdates.month
[1] "20200402" "20200406" "20200409" "20200413" "20200416" "20200420" "20200423"
[8] "20200427" "20200430"
```

Running window

```
# Construct window of 9/7/5/3 start dates
# window
window.len<-3
h <-floor(window.len/2) #4

# Construct array of window length x sdates
hcst.sdays<-array(NA,dim=c(window.len,length(sdates.month))) # 3 x 9
names(dim(hcst.sdays))<-c('sday','sdate')

for (day in 1:length(sdates.month)){
  # sdate, center of the window, position in sdates.seq
  sd.ind<-which(sdates.seq == sdates.month[day])
  # indices of the window in sdates.seq
  inds<-seq(sd.ind-h,sd.ind+h)
  # wrap around indices in sdates.seq
  inds<-(inds-1)%length(sdates.seq)+1
  hcst.sdays[,day]<-sdates.seq[inds]
}
```

```
> hcst.sdays
  [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] "20200330" "20200402" "20200406" "20200409" "20200413" "20200416"
[2,] "20200402" "20200406" "20200409" "20200413" "20200416" "20200420"
[3,] "20200406" "20200409" "20200413" "20200416" "20200420" "20200423"

  [,7]      [,8]      [,9]
[1,] "20200420" "20200423" "20200427"
[2,] "20200423" "20200427" "20200430"
[3,] "20200427" "20200430" "20200504"
```

Load exp data

```
# Load headers:
exp <- Start(dat = ecmwf_path,
  var = var_name,
  sdate = hcst.sdays, # dims: sday sdate
  syear = 'all', # before hdate
  time = 'all',
  ensemble = "all",
  latitude = "all",#indices(1:121),
  longitude = "all",#indices(1:240),
  longitude_reorder = CircularSort(0, 360), # to read negative lons
  latitude_reorder = Sort(),
  syear_depends = 'sdate',
  return_vars = list(latitude = 'dat',
                    longitude = 'dat',
                    time = c('sdate','syear')),
  split_multiselectd_dims = TRUE,
  retrieve = F)
```

```
* Exploring files... This will take a variable amount of time depending
* on the issued request and the performance of the file server...
* Detected dimension sizes:
*   dat: 1
*   var: 1
*   sday: 3
*   sdate: 9
*   syear: 20
*   time: 4
*   ensemble: 11
*   latitude: 121
*   longitude: 240
* Total size of involved data:
*   1 x 1 x 3 x 9 x 20 x 4 x 11 x 121 x 240 x 8 bytes = 5.1 Gb
* Successfully discovered data dimensions.
```

Warning messages:

```
1: ! Warning: Parameter 'pattern_dims' not specified. Taking the first dimension,
! 'dat' as 'pattern_dims'.
2: ! Warning: Could not find any pattern dim with explicit data set descriptions (in
! the form of list of lists). Taking the first pattern dim, 'dat', as
! dimension with pattern specifications.
```

Load obs data

```
#take dates:
```

```
dates <- attr(exp, 'Variables')$common$time
```

```
# the output date from the hindcast indicates the middle of the weekly average
```

```
# while for ERA5 the weekly averages are indicated by the 1st day (monday)
```

```
# To load ERA5 shift 3 days, (ERA5 weekly means are day 1-7,
```

```
#while ncep weekly means are 5-11)
```

```
# so subtract 3 days:
```

```
dates3 <- dates - as.difftime(3, unit="days")
```

```
file_date3 <- sapply(dates3, format, '%Y%m%d')
```

```
dim(file_date3) <- c(window.len, number_of_sd, 20, 4)
```

```
names(dim(file_date3)) <- c('sday', 'sdate', 'syear', 'time')
```

```
obs <- Start(dat = obs_path,
```

```
  var = var_name,
```

```
  file_date = file_date3, # sday sdate syear time
```

```
                # 3 4 20 4
```

```
  latitude = "all", #indices(1:121),
```

```
  longitude = "all", #indices(1:240),
```

```
  longitude_reorder = CircularSort(0, 360), # to read neg
```

```
  latitude_reorder = Sort(),
```

```
  split_multiselected_dims = TRUE,
```

```
  retrieve = F)
```

```
> obs <- Start(dat = obs_path,
+             var = var_name,
+             file_date = file_date3, # sday sdate syear time
+                                     # 3 4 20 4
+             latitude = "all", #indices(1:121),
+             longitude = "all", #indices(1:240),
+             longitude_reorder = CircularSort(0, 360), # to read negative lons
+             latitude_reorder = Sort(),
+             split_multiselected_dims = TRUE,
+             retrieve = F)
* Exploring files... This will take a variable amount of time depending
* on the issued request and the performance of the file server..
* Detected dimension sizes:
*   dat: 1
*   var: 1
*   sday: 3
*   sdate: 9
*   syear: 20
*   time: 4
*   latitude: 121
*   longitude: 240
* Total size of involved data:
* 1 x 1 x 3 x 9 x 20 x 4 x 121 x 240 x 8 bytes = 478.6 Mb
* Successfully discovered data dimensions.
Warning messages:
1: ! Warning: Parameter 'pattern_dims' not specified. Taking the first dimension,
! 'dat' as 'pattern_dims'.
2: ! Warning: Could not find any pattern dim with explicit data set descriptions (in
! the form of list of lists). Taking the first pattern dim, 'dat', as
! dimension with pattern specifications.
```

Do computations and save skill scores

```
score_calc_window<-function(forecast_month,reference_month,cal.method>window.len) {  
  
  # BIAS CORRECTION: #####  
  
  ### ANOMALIES #####  
  
  ##### SKILL SCORE #####  
  
  return(c(list(FairRpss = FairRpss_total),  
            list(FairRpss_sd = FairRpss.sd),  
            list(FairCrpss = FairCrpss_total),  
            list(FairCrpss_sd = FairCrpss.sd),  
            list(Enscorr = my.Enscorr),  
            list(Enscorr.pvalue = my.Enscorr.pvalue))),#,  
        #list(my.prob = my.prob))  
  
} # end function
```

```
  n.members<-11  
  FairRpss<-res$FairRpss  
  FairRpss_sd<-res$FairRpss_sd  
  FairCrpss<-res$FairCrpss  
  FairCrpss_sd<-res$FairCrpss_sd  
  Enscorr <- res$Enscorr  
  Enscorr.pvalue <- res$Enscorr.pvalue  
  #my.prob<-c(1/3,2/3)  
  
  # Obtain values of lat and lon and return them for plotting  
  
  #  
  lons<- attr(exp,"Variables")$dat$longitude  
  
  lats<- attr(exp,"Variables")$dat$latitude  
  
  leadtime.week <- c('5-11','12-18','19-25','26-32')  
  
  print("Saving results...")  
  save(FairRpss, FairRpss_sd, FairCrpss, FairCrpss_sd, Enscorr, yr1.hind, yr2.hind, n.members, lons, lats,  
        leadtime.week, var.name.map, cfs.name,  
        rean.name,  
        file=paste0(work.dir,'/',cfs.name,'/',var_name,  
                    '._','scores_month',month,'_',cal.method,'_By8StartDate_running_window',window.len,'_ERA5.RData'))  
  #tas_scores_operational_test_20200213_calibration_running_window3.RData
```

Q & A

Next meeting: 4th June 2021 (Friday 3pm)