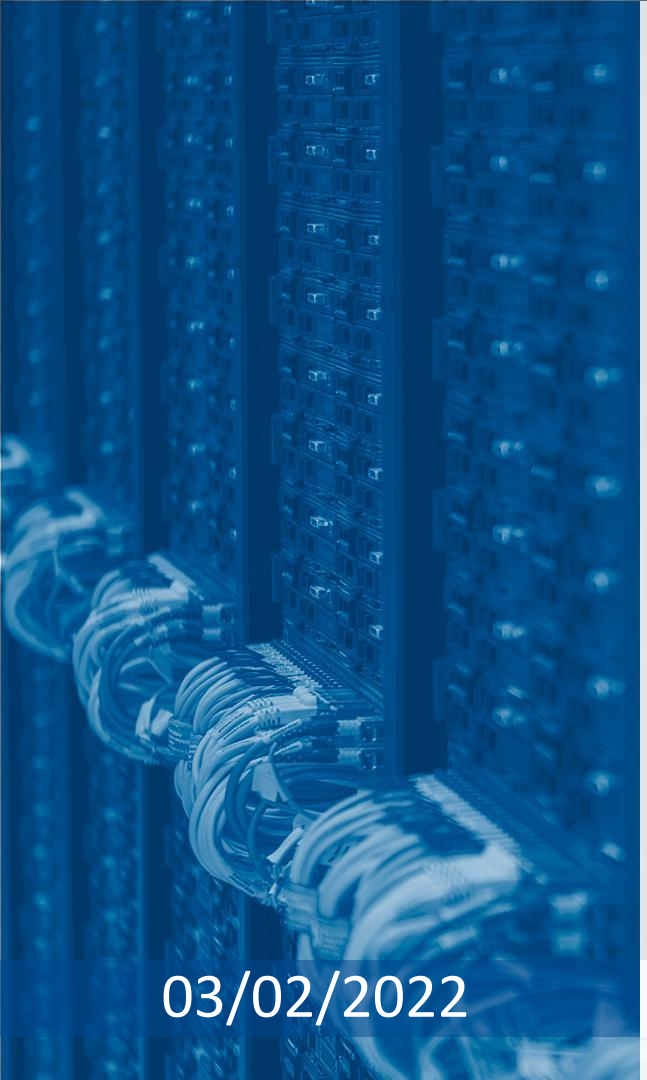


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

R tools users meeting

An-Chi Ho and Núria Pérez-Zanón

contributor: Paolo De Luca

03/02/2022

Agenda

1. Ice-breaker
2. News
    - s2dv
    - startR
3. Bonferroni test [Paolo]
4. Q&A
    - CSTool paper interactive discussion is close [Núria]
      https://gmd.copernicus.org/preprints/gmd-2021-368/

Ice-breaker

Statistical hypothesis test

Terminology: significance testing vs. hypothesis testing?

https://stats.stackexchange.com/questions/16218/what-is-the-difference-between-testing-of-hypothesis-and-test-of-significance
https://link.springer.com/chapter/10.1007/978-1-4612-4414-1_3

Significance testing is created by Fisher and hypothesis testing is created by Neyman and Pearson as an extension of significance testing.

Fisher's significance tests yield a p-value that represents how extreme the observations are under the null hypothesis. That p-value is an index of evidence against the null hypothesis and is the level of significance.

Neyman and Pearson's hypothesis tests set up both a null hypothesis and an alternative hypothesis and work as a decision rule for accepting the null hypothesis. You choose an acceptable rate of false positive inference, α (usually 0.05), and either accept or reject the null based on whether the p-value is larger or smaller than α.

Statistical hypothesis test

Do you have any suggestions for the tests in the s2dv functions?

- **ACC()**: The "parametric" method that provides a confidence interval for the ACC computed by a Fisher transformation and a significance level for the ACC from a one-sided student-T distribution.
- **Corr()**: The confidence interval is computed by the Fisher transformation and the significance level relies on an one-sided student-T distribution. → How about different methods ('spearman', or 'kendall')?
- **RMS()**: CI is computed by the chi2 distribution; no p-value calculated.
- **RMSSS()**: Calculate p-value by one-sided Fisher test; no CI provided.
- **Trend()**: The confidence interval relies on the student-T distribution, and the p-value is calculated by ANOVA.
- **Consist_Trend()** only returns confidence interval but not p-value (but Trend(), which is used inside, does provide p-value)
- **Regression()**: The p-value relies on the F distribution, and the confidence interval relies on the student-T distribution.

s2dv

s2dv

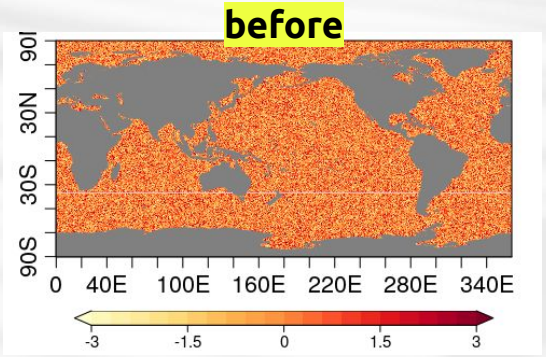
Current version: **1.1.0**

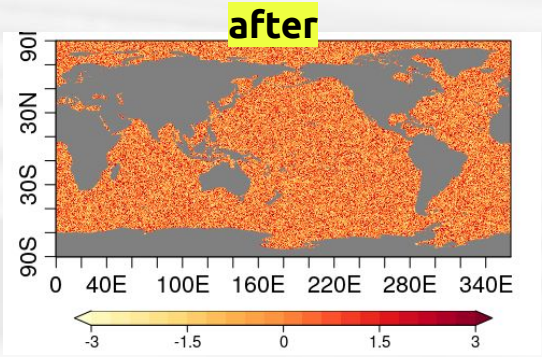
New development

- ACC(): https://earth.bsc.es/gitlab/es/s2dv/-/blob/master/R/ACC.R
  - Add area-weighting into the calculation
  - Ensure the data has a spatial mean of zero.
  - "space_dim" is deprecated and replaced by "lat_dim" and "lon_dim".
  - "dat_dim" can be NULL.
- PlotEquiMap():
  Add useRaster = TRUE in image() if possible (i.e., latitude and longitude are regularly spaced.)

Possible development:

- New function WeightCells()





startR

bugfixes

- Merge the files with different time dimension length together without NAs

  startR used to suppose all the files have the same inner dimension lengths.
  Problematic case: The first chunk has 2 time steps (Nov. - Dec.) and the following chunks have 12 time steps (Jan. - Dec.). To merge all the chunks along time dim, there were extra NAs at the first chunk's position.
  Problem solved: As long as `**largest_dims_length = TRUE**` is used with `merge_across_dims = TRUE` and `merge_across_dims_narm = TRUE`, Start() is able to identify the inner dim length of each file and avoid extra NAs.

Development

The argument `crop` in `transform_params` in Start() for CDORemapper() is now deprecated. The crop value is assigned as the range of lat and lon automatically by Start().

! Warning: Argument 'crop' in ‘transform_params' for CDORemapper() is !deprecated. It is automatically assigned as the selected domain in Start() call.

```
obs <- Start(dat = obs_path,
             var = var_name,
             sdate = '201811',
             time = 'all',
             latitude = values(list(lats.min, lats.max)),
             latitude_reorder = Sort(decreasing = T),
             longitude = values(list(lons.min, lons.max)),
             longitude_reorder = CircularSort(0, 360),
             transform = CDORemapper,
             transform_extra_cells = 2,
             transform_params = list(grid = 'r360x181',
                                     method = 'conservative',
                                     crop = c(lons.min, lons.max, lats.min, lats.max)),
             transform_vars = c('latitude', 'longitude'),
             return_vars = list(time = NULL,
                                latitude = 'dat', longitude = 'dat'),
             retrieve = T)
```

bugfixes

Fix in startR workflow when submitting jobs to a cluster

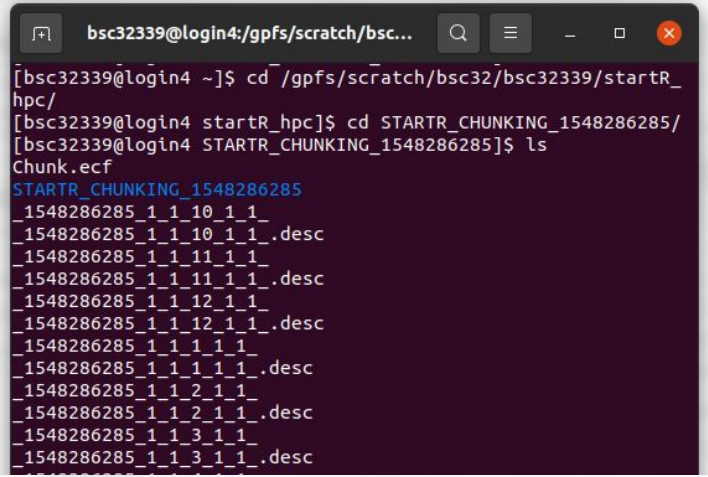
Data declaration → Operation defining → Workflow defining → **Job execution** → Result collection

```
res <- Compute(wf,
               chunks = list(month = 12),
               cluster = list(queue_host = 'nord3', …,
                              tmp_dir = '/gpfs/scratch/bsc32/bsc32XXX', …))
```

The **names** of the temporal files in which the data was being loaded (bigmemory objects) were not correctly created when a dimension was requested to be split in more than 11 pieces.

E.g: chunks 1 and 11 were overwritten by each other, you were able to resubmit the jobs through ecFlow

SOLVED



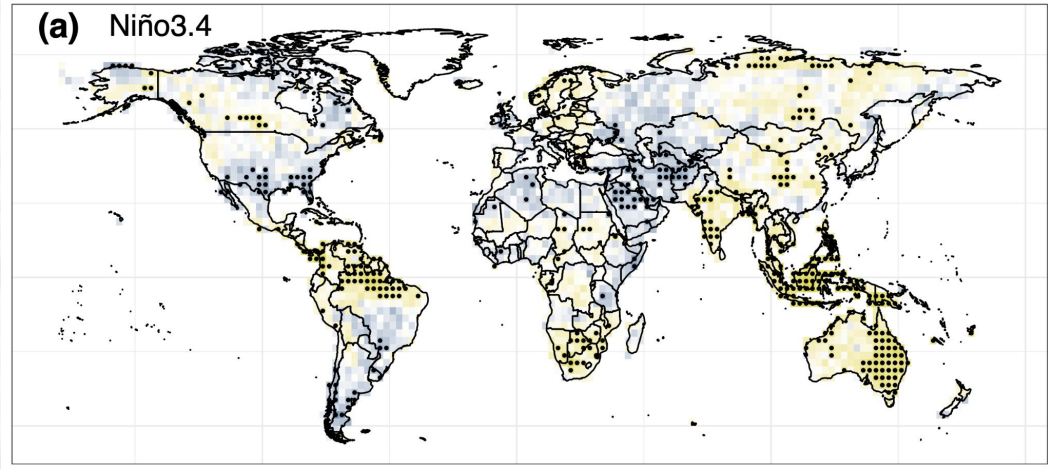
Bonferroni correction

Intro Bonferroni correction

Let's suppose you need to compute a large number p-values.

This could be the case when assessing statistical significance in global climate maps, where a statistical test is computed for each grid-point,

e.g.



De Luca, et al. 2020 (ESD)

Intro Bonferroni correction

In this case, with a large number of statistical tests or multiple comparisons, we may get statistically significant p-values (e.g. $p<0.05$ or $p<0.01$) simply by chance.

These are also called **Type I errors** or "**false positives**". In other words, some of your significant p-values are actually not significant.

Therefore, if not addressed, **Type I errors** may lead to an overestimation of our results.

By doing a **Bonferroni correction** your results will be more robust from a statistical perspective.

What does the Bonferroni correction?

The **Bonferroni correction** (Bonferroni, 1936; Sedgwick, 2014) is a conservative method that addresses the issue of **Type I errors**.

The Bonferroni correction simply **increases the values of the p-values** based on the total number of tests conducted simultaneously.

e.g. by taking one grid-point with **Type I error**, with p-value=0.045 (significant at the 5% level), after the Bonferroni correction the p-value=0.378 (not significant anymore).

Therefore, **some of the p-values which originally were statistically significant, after the correction they are not anymore**.

Bonferroni correction with R

Let's suppose I have a list with 300 data.frames and each data.frame contains two columns x and y:

```
set.seed(182)

# create list

lst=list()

for (i in 1:300) {lst[[i]]=data.frame(x=rnorm(250), y=rnorm(250,mean=0.2))}
```

Now I compute a statistical test, for each data.frame, between x and y:

```
# statistical test

lst_test=list()

for (i in 1:300) {lst_test[[i]]=wilcox.test(lst[[i]]$x,lst[[i]]$y)}
```

Bonferroni correction with R

I extract all p-values and p-values significant at the 5% level, and store them in a data.frame:

```
# get p-values

lst_p=list()

for (i in 1:300) {lst_p[[i]]=lst_test[[i]]$p.value}


p_values=data.frame(p=do.call(rbind,lst_p))


# get significant p-values

p_values_sign=subset(p_values, p <0.05)
```

Bonferroni correction with R

Then I apply the Bonferroni correction to the data.frame with **ALL** p-values and extract the significant p-values (5%) corrected:

```
# Bonferroni correction
p_values_bonferroni=data.frame(p=p.adjust(p_values$p, method = "bonferroni", n=length(p_values$p)))
# get significant p-values Bonferroni
p_values_sign_bonferroni=subset(p_values_bonferroni, p <0.05)
```

Finally, by checking the total number of significant p-values between non-corrected and corrected, we note that Bonferroni's p-values are much less in number.

```
print(nrow(p_values_sign))
[1] 172
print(nrow(p_values_sign_bonferroni))
[1] 16
```

Bonferroni correction with R

References

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni Del R Istituto Superiore Di Scienze Economiche e Commericiali Di Firenze*, *8*, 3–62.

De Luca, P., Messori, G., Wilby, R. L., Mazzoleni, M., & Di Baldassarre, G. (2020). Concurrent wet and dry hydrological extremes at the global scale. *Earth Syst. Dynam.*, *11*(1), 251–266. https://doi.org/10.5194/esd-11-251-2020

Sedgwick, P. (2014). Multiple hypothesis testing and Bonferroni's correction. *BMJ : British Medical Journal*, *349*, g6284. https://doi.org/10.1136/bmj.g6284

Rscript available here:
https://earth.bsc.es/gitlab/pdeluca/landmarc/-/blob/master/scripts/R_bonferroni_pdeluca_3_2_22.R

Q & A

Next meeting: 3rd Mar. 2022 (11 am)