High Performance Computing for Data Science

Academic Year: (2023 / 2024) **Review date: 15/07/2023 13:46:01**

Department assigned to the subject: Statistics Department

Coordinating teacher: UCAR MARQUES, IÑAKI

Type: Electives ECTS Credits : 3.0

Year : 1 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Programming in R
Advanced Programming

OBJECTIVES

- Knowledge of the general principles of high-performance computing.
- Knowledge of automation and scripting strategies and tools.
- Ability to select appropriate tools for program measurement and profiling.
- Ability to connect R code with external C/C++ libraries through Rcpp.
- Ability to parallelize R code and compiled code.
- Ability to produce reproducible execution environments.
- Ability to send workflows to the cloud.

DESCRIPTION OF CONTENTS: PROGRAMME

1. The road to High-Performance Computing
   1.1. HPC overview
   1.2. Tools for automation and scripting
   1.3. Tools for measuring and profiling

2. Making your code run faster: On the shoulders of giants
   2.1. Motivation and main concepts
   2.2. Interfacing C/C++ external libraries via Rcpp
   2.3. Efficient use of linear algebra engines
   2.4. Interfacing other languages and libraries
   2.5. Use cases in Statistics

3. Running multiple things at once: Parallel programming
   3.1. Motivation and main concepts
   3.2. Low-level parallelism: OpenMP, RcppParallel
   3.3. High-level parallelism: the future package
   3.4. Use cases in Statistics

4. Using more resources: Working in the cloud and beyond
   4.1. Motivation and main concepts
   4.2. Containerization: reproducible execution environments
   4.3. Scaling R in the cloud with googleComputeEngineR
   4.4. Use cases in Statistics

The program is subject to modifications due to the course development and/or academic calendar.

LEARNING ACTIVITIES AND METHODOLOGY

* Training activities

- AF1: Theoretical lesson.
- AF2: Practical lesson.
- AF5: Tutorials.
- AF6: Group work.
- AF7: Individual work.
- AF8: On-site evaluation tests.

* Teaching methodologies

- MD1: Class lectures by the professor with the support of computer and audiovisual media, in which the main concepts of the subject are developed and the bibliography is provided to complement the students' learning.
- MD3: Resolution of practical cases, problems, etc. posed by the teacher individually or in groups.
- MD5: Preparation of papers and reports individually or in groups.

ASSESSMENT SYSTEM

% end-of-term-examination/test:	0
% of continuous assessment (assigments, laboratory, practicals…):	100

The evaluation in the ordinary call is done entirely by continuous evaluation. This is done by a mixture of:

(a) two quizzes;
(b) practical exercises.

The continuous evaluation grade (in the scale 0-10) is

0.5 * A + 0.5 * B

where

- A (in the scale 0-10) is the weighted grade of the quizzes;
- B (in the scale 0-10) is the grade of the practical exercises;

Students who have not followed the continuous evaluation may take a final exam in the ordinary call with a value of 60% of the final grade.

The grade in the extraordinary call is established by a quiz and the delivery of practical exercises.

Further details are provided in Aula Global. The evaluation is subject to modifications due to the course development and/or academic calendar.

BASIC BIBLIOGRAPHY

- Chambers, J. M. Software for Data Analysis Programming with R, Springer, 2009

- Chambers, J. M. Extending R, Chapman and Hall/CRC, 2017

ADDITIONAL BIBLIOGRAPHY

- Chapple, S., Troup, E., Forster, T., and Sloan, T. Mastering Parallel Programming with R, Packt Publishing, 2016

- Eddelbuettel, D. Seamless R and C++ integration with Rcpp, Springer, 2013

- McCallum, Q. E. and Weston, S. Parallel R, O'Reilly Media, 2012