Optimisation du traitement de données et bonnes pratiques de programmation avec R

Présentation générale

R est un langage excessivement lent lorsqu'il est mal utilisé. Et il très souvent mal utilisé faute de formation en informatique ou en R tout court. Cela fait que, souvent, un script R peut mettre plusieurs dizaines de minutes à s'exécuter, voire plusieurs heures. Ça n'a pas de bon sens ! Le moins puissant de vos ordinateurs est capable de calculer n'importe lequel de vos codes en quelques millisecondes. Nous verrons pourquoi ces scripts sont lents et comment bien programmer en R pour, d'une part rendre le code beaucoup plus rapide, mais aussi beaucoup plus lisible. En effet, mal utilisé, R est langage dont la syntaxe est assez lourde. Ainsi il arrive souvent qu'on ne soit plus capable de comprendre ses propres codes.

Comme un bon code bien écrit et facile à lire est toujours un code rapide et efficace, à la fin vous aurez les outils pour passer d'un code de 75 lignes incompréhensibles et qui prend 30 minutes à calculer à un code de 15 lignes, très lisible et qui prend 1/10$^{\text{ième}}$ de seconde à calculer.

Cette formation ne requiert aucun prérequis particulier si ce n'est utiliser déjà, même mal (surtout mal), R. Cependant ce **n'est pas** une initiation à R et encore moins une formation aux statistiques.

Informations complémentaires

Tarifs : la formation est **GRATUITE** dans l'esprit de l'entraide, du partage libre des connaissances et du logiciel open source [1].

Dates : la formation aura lieu **fin mars**. Date à définir entre les participants.

Contact : envoyez un mail à <jean-romain.roussel.1@ulaval.ca> avec comme sujet « Prénom - Inscription formation R »

Durée : 4h de théorie le matin, 3 ou 4h de pratique l'après midi. Vous pouvez venir à la théorie uniquement. Ou l'inverse.

Qui suis-je : programmeur autodidacte, optimiseur fou, linuxien non barbu et accessoirement doctorant en télédétection avec la technologie LiDAR.

1. Mon avis est qu'il serait scandaleux et fondamentalement malhonnête de faire payer pour une formation à R alors que l'on doit tout à des gens qui ont donné gratuitement de leur temps. Il s'agit pour moi d'un juste retour à la communauté open source

Description détaillée

Le thème de la présentation est « optimisation et rapidité d'exécution et clarté du code »

Nous verrons certains concepts qui sont **fondamentaux** et à la base de R mais qui ne sont que rarement maîtrisés faute de formation, comme les notions de **programmation vectorielle** ou d'**agrégation de données**. Nous verrons les bonnes pratiques et les bons packages pour écrire du code propre et toujours lisible même 6 mois plus tard. Nous verrons l'importance de la **factorisation du code** et l'élégance des **opérateurs de *piping***. Nous verrons des outils rapides et efficaces vs. des outils lents et lourds. Cette formation contient des notions essentielles + des notions intéressantes mais pas essentielles que je pourrais aborder à la demande.

Que vous soyez confirmé ou non, statisticien ou modélisateur chacun devrait pouvoir y apprendre un grand nombre de notions. Nous verrons essentiellement des choses qui sont **fondamentales en R** mais qui ne sont en fait connues que de peu de gens et souvent même inconnues de ceux qui vous enseignent R (si vous avez eu la chance d'apprendre R).

Éléments que nous étudierons

- Pourquoi R est lent ? Théorie informatique et compréhension d'un ordinateur.
- Programmation vectorielle en R
- Traitement de données en R
  - L'agrégation : LA notion la plus fondamentale
  - Les structures de données : tableaux contingents vs. bases de données
- Bonnes pratiques de programmation
  - Factoriser son code : utilisation des fonctions
  - Limite de longueur d'un code : 80 caractères

Éléments que je peux aborder à la demande

- optimisation du code par la précompilation en bytecode, le C++ ou la parallélisation des calculs.
- une structures de données puissantes : les `data.table`
- notions de programmation fonctionnelle
- autre si vous avez une demande particulière

Les notions que je ne traiterai pas

- la création de package (beaucoup beaucoup trop long)
- la programmation orientée objet dans R (trop complexe)
- les *non-standard evaluations* (trop complexe)