La reproductibilité avec R, ou pourquoi celle-ci est située sur un continuum

Bruno Rodrigues*

Résumé (max 300 mots)

La reproductibilité en science est cruciale, mais elle est assez difficile à mettre en place. Il existe pourtant une multitude d'outils et de paquets pour R qui permettent de rendre un projet reproductible. Dans ma présentation, je vais lister ces outils et fournir des scripts simples, mais fonctionnels, qui vous permettront de commencer rapidement. Je vais aussi expliquer ce qu'il se passe quand l'un d'entre eux, MRAN, cessera d'exister en Juillet 2023. La reproductibilité, ce n'est pas seulement utiliser les bons outils, mais c'est aussi de la gestion des risques. Je vais donc aussi discuter d'un risque majeur qui arrivera je pense bientôt et qui risque d'impacter beaucoup de projets et comment s'en prémunir.
Mots-clefs : Reproductibilité – Gestion des risques

Développement

La reproductibilité est située sur un continuum et, selon la nature de votre projet, différents outils sont nécessaires. Tout d'abord, comprenons ce que je veux dire par "la reproductibilité est sur un continuum." Supposons que vous ayez écrit un script qui produit un document. Voici la liste de tout ce qui peut influencer le compilation de ce document (autre que les différents algorithmes statistiques fonctionnant sous le capot):

- Version du langage de programmation utilisé ;
- Versions des packages/bibliothèques dudit langage de programmation utilisé ;
- Système d'exploitation et sa version ;
- Versions des bibliothèques système sous-jacentes (qui vont souvent de pair avec la version du système d'exploitation, mais pas nécessairement)
- Le matériel sur lequel vous exécutez toute ces logiciels.

Donc, par "la reproductibilité est sur un continuum," ce que je veux dire, c'est que vous pourriez configurer votre projet de manière à ce qu'aucun, un, deux, trois, quatre ou tous les éléments précédents soient pris en considération pour rendre votre projet reproductible.

Toutefois, il ne suffit pas simplement de faire attention à ces quatres éléments, mais idéalement il faut prévoir les choses en amont. En effet, si j'ai un script qui charge les 3 paquets R suivants:

- purrr
- ggplot2
- stringr

Je connais bien évidemment cette liste et si je veux rendre mon script reproductible, je dois prendre note des versions de ces 3 paquets (et éventuellement de leurs propres dépendances). Cependant, que se passe-t-il si vous ne connaissez pas cet ensemble de paquets utilisés ? Cela se produit lorsque vous souhaitez configurer un environnement figé, puis distribuer cet environnement. Les développeurs travailleront alors tous sur le même environnement de base, mais vous ne pouvez pas lister tous les paquets qui vont être utilisés car vous n'avez aucune idée de ce que les développeurs finiront par utiliser (et rappelez-vous que le vous du futur est inclus dans ces développeurs, et vous devriez toujours essayer d'être gentil avec votre futur vous).

Cela signifie donc que nous avons deux scénarios :

*Ministère de l'enseignement supérieur et de la recherche au Luxembourg, bruno.rodrigues@mesr.etat.lu

- Scénario 1 : J'ai un script (ou plusieurs) et je veux m'assurer qu'il produira toujours le même produit ;
- Scénario 2 : Je ne sais pas ce que je (ou mes collègues) développerai, mais nous voulons utiliser le même environnement dans toute l'organisation pour développer et déployer des produits de données.

Il s'avère que les solutions à ces deux scénarios sont différentes, mais elles ont toutes un point commun: Docker. On ne pourra plus bientôt compter sur MRAN, mais d'autres alternatives permettront de s'en passer.

Mais ce n'est pas tout, car jusqu'ici je n'ai parlé que des aspects logiciels. Le dernier point des éléments que je listais plus haut mentionnait le matériel sur lequel on exécutait le code. On a vu récemment que lorsqu'Apple a changé l'architecture de ces ordinateurs, les logiciels compilés pour l'ancienne architecture basée sur des processeurs Intel ne fonctionnaient plus (du moins sans couche de compatibilité) sur leur nouveaux ordinateurs dont l'architecture est basée sur ARM.

Ici, il n'y a pas de solution miracle: il faut avoir accès au code source des différents composants pour pouvoir recompiler tout le projet, si nécessaire, sur n'importe quelle architecture.