

Créez des environnements reproductibles avec **rix**

Bruno Rodrigues

Résumé

S'assurer que nos analyses soient reproductibles est essentiel, et il existe une multitude d'outils pour les utilisateurs de R que nous sommes pour le permettre. Néanmoins, ces outils ne gèrent qu'un seul aspect du continuum de la reproductibilité : `{renv}`, par exemple, permet de "figer" les paquets R pour une analyse, mais pas la version de R elle-même. Le "R Installation Manager" permet d'installer n'importe quelle version de R sur n'importe quel système ; Docker permet de containeriser le tout dans une image depuis laquelle on peut exécuter des conteneurs. Nix, développé par Dolstra et al. (2004) est un outil qui permet de gérer chacune de ces dimensions en même temps : il permet de définir un environnement complet comprenant R, les paquets R et les dépendances système sous-jacentes, et de déployer cet environnement de manière totalement reproductible. Malheureusement, Nix peut sembler très compliqué pour des novices, c'est pourquoi j'ai développé le paquet `{rix}`, qui permet de définir des environnements de développement pour R de manière très simple. Dans cette présentation, j'expliquerai comment Nix assure la reproductibilité d'une analyse et comment on peut l'utiliser simplement grâce à `{rix}`.

Mots-clefs : Package - Reproductibilité

Développement

`{rix}` est un paquet R qui tire parti de Nix, un puissant gestionnaire de paquets axé sur la reproductibilité. Avec Nix, il est possible de créer des environnements spécifiques à un projet contenant une version spécifique de R et des paquets R (ainsi que d'autres outils ou langages, si nécessaire). Vous pouvez utiliser `{rix}` et Nix pour remplacer `{renv}` et Docker par un seul outil. Nix est un logiciel incroyablement utile pour garantir la reproductibilité des projets. Par exemple, il permet d'exécuter des applications web telles que des applications Shiny ou des API plumber dans un environnement contrôlé.

Nix a un coût d'entrée assez élevé cependant. Nix est un logiciel complexe qui dispose de son propre langage de programmation, également appelé Nix. Son objectif est de résoudre

un problème complexe : définir des instructions sur la manière de construire des logiciels et de gérer les configurations de manière déclarative. Cela garantit que le logiciel est installé de manière entièrement reproductible, sur n'importe quel système d'exploitation ou matériel.

{rix} fournit des fonctions pour vous aider à écrire et déployer des expressions écrites dans le langage Nix. Ces expressions seront les entrées du gestionnaire de paquets Nix, pour construire des ensembles de paquets logiciels et les fournir dans un environnement de développement reproductible et cohérent. Ces environnements peuvent être utilisés pour l'analyse de données interactive, ou reproduits lors de l'exécution de pipelines dans des systèmes CI/CD. Dans la collection “nixpkgs” (l'équivalent du CRAN pour Nix), il y a actuellement plus de 80 000 logiciels disponibles via le gestionnaire de paquets Nix. Avec {rix}, vous pouvez définir et construire des environnements R isolés via le gestionnaire de paquets Nix avec facilité. Ainsi, les environnements contiennent R et tous les paquets requis dont vous avez besoin pour votre projet. Vous pouvez également ajouter n'importe quel autre logiciel nécessaire à votre analyse. L'écosystème R de Nix comprend actuellement la quasi-totalité des paquets CRAN et Bioconductor. Il est également possible d'installer des versions antérieures des paquets R, ou d'installer des paquets depuis GitHub à des commits définis.

Le gestionnaire de paquets Nix est extrêmement puissant. Non seulement il gère très bien toutes les dépendances de n'importe quel paquet de manière déterministe, mais il est également possible avec lui de reproduire des environnements contenant des versions anciennes de logiciels. Il est ainsi possible de construire des environnements contenant la version 4.0.0 de R (par exemple) pour exécuter un ancien projet qui a été développé à l'origine sur cette version de R.

Si vous avez besoin d'autres outils ou langages comme Python ou Julia, cela peut également être fait facilement. Nix est disponible pour Linux, macOS et Windows (via WSL2) et {rix} présente les fonctionnalités suivantes :

- permet d'installer n'importe quelle version de R (depuis la version 3.0.2) et des paquets R pour des projets spécifiques ;
- avoir plusieurs versions de R et des paquets R installées en même temps sur le même système ;
- définir des environnements de développement complets en code et les utiliser n'importe où ;
- exécuter des fonctions R individuelles dans un environnement différent (éventuellement avec une version différente de R et des paquets R) depuis une session R interactive, et récupérer la sortie de cette fonction en utilisant `with_nix()`;

{rix} ne nécessite pas que Nix soit installé sur votre système pour générer des expressions. Cela signifie que vous pouvez générer des expressions sur un système sur lequel vous ne pouvez pas facilement installer de logiciel, puis utiliser ces expressions sur le cloud ou dans un environnement CI/CD pour y construire le projet.

Pour définir un environnement il suffit d'utiliser la fonction `rix()`:

```
rix(r_ver = "4.3.1",  
    r_pkgs = c("dplyr", "chronicler"),  
    ide = "other")
```

Ceci va générer un fichier appelé `default.nix` qui sera ensuite utilisé par le gestionnaire de paquets Nix pour installer R version 4.3.1 ainsi que `{dplyr}` et `{chronicler}` (tels qu'ils étaient à la sortie de cette version de R). Toutes les autres dépendances systèmes, telles que des bibliothèques dynamiques ou compilateurs nécessaires pour générer cette environnement seront aussi installés.

Références

Dolstra, Eelco, Merijn De Jonge, Eelco Visser, et al. 2004. "Nix: A Safe and Policy-Free System for Software Deployment." In *LISA*, 4:79–92.