Perception de R et RStudio par des apprenants dans des cours de science des données biologiques

Philippe Grosjean 2* Guyliann Engels 1†

Résumé

Faut-il enseigner les statistiques ou les sciences des données à des apprenants "non techniques", c'est-à-dire, potentiellement réticents à l'utilisation d'un ordinateur et à la programmation, avec un environnement logiciel constitué de R, RStudio, R Markdown et git ? Notre expérience de trois années d'enseignement des sciences de données biologiques en classe inversée et par projets dans un cursus universitaire (voir https://wp.sciviews.org pour le matériel en ligne) donne de bons résultats. Cependant, la question relative à la perception de ces outils par les apprenants n'a, à notre connaissance, pas encore été abordée. Cet exposé présente l'utilisabilité perçue de ces logiciels, la charge cognitive liée à l'utilisation de tutoriels {learnr} pour assurer la transition entre théorie et pratique, ainsi que l'émotion prédominante à l'utilisation d'un tel environnement logiciel. Le caractère pointu de ces outils est clairement perçu et peut générer un stress perceptible chez les primo-apprenants (durant notre premier cours en second Bachelier). Mais une pédagogie sur la durée (au moins trois quadrimestres successifs) et évolutive évitent une confrontation brutale aux fonctions les plus techniques, ce qui amène les apprenants à s'approprier progressivement ces logiciels avec au final, une perception positive de leur expérience.

Mots-clefs : RStudio IDE – Science des données – Apprentissage – Perception – Environnement logiciel

Développement

L'environnement de développement intégré (IDE) RStudio est largement utilisé pour les analyses statistiques autour de R. D'autres interfaces utilisateurs existent, naturellement. Par exemple, R Commander ou JASP) visent à simplifier l'utilisation de R pour les utilisateurs occasionnels et/ou "non techniques", c'est-à-dire ceux qui n'ont pas l'habitude de programmer sur ordinateur. A l'opposé, les IDE comme RStudio conviennent en principe mieux aux programmeurs et utilisateurs plus avancés. Les IDE offrent des fonctionnalités bien plus évoluées qui conduisent à l'adoption de bonnes pratiques, telles que l'utilisation d'un système de gestion de version, la modularisation du code, sa documentation et son test systématique.

Les étudiants suivant un cursus universitaire de biologie ont des cours de (bio)statistiques, voir plus largement de science des données incluant aussi la gestion et le remaniement des données, la recherche reproductible, la présentation des résultats de manière professionnelle, ... Ces étudiants sont néanmoins à classer plutôt dans la catégorie "non technique". Beaucoup sont réticents à l'idée d'aborder la programmation, ainsi que des outils logiciels pointus. Ceci justifie le choix d'interfaces comme R Commander pour les exercices de statistiques par certains professeurs.

A l'Université de Mons, en Belgique, nous enseignons la science des données biologiques à l'aide de R, RStudio, R Markdown, tidyverse, git et GitHub dans un ensemble de trois cours obligatoires dans le cursus de biologie (75h en second Bachelier, 60h en troisième Bachelier et 36h en premier Master), éventuellement complétés par un ou deux cours de perfectionnement à option en second Master. Ces cours sont accessibles en ligne depuis https://wp.sciviews.org via un site Wordpress qui intègre des ouvrages {bookdown} incluant des exercices sous forme H5P et applications {shiny}. La pratique se fait sous forme de projets GitHub Classroom individuels ou en groupes. Le trait d'union entre théorie et pratique est réalisé systématiquement via des tutoriels {learnr} visant à l'auto-évaluation des acquis. Ces enseignements sont donnés en classes inversées et

*Service d'écologie numérique, Institut Complexys & Infortech, Université de Mons, Belgique, philippe.grosjean@umons.ac.be
†Service d'écologie numérique, Institut Complexys & Infortech, Université de Mons, Belgique, guyliann.engels@umons.ac.be

en apprentissage par projets, de sorte que toutes les heures en présentiel sont consacrées à de la pratique sur ordinateur pour analyser des données biologiques. La maîtrise des logiciels se fait de manière progressive durant le premier cours au travers de 12 modules de difficulté croissante qui mélangent apprentissage de la science des données et appropriation des outils.

Les étudiants peuvent utiliser soit les ordinateurs mis à leur disposition dans les salles informatiques de l'Université, soit leur propre ordinateur sous Windows, MacOS ou Linux. Environ les 4/5 des étudiants optent pour cette seconde option. Afin de limiter les différences entre environnements logiciels et les difficultés d'installation, une machine virtuelle sous VirtualBox, totalement préconfigurée, est utilisée. Cette machine virtuelle reste disponible après les cours pour les analyses ultérieures contrairement à une solution dédiée sur un serveur ou sur le cloud qui ne serait plus accessible une fois le cours terminé.

Ces outils logiciels et cette approche de l'enseignement de la science des données à de futurs biologistes donnent satisfaction du point de vue des enseignants, aussi bien au niveau de la gestion quotidienne durant les cours qu'au niveau des acquis constatés pour la majorité des apprenants (faible taux de décrochage inférieur à 10% malgré les exigences élevées, assiduité importante et faible taux d'échec inférieur à 20%). Mais qu'en est-il de la perception de ces outils logiciels par les étudiants eux-même?

Nous avons utilisé des questionnaires d'opinion édumétriquement validés afin d'évaluer l'utilisabilité (degré selon lequel le logiciel peut être utilisé) selon l'échelle SUS, "System Usability Scale" (Brooke (2013)) de R + RStudio + R Markdown, un questionnaire Nasa-TLX (Hart and Staveland (1988)) pour évaluer la charge cognitive (quantité de ressources mentales investies) lors de l'utilisation des tutoriels {learnr} dans RStudio, et enfin, une roue des émotions de Genève (https://www.unige.ch/cisa/gew/) pour déterminer l'état émotionnel général résultant de l'utilisation de ces outils. 99 étudiants des 133 inscrits à l'un des trois cours obligatoires y ont répondu.

L'analyse des résultats obtenus montre que l'utilisabilité perçue par les étudiants évolue peu au fil de l'apprentissage. R + RStudio + R Markdown sont clairement perçus comme moyennement utilisables d'emblée et sont très clairement ressentis comme des outils pointus nécessitant un apprentissage important avant d'être bien maîtrisés. Cependant la roue des émotions indique qu'une certaine fierté, un intérêt, et un contentement se dégagent d'avoir réussi à s'approprier ces outils pour une part majoritaire des étudiants inscrits aux cours 2 et 3, donc les apprenants les plus avancés. Pour les étudiants du premier cours, les émotions de type mépris, dégoût, peur ou colère sont affichées plus souvent. Ceci indique que l'appropriation de ces logiciels ne se fait pas facilement et qu'il en résulte un stress perceptible. Ce n'est donc qu'après environ une centaine d'heures de pratique que l'apprenant "non technique" semble acquérir un degré de maîtrise suffisant pour se sentir à l'aise. Il se peut aussi que l'apprentissage en distanciel lié au confinement COVID-19 sur une bonne partie de la formation ait eu un effet plus néfaste sur les primo-apprenants que sur ceux ayant déjà une certaine maîtrise préalable des logiciels.

Enfin, une charge cognitive raisonnable est observée de manière quasi-unanime chez les apprenants lors de l'utilisation des tutoriels {learnr}. Ces tutoriels sont très appréciés, et ils sont nettement perçus comme indispensables pour faire la transition entre la partie théorique des {bookdown} et la mise en pratique dans les projets GitHub Classroom sous R et RStudio.

En conclusion, l'apprentissage d'un langage tel que R avec un IDE, un système de mise en forme de documents comme R Markdown et un système de gestion de version git sont parfaitement envisageable pour des apprenants "non techniques" *a priori* réticents à la programmation, mais aux conditions suivantes :

1. L'approche pédagogique doit laisser une très large part du temps consacré à la pratique sur ordinateur,
2. L'appropriation doit se faire de manière très progressive,
3. L'utilisation de tutoriels d'auto-apprentissage/auto-évaluation, par exemple sous forme {learnr}, facilite la transition entre théorie et pratique,
4. L'apprentissage doit être continu et s'étaler sur une durée suffisamment longue (dans notre cas, une centaine d'heures réparties sur trois quadrimestres) pour que les apprenants puissent s'approprier les diverses fonctions et se sentent suffisamment à l'aise pour en tirer pleinement profit.

Références

Brooke, John. 2013. "SUS: A Retrospective." *Journal of Usability Studies* 8 (2): 29–40.

Hart, Sandra, and Lowell Staveland. 1988. "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research." *Advances in Psychology* 52: 139–83.