2 Langages de programmation

2.1 Point d'observation: Histoire des langages de programmation pour l'analyse de données

L'histoire des langages de programmation dédiés à l'analyse de données s'inscrit dans une évolution constante des besoins et des méthodes en sciences sociales. Avant l'ère numérique, l'analyse de données se faisait principalement à la main ou à l'aide de calculatrices mécaniques, avec des méthodes statistiques standardisées mais limitées par la capacité humaine à traiter des volumes massifs d'informations. Ces contraintes ont poussé les chercheurs à chercher des moyens plus efficaces de manipuler les données, ce qui a ouvert la voie à l'ère informatique et aux premiers logiciels dédiés à la statistique.

Dans les années 1960 et 1970, les premiers langages et logiciels statistiques, tels que kSPSS (1968), SAS (1976), et STATA (1985), ont vu le jour. Ces outils permettaient aux chercheurs d'automatiser les calculs statistiques complexes sur des ordinateurs de grande taille, en accélérant considérablement les analyses. Ces logiciels propriétaires ont joué un rôle crucial dans la standardisation de l'analyse statistique en sciences sociales, en offrant des plateformes robustes, mais souvent coûteuses et rigides.

L'émergence de l'open source dans les années 1990, avec des projets tels que Linux et d'autres logiciels libres, a commencé à influencer le domaine de la statistique. C'est dans ce contexte que le langage R a été créé, en 1993, par Ross Ihaka et Robert Gentleman, à l'université d'Auckland, en Nouvelle-Zélande. R est basé sur le langage de programmation S, développé chez Bell Labs à la fin des années 1970. Contrairement à ses prédécesseurs propriétaires, R a été conçu pour être gratuit, flexible et extensible, ce qui en a fait un outil populaire parmi les chercheurs qui cherchaient à personnaliser leurs analyses tout en ayant accès à des ressources communautaires.

R a répondu à un besoin pressant de la communauté scientifique : celui de pouvoir accéder à des outils puissants sans avoir à payer des licences onéreuses, tout en bénéficiant d'une liberté totale dans le développement de nouvelles méthodes d'analyse. Grâce à sa structure ouverte, R a rapidement évolué dû aux contributions de statisticiens et de programmeurs du monde entier, devenant l'un des langages de programmation les plus populaires pour l'analyse de données. Aujourd'hui, il est largement utilisé non seulement en sciences sociales, mais aussi en biostatistique, en économie et en science des données.

L'évolution des langages de programmation pour l'analyse de données illustre comment les besoins des chercheurs en termes de flexibilité, d'accessibilité et de partage ont façonné les outils que nous utilisons aujourd'hui. Au fil des décennies, les langages se sont adaptés aux nouvelles méthodes et à l'augmentation des capacités de calcul, offrant des possibilités toujours plus larges pour l'exploration et l'analyse de données.

2.2 Arpentage et choix éditorial: Pourquoi R?

Il existe deux types de langages de programmation pour analyse de données. Les logiciels à licences comme SAS, STATA et SPSS, et les langages *OpenSource* tels que Python et Julia. **R** est un langage de programmation *OpenSource* développé par des statisticiens, pour des statisticiens, dans les années 1990 (Tippmann, 2015). Le langage de programmation **R** a plusieurs avantages qui font de lui un outil puissant et utile pour tout chercheur. L'un de ses grands avantages est qu'il est *OpenSource*. Ayant déjà abordé le sujet dans le chapitre précédent, il sera question ici de simplement rappeler les grandes lignes de l'argument, à savoir que : 1) l'*OpenSource* est gratuit d'utilisation; 2) l'*OpenSource* est développé par les utilisateurs et non par des corporations, ce qui lui procure une grande flexibilité; et 3) il permet aux utilisateurs de créer leurs propres fonctions qui répondent à leurs besoins. À l'inverse, les logiciels à licences sont coûteux, rigides et l'ajout de fonctionnalités se fait par les développeurs internes à la compagnie. Ces formalités rendent le processus plus lent et réduisent l'éventail des possibilités pour la personne chercheuse. Ceci étant dit, certains avanceront que c'est justement ce processus interne lent qui assure la validité et la fiabilité des analyses effectuées par SAS, STATA ou SPSS. Or, dans son livre dédié aux utilisateurs de SPSS et de SAS, Muenchen (2011) soulève le point que bien souvent, ce sont des individus atomisés qui développent les nouvelles fonctionnalités de ces langages et que le processus de révisions se fait ensuite par des comités internes de testeurs. Il en va de même pour le développement des *packages* R dans la mesure où ce dernier se voit testé et amendé par plusieurs programmeurs indépendants dans un processus itératif des plateformes telles que GitHub. De plus, bien des nouvelles techniques statistiques sont développées pour R par des chercheurs qui publient leur travail dans des journaux académiques revus par des pairs, assurant la qualité du procédé. Le fait que SAS et SPSS permettent à leur utilisateur d'intégrer des routines R à leur programme est un indicateur fort ne serait-ce que de l'utilité de R (Muenchen, 2011). Le langage de programmation **R** permet également de réaliser une grande quantité de tâches de recherche. En effet, les personnes programmant en **R** peuvent notamment manipuler et visualiser des données, faire différents types d'analyses, créer des fonctions et faire les automatiser en plus de pouvoir combiner **R** avec certains langages de balisages comme LaTeX, Markdown et HTML.

D'un autre côté, l'utilisation du langage de programmation **R** peut être perçue comme ayant certains inconvénients. Plusieurs disent que la courbe d'apprentissage peut être plus grande que celle de programmes à licences. La véracité de cet argument est discutable. Les programmes demandant des licences ont également un coût d'entrée. De plus, les nouvelles itérations de ces logiciels amènent des changements demandant une période d'adaptation pour la

personne chercheuse. D'autres disent que le développement *OpenSource*, spécifiquement celui du langage de programmation **R**, se fait de façon anarchique. Cela est davantage une question d'opinion et de conception du monde qu'une vérité. Le développement de *package* se fait effectivement de manière décentralisée et toute personne sachant programmer en **R** peut collaborer à cette communauté. Bien qu'il n'y ait pas d'autorité centrale, les *packages* sont regroupés sur le *Comprehensive R Archive Network* (CRAN) (voir le https://cran.r-project.org/ pour plus d'information). Le site a une politique de dépôt stricte, ainsi les *packages* doivent être suffisamment documentés. Il est également possible d'y télécharger le langage de programmation **R**. Ce langage, ainsi que ces différents *packages*, sont disponible sur Windows, macOS et Linux.

2.3 Manuel d'instruction: Apprendre à programmer en R

2.3.1 Où coder en R ?

Un environnement de développement intégré (IDE) permet aux programmeurs de centraliser les différents aspects de l'écriture d'un programme informatique. Il permet de réaliser toutes les activités courantes d'un programmeur – l'édition du code, la construction des exécutables et le débogage – au même endroit. Les environnements de développement intégrés sont conçus pour maximiser la productivité des programmeurs. Ils fournissent de nombreuses fonctionnalités – notamment la coloration syntaxique (Le surlignage différent pour chaques éléments du code) et le contrôle de version – pour créer, modifier et compiler du code. Certains environnements de développement intégré sont dédiés à un langage de programmation spécifique. Par conséquent, ils contiennent des fonctionnalités qui sont plus adaptées aux paradigmes de programmation du langage auquel ils sont associés. Enfin, il existe de nombreux environnements de développement intégré multilingues.

Comme mentionné précédemment, R est l'un des langages de statistiques et d'exploration de données les plus populaires en sciences sociales. R est pris en charge par de nombreux environnements de programmation. Plusieurs ont été spécialement conçus pour la programmation en R – le plus notable étant RStudio – tandis que d'autres sont des environnements de programmation universels – tels que Visual Studio Code – qui prennent en charge R via des plugins. Il est également possible de coder en R à partir d'une interface en ligne de commande. Une telle méthode permet la communication entre l'utilisateur et son ordinateur. Cette communication s'effectue en mode texte : l'utilisateur tape une « ligne de commande » – c'est-à-dire du texte dans la console – pour demander à son ordinateur d'effectuer une opération précise, telle que l'exécution d'un fichier de code R.

La suite du chapitre présente RStudio, l'interface de développement le plus populaire pour l'utilisation de R.

Table 2.1: Résumé des principaux outils de programmation pour l'analyse de données

Critères	Logiciels sous licence (SAS, SPSS, STATA)	Python	Julia	R
Accessibilité (Gratuit ou peu dispendieux)	Non	Oui	Oui	Oui
Existence d'une communauté d'utilisateurs	Modérée	Très élevée	En croissance	Très élevée
Popularité dans le champ	Élevée dans certains secteurs	Très populaire	Modérée	Très populaire en sciences sociales et statistiques
Compatibilité avec d'autres outils	Bonne	Excellente	Bonne	très bonne
Transparence et réplicabilité	Faible	Bonne	Bonne	Très élevée
Adaptabilité et flexibilité	Limitée	Très flexible	Très flexible	Très flexible

2.3.2 Qu'est-ce que RStudio ?

RStudio est un projet open source destiné à regrouper les différentes composantes du langage de programmation R en un seul outil (Allaire, 2011). RStudio fonctionne sur tous les systèmes d'exploitation, y compris Windows, Mac OS et Linux. En plus de l'application de bureau, RStudio peut être déployé en tant que serveur pour permettre l'accès Web aux sessions R s'exécutant sur des systèmes distants (Allaire, 2011). RStudio facilite l'utilisation du langage de programmation R en offrant de nombreux outils permettant à l'utilisateur de réaliser aisément ses tâches. Parmi les outils les plus utiles, on retrouve notamment une fenêtre d'aide, de la documentation sur les différents packages R, un navigateur de l'environnement de travail, une visionneuse de données, ainsi que la prise en charge de la coloration syntaxique (Horton, Kleinman, 2015). De plus, RStudio permet de coder dans plusieurs langages et de gérer une grande variété de formats. Il offre également un support pour plusieurs projets ainsi qu'une interface permettant l'utilisation de systèmes de contrôle de version, tels que GitHub (Horton, Kleinman, 2015).

RStudio présente plusieurs avantages. Son utilisation est facile à apprendre pour les débutants. Les principaux éléments d'un IDE sont intégrés dans une interface à quatre volets (Verzani, 2011). Cette disposition comprend une console, un éditeur de code source à onglets pour organiser les fichiers d'un projet, un espace dédié à l'environnement de travail, et un quatrième volet permettant d'afficher des graphiques ou de consulter la documentation sur les différents packages. Ce volet permet également d'accéder au répertoire des packages disponibles pour R et à l'arborescence des fichiers de l'utilisateur. De plus, il est possible de créer plusieurs espaces de travail – appelés projets – facilitant l'organisation des différents workflows.

Il existe plusieurs autres aspects de RStudio que les programmeurs apprécient. Parmi eux, le fait que l'application peut être utilisée via un navigateur Web pour un accès à distance (Verzani, 2011). De plus, RStudio prend en charge plusieurs langages de programmation ainsi que différents langages de balisage. Qui plus est, de nouvelles fonctionnalités sont régulièrement ajoutées pour répondre aux besoins de la communauté scientifique. Enfin, le logiciel R lui-même est souvent mis à jour.

Parmi ce que certains considèrent comme les points faibles de RStudio, on retrouve des éléments liés à la configuration. Certains utilisateurs trouvent que le nombre de raccourcis est limité. D'autres jugent que l'organisation des différents panneaux n'est pas ergonomique, ou que la personnalisation de l'environnement de programmation est insuffisante. De plus, certains utilisateurs ont rapporté que RStudio était plus lent que d'autres alternatives pour certaines opérations, notamment celles impliquant de longs scripts.

2.3.3 Comment utiliser RStudio ?

Bien que de nombreux éléments puissent être personnalisés, la disposition par défaut de RStudio est composée de quatre volets principaux (Verzani, 2011). Dans le coin supérieur gauche se

trouve le volet principal. C'est dans celui-ci que l'utilisateur passera la majeure partie de son temps. On y modifie des fichiers de différents formats et il est possible d'y afficher des bases de données. Dans le coin inférieur gauche se trouvent la console et le terminal. Dans la console, on peut interagir avec R de la même manière que dans le volet principal, mais le code ne sera pas enregistré. Le terminal, quant à lui, est le point d'accès pour la communication entre l'utilisateur et son ordinateur. Bien que les différents systèmes d'exploitation soient livrés avec un terminal intégré, il est également possible d'y accéder à partir de RStudio.

Dans le coin supérieur droit, on retrouve l'espace de travail. Ce volet contient trois éléments : l'environnement global, l'historique et les connexions. L'environnement global est l'endroit où l'utilisateur peut voir les bases de données, les fonctions et les différents autres objets R actifs. Il peut cliquer sur les divers éléments actifs pour les consulter. L'onglet historique permet à l'utilisateur de consulter les derniers morceaux de code R qu'il a exécutés ainsi que les dernières commandes saisies dans la console. L'onglet connexions, quant à lui, permet de connecter l'IDE à une variété de sources de données et d'explorer les objets et données qu'elles contiennent. Il est conçu pour fonctionner avec divers outils permettant de travailler avec des bases de données en R dans RStudio.

Le volet dans le coin inférieur droit, quant à lui, contient plusieurs outils très utiles pour les utilisateurs de RStudio. L'onglet Files permet de naviguer dans les fichiers présents sur l'ordinateur sans avoir à quitter RStudio. L'onglet Plots permet de visualiser les graphiques générés à partir de R, que ce soit en utilisant ggplot2, lattice ou base R. L'onglet Packages permet de consulter les packages installés précédemment par l'utilisateur et d'en consulter la documentation. C'est aussi un des endroits où il est possible d'installer des packages avec RStudio. L'onglet Help permet à l'utilisateur de rechercher et de consulter de la documentation sur de nombreux sujets, notamment sur les différentes fonctions en R ainsi que sur les packages. L'onglet Viewer, quant à lui, permet de visualiser du contenu web local.

Enfin, l'utilisateur peut modifier les dimensions par défaut de chacun des quatre volets principaux. En cliquant sur la séparation entre les sections, il est possible d'ajuster la répartition horizontale de l'espace. De plus, chaque côté dispose d'une autre séparation permettant d'ajuster l'espace vertical. Qui plus est, la barre de titre de chaque volet comporte des icônes pour réduire un composant, maximiser un volet verticalement ou modifier la taille de l'espace de travail (Verzani, 2011 ; Nierhoff et Hillebrand, 2015).

2.4 Conclusion

Le langage de programmation R est un outil très utile pour toutes sortes de tâches, notamment liées aux statistiques et à la visualisation graphique. Sa maîtrise est requise pour accéder à plusieurs emplois, tant dans le monde académique que dans les secteurs public et privé. Nous espérons que le présent chapitre vous a éclairé sur son utilité et sa pertinence dans le monde du travail contemporain. Bien que le langage de programmation R ne doive pas obligatoirement être utilisé avec RStudio, nous pensons que, pour la plupart des utilisateurs,

leur utilisation conjointe est bénéfique et recommandée. RStudio permet également d'utiliser différents langages de balisage compatibles avec R, facilitant ainsi l'utilisation de plusieurs outils complémentaires. L'apprentissage du langage de programmation R apparaît également comme une valeur sûre. Sa longévité dans plusieurs domaines ainsi que la forte croissance de sa base d'utilisateurs laissent présager que connaître au moins les bases de R constitue un énorme avantage pour tout le monde. Pour ceux qui sont particulièrement intéressés par le langage de programmation R et qui souhaitent s'impliquer dans sa communauté, il existe plusieurs conférences internationales et nationales sur R – notamment RConference et useR! – ainsi qu'un journal académique, The R Journal. On retrouve également différentes communautés, telles que R-Ladies, qui mettent de l'avant la diversité des genres dans la communauté du langage de programmation R. Le langage de programmation R est plus qu'un simple outil statistique, il est au centre d'une grande communauté de personnes qui ont à cœur des principes liés à l'inclusion et à l'avancement humain.