

Outils de recherche en sciences sociales numériques

Chaire de leadership en enseignement des sciences sociales numériques (CLESSN)

2023-09-25

Table of contents

| | |
|--|-----------|
| Avant-propos | 1 |
| Introduction | 3 |
| 1 Comment les données massives affectent-elles les sciences sociales? Changements actuels et quelques réflexions sur l'avenir | 5 |
| 1.1 Définition des données massives | 6 |
| 1.2 Les données massives et les sciences sociales<!--AFC: La structure du chapitre est mêlante. Certaines sous-sections devraient-elles être des sous-sous-sections? Pourquoi la section actuelle est-elle aussi courte, devrait elle inclure plus de choses?--> | 8 |
| 1.3 La validité de la mesure en sciences sociales | 9 |
| 1.4 La validité des données massives | 10 |
| 1.4.1 Validité interne des données massives | 11 |
| 1.4.2 Validité externe des données massives | 12 |
| 1.4.3 Données expérimentales | 13 |
| 1.4.4 Données observationnelles | 13 |
| 1.4.5 Validité écologique et observation par sous-groupes . | 15 |
| 1.5 Pourquoi ce qui se passe actuellement mérite-t-il que l'on s'y attarde? | 15 |
| 1.6 En guise de conclusion : trois questions ouvertes pour le futur | 17 |
| 2 1. Le monde du libre | 19 |
| 2.1 1.1 Émergence et sémantique du <i>libre</i> | 21 |
| 2.2 1.2 Logiciel libre et code ouvert | 22 |

Table of contents

| | | |
|----------|---|-----------|
| 2.3 | 1.3 Cas d'étude: R | 24 |
| 3 | 2. Principaux avantages et inconvénients | 27 |
| 3.1 | 2.1 Avantages | 27 |
| 3.1.1 | 2.1.1. Le partage et co-construction des connaissances | 27 |
| 3.1.2 | 2.1.2. Avantages économiques : Une plus grande accessibilité pour tous | 28 |
| 3.2 | 2.2. Inconvénients et défis: | 30 |
| 3.2.1 | 2.2.1. Coûteux en temps | 30 |
| 3.2.2 | 2.2.2. Problème de transparence | 32 |
| 3.2.3 | 2.2.3. Appropriation capitaliste | 33 |
| 3.3 | 3. Les sciences sociales à l'ère du numérique: les enseigne- ments de la philosophie du logiciel libre | 33 |
| 4 | Bibliographie | 35 |
| 5 | Les outils de collecte de données | 37 |
| 5.1 | Le Big Data et les différents acteurs de la société : . | 38 |
| 5.2 | Plateformes de sondages et collecte de données | 39 |
| 5.2.1 | Les principales plateformes web. | 40 |
| 5.2.2 | Les limites des sondages en ligne | 42 |
| 5.3 | Factiva : outils de récolte de données médiatiques . | 45 |
| 5.4 | Les extracteurs : avoir accès à des données massives via du code. | 49 |
| 5.5 | Covidence : outil de récolte d'articles scientifiques . | 50 |
| 5.6 | Conclusion et discussion: | 52 |
| 6 | R ou ne pas R? | 55 |
| 6.1 | Pourquoi R? | 55 |
| 6.2 | Où coder en R ? | 57 |
| 6.3 | Qu'est-ce que RStudio ? | 58 |
| 6.4 | Comment utiliser RStudio ? | 60 |
| 6.5 | Personnaliser son RStudio | 61 |

| | | |
|-----------|--|------------|
| 7 | Baliser les sciences sociales : langages et pratiques | 63 |
| 7.1 | Qu'est-ce qu'un langage de balisage? | 64 |
| 7.2 | Qu'est-ce qu'un langage de balisage? | 67 |
| 7.3 | Quand et pourquoi utiliser un langage de balisage? | 71 |
| 7.3.1 | Avantages | 74 |
| 7.3.2 | Inconvénients | 86 |
| 7.4 | Comment utiliser un langage de balisage? | 89 |
| 8.0.1 | Environnements d'édition et de compilation | 93 |
| 8.1 | Conclusion | 95 |
| 8.2 | Références | 96 |
| 9 | La gestion des références | 97 |
| 9.1 | Pourquoi citer ? | 97 |
| 9.2 | À quoi sert un logiciel de gestion bibliographique ? | 97 |
| 9.3 | Pourquoi Zotero? | 100 |
| 9.4 | Pourquoi BibLaTeX? | 101 |
| 9.5 | Installation et configuration de Zotero | 102 |
| 9.5.1 | Zotero | 102 |
| 9.6 | Conclusion | 106 |
| 10 | Une image vaut mille mots | 107 |
| 10.1 | Introduction | 107 |
| 10.2 | Réflexion théorique | 108 |
| 10.2.1 | Les options disponibles | 108 |
| 10.3 | Réflexion méthodologique | 113 |
| 10.3.1 | Comment utiliser ggplot2 | 113 |
| 10.3.2 | Exemples et fonctionnalités | 113 |
| 10.4 | Trucs et astuces | 113 |
| 10.5 | Pour aller plus loin | 113 |
| 10.6 | Références | 114 |
| 11 | À la quête de l'optimisation | 115 |
| 11.1 | L'importance d'une méthode de travail efficace | 115 |

Table of contents

| | | |
|-----------|---|------------|
| 11.2 | Logiciel de gestion de communication (Slack) | 116 |
| 11.2.1 | Pourquoi utiliser une de ces plateformes | 117 |
| 11.2.2 | Comment utiliser votre logiciel efficacement | 118 |
| 11.3 | Logiciel de gestion de versions décentralisé | 122 |
| 11.3.1 | Pourquoi choisir Git et GitHub? | 122 |
| 11.3.2 | Comment les utiliser efficacement (en parallèle à Dropbox, etc.) | 124 |
| 11.3.3 | Pourquoi prioriser Git et GitHub pour les chercheurs en sciences sociales | 125 |
| 11.3.4 | Pratiques à éviter sur GitHub pour les chercheurs en sciences sociales | 126 |
| 11.3.5 | Exemple d'utilisation de Git et de GitHub pour un chercheur en sciences sociales | 128 |
| 11.3.6 | GitHub Desktop | 131 |
| 11.4 | Conclusion | 133 |
| 11.5 | Outils d'entreposage des données | 133 |
| 11.5.1 | Entreposage de données non sensibles | 133 |
| 11.5.2 | Entreposage de données sensibles | 136 |
| 11.5.3 | Conclusion | 139 |
| 12 | Outils d'intelligence artificielle | 141 |
| 12.1 | Définition et différents type d'IA | 142 |
| 12.2 | Utilisation du package OpenAI | 143 |
| 12.2.1 | Installation et chargement du package | 143 |
| 12.3 | Configuration de l'API | 143 |
| 12.4 | Utilisation de l'API | 144 |
| 12.5 | Notes | 145 |
| 13 | Serpents et échelles | 147 |
| 13.1 | Introduction | 147 |
| 13.1.1 | Datacamp | 148 |
| 13.2 | Débutant | 149 |
| 13.2.1 | Environnements de programmation | 149 |
| 13.2.2 | Les alternatives à Word : les langages de balisage . . | 149 |

Table of contents

| | | |
|-------------------|--|------------|
| 13.2.3 | Serpents | 149 |
| 13.3 | Intermédiaire | 151 |
| 13.3.1 | La gestion des références | 151 |
| 13.3.2 | Visualisation graphique en R | 151 |
| 13.3.3 | Serpents : | 151 |
| 13.4 | Avancé | 153 |
| 13.4.1 | Gestion de projet et de données | 153 |
| 13.4.2 | Outils d'intelligence artificielle | 153 |
| 13.4.3 | Snakes | 153 |
| 13.5 | Conclusion | 154 |
| References | | 155 |

Avant-propos

Ceci est un exemple de citation Adcock and Collier (2001) .

Introduction

1 Comment les données massives affectent-elles les sciences sociales? Changements actuels et quelques réflexions sur l'avenir

L'apparition des données massives (*big data*) dans le paysage technologique représente un de ces cas de plus en plus communs de phénomène hautement technique dont les effets politiques et sociaux sont remarquables. La discussion publique s'est en effet rapidement emparée du sujet, au point de transformer un moment technologique en phénomène social. Les données massives se trouvent ainsi régulièrement présentées dans l'espace public à la fois comme un moyen puissant de développement et d'innovation technoscientifique, de même que comme une menace à la stabilité de certaines normes sociales telles que la confidentialité des informations privées. Il n'est d'ailleurs pas rare que le discours public s'inquiète du danger que poseraient les données massives à la séparation des sphères publique et privée, pourtant centrale à la conception libérale du rôle de la politique qui structure la majorité des débats sociaux, en amalgamant parfois de manière trop rapide l'objet et l'utilisation qui en est faite. Toutefois, ce même discours public s'emporte aussi rapidement à propos des gains technologiques monumentaux réalisés par l'utilisation des données massives.

Dans le domaine des sciences sociales, les avancées dues à l'utilisation des données massives se font de plus en plus fréquentes et l'impact des données massives dans le domaine de la recherche sociale est en ce sens indéniable. Toutefois, d'un point de vue épistémologique, l'utilisation des données

massives en recherche en sciences sociales dans les dernières années laisse plusieurs questions ouvertes dans son sillage.

Comment l'utilisation des données massives change-t-elle la pratique des sciences sociales? Les données massives causeront-elles un changement de paradigme scientifique? Quels impacts auront-elles sur les traditions scientifiques dominantes telles que le béhavioralisme ou l'individualisme méthodologique en sciences sociales?

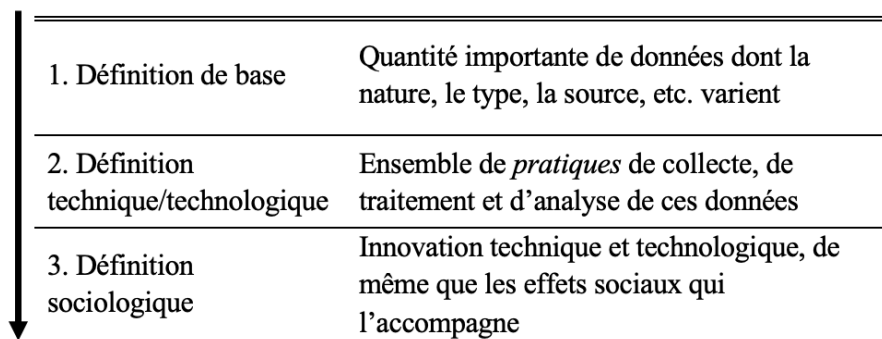
Ce chapitre ne prétend pas offrir de réponses définitives à ces questions, mais plutôt des pistes de réflexion par le biais d'une introduction critique de certains points relatifs aux impacts des données massives sur la recherche en sciences sociales. Premièrement, nous présentons une conceptualisation des données massives. Deuxièmement, nous nous penchons sur les impacts des données massives en sciences sociales et soulignons tout particulièrement comment elles affectent les enjeux de la *validité* interne et externe dans le domaine des sciences sociales. Finalement, nous explorons quelques pistes de réflexion sur l'avenir des données massives en sciences sociales en analysant quelques changements *épistémologiques* que ces données pourraient potentiellement entraîner.

1.1 Définition des données massives

Ce qui définit les données massives comme concept est souvent mêlé avec le phénomène social qui l'accompagne. Il est toutefois possible de démêler le tout en distinguant trois approches conceptuelles des données massives qui sont décrites dans la Figure 1.1.

1. Premièrement, les données massives représentent une ***quantité importante de points d'information*** qui varient selon la nature, le type, la source, etc. Ici, la distinction est simplement quantitative. Il s'agit d'une première dimension à la définition des données massives.

1.1 Définition des données massives



| | |
|---------------------------------------|--|
| 1. Définition de base | Quantité importante de données dont la nature, le type, la source, etc. varient |
| 2. Définition technique/technologique | Ensemble de <i>pratiques</i> de collecte, de traitement et d'analyse de ces données |
| 3. Définition sociologique | Innovation technique et technologique, de même que les effets sociaux qui l'accompagne |

Figure 1.1: image1_1

2. Deuxièmement, d'une perspective technique et technologique, les données massives constituent un ensemble de ***pratiques*** de collecte, de traitement et d'analyse de ces points d'information. Les données massives représentent donc une technique ou une méthode nouvelle de recherche.
3. Finalement, d'une perspective sociologique, les données massives représentent un phénomène incorporant à la fois la dimension propre aux ***développements technologiques, ainsi que les impacts sociétaux de ces développements*** — i.e., les risques à la confidentialité des données, les enjeux relatifs au consentement et à l'autorisation de collecte des informations, les innovations en intelligence artificielle, etc. Cette perspective souligne le caractère essentiellement social des données massives.

Dans les domaines scientifiques et technologiques, la définition courante donnée aux données massives intègre des éléments de ces trois niveaux d'analyse en se référant à la composition et à la fonction des données. Premièrement, la *composition* des données massives est généralement conceptualisée comme comprenant « 4V » : le volume, la variété, la vitesse et

la véracité. Cette conceptualisation jouit d'un large consensus scientifique (Chen, Mao et Liu, 2014; Gandomi et Haider, 2015; Kitchin et McArdle 2016). Par ailleurs, plusieurs chercheurs ont élargi cette définition de la composition des données massives en y incluant, par exemple, la variabilité et la valeur des points de données (CITE). Deuxièmement, la *fonction* des données massives comprend les innovations relatives à l'optimisation, à la prise de décision et à l'approfondissement des connaissances qui résultent de leur utilisation. Ces fonctions touchent des domaines sociaux disparates, incluant le souci d'efficacité et de rendement des secteurs privé et public ainsi que la recherche scientifique pure (Gartner 2012).

1.2 Les données massives et les sciences sociales<!--AFC: La structure du chapitre est mêlante. Certaines sous-sections devraient-elles être des sous-sous-sections? Pourquoi la section actuelle est-elle aussi courte, devrait elle inclure plus de choses?-->

Dans le domaine des sciences sociales, les changements causés par l'utilisation des données massives en recherche sont significatifs. Plusieurs n'hésitent d'ailleurs pas à les qualifier de changements de paradigme dans l'étude des phénomènes sociaux (Anderson 2008; Chandler 2015; Grimmer 2015; Kitchin 2014; Monroe et al. 2015). Dans le cas qui nous intéresse, deux dimensions majeures méritent d'être abordées : (1) une première relative à la validité (interne et externe) des données massives et (2) une seconde, plus large, relative au potentiel changement de posture ou d'orientation épistémologique causé par l'utilisation de ces données en recherche.

1.3 La validité de la mesure en sciences sociales

La validité de la mesure constitue une exigence méthodologique centrale à la recherche en sciences sociales. Les scientifiques cherchent effectivement à s'assurer que ce qui est mesuré — par un sondage, une entrevue, un thermostat ou tout autre outil de mesure — constitue bel et bien ce qui est censé être mesuré. Adcock et Collier définissent plus spécifiquement l'application de la validité de la mesure en sciences sociales par le biais de « scores (including the results of qualitative classification) [that] meaningfully capture the ideas contained in the corresponding concept » (2001: 530).

Toutefois, les problèmes liés à la validité de la mesure sont nombreux et ont une importance considérable. Dans l'étude des phénomènes sociaux et humains, la validité de la mesure prend d'ailleurs une complexité supplémentaire du fait que les données collectées par le biais d'une mesure constituent le *produit de l'observation* d'un phénomène, mais non pas le phénomène en soi. Ainsi, lorsque, dans le contexte d'une recherche, on propose de mesurer l'humeur de l'opinion publique (le phénomène en soi) sur un enjeu politique, on utilise généralement un sondage qui a pour fonction de mesurer le pouls d'un échantillon de la population d'intérêt (ce qui est réellement observé). Cependant, ce que ce sondage mesure ne constitue pas tout à fait l'opinion publique elle-même, mais plutôt un segment populationnel qui se veut représentatif de l'humeur de l'opinion publique. Autrement dit, la mesure et les données collectées ne représentent pas le phénomène — l'opinion publique — en soi.

On a déjà mentionné que la validité de la mesure a de l'importance puisqu'elle garantit que ce qui est mesuré représente réellement ce qu'on croit mesurer. Toutefois, pour être plus spécifique, dans une approche positiviste, la validité de la mesure se traduit généralement par une logique de classification des valeurs attribuées aux différentes manifestations distinctes d'un même phénomène. Par exemple, une mesure de la démocratie comme celle proposée par *Freedom House*, fréquemment

utilisée en science politique, classe les libertés civiles et les droits politiques des États du monde par degré afin de construire un index allant d'un autoritarisme complet à une démocratie parfaite. Les scores représentent, dans ce contexte, une mesure artificielle, mais ordonnée et logique, des idées contenues dans le concept de démocratie telles que libertés civiles et droits politiques. On peut ainsi dire que le souci avec la validité de la mesure traverse les connexions entre (1) le phénomène social étudié (la démocratie), (2) son opérationnalisation (via les libertés civiles et droits politiques) et (3) la méthode de mesure utilisée pour observer et classer d'une certaine façon le phénomène et les données qui en découlent (dans le cas de *Freedom House*, des codeurs indépendants).

1.4 La validité des données massives

En ce qui a trait aux données massives, la question de la validité de la mesure constitue un défi nouveau. Les données massives ont en effet comme avantage d'offrir aux chercheurs soit de nouveaux phénomènes à étudier, soit de nouvelles manifestations et nouvelles formes à des phénomènes déjà étudiés. Les données massives permettent donc d'agrandir la connaissance scientifique.

L'étude de King et al. (2013) représente un cas éclairant de phénomène social que l'utilisation des données massives a rendu possible d'étudier. En se basant sur la collecte de plus de 11 millions de publications sur les réseaux sociaux chinois, King et ses collègues ont pu mesurer la censure exercée par le gouvernement chinois sur les réseaux sociaux ont donc pu observer une manifestation inédite de censure massive qui, sans de telles données, serait probablement demeurée mal comprise d'une perspective scientifique. Le nombre de recherches basées sur l'utilisation des données massives similairement innovantes en sciences sociales est par ailleurs en croissance constante (Beauchamp 2017; Bond et al. 2012; Poirier et al. 2020).

1.4 La validité des données massives

Cependant, il faut aussi souligner que les données massives, en raison de leur complexité, peuvent avoir pour désavantage d’embrouiller l’étude des phénomènes sociaux. Les opportunités scientifiques liées aux données massives s’accompagnent en effet de certaines difficultés méthodologiques. Parmi ces difficultés, trois enjeux sont particulièrement cruciaux : (1) la validité interne, (2) la validité externe et (3) la question d’un changement de posture ou d’orientation épistémologique en sciences sociales causé par les données massives.

1.4.1 Validité interne des données massives

Premièrement, les données massives peuvent représenter un défi à la validité interne des études en sciences sociales en rendant ***pragmatiquement difficile l’établissement de mécanismes causaux clairs***. Ce défi est notamment une conséquence du fait que la plupart des données sont présentement issues d’un processus de génération (*data-generating process*) qui est hors du contrôle des chercheur.e.s. Les données massives proviennent en effet habituellement de sources diverses qui sont externes aux projets de recherche qui les utilisent. Elles ne sont pas donc générées de manière aléatoire sous le contrôle des chercheur.e.s.

Un des problèmes liés à cette situation est qu’il est difficile de garantir une source *exogène* de variation par laquelle les chercheur.e.s éliminent l’effet potentiel des facteurs confondants (*confounders*). La distribution aléatoire d’un traitement et d’un contrôle dans une expérience en laboratoire ou sur le terrain représente le standard le plus élevé permettant de fournir cette source exogène de variation.

Pour le dire autrement, le défi de validité interne avec les données massives constitue un enjeu relatif à la qualité des données. Ce n’est évidemment pas un défi propre ou unique aux données massives. Ce défi s’applique également aux autres types de données. Cependant, dans l’état actuel

des choses, le volume et la variété — deux des 4V — des données massives — textuelles, numériques, vidéos, etc. — peuvent miner la qualité de l'inférence causale entre une cause et une conséquence que permet habituellement un processus contrôlé de génération des données. En somme, la validité interne des données massives est une fonction de la qualité de ces mêmes données.

1.4.2 Validité externe des données massives

Deuxièmement, les données massives représentent un défi plus important pour la validité externe des recherches en sciences sociales (Tufekci 2014; Lazer et Radford 2017; Nagler et Tucker 2015). La préoccupation la plus évidente concerne la **représentativité** des données massives collectées. Comme le soulignent Lazer et Radford (2017), la quantité ne permet pas de corriger pour la non-représentativité des données. Les données massives sont ainsi soumises au même problème de biais de sélection que les autres types de données observationnelles, tels un sondage ou une série d'entrevues, traditionnellement utilisés en sciences sociales.

Le cas célèbre de l'erreur de prédiction du *Literary Digest* lors de la campagne présidentielle américaine de 1936 illustre bien ce problème récurrent. Lors de cette campagne, le *Literary Digest* a prédit à tort la victoire du candidat républicain Alf Landon sur le président démocrate sortant Franklin D. Roosevelt, puisque son échantillon de répondants surreprésentait les électeurs plus aisés, traditionnellement plus républicains, au détriment des électeurs moins aisés, plus généralement proches du Parti démocrate. Cette erreur de surreprésentation dans l'échantillon est due au fait que le *Literary Digest* a effectué un échantillonnage basé sur les listes téléphoniques et le registre des propriétaires de voitures, biaisant par le fait même l'échantillon au détriment des électeurs plus pauvres ne possédant pas de téléphone ou d'automobile, mais qui constituaient un électorat favorable à Roosevelt (Squire 1981). Le biais de sélection du sondage a

1.4 La validité des données massives

ainsi sous-estimé le soutien populaire de Roosevelt de plus de 20 points de pourcentage.

Aujourd’hui, l’utilisation des données massives est soumise aux mêmes risques méthodologiques. L’accumulation massive de données ne permet pas de compenser pour la qualité des données. Les données massives, comme les données plus traditionnelles, sont soumises aux conséquences induites par le processus de génération des données (*data generating process*) comme un échantillonnage.

1.4.3 Données expérimentales

La question du *processus de génération* des données est plus claire quand on considère comment les *données observationnelles* et les *données expérimentales* permettent d’effectuer des *inférences* de manière distincte.

Premièrement, les données massives ne peuvent pas résoudre les enjeux liés aux inférences causales ou explicatives (Grimmer, 2015). En effet, le processus de génération de données expérimentales assure idéalement la validité de l’inférence causale sur l’ensemble de la population visée. Cela prend plus spécifiquement la forme d’un processus de génération des données au sein duquel les chercheur.e.s assurent la distribution aléatoire du traitement entre les deux groupes — traitement et contrôle — garantissant par le fait même une source exogène de variation qui permet d’éliminer l’endogénéité entre la variable indépendante (x) et le résidu (e) et qui assure donc que l’effet observé n’est pas dû à une variable confondante.

1.4.4 Données observationnelles

En ce qui a trait aux données observationnelles, il y a deux points importants. Premièrement, des méthodes d’inférence basées sur des approches par design (*design-based methods*) comme une méthode de régression sur discontinuité ou de variable instrumentale peuvent également garantir

des inférences explicatives et causales valides. Elles nécessitent toutefois plusieurs postulats plus restrictifs dont l’objectif est d’imiter ou de recréer, de la manière la plus fidèle possible, une distribution aléatoire du traitement – ce que la littérature appelle un *as-if random assignment* (Dunning, 2008).

Dans un contexte observationnel, les données massives peuvent donc permettre d’augmenter la précision des estimations causales. Effectivement, comme dans un modèle de régression linéaire, plus l’échantillon est grand, plus l’estimation du coefficient causal ou probabiliste est précise. Par exemple, un échantillon large dans un modèle de régression sur discontinuité permet de restreindre la largeur de bande autour du seuil, garantissant ainsi une distribution presque parfaitement aléatoire des données et une validité plus élevée à l’estimation de l’effet causal.

Deuxièmement, un échantillon de données massives observationnelles issues d’une plateforme comme X — anciennement Twitter — ou Facebook peut fournir une *description* plus fine de certaines dynamiques sociales observées sur les réseaux sociaux. Cependant, c’est la manière dont sont collectées les données de cet échantillon de données massives qui garantit la représentativité de l’échantillon — avec pour objectif un biais de sélection = 0 — et non pas la quantité de données. Généralement, le biais d’un échantillon est une conséquence de la non-représentativité des répondants; dans notre exemple, les utilisateurs des médias sociaux ne sont généralement pas représentatifs de la population entière.

Dans un tel cas, des méthodes de pondération sur des données observationnelles peuvent compenser pour la sur- ou la sous-représentativité de sous-groupes dans un échantillon afin d’assurer la validité de l’inférence entre échantillon et population. Les données massives ont ici une importance puisqu’une pondération fiable nécessite une quantité substantielle d’observations. Une pondération *a posteriori* sera donc plus fiable plus l’échantillon est grand. Les données massives ont ainsi une valeur ajoutée afin d’établir des inférences descriptives plus précises et sophistiquées.

1.5 Pourquoi ce qui se passe actuellement mérite-t-il que l'on s'y attarde?

1.4.5 Validité écologique et observation par sous-groupes

Les données massives peuvent aussi jouer d'autres rôles importants relatifs à la validité externe. Premièrement, les données massives facilitent effectivement la validité externe de certaines études en accroissant la validité écologique (*ecological validity*) des tests expérimentaux, c'est-à-dire le réalisme de la situation expérimentale (Grimmer, 2015: 81). En effet, la variété des sources et des formats de données permet aux chercheurs d'imiter plus concrètement la réalité sur le terrain vécue par les participants aux études.

Deuxièmement, la quantité importante de données rend possible l'observation d'effets précis, spécifiques et inédits par sous-groupes (Grimmer 2015: 81). Alors qu'auparavant, la taille réduite des échantillons ne permettait pas d'effectuer des inférences valides pour des sous-groupes de la population — les écarts-types par sous-groupes étaient trop grands, rendant difficile l'estimation précise d'un paramètre comme la moyenne et impossible celle d'un coefficient —, la taille énorme des échantillons de données massives permet aux chercheurs d'estimer des paramètres qui étaient demeurés extrêmement imprécis jusqu'à aujourd'hui. Notre compréhension des phénomènes sociaux s'en trouve par le fait même approfondie de façon considérable.

1.5 Pourquoi ce qui se passe actuellement mérite-t-il que l'on s'y attarde?

Appréhender l'impact actuel des données massives se révèle d'une importance cruciale pour se préparer à l'avenir. Tout d'abord, cela s'avère propice à une prise de décision éclairée. En scrutant comment ces données ont été rassemblées, traitées et interprétées dans le passé, nous pouvons rehausser la qualité des choix que nous effectuons aujourd'hui dans des domaines aussi diversifiés que la santé, l'économie et l'environnement. De sur-

1 Comment les données massives affectent-elles les sciences sociales? Changements actuels et

| | Données observationnelles | Données expérimentales |
|-------------------------------------|--|--------------------------------------|
| Processus de génération des données | Non contrôlé par le chercheur | Contrôlé par le chercheur |
| Type d'inférence causale | Locale (LATE) ou populationnelle (ATE) | Populationnelle (ATE) |
| Méthodes | Approches par design | Distribution aléatoire du traitement |
| Exemples | Régression sur discontinuité, variable instrumentale | Expérience de terrain, laboratoire |

Figure 1.2: image2__2

croît, l'analyse des données massives met en lumière des tendances et des motifs subtils échappant aux ensembles d'informations plus restreints. Ces découvertes pavent la voie à des concepts innovants et à des avancements technologiques répondant aux mutations des besoins sociétaux. D'autre part, la préoccupation grandissante liée à la préservation de la vie privée et à l'éthique requiert une appréhension approfondie des erreurs passées dans la manipulation de ces données massives. Évitant la réitération de telles erreurs, nous pouvons ériger des cadres réglementaires plus responsables et instaurer des pratiques de traitement respectueuses des droits individuels. Somme toute, la compréhension de l'incidence actuelle des données massives offre une opportunité inestimable pour contrecarrer les égarements passés et façonner un avenir où l'utilisation de ces données s'inscrit dans une démarche éclairée, éthique et propice au bien-être de l'ensemble de la société.

1.6 En guise de conclusion : trois questions ouvertes pour le futur

Comme nous venons de le voir, la quantité et la variété nouvelle des données massives permettent à la fois un approfondissement de l'analyse de certains phénomènes et l'ouverture de nouvelles avenues de recherche. Il faut toutefois souligner que d'une perspective non pas seulement méthodologique/technique, mais plutôt *épistémologique*, les données massives représentent une *complexification* de l'analyse des phénomènes en sciences sociales. Cela soulève au moins trois questions d'importance, dont les réponses ne nous sont pas encore accessibles, pour l'avenir de la recherche en sciences sociales : (1) les données massives entrent-elles (partiellement du moins) en conflit avec l'impératif de parcimonie qui caractérise la science moderne?; (2) ces données sont-elles dans la continuité ou représentent-elles une coupure dans la tradition béhavioraliste en sciences sociales (et en science politique tout particulièrement)?; (3) et finalement, de manière reliée, les données massives proposent-elles ou non une manière de dépasser l'individualisme méthodologique qui caractérise les sciences sociales contemporaines?

2 1. Le monde du libre

”Vers une science numérique plus transparente: l’apport du logiciel libre et du code ouvert dans les sciences sociales” author: ”Catherine Ouellet et Jozef Rivest”

Catherine Ouellet et Jozef Rivest

Ce chapitre vise à initier les lecteurs et lectrices aux concepts fondamentaux du logiciel libre. Pour ce faire, nous présenterons, dans un premier temps, l’historique de ce mouvement afin de pouvoir le situer temporellement. De cette façon, nous pourrions mieux comprendre les motivations derrière ce mouvement, mais aussi ses influences actuelles. Ensuite, nous distinguerons le logiciel libre du code ouvert. Bien que les deux soient très près l’un de l’autre, il est important de les distinguer puisqu’ils ne renvoient pas aux mêmes caractéristiques et aux mêmes fondements. Après coup, nous utiliserons un exemple concret pour illustrer le propos: R, et ses différentes librairies. La dernière section du chapitre présentera certains avantages, certains inconvénients ainsi que des défis qui se posent pour ce monde. En guise de conclusion, nous souhaitons mettre l’emphase sur l’apport du logiciel libre et du code ouvert afin d’assurer la transparence, la reproductibilité ainsi que la qualité des recherches scientifiques.

« Vous n’avez pas à suivre une recette avec précision. Vous pouvez laisser de côté certains ingrédients. Ajouter quelques champignons parce que vous en raffolez. Mettre moins de sel car votre médecin vous le conseille — peu importe. De surcroît, logiciels et recettes sont faciles à partager. En donnant une recette à un invité, un cuisinier n’y perd que du temps et le

2 1. Le monde du libre

coût du papier sur lequel il l'inscrit. Partager un logiciel nécessite encore moins, habituellement quelques clics de souris et un minimum d'électricité. Dans tous les cas, la personne qui donne l'information y gagne deux choses : davantage d'amitié et la possibilité de récupérer en retour d'autres recettes intéressantes. » - Richard Stallman (**williams2010?**)

Cette analogie illustre bien trois concepts au coeur de la philosophie de Richard Stallman, souvent considéré comme le père fondateur du logiciel libre : liberté, égalité, fraternité. Les utilisateurs de ces logiciels sont libres, égaux, et doivent s'encourager mutuellement à contribuer à la communauté. Ainsi, un logiciel libre est généralement le fruit d'une collaboration entre développeurs qui peuvent provenir des quatre coins du globe. Une réflexion éthique est au coeur du mouvement du logiciel libre, dont les militants font campagne pour la liberté des utilisateurs dès le début des années 1980. La Free Software Foundation (FSF), fondée par Richard Stallman en 1985, définit rapidement le logiciel «libre» [free] comme étant garant de quatre libertés fondamentales de l'utilisateur: la liberté d'utiliser le logiciel sans restrictions, la liberté de le copier, la liberté de l'étudier, puis la liberté de le modifier pour l'adapter à ses besoins puis le redistribuer¹[La redistribution doit évidemment respecter certaines conditions précises, dont l'enfreint peut mener à des condamnations [http://www.softwarefreedom.org/resources/2008/shareware.html]. Il s'agit ainsi d'un logiciel dont le code source¹ est disponible, afin de permettre aux internautes de l'utiliser tel quel ou de le modifier à leur guise. Puisque le langage machine est difficilement lisible par l'homme et rend la compréhension du logiciel extrêmement complexe, l'accès au code source devient essentiel afin de permettre à l'utilisateur de savoir ce que le fait programme fait réellement. Seulement de cette façon, l'utilisateur peut *contrôler* le logiciel, plutôt que de se faire contrôler par ce dernier (**stallman1986?**).

¹Pour rester dans les analogies culinaires, le code source est au logiciel est ce que la recette est à un plat: elle indique les actions à effectuer, une par une, pour arriver à un résultat précis. Encore une fois, cette dernière peut-être adaptée, modifiée, bonifiée.

2.1 1.1 Émergence et sémantique du *libre*

Plusieurs situent les débuts du mouvement du logiciel libre avec la création de la licence publique générale GNU, en 1983, à partir de laquelle va se développer une multitude de programmes libres. Parmi les plus populaires, on retrouve notamment le navigateur Firefox, la suite bureautique OpenOffice et l’emblématique système d’exploitation Linux, qui se développe d’ailleurs à partir de la licence GNU. Aujourd’hui, il s’agit d’un véritable phénomène sociétal : des milliers d’entreprises, d’organisations à but non lucratif, d’institutions ou encore de particuliers adoptent tour à tour ces logiciels, dont la culture globale et les valeurs (entraide, collaboration, partage) s’arriment avec le virage technologique de plusieurs entreprises. Les logiciels libres ont différents usages, en passant par la conception Web, la gestion de contenu, les systèmes d’exploitation, la bureautique, entre autres. Ils permettent donc de répondre à plusieurs types de besoins numériques et informatiques.

“Les principes du logiciel libre ont également inspiré de nombreuses initiatives non directement liées à l’informatique et au développement des logiciels libres. La plus connue est sans aucun doute Wikipédia, qui se définit comme une encyclopédie libre, s’inspirant en cela explicitement du modèle du logiciel libre. Soulignons également les licences Creative Commons et le mouvement des archives ouvertes et de libre accès aux revues scientifique”

Attention, le logiciel libre est avant tout une philosophie, voire un mouvement de société. C’est une façon de concevoir la communauté du logiciel, où le respect de la liberté de l’utilisateur est un impératif éthique (**williams2010?**). Par conséquent, le terme libre, *free* en anglais, porte à confusion. Celui-ci ne signifie pas qu’un logiciel libre est nécessairement gratuit. Certes, plusieurs sont effectivement téléchargeables gratuitement. Toutefois, il est aussi possible de (re)distribuer des logiciels libres payant. Par ailleurs, aucun logiciel libre n’est réellement « gratuit » dans la mesure où son déploiement et son utilisation nécessitent généralement différents

2 1. Le monde du libre

coûts, dont les degrés sont variables en fonction des compétences et de l'infrastructure dont disposent les utilisateurs (coût d'apprentissage, coûts d'entretien, etc.). Enfin, il est important de garder en tête que les logiciels libres possèdent eux aussi une licence - cette dernière est d'ailleurs garante des libertés que confèrent les logiciels libres aux utilisateurs.

2.2 1.2 Logiciel libre et code ouvert

Parallèlement au logiciel libre, il y a aussi le code ouvert, ou *open source*. *A priori*, la dénomination du logiciel libre et celle de l'*code ouvert* semble suggérer qu'il s'agit de synonymes. Dans les deux cas on dirait que l'on fait référence à des logiciels, par exemple, qui sont exempts de restrictions d'utilisations et auxquels les utilisateurs peuvent participer au développement. Cependant, il y a une distinction importante entre les deux.

Bien que les deux renvoient sensiblement aux mêmes types de logiciels, les tenants de ces approches ne partagent pas la même perspective. Comme (stallman2022?) l'explique, le logiciel libre est d'abord et avant tout un mouvement qui fait "campagne pour la liberté des utilisateurs de l'informatique". Le code ouvert, quant à lui, met l'accent sur les avantages pratiques, plutôt que de militer pour des principes.

Le terme *code ouvert* sera introduit seulement en 1998 afin de clarifier l'ambiguïté dans la dénomination "logiciel libre"², *free software* en anglais, afin de spécifier que le code source était accessible, et non pas que le logiciel était "gratuit"(ballhausen2019?). De plus, les logiciels code ouvert, doivent respecter certains critères quant à la distribution de leurs logiciels (opensourceinitiative2006?). Nous aborderons ces critères dans le prochain paragraphe.

²Soit ceux qui ont été conçus suivant les principes philosophiques et "moraux" qui sous-tendent ce mouvement.

2.2 1.2 Logiciel libre et code ouvert

Afin de mieux distinguer les deux, il est utile de faire référence aux critères qui composent ces deux éléments, et qui constituent la base de leur définition. Tout d’abord, le logiciel libre se définit sur la base de quatre libertés: 1) liberté d’utiliser le programme tel que désiré; 2) liberté d’étudier le fonctionnement du programme et de le modifier pour ses propres besoins; 3) liberté de re-distribuer des copies; 4) liberté de distribuer des copies de la version “améliorer” du programme pour ses pairs (**ballhausen2019?**). Concernant le *code ouvert*, tout logiciel qui souhaite être inclut sous cette appellation doit respecter dix critères: 1) Redistribution gratuite; 2) doit inclure le code source; 3) doit permettre les modifications et les travaux dérivés; 4) intégrité du code source; 5) ne doit pas discriminer des personnes et/ou groupes; 6) ne doit pas restreindre personne dans l’utilisation du logiciel pour un domaine d’activité; 7) distribution d’une license pour l’utilisation; 8) la license ne doit pas être spécifique pour un produit; 9) la license ne doit pas placer de restriction sur d’autres programmes; 10) la license doit être technologiquement neutre³ (**opensourceinitiative2006?**).

Il est aussi utile de les distinguer des logiciels “non-libres”, soit les logiciels propriétaires: “Son utilisation, sa redistribution ou sa modification sont interdites, ou exigent une autorisation spécifique, ou sont tellement restreintes qu’en pratique vous ne pouvez pas le faire librement” (**systemedexploitationgnu2023?**). Par contraste, la licence libre confère des droits de propriétaire. L’utilisateur a le droit d’installer le logiciel sur autant d’ordinateurs que désiré, le modifier selon ses besoins et le distribuer avec ou sans ses modifications. Il peut même demander d’être payé pour distribuer des copies, avec ou sans ses modifications. Par exemple, le logiciel Ubuntu, une version de Linux, peut être téléchargé gratuitement du site Ubuntu.com. Il est aussi vendu par Amazon.com pour 12\$ la copie, plus les frais d’expédition!

Comme nous le constatons, le logiciel libre et le *code ouvert* ont certaines

³Pour plus d’informations sur ces caractéristiques, nous encourageons les lecteurs à se référer au lien web de la source. Ils y trouveront un contenu détaillé pour chacune des caractéristiques sus-mentionnées.

similitudes puisqu'ils adhèrent tous les deux à la même vision du logiciel, ainsi que de son accessibilité. Toutefois, il est important tout de même de les distinguer puisqu'ils ont des origines différentes, et qu'ils mènent à certaines pratiques qui sont différentes. La prochaine section utilise un cas concret afin d'expliquer l'effet du libre, et l'utilité que cela peut avoir.

2.3 1.3 Cas d'étude: R

Afin d'illustrer le tout plus concrètement, nous utiliserons ici le cas du logiciel R. Il s'agit d'un logiciel statistique que tous les utilisateurs peuvent télécharger gratuitement, et dans lequel il n'y a pas d'achats supplémentaires pour avoir accès à des fonctionnalités supplémentaires par exemple. Bien que ce logiciel soit déjà riche en fonctions et commandes, plusieurs utilisateurs ont développé des *packages*, des librairies externes, afin de bonifier les fonctions de base (arel-bundock2021?).

Utilisons un cas d'étude afin de démontrer l'apport des librairies externes. Par exemple, je souhaite savoir la probabilité de survie à bord du Titanic en fonction du genre. Je pourrais résumer mon intérêt avec sous la notation suivante: $P(Y = \text{Survie} | X = \text{Femme})$. Cela se lit "la probabilité de survie étant donné que nous soyons une femme". Pour ce faire, je dois utiliser l'ensemble de données `titanic`, disponible en format csv. Je dois donc installer et télécharger la librairie `readr` afin que R puisse importer et lire les données. Ensuite je vais utiliser la commande `table`, offerte dans celles de base, afin de visionner mes données. Cette dernière commande affichera un tableau croisé.

```
library(readr) ①  
  
dat <- read_csv("data/titanic.csv") ②  
  
table(dat$survie, dat$femme) ③
```


- ① Téléchargement de la librairie **readr** qui nous permettra de lire des ensembles de données en format **.csv**.
- ② Importation d'une banque de données en format **.csv**.
- ③ Impression d'un tableau croisé afin d'observer la distribution des hommes et des femmes (colonnes), croisé avec la survie (lignes).

```

      0    1
0 709 154
1 142 308

```

Comme nous le voyons ici, la librairie **readr**, développé par plusieurs individus⁴, nous a permis d'importer l'ensemble de données sur le Titanic. Toutefois, le format du tableau n'est pas très esthétique. Pour remédier à ce problème, nous pouvons installer et utiliser la librairie **modelsummary** qui nous permettra de créer rapidement des tableaux croisés plus esthétique, et qui contiendront davantage d'informations, facilitant la lecture et notre compréhension de la relation qui nous intéresse.

```

library(modelsummary) ①

Tableau.2 <- datasummary_crosstab(survie ~ femme, data = dat) ②

Tableau.2

```

- ① Téléchargement de la librairie **modelsummary**.
- ② Création d'un tableau croisé à l'aide de la commande **datasummary_crosstab()**.

Comme nous le voyons, la commande **datasummary_crosstab()** permet facilement de créer des tableaux non seulement plus esthétiques, mais aussi plus informatif. C'est très utile si l'on souhaite incorporer des

⁴Pour avoir la liste complète des contributeurs, les lecteurs peuvent utiliser la commande **?readr** dans **R**, ou bien consulter le lien suivant <https://readr.tidyverse.org>

2 1. Le monde du libre

| survie | | 0 | 1 | All |
|--------|-------|------|------|-------|
| 0 | N | 709 | 154 | 863 |
| | % row | 82.2 | 17.8 | 100.0 |
| 1 | N | 142 | 308 | 450 |
| | % row | 31.6 | 68.4 | 100.0 |
| All | N | 851 | 462 | 1313 |
| | % row | 64.8 | 35.2 | 100.0 |

tableaux dans notre rapport finale, surtout que cette commande nous permet d’exporter les tableaux sous différents format (.docx, LaTeX, .qmd, etc.)

Ces deux librairies que nous venons de présenter en exemple, ne sont que deux des 19 897 disponibles pour R. Elles illustrent très bien la contribution que les utilisateurs peuvent faire au logiciel. Surtout, ces *add on* ont été développés de manière bénévole. Les contributeurs le font par “passion”, et pour en faire profiter la collectivité d’utilisateurs.

Les logiciels libres permettent aux utilisateurs de jouir d’une plus grande liberté dans leur utilisation, ce qui génère des externalités positives puisque ces gens peuvent créer de nouvelles commandes ou fonction et en faire bénéficier toute la collectivité. L’exemple que nous avons utilisé avec R ici reflète très bien cet avantage. La prochaine section de ce chapitre se penche plus en profondeur sur les autres avantages ainsi que sur les inconvénients de ces logiciels.

3 2. Principaux avantages et inconvénients

Dans cette section, nous ne dresserons pas un portrait exhaustif de tous les avantages et les inconvénients du logiciel libre. Cette tâche serait fastidieuse et peu intéressante pour les lecteurs. Notre but ici est de présenter les certains avantages qui sont propre aux logiciels, ainsi que les inconvénients de ceux-ci.

3.1 2.1 Avantages

3.1.1 2.1.1. Le partage et co-construction des connaissances

La grande liberté que ce type de logiciel offre favorise la collaboration entre les utilisateurs, et ce à une échelle pouvant être internationale. Les interactions entre les chercheurs créent une dynamique d'« innovation ascendante » et d'entraide (couture2014?). Ce résultat constitue un important avantage pour le développement de ces logiciels. Selon certains, et comparativement aux logiciels privés, les logiciels libres ont un niveau plus élevé d'innovation (smith2002?). Contrairement à ceux qui se développent de manière privée et fermée, les logiciels libres permettent à tous les utilisateurs de participer au développement. Ceux-ci partagent ensuite leurs améliorations, ce qui stimule à son tour de nouvelles initiatives. Ainsi, un certain savoir est généré dans cette situation. De plus, il est raisonnable de penser que l'utilité des améliorations, ainsi que leur utilisation par les utilisateurs

3 2. Principaux avantages et inconvénients

en fonction de leur besoin, comme dans le cas de la recherche sociale avec R permet de générer un savoir collaboratif (**couture2020?**). Amélioration constante, entraide, savoir partagé et plusieurs milliers de contributeurs (**couture2014?**), ces éléments résument très bien la philosophie du logiciel libre.

Comme nous le verrons dans la section suivante, cet avantage est couplé avec ceux économiques. Les bas coûts démocratise l'accès à plusieurs logiciels qui sont utiles pour mener des analyses scientifiques. Et ce, pour tous les utilisateurs dans le monde.

3.1.2 2.1.2. Avantages économiques : Une plus grande accessibilité pour tous

Nous pouvons aussi mentionner certains avantages économique dans l'utilisation de logiciels libres. Le principal avantage économique des logiciels libres est son faible d'acquisition et de renouvellement pour les particuliers. Cet avantage individuel génère plusieurs externalités positives.

Tout d'abord, certains logiciels statistiques et programmes informatiques, tel que Stata et SPSS, coûtent plusieurs centaines, voir des milliers de dollar. De plus, la license doit être renouvelée annuellement, ce qui limite l'accès à ces logiciels. Comparativement, pour les logiciels libres, la license d'acquisition coûte bien souvent moins cher, et aucun renouvellement de licence n'est demandé dans la plus part des cas. Étant donné que les chercheurs doivent souvent faire face à des contraintes budgétaires, les logiciels libres deviennent des outils intéressants afin de minimiser les coûts de la recherche (**yu2022?**). Avantage encore plus important pour les chercheurs dans les pays du Sud global (**santillán-anguiano2023?**). L'accessibilité de ces ressources permet donc de réduire l'écart dans la production scientifique entre les pays du Sud et ceux du Nord. De plus, elle permet à tous de bénéficier d'outils pédagogiques accessibles,

3.1 2.1 Avantages

ce qui favorise l'acquisition ainsi que le développement de compétences méthodologiques.

Dans le cadre d'une formation universitaire, il peut être pertinent d'enseigner aux étudiants à se servir de logiciel statistique ou d'analyse de texte. L'acquisition de ces compétences peut être précieuse tant pour ceux et celles qui souhaitent se diriger vers le milieu académique, que pour ceux et celles qui visent le marché professionnel. D'ailleurs sur le site web de la banque d'emplois du gouvernement du Canada¹, les conditions d'emploi sont en ce moment² très bonnes, et une pénurie de main d'oeuvre est anticipé dans ce secteur entre 2022-2031. Ces compétences sont d'autant plus précieuses aujourd'hui, dans le monde de données dans lequel nous vivons.

Ensuite, le logiciel libre est adaptable et modifiable. Ces coût techniques de développement restent néanmoins nettement inférieurs aux coûts de renouvellement et de mise à jour des logiciels propriétaires dans bien des cas. L'argent sauvée des licences peut alors être investie dans le développement du logiciel libre (**béraud2007?**).

Cependant, une transition vers les logiciels libres ne doit pas se faire seulement sur des bases économiques, mais dans une perspective globale de changement de cultures. Changer pour des raisons purements économiques viendrait à violer l'essence même de la philosophie du logiciel libre, qui se veut davantage être un esprit de collaboration et de transparence. Par conséquent, il est important d'incorporer aussi les valeurs et la philosophie dans notre utilisation

Pour résumer, les logiciels libres permettent donc une plus grande égalité dans l'accès aux nouvelles technologies, puisqu'ils ont dans la majorité des cas, des coûts d'acquisition nettement moindre. (Oui et non, l'acquisition financière est une chose, mais il y a d'autres barrières à l'utilisation tel

¹Ces informations proviennent du site web suivant:
<https://www.jobbank.gc.ca/marketreport/outlook-occupation/17882/ca>

²En date d'écire ces lignes, septembre 2023.

3 2. Principaux avantages et inconvénients

que l'apprentissage à faire pour apprendre un langage de programmation, l'achat de matériel informatique, etc.) Cependant, considérant cela, donner l'exemple de l'étude qui montre que c'est beaucoup plus économique, même si l'on doit compter les coûts de formation, le soutien technique, l'entretien et la maintenance. (couture2014?; karjalainen2010?).

3.2 2.2. Inconvénients et défis:

3.2.1 2.2.1. Coûteux en temps

Dans leur texte, (paura2012?) soulèvent une critique faite envers certains logiciels libres, notamment envers R. Le problème principal d'enseigner les statistiques avec des logiciels libres est qu'ils sont compliqués à apprendre ainsi qu'à utiliser; par conséquent, les étudiants passeraient plus de temps à tenter de résoudre les erreurs de programmation plutôt que d'apprendre les statistiques. Il est vrai que ces logiciels demandent un investissement en temps, afin d'être en mesure de mener ses propres analyses statistiques. Par exemple, R demande l'apprentissage d'un langage de programmation afin de pouvoir utiliser le logiciel à son plein potentiel.

La syntaxe de certaines libraries demandent aussi un certain temps d'adaptation. Par exemple, je souhaite recoder la variable femme, de l'ensemble de données `titanic`, afin de remplacer les valeurs numériques actuelles (0, 1) par des valeurs nominales (homme, femme). La section de code ci-dessous réalise cette tâche avec les commandes de base de R et celle du `tidyverse`.

```
dat$femme[dat$femme == 0] <- "Homme" ①  
dat$femme[dat$femme == 1] <- "Femme"  
table(dat$femme) ②
```

- ① Utilisation de la commande de base dans R pour recoder la variable femme.

3.2 2.2. Inconvénients et défis:

- ② Vérification que la manipulation a bien fonctionné.

```
Femme Homme  
462      851
```

```
library(tidyverse) ③  
dat$femme <- recode(dat$femme, `0` = "Homme", `1` = "Femme") ④  
table(dat$femme) ⑤
```

- ③ Téléchargement de la librairie `tidyverse`.
④ Recodage de la variable `femme` à l'aide de la commande `recode`.
⑤ Vérification que la manipulation a bien fonctionné.

```
Femme Homme  
462      851
```

Toutefois, l'orsque l'on compare le coût d'apprentissage avec les bénéfices tirés, il est plus difficile de soutenir qu'il s'agit d'un désavantage. L'habileté que nous développons devient très utile par la suite, puisqu'elle nous permet de manipuler ainsi que d'analyser des données. Surtout, ces compétences s'inscrivent dans la longue durée, alors que l'apprentissage est plutôt de courte à moyenne durée. Surtout, la logique derrière la syntaxe de base de R et celle d'une nouvelle librairie reste sensiblement inchangée. Par conséquent, lorsque nous avons une bonne compréhension du fonctionnement de base de R, l'apprentissage d'une nouvelle librairie se fait relativement rapidement. Certaines, comme `dplyr` du `tidyverse` facilite grandement la manipulation des données comparativement aux commandes de base.

Pour résumer, bien que l'apprentissage d'un langage de programmation demande un investissement en temps, les bénéfices générées par ces nouvelles compétences dépassent le coût initial.

3.2.2 2.2.2. Problème de transparence

L'arrivée des sciences informatiques a fait émerger des problèmes de reproductibilité des protocoles scientifiques (**janssen2017?**). Le problème principal est relatif à l'accès au code utilisé par les chercheurs. Par exemple, il est possible de réaliser des analyses statistiques avec R sans partager le code utilisé, ce qui limite la transparence du processus scientifique. Dans cette situation, il est difficile de savoir si des erreurs de codage ont été commises, volontairement ou involontairement, affectant ainsi les résultats partagés.

Afin de remédier à ce problème, certains logiciels tel que GitHub³ participent à la transparence des résultats scientifiques (**fortunato2021?**). Ce logiciel permet aux chercheurs de partager leur code afin qu'il puisse être accessible pour tous. Il est important de mentionner ici que l'installation et la configuration de GitHub peut s'avérer difficile pour ceux et celles qui ne sont pas initiés à l'informatique. Cela constitue une certaine barrière dans l'utilisation de ce logiciel. Toutefois, nous souhaitons tout de même présenter l'utilité de ce logiciel puisqu'il permet de rendre les processus ainsi que les résultats de recherche plus transparents.

Par exemple, si l'on réalise une analyse statistique de la relation entre l'économie et le vote, nous pourrions partager l'ensemble du code que nous avons utilisé sur GitHub. D'une part cela permettrait aux utilisateurs de vérifier si les résultats sont honnêtes, et d'autre part de réutiliser le code pour mener leurs propres analyses.

Cependant, le partage du code utilisé reste encore majoritairement volontaire. (**janssen2020?**) soutiennent que plus d'effort et d'actions concertées doivent être mise en place afin d'améliorer l'accessibilité aux codes. Toujours selon ces auteurs, les journaux scientifiques pourraient exiger que les auteurs rendent leur code public lors du processus de publication.

³une plateforme publique *code ouvert* sur laquelle nous pouvons héberger et partager notre code.

3.3 3. Les sciences sociales à l'ère du numérique: les enseignements de la philosophie du logiciel libre

D'ailleurs, les résultats d'une expérience sur les facteurs qui influencent les chercheurs à partager leur code démontre que les initiatives individuelles ne seront pas suffisantes pour une agmentation du partage du code (**krähmer2023?**). Par conséquent, rendre le code accessible devrait devenir un standard institutionnalisé.

3.2.3 2.2.3. Appropriation capitaliste

Dans ce cas-ci, il s'agit plutôt d'un défis auquel le logiciel libre est confronté plutôt qu'une critique quant aux limites de son utilisation. En fait, l'accès au code source ainsi que la liberté et la possibilité de contribuer au développement du logiciel constitue un avantage intéressant pour les compagnies privées. Par conséquent, nous avons assisté à une intégration partielle du logiciel libre dans la logique capitaliste (**broca2013?**; **bessen2002?**). Certaines d'entre elles utilisent les utilisateurs comme une main d'oeuvre gratuite afin de bonifier leur logiciel, ce qui permet, dans certains cas, de générer des revenus commerciaux dont l'entreprise est la seule bénéficiaire (**couture2020?**). Attention, il ne faut pas penser que toutes les compagnies agissent de manière prédatrice. Le but ici est de souligner que certaines pratiques commerciales trouble l'essence du mouvement du logiciel libre, qui se veut davantage être un outil de collaboration accessible, plutôt qu'un moyen pour générer des profits. Il est important de garder en tête les valeurs et la philosophie qui a donné lieu à ce mouvement.

3.3 3. Les sciences sociales à l'ère du numérique: les enseignements de la philosophie du logiciel libre

Ce chapitre à voulu mettre de l'avant le logiciel libre afin d'initier les lecteurs et lectrices à ce monde. Le but n'était pas de présenter de manière exhaustive tout ce champ. Plutôt, nous avons préféré nous limiter aux

3 2. Principaux avantages et inconvénients

bases de compréhension, ainsi qu'à quelques exemples. Par conséquent, nous souhaitons qu'à la lecture du chapitre, les lecteurs et lectrices soient mieux outillés pour comprendre et réfléchir par rapport à ce monde, et ainsi insérer ces réflexions dans leurs démarches scientifiques. Générer des idées et des débats nous paraît bien plus promoteur pour l'avenir que d'apprendre par coeur.

En guise de conclusion, nous souhaitons résumer ce chapitre tout en situant ces différents éléments dans les sciences sociales à l'ère du numérique. Le livre de (marres2017?) est très intéressant à ce sujet. Face au constat que la vie sociale se trouve affectée par les changements numériques, il nous faut en tant que chercheur du monde social réfléchir à notre façon de comprendre les changements qui sont entrain de s'opérer. Bien que ces réflexions ratissent large ⁴, nous nous concentrons ici sur la dimension méthodologique.

Comme nous l'avons présenté ci-haut, les bas coûts associés à l'utilisation ainsi que la facilité du partage avec la communauté nous semble être deux avantages importants pour l'avenir des sciences sociales numériques. Notamment parce qu'ils ont le potentiel d'améliorer la transparence des protocoles scientifiques. Dans *Designing Social Inquiry*, l'un des livres les plus influents en science politique depuis les trente dernières années, les auteurs définissent quatre caractéristiques que chaque recherche doit posséder afin d'être considérée comme scientifique (king2021?). L'une d'elles, est que la *procédure doit être publique*: "La recherche scientifique utilise des méthodes explicites, codifiées et publiques afin de générer et analyser des données sur lesquelles la fiabilité peut ensuite être déterminer" (king2021?). Chaque individu qui souhaite contribuer à la connaissance et à la compréhension globale que nous avons de la réalité sociale doit garder en tête cette caractéristique fondamentale. Comme nous l'avons exposé, le partage du code devient un impératif pour assurer la transparence, la répliquabilité ainsi que la qualité des recherches.

⁴Allant de nos postulats ontologiques, épistémologiques et méthodologiques.

4 Bibliographie

5 Les outils de collecte de données

La révolution numérique engendrée par l'émergence du Big Data représente un important défi pour le monde des sciences sociales (Manovich, 2011; Burrows et Savage, 2014). Elle constitue également une opportunité de recherche enrichissante et innovante permettant une compréhension plus accrue des phénomènes sociaux étudiés par la communauté scientifique (Connelly et al., 2016). Cette meilleure compréhension est permise, entre autres, par l'accès à des données massives concernant les trois acteurs clés de la société démocratique: les citoyens, les médias et les décideurs (Schroeder, 2014; Kramer, 2014). Si l'accès à ces données représente un défi éthique et théorique, tel qu'explicité lors des chapitres précédents, elle représente également un défi technique pour les chercheurs.euses voulant exploiter le potentiel et les opportunités offertes par les données massives (Burrows et Savage, 2014). Le chapitre qui suit vise à offrir un portrait de certains outils de collecte de données pouvant être exploitées par les chercheurs.euses en sciences sociales visant à tirer profit de la révolution numérique. À travers ce chapitre, il sera question d'outils permettant de collecter des données de sondages, des données médiatiques, de même qu'une panoplie de données par le biais d'extracteurs. Ce chapitre offre donc un tour d'horizon de certains outils de collecte de données à la disposition des chercheurs.euses qui souhaitent entamer des recherches en sciences sociales numériques.

5.1 Le Big Data et les différents acteurs de la société :

Le champ d'étude de la science politique repose sur l'étude de trois types d'acteurs distincts ayant un impact sur la condition socio-économique et politique d'une société : les décideurs, les médias et les citoyens. La recherche sur les décideurs comprend entre autres l'analyse des politiques publiques, des partis politiques, de stratégies électorales ou encore l'analyse de discours de politiciens ou d'organisations. L'étude des médias repose largement sur le rôle des médias dans la formation des priorités et des jugements des citoyens quant aux enjeux politiques, de même que sur leur capacité d'influencer l'agenda des politiciens. En ce qui concerne les citoyens, le champ d'étude de l'opinion publique se consacre à l'analyse des comportements et des attitudes politiques des individus. De plus, de nombreuses recherches visant à comprendre le rôle des citoyens dans une société démocratique portent sur l'influence de la société civile de même que sur l'effet des mouvements sociaux.

Chacun de ces champs de recherches se voit confronté à une panoplie de défis théoriques et techniques en lien avec l'émergence des données massives. La révolution technologique permet une étude plus approfondie des phénomènes auxquels sont confrontés les différents acteurs de la société démocratique, en raison de l'importante quantité de données accessible aux chercheurs.euses. Toutefois, la collecte de données permettant de mener à terme de telles études peut s'avérer complexe. Pour chacun des trois acteurs démocratiques énumérés précédemment, les sections suivantes énumèrent et expliquent les capacités techniques d'outils permettant aux chercheurs.euses d'accéder à des données massives. Bien que d'autres outils existent et offrent des résultats satisfaisants, les méthodes suivantes sont particulièrement pertinentes dans une optique d'étude des sciences sociales numériques en raison de leur capacités techniques de même que par la relative simplicité de leur utilisation.

5.2 Plateformes de sondages et collecte de données

Malgré certaines différences méthodologiques, toute recherche doit analyser et interpréter des données fiables et de qualité afin d'émettre des résultats (Nayak & K. A., 2019). Notamment lorsqu'il est question d'étudier les citoyens et l'opinion publique, il est nécessaire d'accumuler suffisamment de données auprès d'un échantillon assez grand afin d'inférer des conclusions sur la population.

Une des méthodes couramment utilisées est le sondage, également nommé panel, enquête, questionnaire, etc. Ils peuvent être manuels ou électroniques, et dans le second cas, peuvent être administrés par ordinateur, par courriel ou via le web (Nayak & K. A., 2019). La différence majeure entre les méthodes manuelles et les méthodes numériques réside dans le fait que les premières impliquent un contact direct entre le chercheur et le répondant, tandis que dans le cas des secondes le contact est indirect (Evans & Mathur, 2018). L'arrivée des données massives et des outils numériques offre une panoplie de nouvelles opportunités de collecte de données pour la communauté scientifique. Lorsqu'exécutée manuellement, la collecte de données et la réalisation de sondages peuvent devenir des tâches lourdement fastidieuses, et de facto, demander énormément de ressources pour mener une recherche à grande échelle. C'est pourquoi les technologies du numérique peuvent faciliter cet aspect de la recherche en fournissant des plateformes de sondages et de collecte de données. De plus, les sondages représentaient en 2016 environ 20% du chiffre d'affaires de l'industrie globale du marketing (Evans & Mathur, 2018). Ces chiffres montrent la pertinence de l'acquisition de compétences nécessaires à la formation de sondages, tant dans le monde académique que professionnel. Le numérique permet donc de créer un questionnaire, de cibler une population et de la contacter, d'entreposer les données des répondants pour ainsi les visualiser, le tout à un coût réduit et plus rapidement que s'il avait été conduit manuellement (Nayak & K. A., 2019). Ainsi, les sondages en ligne ont une portée internationale, permettent le suivi de la ligne du temps,

5 Les outils de collecte de données

offrent des options qui contraignent le répondant à répondre à certaines questions et permettent d'utiliser des arbres de logique avancés que les sondages manuels ne permettent pas.

5.2.1 Les principales plateformes web.

Il existe un large éventail de plateformes de sondages et de collecte de données qui peuvent être utiles dans un contexte académique. Cet ouvrage se limite à cinq d'entre elles : Qualtrics, REDCap, SurveyMonkey, Google Forms et Typeform. Cependant, il n'est pas déconseillé de se renseigner sur les autres plateformes disponibles en fonction de ses besoins et de ses ressources. Voici une liste non-exhaustive d'autres plateformes de collecte de données en ligne : LimeSurvey, Zoho Survey, Qualaroo, Formstack, Wufoo, Checkbox Survey, SmartSurvey, QuickTapSurvey, SoGoSurvey, Snap Surveys, AskNicely, Opinio, Alchemer, Cognito Forms, Feedbackify.

5.2.1.1 Qualtrics (<https://www.qualtrics.com/>)

Cette plateforme est une des plus reconnues et utilisées à l'international, tant dans le milieu académique que dans le secteur privé. En plus d'offrir des outils de collecte de données et de sondages, Qualtrics est utilisé dans le marketing et dans la gestion de l'expérience client. Il est donc pertinent de se familiariser avec cet outil, car il offre des compétences pratiques pour la recherche, mais également pour obtenir des opportunités de carrière. Qualtrics offre plusieurs services pratiques pour la collecte de données, avec des options flexibles pour la programmation et l'administration des sondages. Par exemple, Qualtrics s'adapte à différents formats en fonction de l'appareil du répondant (Evans & Mathur, 2018).

5.2 Plateformes de sondages et collecte de données

5.2.1.2 REDCap (<https://www.project-redcap.org/>)

Research Electronic Data Capture (REDCap) permet de construire et de gérer des sondages ainsi que des bases de données. Pour accéder aux services de cette application, il est nécessaire d'être un partenaire du REDCap Consortium ou membre d'une organisation qui en fait partie. Seules les organisations à but non lucratif peuvent adhérer au Consortium. Les données et les sondages qui y sont produits peuvent être partagés et utilisés par différents chercheurs issus de diverses institutions. L'exportation vers différents types de fichiers (Excel, PDF, SPSS, SAS, Stata, R) est possible. Ce qui distingue REDCap des autres applications est sa compatibilité avec les dossiers médicaux, sa sécurité pour les données sensibles, ainsi que son approche académique à la collecte de données par sondage.

5.2.1.3 SurveyMonkey (<https://www.surveymonkey.com/>)

SurveyMonkey se distingue des autres applications en permettant de construire et gérer des sondages/formulaires à l'aide d'une interface conviviale sans toutefois perdre de ses fonctionnalités. En plus d'avoir recours aux nouvelles technologies de l'I.A. pour aider à construire des sondages adaptés à vos besoins, cette application propose plusieurs centaines de modèles personnalisables élaborés par des experts dans le domaine. SurveyMonkey permet également l'analyse des données et la création de rapports directement sur l'application, en plus de permettre l'exportation vers d'autres types de programmes. Les forfaits varient en gamme de tarifs, allant du gratuit avec des fonctionnalités restreintes, jusqu'aux options payantes destinées aux particuliers et aux entreprises.

5.2.1.4 Google Forms (<https://docs.google.com/>)

Cette application se distingue par sa simplicité et son accessibilité, en grande partie grâce à l'omniprésence de google tant dans le monde académique que dans la vie courante. Google Forms est inclus dans le forfait de base du Google Workspace, ce qui le rend largement compatible avec les autres applications de Google, en plus d'être disponible gratuitement. Bien que ses fonctionnalités soient moins avancées que celles de ses concurrents, Google Forms peut convenir pour des sondages plus simples et rapides grâce à son interface conviviale, à sa fonction d'analyse de données directement sur la plateforme, ainsi qu'à ses modèles préfabriqués.

5.2.1.5 TypeForm (<https://www.typeform.com/>)

Si votre objectif est de produire des formulaires avec une esthétique attrayante, moderne et interactive, TypeForm est la plateforme idéale. Elle permet de se concentrer sur l'expérience de l'utilisateur et de l'impliquer dans le sondage grâce à son aspect visuel. Cette plateforme dispose d'une option gratuite, ainsi que plusieurs forfaits payants. Typeform est également compatible avec plusieurs applications de gestion du flux de travail (Zapier, Google Sheets, Slack, etc).

5.2.2 Les limites des sondages en ligne

Néanmoins, les sondages en ligne comportent des défis, notamment en ce qui concerne l'échantillonnage, les taux de réponse et les caractéristiques des non-répondants. Il est également nécessaire de se méfier des enjeux

5.2 Plateformes de sondages et collecte de données

éthiques et de confidentialité (Nayak & K. A., 2019). Comme la généralisation essentielle pour conférer une valeur scientifique à ses résultats de recherche, les sondages en ligne ont leurs limites. En effet, il est crucial de connaître la population cible pour effectuer des inférences fiables, et l'échantillonnage doit reposer sur des caractéristiques précises. Même si des informations démographiques peuvent être collectées et des quotas utilisés, il n'est toutefois pas réellement possible de confirmer les informations sur le répondant (Andrade, 2020). Les sondages traditionnels où l'on retrouve un contact direct sont plus susceptibles de permettre de brosser un portrait plus complet du répondant (Evans & Mathur, 2018). Les répondants avec des biais peuvent également plus facilement répondre aux sondages en ligne et limiter la généralisation (Andrade, 2020). Les sondages en ligne sont également souvent perçus comme des pourriels, ont généralement de faibles taux de réponse, sont impersonnels et peuvent avoir des instructions peu claires. Ils ont également leurs lots d'enjeux de confidentialité (Evans & Mathur, 2018).

Conseils méthodologiques à la réalisation d'un sondage numérique
(Evans & Mathur, 2018)

L'article de Evans et Mathur (2018) est une revue de littérature observant l'évolution des sondages numériques depuis la parution de leur dernier article sur le sujet en 2005. À travers cet article, les auteurs offrent des conseils méthodologiques en fonction de leur analyse de contenu de la littérature scientifique. Les conseils d'Evans et Mathur (2018) sont résumés ci-dessous. Afin d'obtenir plus de détails, n'hésitez pas à approfondir votre lecture de l'article. De plus, bien qu'il s'agisse d'un article crédible et largement documenté, il est toujours pertinent de consulter des sources spécifiques à vos besoins.

1. Définir le but du sondage avant la méthodologie. Lorsque possible, inclure des hypothèses testables et des méthodes basées sur des fondations théoriques.
2. Choisir le type de sondage.

5 *Les outils de collecte de données*

3. Décider des méthodes d'échantillonnage, des quotas et des échéances.
4. Déterminer le responsable de la construction du sondage.
5. Soyez transparent en divulguant le but du sondage, la façon dont les données seront utilisées ainsi que l'auteur du sondage.
6. Les questions et les catégories de réponses doivent être élaborées de manière objective et dans une perspective de convivialité.
7. Les sondages doivent être assez légers pour favoriser un taux de réponse positif, mais assez complet pour avoir l'information nécessaire.
8. Ils doivent également être attrayant afin de favoriser leur complétion par le répondant.
9. S'assurer de l'anonymat du répondant.
10. Il faut régulièrement procéder à des tests afin de corriger les faiblesses du questionnaire.
11. Déterminer qui administre le sondage, qui collecte l'information, et qui analyse les données.
12. Établir un échancier pour les différentes étapes de l'étude.
13. Suite à la collecte de données, entreposer les données brutes dans un fichier électronique.
14. Utiliser les méthodes appropriées (qualitatif ou quantitatif), et analyser les données selon les buts de l'étude.
15. Dans le cas d'une recherche académique, il est important d'avoir une section dédiée aux limites de l'étude.
16. Conserver l'anonymat des répondants lors de l'analyse et de la publication.

5.3 Factiva : outils de récolte de données médiatiques

17. Agir sur les résultats. Rien ne sert de conduire un sondage qui ne contribue pas à la croissance du savoir ou n'apporte pas de changement stratégique ou organisationnel.
18. Toujours se plier à un code d'éthique rigide.

Il s'agit donc ici d'un court résumé des plateformes de sondages et de la collecte de données en ligne, tentant de couvrir l'essentiel de cet outil afin de vous aider lors de votre parcours académique, ou simplement comme aide-mémoire pour la réalisation d'un sondage. Les outils énumérés précédemment permettent une étude approfondie de phénomènes concernant les citoyens. Bien sûr, il n'est pas possible de couvrir l'entièreté de cet outil très complexe et ayant évolué dans le temps, cette section ne sert donc que de point de départ si vous vous intéressez à l'élaboration d'un sondage numérique, ou si vous avez besoin de récolter des données. Il vous est donc recommandé de vous renseigner davantage avec d'autres ressources afin de compléter ce qui est indiqué dans cet ouvrage.

5.3 Factiva : outils de récolte de données médiatiques

L'émergence de nouvelles technologies de même que la fragmentation médiatique, causée notamment par l'apparition de chaînes de nouvelles en continu, ébranlent considérablement les écosystèmes médiatiques occidentaux (Chadwick, 2017). Un récent courant de recherche se penche sur le rôle des médias relativement aux comportements des individus dans une perspective de fragmentation médiatique. Ces changements de dynamique médiatiques permettent aux individus de choisir leurs sources d'information. Cette fragmentation aurait conséquemment pour effet de contribuer à la formation de chambres d'écho. Ainsi, les études sur les effets des médias visent à comparer les agendas de différentes organisations médiatiques de même que de comprendre le cadrage de la nouvelle qu'ils offrent aux citoyens. Pour effectuer de telles études comparées, l'accès

5 Les outils de collecte de données

à des données médiatiques est essentiel. L'arrivée de données massives permet de nouvelles avenues de recherche pour les chercheurs.euses en sciences sociales en raison de l'importante quantité de données accessibles aux chercheurs.euses, ce qui permet une compréhension accrue des réalités médiatiques modernes.

L'outil Factiva offre un accès à l'ensemble des articles d'une panoplie de médias provenant d'une vaste sélection de pays. Le moteur de recherche est opéré par Dow Jones et offre également l'accès à des documents d'entreprises. En revanche, l'accès qu'il offre aux contenus médiatiques est particulièrement pertinent pour la communauté scientifique en communication et en sciences sociales. Il offre l'accès à plus de 15 000 sources médiatiques provenant de 120 pays. Il permet de télécharger une quantité illimitée de documents RTF, un format de fichier de texte, pouvant contenir jusqu'à 100 articles chacun. En outre, ils peuvent être sélectionnés automatiquement en cochant le bouton proposant de sélectionner les 100 articles de la page de résultat. Chaque page de résultat contient 100 articles à la fois. Enfin, Factiva permet également de filtrer les doublons.

En outre, cet outil permet également de lancer une requête de recherche par mots-clés et par date qui permet, par exemple, de récolter les articles médiatiques concernant un sujet précis dans une ligne de temps déterminée. De manière plus précise, Factiva permet de filtrer la recherche d'articles par source, par date, par auteur, par sociétés, par sujet, par secteur économique, par région et par langue. Disons qu'un.e chercheur.euse désire comparer la couverture médiatique d'une élection donnée. Il peut, par le biais de Factiva, sélectionner tous les articles contenant le mot « élection » dans une sélection de médias, et ce, durant la période de l'élection. Les mots clés sélectionnés peuvent être adaptés aux désirs de la personne chercheuse de manière à inclure des mots qui peuvent être mis ensemble ou à un maximum d'intervalle de mot. L'utilisation des signes « and » et « or », aussi connus sous le nom d'opérateurs booléens, permettent d'ajouter un mot dans la requête de recherche. En ajoutant `near5`, l'on peut spécifier qu'il doit y avoir un maximum de 5 mots entre les deux mots

5.3 Factiva : outils de récolte de données médiatiques

recherchés. L'on peut également mettre certains signes à la fin de mots, ce qui permet de préciser le champ de recherche. Par exemple, dans une étude récoltant des articles sur les immigrants, le mot immigrant pourrait être écrit de la manière suivante : `immigra*`. Ainsi, tous les mots débutant par ce suffixe seraient inclus de la recherche d'article, ce qui comprend donc : immigrant, immigration, immigrants, immigrante, etc. La Figure 1 est une capture d'écran de l'interface de recherche de Factiva. Ainsi, en ajoutant un opérateur booléen, l'on peut préciser un champ de recherche. La personne chercheuse pourrait, par exemple, rechercher des articles sur les immigrants syriens, et rajoutant les opérateurs "and" ou encore "or", de même que le mot « `syri*` », l'étoile étant rajoutée pour inclure le plus de mots possible.

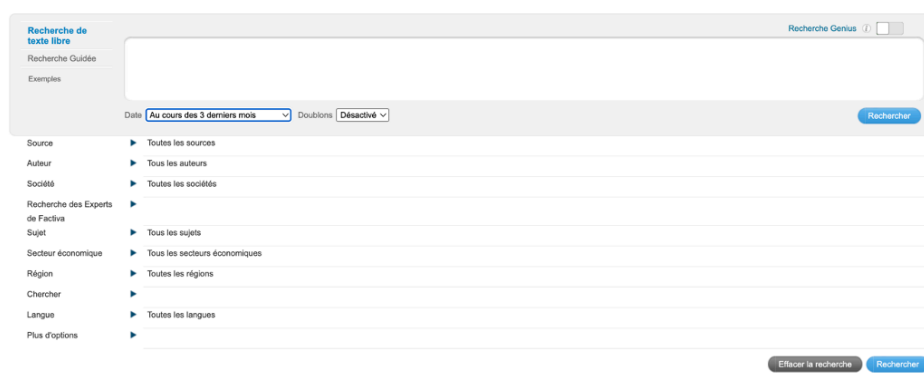


Figure 5.1: image3__1

Ainsi, Factiva permet d'avoir accès facilement à des données utiles pour de l'analyse textuelle d'articles médiatiques. Comme les textes deviennent accessibles aux chercheurs.euses, ils permettent de faire facilement une analyse de contenu par thèmes ou par ton.

Cependant, ce ne sont pas tous les médias qui sont accessibles sur Factiva. Dans l'optique où un média recherché n'est pas trouvable sur Factiva, le logiciel Eureka représente une bonne alternative. Eureka se concentre prin-

5 Les outils de collecte de données

cipalement sur les médias francophones (autant au Québec qu'en Europe). La structure d'Eureka est similaire à celle de Factiva. En effet, Eureka permet de filtrer des articles médiatiques par requête de recherche adaptée à la source, la date ou encore l'auteur. Toutefois, les requêtes de recherche doivent être formulées d'une manière quelque peu différente. Elles doivent donc être adaptées au fonctionnement d'Eureka. Les articles doivent être sélectionnés à la main, et peuvent être téléchargés dans un document PDF pouvant contenir un maximum de 50 articles à la fois. La Figure 2 contient l'interface de recherche d'Eureka.

The screenshot displays the Eureka search interface. At the top, there is a navigation bar with the Eureka logo and tabs for 'RECHERCHER', 'DOSSIERS', and 'PUBLICATIONS PDF'. The main heading is 'Recherche avancée'. Below this, there is a section for 'Mots clés dans tout le texte' with a large text input field. To the right of this field are three rows of search operators: 'ET', 'OU', and 'SANS', each with a corresponding input field and a dropdown menu for search location (e.g., 'dans le titre', 'dans l'introduction', 'dans le nom de l'auteur'). Below the operators is a button 'Ajouter une zone de mots clés'. The 'Sources' section allows selecting sources by 'groupe de sources', 'critères de sources', or 'nom de source'. There is a 'Domaine de recherche' dropdown menu set to 'Tout le contenu'. The 'Période' section has a dropdown menu set to 'Depuis 30 jours'. At the bottom right, there are buttons for 'Recommencer' and 'Recherche'. On the right side of the interface, there is a sidebar titled 'Astuces de recherche' containing several search tips with examples like 'pomme verte', 'blanc & noir', 'rouge | vert', 'pomme & (verte | rouge)', 'bières ! "bières blondes"', 'voiture \$2 sport', and 'automobile %2 salon'.

Figure 5.2: image3_2

Il existe aussi une panoplie d'outils permettant un accès à des données médiatiques. Quoique Factiva soit intuitive et que de nombreuses universités possèdent des licences permettant d'exploiter la plateforme, plusieurs alternatives existent pour les personnes chercheuses. NexisUni, qui comprend entre autres l'outil LexisNexis Academic particulièrement prisé par le champ d'études de communication aux États-Unis, représente une excellente alternative. C'est également le cas de NewsBank qui permet lui aussi

5.4 Les extracteurs : avoir accès à des données massives via du code.

un accès à un vaste répertoire d'articles médiatiques. Les chercheurs.euses peuvent choisir la plateforme qui leur convient le mieux, en prenant en compte notamment l'accès qui peut leur être fourni par l'institution universitaire les employant.

En somme, la révolution numérique permet un accès sans précédent aux données médiatiques, ce qui permet des analyses approfondies du rôle des médias traditionnels dans une société démocratique.

5.4 Les extracteurs : avoir accès à des données massives via du code.

Chacun des acteurs démocratiques énumérés précédemment peut également être étudié par le biais d'extracteurs qui offrent un accès à des données numériques massives. Les extracteurs de données numériques sont des infrastructures de code permettant d'extraire des données brutes d'une source définie. La section suivante explique comment les extracteurs peuvent être utiles dans un contexte de recherche en sciences sociales numériques.

Les données en lien avec les décideurs sont souvent accessibles sur des sites gouvernementaux. Toutefois, certaines identifications peuvent être nécessaires et l'accès peut être compliqué, particulièrement dans une perspective de données massives. C'est dans cette optique que les extracteurs de données numériques peuvent être utiles. Un code peut extraire de manière automatisée les débats des parlements, les communiqués de presse des gouvernants, les plateformes électorales des partis politiques, ce qui offre un accès inégalé aux chercheurs.euses aux données de décideurs. Dans une autre optique, des extracteurs peuvent également offrir l'accès aux données provenant de médias socionumériques comme Twitter (maintenant X) ou Facebook . Un extracteur peut, par exemple, être en mesure de répertorier l'ensemble des Tweets de journalistes, de politiciens ou encore de citoyens de manière automatisée, offrant un accès inégalé aux chercheurs.euses à

des données massives exclusives. L'élaboration d'extracteurs est toutefois facilitée par l'existence d'API (Application programming interface) sur les plateformes exploitées. L'API d'un site ou d'une application permet à un tiers parti d'avoir accès à du code expliquant le fonctionnement de la plateforme étudiée, ce qui en facilite l'extraction de données. Par exemple, Twitter possédait avant les changements de directions récents un API qui facilite l'élaboration d'un extracteur. En contrepartie, Facebook ne possède pas d'API, ce qui rend l'accès à ses données beaucoup plus complexe. Un extracteur peut également offrir l'accès à des données médiatiques, en codant un accès à des fils RSS ou encore aux HTML des médias extraits.

L'élaboration d'un extracteur est toutefois une tâche complexe qui requiert un certain nombre de connaissances en lien avec les langages de programmation. Les chapitres 4 et 5 du présent ouvrage offrent justement un survol du langage fonctionnel R, qui est utilisé par de nombreux développeurs lors de l'écriture d'extracteurs. R est également reconnu pour ces fonctionnalités statistiques qui sont, elles aussi, abordées ultérieurement dans ce livre.

5.5 Covidence : outil de récolte d'articles scientifiques

Comme mentionné précédemment, les outils numériques de données massives facilitent le travail des personnes chercheuses lors de la récolte de données dans le cadre d'analyses empiriques. Cependant, la révolution technologique offre également des outils pouvant être utiles lors d'autres étapes du cycle de la recherche. Il s'agit notamment du cas de la revue de littérature, alors que de nombreux outils offrent aux personnes chercheuses des ressources permettant d'élaborer un cadre théorique exhaustif par le biais de données massives sur la littérature scientifique. L'outil Covidence, géré par une compagnie sans but lucratif, en est un exemple particulière-

5.5 *Covidence : outil de récolte d'articles scientifiques*

ment prisé du monde académique lors de l'entreprise de revues de littérature.

La plateforme en ligne Covidence est utilisée pour faciliter les revues systématiques de littérature. Cette dernière permet de réduire drastiquement le temps d'accomplissement du travail en plus de le rendre plus simple et plus intuitif. L'outil a été développé pour mieux gérer et organiser l'évaluation de quantité importante d'études scientifiques. L'exécution d'une revue de littérature sur Covidence se fait par le biais d'un double codage. C'est-à-dire que l'évaluation des études se fait manuellement par deux codeurs travaillant de manière autonome et qui mettront en commun leurs résultats à la fin de l'exercice. L'outil est reconnu pour ses trois étapes précises : « Title and abstract screening », « Full text review » et « Extraction ». Covidence permet d'importer des données massives provenant de base de données bibliographiques. En effet, l'outil lance des requêtes auprès de multiples bibliothèques, ce qui offre l'accès à des milliers d'études sur le champ étudié par les personnes chercheuses. Ces requêtes sont adaptées aux besoins spécifiques de la personne chercheuse voulant explorer en profondeur un domaine de la littérature scientifique.

La première étape, soit le « Title and abstract screening », consiste en la révision des titres et des résumés des articles récoltés. Pour rendre le travail davantage efficace, il est nécessaire d'inclure des critères précis pour analyser les titres et résumés d'articles. En se servant du jugement et des critères qui étaient recherchés, les individus doivent éliminer ou accepter selon la pertinence de l'article quant à la littérature étudiée. Cette partie est souvent longue, puisque la littérature existante est souvent massive. Il est donc important pour les personnes chercheuses de se rencontrer à maintes reprises pour discuter des conflits de jugement et pour trouver des compromis. En outre, cette étape, plutôt longue, s'avère très utile et motivante, puisqu'il est possible de développer un jugement critique davantage raffiné et de s'instruire dans une littérature continuellement plus précise.

Une fois avoir complété la revue des titres et des résumés, il faut entamer

5 Les outils de collecte de données

le « Full text review » qui, comme l'indique le nom, consiste à la révision complète des textes sélectionnés. Cette étape demande d'analyser chaque texte, puis de voter « oui », « non » ou « peut-être » quant à la conservation du texte dans la revue de littérature. Le vote permet donc soit d'exclure l'article, de le retenir ou de l'envoyer à la prochaine étape. D'un autre côté, les conflits rendent le travail beaucoup plus long, puisque les codeurs.euses ont un texte entier à argumenter. Ainsi, cette partie du travail, bien qu'elle comporte beaucoup moins de documents, est assez longue et exigeante.

La dernière étape, soit celle de l'extraction, consiste à recueillir toute donnée étant utile à l'étude de la littérature désignée. Cette étape est demandante, car les chercheur.euse.s doivent se conformer à une grille de codification prédéfinie. Le but est qu'un consensus entre les codeurs émerge de ce processus. L'extraction permet de faire ressortir les théories, les méthodologies et les conclusions présentent dans les études retenues.

Une fois les étapes de la revue systématique terminées, Covidence facilite l'exportation des résultats de l'extraction sous forme de tableaux, de graphiques et de rapports pour la méta-analyse ou pour la rédaction d'articles scientifiques. De nombreuses universités offrent un accès à Covidence par le biais de licences, et l'outil est particulièrement utile et bien construit. Toutefois, il existe d'autres alternatives à Covidence. Le choix de l'outil dépend des coûts de même que des besoins spécifiques des personnes chercheuses. Les plateformes DistillerSR, Archie et Rayyan sont notamment largement utilisées par les personnes chercheuses.

5.6 Conclusion et discussion:

Le précédent chapitre portait sur les différents outils de collecte de données massives mis à la disposition des chercheur.euse.s s'intéressant au champ des sciences sociales numériques. Les outils relevés se démarquent par leur capacité d'accorder l'accès à des données permettant d'étudier les trois principaux acteurs de la société démocratique, soit: les citoyens,

5.6 Conclusion et discussion:

les décideurs et les médias. Comme mentionné à plusieurs reprises lors du chapitre, le but de ce dernier n'est pas d'offrir une liste complète des outils disponibles. Toutefois, les outils énumérés ont été sélectionnés en raison de leur intuitivité, leur relative simplicité d'accès de même que leurs capacités techniques considérées par les auteurs comme étant particulièrement pertinentes dans une optique de recherche en sciences sociales numérique. Ainsi, ce chapitre démontre que la possibilité d'effectuer des recherches en sciences sociales numériques par le biais de données massives est plus que jamais accessible à la communauté scientifique, particulièrement en ce qui a trait à la collecte de données permettant de tels travaux. Une fois les données collectées, le travail d'analyse représente un défi technique supplémentaire se dressant devant les personnes chercheuses. Les chapitres suivants visent à familiariser les chercheurs.euses à des outils méthodologiques permettant l'analyse et la visualisation de données massives au sein des sciences sociales.

Bibliographie:

Schroeder, R. (2014). Big data and the brave new world of social media research. *Big Data & Society*, 1(2), 2053951714563194.

Chadwick, A. (2017). The hybrid media system: Politics and power. Oxford University Press.

Connelly, R., Playford, C. J., Gayle, V., & Dibben, C. (2016). The role of administrative data in the big data revolution in social science research. *Social science research*, 59, 1-12.

Manovich, L. (2011). Trending: The promises and the challenges of big social data. *Debates in the digital humanities*, 2(1), 460-475.

Burrows, R., & Savage, M. (2014). After the crisis? Big Data and the methodological challenges of empirical sociology. *Big data & society*, 1(1), 2053951714540280.

Kramer, A. D., Guillory, J. E., & Hancock, J. T. (2014). Experimental evidence of massive-scale emotional contagion through social networks. *Pro-*

5 Les outils de collecte de données

ceedings of the National academy of Sciences of the United States of America, 111(24), 8788.

Andrade, C. (2020). The Limitations of Online Surveys. *Indian Journal of Psychological Medicine*, 42(6), 575-576. <https://doi.org/10.1177/0253717620957496>

Evans, J. R., & Mathur, A. (2018). The value of online surveys: A look back and a look ahead. *Internet Research*, 28(4), 854-887. <https://doi.org/10.1108/IntR-03-2018-0089>

Nayak, M., & K A, N. (2019). Strengths and Weakness of Online Surveys. 24, 31-38. <https://doi.org/10.9790/0837-2405053138>

6 R ou ne pas R?

Plusieurs notions liées à l'ère numérique, notamment à ce qui a trait aux opportunités et difficultés que cette dernière peut amener, ont été présentées par l'entremise du chapitre précédent. C'est un monde de possibilité qui s'offre à ceux qui maîtrisent les nouveaux outils des temps modernes. Mais comment en arriver là ? Le présent chapitre a pour but de présenter certains outils flexibles et péreins permettant la réalisation de nombreuses tâches. Une des premières étapes permettant de notamment réaliser la collecte, l'analyse et la visualisation graphique de données ainsi que la rédaction de documents est l'apprentissage d'un langage de programmation. Bien que plusieurs langages de programmation existent, le présent ouvrage priorise le langage **R**. Les sections suivantes présentent ce langage de programmation, ces forces et ces faiblesses ainsi que les raisons de son utilisation. Enfin, la dernière section présente un environnement de programmation qui se prête bien à son utilisation.

6.1 Pourquoi R?

Comme mentionné précédemment, il existe plusieurs langages de programmation. **R** a deux types de compétiteurs : 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). **R** prend ses racines dans le langage de programmation S, créé notamment par Ross Ihaka et Robert Gentleman. Ces derniers ont fait des choix non

6 *R* ou ne pas *R*?

orthodoxes lors de l'élaboration du langage, qui font aujourd'hui la popularité de ce logiciel auprès d'un large pan de la communauté académique. En effet, Morandat et al. (2012) rapporte que le langage a été élaboré afin qu'il soit intuitif et qu'il permette aux nouveaux utilisateurs de rapidement réaliser des analyses.

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 de façon bottom-up, ce qui lui procure une grande flexibilité; et 3) il permet aux utilisateurs de créer leurs propres fonctions. 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,

6.2 Où coder en R ?

créer des fonctions et faire des boucles en plus de pouvoir combiner **R** avec certains langages de balisages.

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.

6.2 Où coder en R ?

Un environnement de développement intégré (IDE) permet aux programmeurs de consolider les différents aspects de l'écriture d'un programme informatique. Ils permettent 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 du programmeur. Ils fournissent de nombreuses fonctionnalités – notamment la coloration syntaxique ainsi que le contrôle de version – pour créer, modifier et compiler du code. Certains environnements de développement intégré sont dédiés

6 *R* ou ne pas *R*?

à un langage de programmation spécifique. Par conséquent, ils contiennent des fonctionnalités qui sont plus compatibles avec les paradigmes de programmation du langage auquel ils sont associés. Enfin, il existe de nombreux environnements de développement intégré multilingues.

Comme mentionné précédemment, *R* est 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 – et 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 le *terminal* – pour demander à son ordinateur d'effectuer une opération précise, telle que rouler un fichier de code *R*.

La suite du chapitre présente RStudio, notamment à travers ses avantages et inconvénients, mais également des exemples de ses fonctionnalités ainsi que des conseils sur comment l'utiliser et le personnaliser.

6.3 Qu'est-ce que RStudio ?

RStudio est un projet open source destiné à combiner 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 son utilisateur d'aisément réaliser ses tâches. Parmi les plus utiles, on retrouve notamment une fenêtre d'aide, de la

6.3 Qu'est-ce que RStudio ?

documentation sur les différents packages R, un navigateur d'espace de travail, une visionneuse de données et une prise en charge de la coloration syntaxique (Horton, Kleinman, 2015). De plus, RStudio permet de coder dans plusieurs langages et de supporter une grande quantité de formats. Il fournit également un support pour plusieurs projets ainsi qu'une interface pour utiliser des systèmes de contrôle, tels que GitHub (Horton, Kleinman, 2015).

RStudio a plusieurs avantages. Son utilisation est facile à apprendre pour les débutants. Les principaux éléments d'un IDE sont intégrés dans une disposition 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 pour l'environnement de travail et un quatrième volet où il est notamment possible d'afficher des graphiques ou de la documentation sur différents packages. Ce volet permet d'ailleurs d'accéder au répertoire des *packages* disponibles pour R en plus de permettre à l'utilisateur de consulter l'arborescence de ses fichiers. De plus, on y retrouve la possibilité de créer plusieurs espaces de travail – appelés projets – qui facilitent l'organisation de différents *workflows*.

Il y a plusieurs autres aspects de RStudio que les programmeurs apprécient. Parmi ceux-ci se trouve le fait qu'il peut être utilisé via un navigateur Web pour un accès à distance (Verzani, 2011). De plus, RStudio supporte plusieurs langages de programmation ainsi que différents langages de balisage. Qui plus est, de nouvelles fonctionnalités sont régulièrement ajoutées pour satisfaire les besoins de la communauté scientifique. Enfin, R logiciel est également souvent mis à jour.

Parmi ce que certains considèrent comme étant 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 trouvent que le *set up* des différents panneaux n'est pas ergonomique, ou même qu'il n'est pas possible de pouvoir suffisamment personnaliser l'environnement de programmation. De plus, certains utilisateurs ont rapporté que RStudio

était plus lent que d'autres alternatives pour quelques opérations, surtout celles comprenant de longs codes.

6.4 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 cadran principal. C'est dans celui-ci que l'utilisateur passera la plus grande 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 trouve la console ainsi que le terminal. Dans cette première, on peut interagir avec R de la même manière que dans le cadran principal, mais le code ne sera pas enregistré. Le terminal, pour sa part, est le point d'accès de communication entre un usager et son ordinateur. Bien que les différents systèmes d'exploitation viennent avec un terminal déjà intégré, il est aussi possible d'y accéder à partir de RStudio.

On retrouve, dans le coin supérieur droit, l'espace de travail. Ce cadran contient trois éléments : *l'environnement global*, *l'historique* et *les connections*. *L'environnement global* est l'endroit où l'utilisateur peut voir les bases de données, les fonctions et les différents autres objets R qui sont 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 roulé ainsi que les dernières commandes écrites dans la console. L'onglet *connections*, pour sa part, permet de connecter son IDE à une variété de sources de données et d'explorer les objets et les données qui la composent. Il est conçu pour fonctionner avec une variété d'autres outils pour travailler avec des bases de données en R dans RStudio.

Le cadran dans le coin inférieur droit, pour sa part, contient plusieurs outils très utiles pour les usagers de RStudio. L'onglet *Files* permet à l'utilisateur de naviguer dans les fichiers que contient son ordinateur

6.5 Personnaliser son RStudio

sans avoir à sortir de 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 en plus de pouvoir en consulter la documentation. C'est aussi un des différents endroits à partir d'où il est possible d'installer des packages avec RStudio. L'onglet *Help* permet à l'utilisateur de chercher et de consulter de la documentation sur de nombreux sujets, notamment sur les différentes fonctions en R ainsi que sur les packages. Pour sa part, l'onglet *Viewer* permet la visualisation de contenu web local.

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

6.5 Personnaliser son RStudio

7 Baliser les sciences sociales : langages et pratiques

Lorsque vous lisez une page Web, un article scientifique ou un curriculum vitae professionnel, vous vous doutez peut-être que le texte n'est pas toujours produit à l'aide d'un simple logiciel de traitement de texte comme Microsoft Word, Apple Pages ou LibreOffice Writer. La mise en page complexe réglée au millimètre près, la qualité des figures et des tableaux, l'utilisation de gabarits professionnels, le style des références ou encore la présence d'éléments interactifs sont difficiles et parfois impossibles à reproduire à l'aide d'un logiciel de traitement de texte régulier. L'ajout d'extraits de code, de tableaux de régression ou encore de figures de haute qualité graphique, ainsi que leur personnalisation, nécessitent une interface particulière.

Pour ces raisons et plusieurs autres, les chercheurs en sciences sociales font souvent appel aux langages de balisage, ou *markup languages*. Ceux-ci permettent de produire des documents et pages Web sans les limitations des logiciels de traitement de texte. Le présent livre, par exemple, est écrit à l'aide du langage de balisage Markdown et de la plateforme de publication Quarto. D'entrée de jeu, vous vous demandez peut-être quelle est l'utilité d'apprendre ces langages alors que les logiciels de traitement de texte sont nombreux, simples d'approche et en amélioration constante. Ce chapitre tentera donc de répondre, tour à tour, aux trois grandes questions suivantes : *Qu'est-ce qu'un langage de balisage ? Quand et pourquoi utiliser un langage de balisage ? Comment utiliser un langage de balisage ?* L'accent

sera mis sur la plateforme Quarto ainsi que sur les langages Markdown et \LaTeX , bien que d'autres langages soient aussi abordés.

7.1 Qu'est-ce qu'un langage de balisage?

Un langage de balisage constitue un ensemble de commandes qui peuvent être entremêlées à du texte afin de produire une action informatique. Chaque langage contient son ensemble de commandes cohérentes et complémentaires. De manière plus formelle, ces commandes sont nommées *balises* (*tags* en anglais) et inscrites par le chercheur lui-même au travers du texte. Les balises constituent une manière de communiquer avec le logiciel que vous utilisez dans un langage qu'il peut comprendre, par exemple pour lui indiquer que vous désirez qu'une section du texte soit écrite en caractères gras, en italique, à double interligne ou encore que vous souhaitez positionner une image d'une certaine manière au travers du texte. Cette interaction est rendue possible par la standardisation des langages de balisage : chaque balise correspond à une action précise, peu importe le logiciel utilisé, la langue dans laquelle le texte est rédigé, le type d'ordinateur utilisé, etc. Dans votre document source, les balises sont entremêlées au contenu de votre document, puis au moment de compiler ce dernier, les balises produisent les actions informatisées qu'elles commandent et laissent comme document final le contenu mis en page tel que vous l'avez défini via les balises utilisées. La compilation est le processus par lequel un document écrit en langage de balisage est transformé en fichier textuel, en format PDF dans le cas de \LaTeX par exemple.

Le premier langage de balisage, le Generalized Markup Language (GML), a été inventé en 1969 par les chercheurs Charles F. Goldfarb, Ed Mosher et Ray Lorie pour la compagnie IBM. Goldfarb et ses collègues devaient intégrer trois applications créées avec des langages différents et avec une logique différente pour les besoins d'un bureau de droit. Même après avoir créé un programme qui permettait aux trois applications d'interagir, ces

7.1 Qu'est-ce qu'un langage de balisage?

langages demeuraient différents et avaient chacun leur propre fonctionnement. Le développement de GML a permis de résoudre ce problème en standardisant et en structurant le langage : les mêmes commandes étaient utilisées pour accomplir les mêmes tâches dans chaque programme (Goldfarb 1996). GML a été amélioré durant les décennies suivantes et a été suivi par d'autres langages de balisage, dont \LaTeX (1985), \BibTeX (1988), HTML (1993), XML (1998), Markdown (2004) et R Markdown (2012) (Encyclopaedia Britannica 2023; Hameed 2023; Markdown Guide 2023; World Wide Web Consortium (W3C) 1998; Xie 2023).

Les langages de balisage permettent d'effectuer différentes tâches. HTML, qui est sans doute le plus connu des langages de balisage, permet de formater des sites Web. XML, quant à lui, permet de structurer de larges volumes de données. \LaTeX permet pour sa part de formater du texte et de créer des documents en format PDF. Markdown permet également de créer des documents de format PDF, mais aussi en format HTML ou DOCX (format utilisé pour les documents Word), contrairement à \LaTeX . R Markdown permet d'ajouter des extraits de code R à un fichier en langage Markdown. Enfin, depuis 2022, le système de publication scientifique et technique multilingue Quarto permet d'intégrer des extraits de code R, \LaTeX , Python, Julia ou JavaScript, créés dans différents types d'environnements, à un fichier en langage Markdown (Allaire 2022). \LaTeX , Markdown, R Markdown et Quarto permettent aussi d'intégrer les références bibliographiques du système de traitement de références \BibTeX . Les langages de balisage communiquent ainsi souvent les uns avec les autres au sein d'un même fichier. Le Chapter 9 explique la manière de citer les références en langage \BibTeX par le biais de Zotero et de Better \BibTeX .

Les balises constituent une manière de donner manuellement des commandes au logiciel que vous utilisez. Si vous utilisez Microsoft Word, vous avez accès à une panoplie de boutons qui vous permettent de formater votre texte. Les balises exercent les mêmes fonctions de formatage pour les fichiers produits en \LaTeX ou en Markdown, mais doivent être ajoutées à l'écrit par l'utilisateur. Lorsque vous appuyez sur un bouton dans Word, celui-ci ajoute des balises au travers de votre texte, mais rend celles-ci

invisibles dans l'interface que vous utilisez. Cela permet d'avoir un texte élégant et facile à lire, mais comporte aussi plusieurs inconvénients. Le principal inconvénient est que vous êtes condamné à avoir un pouvoir limité sur le formatage de votre texte. En effet, si les boutons à votre disposition ne vous permettent pas de réaliser une opération, celle-ci sera éternellement impossible à réaliser pour vous. A contrario, les langages de balisage permettent un contrôle presque infini sur les opérations que vous souhaitez réaliser. Incidemment, dans la mesure où vous utilisez le langage approprié pour la tâche que vous souhaitez accomplir, vous devriez être capable de donner exactement la commande nécessaire à votre logiciel. Les langages de balisage, bien qu'ils aient un coût d'apprentissage qui peut s'avérer important et que l'interface de travail soit moins élégante qu'un simple document Word, vous offrent une plus grande flexibilité.

Afin d'utiliser un langage de balisage, il est impératif que le logiciel que vous utilisez puisse prendre en compte ce langage. Un logiciel permet rarement d'utiliser n'importe quel langage. Il est aussi impératif de bien utiliser le langage de balisage. En effet, comme pour les langages de programmation, les langages de balisage ne peuvent pas déduire ce que vous souhaitez leur faire comprendre. Si vous souhaitez mettre du texte en gras, vous devez utiliser les bonnes balises. La moindre erreur peut être fatale, puisqu'une erreur dans la balise que vous utilisez risque de produire une commande incompréhensible et un message d'erreur, le logiciel ne réussissant pas à associer votre balise mal inscrite à une action informatisée. Conséquemment, il est impératif de bien vérifier les balises utilisées afin d'éviter toute erreur qui empêcherait votre document d'être compilé, c'est-à-dire d'être traduit dans son format final¹. Chaque caractère dans une balise est important et il y a rarement plus d'une seule manière de commander une action. Le positionnement des balises est lui aussi critique : il délimite la portion de texte à laquelle doit être appliquée l'action

¹Les logiciels permettent plus ou moins efficacement d'identifier les balises problématiques. Certains ne produisent qu'un message d'erreur sans donner d'indication sur la source du problème, alors que d'autres ciblent très spécifiquement la ligne de syntaxe où se situe la balise problématique.

7.2 Qu'est-ce qu'un langage de balisage?

commandée par la balise.

Pour ces raisons et plusieurs autres, les chercheurs en sciences sociales font souvent appel aux langages de balisage, ou *markup languages*. Ceux-ci permettent de produire des documents et pages Web sans les limitations des logiciels de traitement de texte. Le présent livre, par exemple, est écrit à l'aide du langage de balisage Markdown et de la plateforme de publication Quarto. D'entrée de jeu, vous vous demandez peut-être quelle est l'utilité d'apprendre ces langages alors que les logiciels de traitement de texte sont nombreux, simples d'approche et en amélioration constante. Ce chapitre tentera donc de répondre, tour à tour, aux trois grandes questions suivantes : *Qu'est-ce qu'un langage de balisage? Quand et pourquoi utiliser un langage de balisage? Comment utiliser un langage de balisage?* L'accent sera mis sur la plateforme Quarto de même que sur les langages Markdown et L^AT_EX, bien que d'autres langages soient aussi abordés.

7.2 Qu'est-ce qu'un langage de balisage?

Un langage de balisage constitue un ensemble de commandes qui peuvent être entremêlées à du texte afin de produire une action informatique. Chaque langage contient son propre ensemble de commandes cohérentes et complémentaires. De manière plus formelle, ces commandes sont nommées *balises* (*tags* en anglais) et inscrites par la personne chercheuse elle-même au travers du texte. Les balises constituent une manière de communiquer avec le logiciel que utilisé dans un langage qu'il peut comprendre. Par exemple, une balise permet d'indiquer au logiciel que vous désirez qu'une section du texte soit écrite en caractères gras, en italique, à double interligne ou encore que vous souhaitez positionner une image d'une certaine manière au travers du texte. Cette interaction est rendue possible par la standardisation des langages de balisage : chaque balise correspond à une action précise, peu importe le logiciel utilisé, la langue dans laquelle le texte est rédigé, le type d'ordinateur utilisé, etc. Dans votre document source, les balises sont entremêlées au contenu de votre document, puis au

moment de compiler ce dernier, les balises produisent les actions informatisées qu'elles commandent et laissent comme document final le contenu mis en page tel que vous l'avez défini via les balises utilisées. La compilation est le processus par lequel un document écrit en langage de balisage est transformé en fichier textuel, en format PDF dans le cas de \LaTeX par exemple.

Le premier langage de balisage, le Generalized Markup Language (GML), fut inventé en 1969 par les chercheurs Charles F. Goldfarb, Ed Mosher et Ray Lorie pour la compagnie IBM. Goldfarb et ses collègues devaient intégrer trois applications créées avec des langages différents et avec une logique différente pour les besoins d'un bureau de droit. Même après avoir créé un programme qui permettait aux trois applications d'interagir, ces langages demeuraient différents et avaient chacun leur propre fonctionnement. Le développement de GML permis de résoudre ce problème en standardisant et en structurant le langage : les mêmes commandes étaient utilisées pour accomplir les mêmes tâches dans chaque programme (Goldfarb 1996). GML a été amélioré durant les décennies suivantes et a été suivi par d'autres langages de balisage, dont \LaTeX (1985), \BibTeX (1988), HTML (1993), XML (1998), Markdown (2004) et R Markdown (2012) (Encyclopaedia Britannica 2023; Hameed 2023; Markdown Guide 2023; World Wide Web Consortium (W3C) 1998; Xie 2023).

Les langages de balisage permettent d'effectuer différentes tâches. HTML, qui est sans doute le plus connu des langages de balisage, permet de formater des sites Web. XML, quant à lui, permet de structurer de larges volumes de données. \LaTeX permet pour sa part de formater du texte et de créer des documents en format PDF. Markdown permet également de créer des documents de format PDF, mais aussi en format HTML ou DOCX (format utilisé pour les documents Word), contrairement à \LaTeX . R Markdown permet d'ajouter des extraits de code R à un fichier en langage Markdown. Enfin, depuis 2022, le système de publication scientifique et technique multilingue Quarto permet d'intégrer des extraits de code R, \LaTeX , Python, Julia ou JavaScript, créés dans différents types d'environnements, à un fichier en langage Markdown (Allaire 2022). \LaTeX , Markdown,

7.2 *Qu'est-ce qu'un langage de balisage?*

R Markdown et Quarto permettent aussi d'intégrer les références bibliographiques du système de traitement de références `BIBTEX`. Les langages de balisage communiquent ainsi souvent les uns avec les autres au sein d'un même fichier. Le Chapter 9 explique la manière de citer les références en langage `BIBTEX` par le biais de Zotero et de Better `BIBTEX`.

Les balises constituent une manière de donner manuellement des commandes au logiciel que vous utilisez. Si vous utilisez Microsoft Word, vous avez accès à une panoplie de boutons qui vous permettent de formater votre texte. Les balises exercent les mêmes fonctions de formatage pour les fichiers produits en `LATEX` ou en Markdown, mais doivent être ajoutées à l'écrit par l'utilisateur. Lorsque vous appuyez sur un bouton dans Word, celui-ci ajoute des balises au travers de votre texte, mais rend celles-ci invisibles dans l'interface que vous utilisez. Cela permet d'avoir un texte élégant et facile à lire, mais comporte aussi plusieurs inconvénients. Le principal inconvénient est que vous êtes condamné à avoir un pouvoir limité sur le formatage de votre texte. En effet, si les boutons à votre disposition ne vous permettent pas de réaliser une opération, celle-ci sera éternellement impossible à réaliser pour vous. A contrario, les langages de balisage permettent un contrôle presque infini sur les opérations que vous souhaitez réaliser. Incidemment, dans la mesure où vous utilisez le langage approprié pour la tâche que vous souhaitez accomplir, vous devriez être capable de donner exactement la commande nécessaire à votre logiciel. Les langages de balisage, bien qu'ils aient un coût d'apprentissage qui peut s'avérer important et que l'interface de travail soit moins élégante qu'un simple document Word, vous offrent une plus grande flexibilité.

Afin d'utiliser un langage de balisage, il est impératif que le logiciel que vous utilisez puisse prendre en compte ce langage. Un logiciel permet rarement d'utiliser n'importe quel langage. Il est aussi impératif de bien utiliser le langage de balisage. En effet, comme pour les langages de programmation, les langages de balisage ne peuvent pas déduire ce que vous souhaitez leur faire comprendre. Si vous souhaitez mettre du texte en gras, vous devez utiliser les bonnes balises. La moindre erreur peut être fatale, puisqu'une erreur dans la balise que vous utilisez risque de pro-

duire une commande incompréhensible et un message d'erreur, le logiciel ne réussissant pas à associer votre balise mal inscrite à une action informatisée. Conséquemment, il est impératif de bien vérifier les balises utilisées afin d'éviter toute erreur qui empêcherait votre document d'être compilé, c'est-à-dire d'être traduit dans son format final². Chaque caractère dans une balise est important et il y a rarement plus d'une seule manière de commander une action. Le positionnement des balises est lui aussi critique : il délimite la portion de texte à laquelle doit être appliquée l'action commandée par la balise.

Afin d'utiliser un langage de balisage, il est impératif que le logiciel que vous utilisez puisse prendre en compte ce langage. Un logiciel permet rarement d'utiliser n'importe quel langage. Il est aussi impératif de bien utiliser le langage de balisage. En effet, comme pour les langages de programmation, les langages de balisage ne peuvent pas déduire ce que vous souhaitez leur faire comprendre. Si vous souhaitez mettre du texte en gras, vous devez utiliser les bonnes balises. La moindre erreur peut être fatale, puisqu'une erreur dans la balise que vous utilisez risque de produire une commande incompréhensible et un message d'erreur, le logiciel ne réussissant pas à associer votre balise mal inscrite à une action informatisée. Conséquemment, il est impératif de bien vérifier les balises utilisées afin d'éviter toute erreur qui empêcherait votre document d'être compilé, c'est-à-dire d'être traduit dans son format final³. Chaque caractère dans une balise est important et il y a rarement plus d'une seule manière de commander une action. Le positionnement des balises est lui aussi critique : il délimite la portion de texte à laquelle doit être appliquée l'action commandée par la balise.

²Les logiciels permettent plus ou moins efficacement d'identifier les balises problématiques. Certains ne produisent qu'un message d'erreur sans donner d'indication sur la source du problème, alors que d'autres ciblent très spécifiquement la ligne de syntaxe où se situe la balise problématique.

³Les logiciels permettent plus ou moins efficacement d'identifier les balises problématiques. Certains ne produisent qu'un message d'erreur sans donner d'indication sur la source du problème, alors que d'autres ciblent très spécifiquement la ligne de syntaxe où se situe la balise problématique.

7.3 Quand et pourquoi utiliser un langage de balisage?

Il est important de distinguer les langages de balisage des langages de programmation, qui sont abordés plus en détail dans le Chapitre 6. En effet, ceux-ci sont similaires à certains égards, mais ont des vocations différentes. Les deux s'appuient sur un langage informatisé, mais les langages et leurs objectifs diffèrent. Un langage de programmation définit des processus informatisés alors qu'un langage de balisage permet d'encoder du contenu de manière à ce que celui-ci soit lisible tant pour l'humain que pour son ordinateur.

Dans le contexte de la recherche en sciences sociales, la programmation est généralement utilisée afin de récolter, d'analyser et de présenter visuellement des données. Une fois cartes, tableaux et graphiques produits, ceux-ci peuvent être enregistrés — par exemple en format PDF ou PNG — et inclus au sein d'un document qui sera formaté en utilisant un langage de balisage. En R Markdown et en Quarto, des extraits de langage de programmation peuvent être inclus dans des sections bien délimitées de documents écrits en langage de balisage. Plus généralement, le langage de programmation contribue à l'analyse alors que le langage de balisage est essentiellement utile afin de présenter les travaux de recherche, que ce soit dans un document écrit ou sur un site Web. C'est principalement de cette manière que sont utilisés les langages de programmation et de balisage dans le cadre de la recherche en sciences sociales.

7.3 Quand et pourquoi utiliser un langage de balisage?

La plupart des langages de balisage permettent de remplir l'une des deux fonctions suivantes, qui sont particulièrement importantes dans le contexte de la recherche en sciences sociales : produire des documents écrits et formater des pages Web. Dans les deux cas, ces actions peuvent être réalisées à partir de logiciels simples, mais ces logiciels ont des limites importantes qui ne sont pas présentes en langage de balisage.

Pour l'écriture de documents très simples comme une liste d'épicerie ou des notes rapides pendant une conférence, les logiciels de traitement de texte sont tout à fait convenables : ils sont simples et rapides à utiliser, un formatage professionnel du document n'est pas de mise. Utiliser un langage de balisage pour des tâches de base peut en effet rendre la tâche inutilement longue. Par contre, plus la complexité d'un document augmente, plus il devient difficile d'obtenir un résultat satisfaisant en utilisant un logiciel de traitement de texte tel que Word, Pages ou Writer. A contrario, \LaTeX permet de produire des documents de tous les niveaux de complexité, tel que démontré sur la Figure 7.1. Quant à Markdown, sa courbe se situerait logiquement entre celles de \LaTeX et de Word. Plus généralement, utiliser un langage de balisage comme \LaTeX ou Markdown⁴ comporte plusieurs avantages par rapport aux logiciels de traitement de texte traditionnels. Ces avantages peuvent se résumer en quatre concepts : automatisation, personnalisation, flexibilité et qualité graphique.

La plupart des langages de balisage permettent de remplir l'une des deux fonctions suivantes, qui sont particulièrement importantes dans le contexte de la recherche en sciences sociales : produire des documents écrits et formater des pages Web. Dans les deux cas, ces actions peuvent être réalisées à partir de logiciels simples, mais ces logiciels ont des limites importantes auxquelles les langages de balisages apportent des solutions.

Pour l'écriture de documents très simples comme une liste d'épicerie ou des notes rapides pendant une conférence, les logiciels de traitement de texte sont tout à fait convenables : ils sont simples et rapides à utiliser, un formatage professionnel du document n'étant pas de mise. Utiliser un langage de balisage pour des tâches de base peut en effet rendre la tâche inutilement longue et complexe. Toutefois, plus la complexité d'un document s'avère grande, plus il devient difficile d'obtenir un résultat satisfaisant en utilisant un logiciel de traitement de texte tel que Word, Pages ou Writer.

⁴Les avantages et désavantages de Markdown cités dans cette section s'appliquent également à Quarto et à R Markdown, puisque ces derniers font appel au langage Markdown.

7.3 Quand et pourquoi utiliser un langage de balisage?

A contrario, \LaTeX permet de produire des documents de tous les niveaux de complexité, tel que démontré sur la Figure 7.1. Quant à Markdown, sa courbe se situerait logiquement entre celles de \LaTeX et de Word. Plus généralement, utiliser un langage de balisage comme \LaTeX ou Markdown⁵ comporte plusieurs avantages par rapport aux logiciels de traitement de texte traditionnels. Ces avantages peuvent se résumer en quatre concepts : automatisation, personnalisation, flexibilité et qualité graphique.

La plupart des langages de balisage permettent de remplir l’une des deux fonctions suivantes, qui sont particulièrement importantes dans le contexte de la recherche en sciences sociales : produire des documents écrits et formater des pages Web. Dans les deux cas, ces actions peuvent être réalisées à partir de logiciels simples, mais ces logiciels ont des limites importantes auxquelles les langages de balisages apportent des alternatives.

Pour l’écriture de documents très simples comme une liste d’épicerie ou des notes rapides pendant une conférence, les logiciels de traitement de texte sont tout à fait convenables : ils sont simples et rapides à utiliser, un formatage professionnel du document n’étant pas de mise. Utiliser un langage de balisage pour des tâches de base peut en effet rendre la tâche inutilement longue et complexe. Toutefois, plus la complexité d’un document s’avère grande, plus il devient difficile d’obtenir un résultat satisfaisant en utilisant un logiciel de traitement de texte tel que Word, Pages ou Writer. A contrario, \LaTeX permet de produire des documents de tous les niveaux de complexité, tel que démontré sur la Figure 7.1. Quant à Markdown, sa courbe se situerait logiquement entre celles de \LaTeX et de Word. Plus généralement, utiliser un langage de balisage comme \LaTeX ou Markdown⁶ comporte plusieurs avantages par rapport aux logiciels de traitement de

⁵Les avantages et désavantages de Markdown cités dans cette section s’appliquent également à Quarto et à R Markdown, puisque ces derniers font appel au langage Markdown.

⁶Les logiciels permettent plus ou moins efficacement d’identifier les balises problématiques. Certains ne produisent qu’un message d’erreur sans donner d’indication sur la source du problème, alors que d’autres ciblent très spécifiquement la ligne de syntaxe où se situe la balise problématique.

texte traditionnels. Ces avantages peuvent se résumer en quatre concepts : automatisation, personnalisation, flexibilité et qualité graphique.

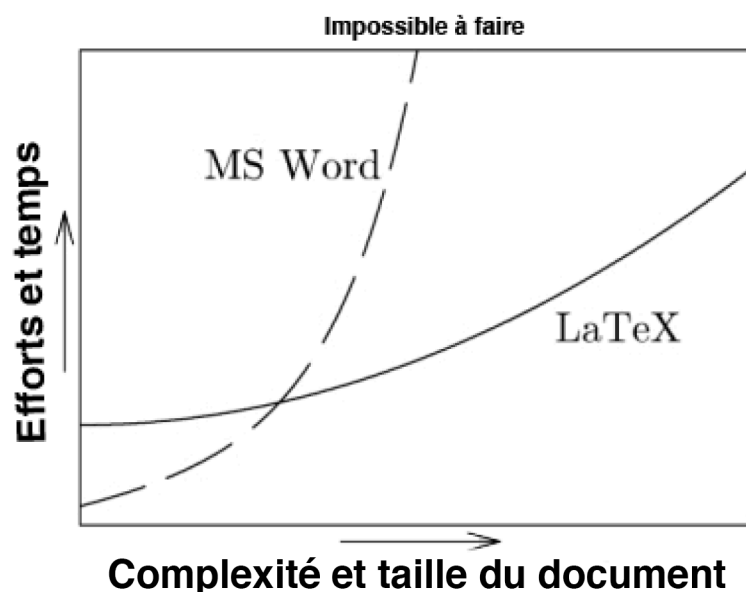


Figure 7.1: Utilité relative de Word et de \LaTeX selon la complexité et la taille du document

Source : Yannick Dufresne (2015).

7.3.1 Avantages

Premièrement, \LaTeX et Markdown permettent d'intégrer une bibliographie *automatique* et professionnelle en utilisant \BibTeX . Cette bibliographie peut être adaptée très facilement en différents styles bibliographiques reconnus ou en un style bibliographique *personnalisé* à partir d'un des nombreux gabarits professionnels disponibles. Avec \BibTeX , plus besoin de vérifier si le titre de l'article est toujours en italique, si le numéro de

7.3 Quand et pourquoi utiliser un langage de balisage?

volume est toujours entre parenthèses ou si le nom de famille des deuxièmes auteurs est toujours avant ou après le prénom puisque toutes ces opérations sont effectuées de manière *automatique*. BIB_TE_X comprend également les différences entre les types de sources (articles scientifiques, livres, sites Internet, etc.) et ajuste leur présentation en conséquence. De plus, si une des sources que vous citez n'est pas incluse dans la bibliographie, une erreur s'affiche, vous permettant d'identifier le problème plutôt que de vous retrouver avec une référence manquante. À l'inverse, si une source est retirée du texte, elle disparaît *automatiquement* de la bibliographie dans le document final mais demeure présente dans le fichier où se trouvent les références bibliographiques. Cela évite les aller-retour pour vérifier que chaque source de la bibliographie se trouve au moins une fois dans le texte et que chaque source dans le texte est citée en bibliographie. Grâce aux balises, en cliquant sur les références incluses dans le document, vous vous retrouverez immédiatement plus loin dans le document, à l'endroit où se trouve l'entrée bibliographique associée. Les références BIB_TE_X pour articles scientifiques peuvent être copiées-collées à partir de Google Scholar. BIB_TE_X rend donc extrêmement simple et efficace l'utilisation des références bibliographiques grâce à sa capacité à *personnaliser* et *automatiser* leur présentation.

L'intégration de figures et de tableaux dans le texte est aussi rendue très simple et professionnelle grâce à L^AT_EX et à Markdown. La taille de la figure ou du tableau, son positionnement et son intégration par rapport au texte environnant peuvent être réglés de telle sorte que l'ajout de texte avant ou après la figure ou le tableau ne produira pas des résultats inattendus tels qu'une demi-page vide avant un graphique ou un titre de tableau complètement en bas d'une page. En définissant des paramètres pour l'ensemble du texte, le chercheur peut *personnaliser* entièrement la présentation des figures et des tableaux. De plus, la *qualité des figures et des tableaux* ne diminue pas lors de leur intégration : les figures restent aussi belles qu'elles l'étaient originalement, ce qui n'est pas toujours le cas dans les logiciels de traitement de texte. Les figures et les tableaux sont aussi numérotés *automatiquement*, ce qui veut dire que vous n'aurez jamais à

vous préoccuper de modifier le numéro en ajoutant une figure ou un tableau dans le texte. Grâce aux balises, en cliquant sur le numéro associé à la figure ou au tableau dans le texte, le document se retrouve automatiquement à l'endroit où se trouve le graphique ou le tableau. De plus, les figures peuvent être intégrées en format PDF, ce qui permet au lecteur de copier-coller ou de surligner de l'information se trouvant sur le graphique directement, incluant les titres des axes et les annotations.

Surtout, l'intégration de graphiques produits par R au texte en langage de balisage est simplifiée et *automatisée*. En effet, même lorsque les données ou le code pour produire un graphique changent, R resauvegarde le fichier dans le même chemin d'arborescence (*path*) particulier que vous avez indiqué, par exemple `C:/Users/Jean/Dropbox/projet1/graphs/Figure1.pdf`. Le langage de balisage peut ensuite indiquer le même chemin d'arborescence, de sorte qu'il n'est pas nécessaire de recopier-coller la figure à l'intérieur du document chaque fois que des changements y sont apportés; la figure est mise à jour *automatiquement*.

L'intégration de figures et de tableaux est particulièrement simple et *flexible* avec Quarto. Contrairement à \LaTeX , qui nécessite la production de tableaux et de figures dans un document en langage de programmation (comme R), Quarto permet de créer une figure grâce à du code R et d'intégrer celle-ci au texte dans un seul document. Cela se fait grâce à l'intégration de blocs de code R (*code chunks*) dans le document. Le code est produit dans le bloc de code et la figure ou le tableau qui en résulte apparaît à la fois dans le document Quarto, où des balises supplémentaires permettent d'adapter le formatage, et sur le document fini.

\LaTeX permet également d'ajouter des équations mathématiques poussées. En effet, il existe des balises pour chaque symbole mathématique, et ceux-ci peuvent être agencés de manière à former des équations cohérentes. Ces équations peuvent être intégrées au sein même d'une phrase ou être mises de l'avant dans un paragraphe à part centré.

Markdown et \LaTeX permettent aussi la gestion *automatisée* de la table des matières, et les références aux pages appropriées à partir de la table

7.3 Quand et pourquoi utiliser un langage de balisage?

des matières se mettent à jour en continu. La table des matières prend en compte l'architecture du texte choisie manuellement par le chercheur, qui est définie par des balises définissant différents niveaux hiérarchiques de sections, sous-sections ou chapitres. Des manières *automatiques* de référencer les figures et les tableaux dans des sections distinctes de la table des matières sont également offertes, encore une fois *personnalisables* au goût du chercheur.

Bien que la mise en page de documents produits via Markdown et L^AT_EX puisse être définie entièrement manuellement par un.e utilisateur.ice expérimenté.ee, les débutants.antes apprécieront les nombreux gabarits (*templates*) qui permettent de gérer *automatiquement* la mise en page de manière « clé-en-main ». Ceux-ci permettent de rendre l'apparence d'un document plus esthétique et uniforme et peuvent être utilisés tels quels ou servir de point de départ pour un.e chercheur.euse souhaitant y apporter certaines modifications sans toutefois partir d'une feuille blanche. La majorité des utilisateurs.rices, même les plus expérimentés.es, utilisent ces gabarits comme base lorsqu'ils.elles rédigent un document. Ceux-ci constituent une mine d'or puisqu'ils rendent accessible le code Markdown ou L^AT_EX ayant servi à la conception du gabarit, permettant à la personne chercheuse de comprendre comment est obtenu le résultat que lui offre le gabarit. Incidemment, la personne chercheuse peut identifier les sections de code produisant certains éléments de mise en page (ex : positionnement des numéros de page, positionnement du nom des auteurs en début de document, etc.) et les modifier ou s'en inspirer afin de modifier d'autres gabarits. L'utilisation de ces gabarits peut s'avérer complexe pour les non-initiés.ées, mais il s'agit d'une complexité qui s'avère ultimement extrêmement productive puisqu'elle permet à la personne chercheuse de devenir autonome et d'ajuster les gabarits à sa convenance afin de produire exactement le résultat désiré en termes de mise en page. En comparaison, les logiciels de traitement de texte rendent souvent très ardue la mise en page uniforme d'un document, puisque cet élément ne peut pas être *automatisé*. La liste des gabarits disponibles est extrêmement large, et ceux-ci ont une variété de fonctions. En effet,

7 Baliser les sciences sociales : langages et pratiques

une variété de gabarits professionnels et de haute *qualité graphique* sont offerts gratuitement en ligne pour des articles, des livres, des rapports, des *curriculum vitæ*s ou encore des feuilles de temps pour des contrats rémunérés.

Les images ci-haut ne sont qu'un exemple parmi tant d'autres de gabarits disponibles en ligne. L'interface en ligne du logiciel Overleaf offre une grande variété de ces exemple de gabarits. Libre au chercheur et à la chercheuse d'y naviguer et de voir quel gabarit convient le mieux à ses besoins. ===== *Images de CVs et feuilles de temps professionnels*

7.3 Quand et pourquoi utiliser un langage de balisage?

Your Name Here, Ph.D.

✉ example@gmail.com [@overleaf_example](#) [example](#)
<http://example.example.org/>



Employment History

- 2014 – **Community Witch**, Village of Frying Pans.
2013 – 2015 **Lecturer**, Information Technology Department, School of Engineering, Science and Technology, XYZ College.

Education

- 2009 – 2013 **Ph.D., Unseen University** High Energy Magic.
Thesis title: *Low-Cost Mana Generation in Under-Resourced Environments*.
2003 – 2006 **M.Sc. Computer Science, Unseen** in High Energy Magic.
Thesis title: *Applying ant algorithms in automatic design of novel magic charms*.
 M.Sc. Computer Science, Unseen in High Energy Magic.
Thesis title: *Applying ant algorithms in automatic design of novel magic charms*.

Research Publications

Journal Articles

- 1 L. T. Lim, R. T. Chiew, E. K. Tang, A. G. Rusli, and Y. Naimah, "Digitising a machine-tractable version of Kamus Dewan with TEL-P5," *PeerJ Preprints*, vol. 4, e2205v1, 2016, ISSN: 2167-9843. DOI: 10.7287/peerj.preprints.2205v1.
- 2 F. Bond, L. T. Lim, E. K. Tang, and H. Riza, "The combined Wordnet Bahasa," *Nusa: Linguistic studies of languages in and around Indonesia*, vol. 57, pp. 83–100, 2014. URL: <http://hdl.handle.net/10108/79286>.
- 3 L. T. Lim, L.-K. Soon, T. Y. Lim, E. K. Tang, and B. Ranaivo-Malançon, "Lexicon+TX: Rapid construction of a multilingual lexicon with under-resourced languages," *Language Resources and Evaluation*, vol. 48, no. 3, pp. 479–492, 2014, ISSN: 1574-020X. DOI: 10.1007/s10579-013-9253-0.
- 4 L. T. Lim, B. Ranaivo-Malançon, and E. K. Tang, "Low cost construction of a multilingual lexicon from bilingual lists," *Polibits*, vol. 43, pp. 45–51, 2011.
- 5 L. T. Lim, B. Ranaivo-Malançon, and E. K. Tang, "Symbiosis between a multilingual lexicon and translation example banks," *Procedia: Social and Behavioral Sciences*, vol. 27, pp. 61–69, 2011.
- 6 L. T. Wong and E. Someone, "A non-existent paper," *Journal of Carrying On*, vol. 12, 2011.
- 7 E. Someone and L. T. Lim, "Another paper something something," *Journal of Carrying On*, vol. 11, 2010.
- 8 E. Someone and T. Lim, "A fictional research," *Journal of Carrying On*, vol. 10, 2010.

Conference Proceedings

- 1 K. M. Boon and L. T. Lim, "An examination question paper preparation system with content-style separation and Bloom's taxonomy categorisation," in *Proceedings of the 3rd International Conference on E-Learning and E-Technologies in Education (ICEEE 2014)*, Kuala Lumpur, Malaysia, 2014, pp. 39–47. URL: <http://goo.gl/pfdUfm>.

7 Baliser les sciences sociales : langages et pratiques

Comp.

thank you for your confidence!

Invoice

N° 001
August 28, 2023

Company Foo,
Temple Bar,
Dublin.

Title of the invoice

| | product | unit price | qty. | price |
|--------------------|---------|------------|------|--------|
| My product 1 | | 15 € | 10 | 150 € |
| My product 2 | | 25 € | 10 | 250 € |
| My product 3 | | 35 € | 10 | 350 € |
| My product 4 | | 45 € | 10 | 450 € |
| Total ¹ | | | | 1200 € |

Comp.,
Auto-entrepreneur (APE XXXXX),
foo, bar street, XXXXX City,
SIREN: XXX XXXX XXXX,

Tél: XX XX XX XX XX,
Mét: xxx@ccc.xxx,
IBAN: XXXX XXXX XXXX XXXX XXXX XXXX XXXX,
BIC: XXX XXX XXX

Conditions de paiement: write the sell conditions here
on several lines

¹example of footnote

Les

7.3 Quand et pourquoi utiliser un langage de balisage?

images ci-haut ne sont qu'un exemple parmi tant d'autres de gabarits disponibles en ligne. L'interface en ligne du logiciel Overleaf offre une grande variété de ces exemple de gabarits. Libre au chercheur et à la chercheuse d'y naviguer et de voir quel gabarit convient le mieux à ses besoins.

7.3 Quand et pourquoi utiliser un langage de balisage?

Comp.
thank you for your confidence!

Invoice
N° 001
August 28, 2023

Company Foo,
Temple Bar,
Dublin.

Title of the invoice

| | product | unit price | qty. | price |
|--------------------|---------|------------|------|--------|
| My product 1 | | 15 € | 10 | 150 € |
| My product 2 | | 25 € | 10 | 250 € |
| My product 3 | | 35 € | 10 | 350 € |
| My product 4 | | 45 € | 10 | 450 € |
| Total ¹ | | | | 1200 € |

Comp.,
Auto-entrepreneur (APE XXXXX),
foo, bar street, XXXXX City,
SIREN: XXX XXXX XXXX,

Tél: XX XX XX XXXX,
Mél: xxx@foo.xxx,
IBAN: XXXX XXXX XXXX XXXX XXXX XXXX,
BIC: XXX XXX XXXX

Conditions de paiement: write the sell conditions here
on several lines

¹example of footnote

Les

images ci-haut ne sont qu'un exemple parmi tant d'autres de gabarits disponibles en ligne. L'interface en ligne du logiciel Overleaf offre une grande variété de ces exemple de gabarits. Libre au chercheur et à la chercheuse d'y naviguer et de voir quel gabarit convient le mieux à ses besoins.

Un autre avantage non-négligeable de Markdown — qui le distingue à cet égard de \LaTeX — est la *flexibilité* qu'il offre à ses utilisateurs.rices. En effet, en utilisant Pandoc Markdown, qui est une extension du langage Markdown de base permettant de combiner plusieurs langages de balisage différents en un seul document, il est possible d'intégrer dans un seul document plusieurs langages de balisage différents tels que Markdown, \LaTeX ou HTML. Quarto est également habilité en plus à travailler avec des extraits de code R ou Python. Ceci permet donc à l'utilisateur.rice de bénéficier des fonctionnalités de différents langages dans un seul document, rendant ainsi possible une variété de *personnalisations* qui ne seraient pas possible autrement. Qui plus est, puisque Markdown permet de créer des fichiers Word réguliers, PDF professionnels et HTML à partir d'un même document, l'utilisateur.rice peut choisir à sa convenance et à tout moment de quelle manière sera compilé le document rédigé. Cette possibilité de créer des documents Word est particulièrement pratique dans le cadre de collaboration avec des chercheurs.euses n'utilisant pas les langages de balisage ainsi que lors de l'envoi de manuscrits à des revues scientifiques, puisque certaines d'entre elles exigent de recevoir ceux-ci sous forme de document Word.

Bien que l'apprentissage de \LaTeX et de Markdown puisse être parsemé de nombreuses embuches, ces deux langages bénéficient d'une communauté d'utilisateurs.rices en ligne sur laquelle il est possible de s'appuyer afin de résoudre tout problème rencontré. Les utilisateurs.rices — particulièrement les plus expérimentés.ées — sont nombreux.euses à partager leur expérience à leurs collègues rencontrant des problèmes afin de contribuer à régler ceux-ci. Cette communauté est présente sur une multitude de sites Web, bien que le point de rencontre principal soit le forum Stack Overflow (2023), qui est également utilisé pour régler des prob-

7.3 Quand et pourquoi utiliser un langage de balisage?

lèmes de programmation et est abordé plus en détail dans le Chapter 6. Une simple recherche sur Google d'un problème rencontré avec \LaTeX ou Markdown offrira à l'utilisateur.rice des liens vers des échanges pertinents ayant eu lieu sur Stack Overflow ou encore vers de la documentation technique. L'utilisateur.rice pourra donc filtrer les résultats et observer les nombreuses solutions envisageables à son problème afin de définir laquelle est la plus appropriée dans sa situation. Il est important de noter, toutefois, que cette communauté est nettement plus développée pour les utilisateurs de \LaTeX que de Markdown, puisque ce dernier langage est moins répandu que le premier.

Également, avec l'émergence de l'intelligence artificielle (IA), de nombreux modèles commencent à émerger comme des ressources d'aides utiles pour les chercheuses et les chercheurs. Au moment de la rédaction du présent chapitre, le *chatbot* ChatGPT, développé par OpenAI et basé sur le grand modèle de langage GPT-3.5, est déjà une ressource d'aide en ce qui a trait aux langages de balisage. Le corpus de données sur lequel il a été formé inclut une grande variété de langages et de styles d'écriture, incluant \LaTeX et Markdown. Ainsi, il est possible de poser des questions à ce *chatbot* lorsque des problèmes de balisage sont rencontrés, et celui-ci fournira en réponse le texte avec les balises adéquates pour régler le problème — y compris pour des problèmes pour lesquels la réponse n'est pas directement indiquée sur Stack Overflow, lorsque la logique des langages est comprise par ces modèles basés sur l'IA. ChatGPT est toutefois plus outillé en \LaTeX qu'en Markdown ou en Quarto en raison de la plus grande abondance de ressources en \LaTeX disponibles en ligne. Il arrive cependant régulièrement que les réponses des modèles de langage comme ChatGPT soit erronées — tout comme certaines réponses sur Stack Overflow peuvent ne pas être adaptées à régler le problème d'un.e utilisateur.rice. Il est donc important de vérifier que le code compilé affiche bien la balise suggérée dans la réponse. Ainsi, il est utile de s'appuyer autant sur la communauté d'utilisateurs.rices de langages de balisage qui échange des ressources en ligne que sur les modèles de langage basés sur l'IA.

Certaines manières plutôt spécifiques de formater le texte sont présen-

tement disponibles avec \LaTeX ou Markdown bien que non disponibles en Word, ce qui constitue une autre preuve de leur grande *flexibilité* et capacité de *personnalisation*. Bien qu'il soit rare que nous ayons absolument besoin de personnaliser le texte ainsi, ces possibilités peuvent s'avérer utiles lorsque vous rédigez un texte qui doit se conformer en tout point à un gabarit spécifique. En effet, certaines revues scientifiques, maisons d'édition ou universités, dans le cadre de la rédaction d'articles, de mémoires et de thèses par exemple, imposent ce type de gabarit inflexible et parfois plutôt capricieux. C'est dans ce type de contexte que la *flexibilité* de Markdown peut s'avérer utile.

Les langages de balisage permettent également de créer des pages Web. Bien que les pages Web puissent être créées à partir de sites Web comme WordPress, le langage HTML permet de produire des résultats plus *personnalisables*, plus *automatisables* et avec une plus *grande qualité graphique*.

Finalement, il est important de mentionner que Markdown, Quarto et \LaTeX sont entièrement gratuits et accessibles aux utilisateurs de tous les systèmes d'exploitation.

7.3.2 Inconvénients

Il existe toutefois des désavantages inhérents à l'utilisation des langages de balisage. L'un des principaux désavantages de Markdown et de \LaTeX est le fait qu'ils ne comportent aucun système de suivi des modifications lors de travaux collaboratifs. Pour réviser un travail fait en langage de balisage, des commentaires peuvent être ajoutés sur le fichier sortant — nécessairement PDF pour un fichier sortant produit avec \LaTeX . Des commentaires peuvent aussi être faits directement dans le document \LaTeX ou Markdown, à l'aide de balises spécifiques. Ces commentaires n'apparaissent cependant pas dans le fichier sortant. Le suivi des modifications en \LaTeX et Markdown nécessite donc souvent l'utilisation de Git et de GitHub, qui sont abordés plus en détail dans le Chapitre 11. Même avec GitHub, les longs paragraphes ayant fait l'objet de plusieurs modifications peuvent être longs

7.3 Quand et pourquoi utiliser un langage de balisage?

à comparer par rapport à Word, qui permet de visualiser les propositions d'ajouts et de retraits de caractères de manière plus intuitive. Le suivi des modifications en Word permet également de distinguer les auteurs de différents commentaires par leurs noms, ce que GitHub ne permet pas de faire. Pour ces raisons, et aussi pour faciliter la mise en page par les éditeurs, certaines revues scientifiques refusent les fichiers PDF et demandent que les soumissions soient faites en format DOCX — ce qui pose problème pour les utilisateurs de \LaTeX mais pas ceux de Markdown.

Les langages de balisage comportent également un autre désavantage important dans certains cas : l'absence d'un correcteur de fautes de français complet, en particulier pour corriger les fautes autres que celles d'orthographe. Parmi les principaux endroits permettant l'édition en langages de balisage, Visual Studio Code (VS Code) et Overleaf comprennent tous deux une extension Grammarly, et VS Code possède également une extension Antidote. Cependant, RStudio ne possède qu'un correcteur orthographique de base, disponible en plusieurs langues. Ce correcteur ne repère pas les erreurs de syntaxe, de grammaire ou de forme, entre autres. Ces éléments sont pourtant essentiels pour la rédaction de textes académiques. Pour les utilisateurs de RStudio, il est donc souvent nécessaire de copier-coller le texte dans un logiciel externe pour faire une révision linguistique complète, puis d'intégrer les corrections en collant le texte corrigé dans le document original Markdown ou en \LaTeX .

Enfin, les langages de balisage, contrairement aux logiciels de traitement de texte, nécessitent d'être compilés, ce qui implique que deux fichiers coexistent : le fichier où le langage de balisage est utilisé (format `.tex` pour \LaTeX , `.md` pour Markdown ou encore `.qmd` pour Quarto) ainsi que le fichier où le texte final balisé apparaît (généralement `.pdf`, `.docx` ou `.html`). La compilation peut prendre un temps variable selon la complexité du document, mais dure typiquement une quinzaine de secondes. Le fait de devoir travailler avec deux fichiers en parallèle et de ne pas voir immédiatement l'effet des balises sur le document final constitue ainsi un autre désavantage des langages de balisage.

L^AT_EX comporte aussi quelques difficultés techniques particulières qui peuvent être réglées ou diminuées en travaillent en Markdown. Premièrement, L^AT_EX est difficile à apprendre. Certaines tâches qui peuvent sembler simples comme l'ajout d'un tableau peuvent nécessiter de nombreuses lignes de code. De plus, à la moindre erreur de frappe dans l'utilisation d'une balise, le code risque de planter et de ne pas produire le document PDF souhaité. C'est ce qu'on appelle une erreur de compilation. Markdown est un langage plus simple à apprendre, avec des balises plus courtes et intuitives. Il occasionne donc moins d'erreurs de compilation.

Deuxièmement, L^AT_EX est incompatible avec les logiciels de traitement de texte. Pour transférer un fichier créé à partir d'un logiciel de traitement de texte vers L^AT_EX, les balises doivent être ajoutées manuellement une par une. À l'inverse, pour transférer un document L^AT_EX vers un fichier de traitement de texte, les balises doivent être retirées une par une et le formatage doit être refait en utilisant les boutons fournis sur le logiciel de traitement de texte. Il est aussi possible de copier le texte directement à partir du fichier PDF produit par L^AT_EX vers Word, mais les fins de ligne sont interprétées par Word, Pages ou Writer comme des retours plutôt que des espaces, et les accents sont souvent mal copiés et doivent être réécrits manuellement. Encore une fois, Markdown évite ce problème en permettant d'écrire un fichier DOCX à partir du langage de balisage. Le formatage du fichier DOCX demeure un peu compliqué cependant et doit être fait à partir du modèle d'un autre document DOCX formaté tel que souhaité. De plus, les fichiers DOCX ne peuvent pas être transformés en format Markdown. Quarto permet d'écrire un texte en format Markdown et de produire un fichier DOCX à partir d'un gabarit Word. De plus, pour les fichiers Word à transformer en format Markdown, les balises plus simples en Markdown qu'en L^AT_EX rendent la tâche plus simple.

Somme toute, Word n'est pas à antagoniser et demeure très utile pour des tâches simples. Cependant, dans le monde académique, la production de fichiers de qualité faisant appel à des graphiques, tableaux et blocs de code personnalisés de qualité et automatisés est simplifiée en utilisant des langages de balisage.

7.4 Comment utiliser un langage de balisage?

En pratique, comment utilise-t-on Markdown, \LaTeX et \BibTeX ? D'emblée, \LaTeX a une syntaxe particulière qui demande un certain temps d'adaptation. Pour écrire une phrase simple comme celle-ci, la phrase peut être écrite telle quelle. Par contre, pour mettre un **mot** en caractères gras, il faut utiliser la balise suivante: `\textbf{mot}`. Pour mettre le **mot** en rouge, la balise est `\textcolor{red}{mot}`. Pour le mettre en italique et en note de bas de page⁷, les balises `\footnote{\emph{mot}}` peuvent être utilisées. Ainsi, des balises peuvent contenir d'autres balises. En langage \LaTeX , une balise commence toujours par une barre oblique inversée. Par la suite, le nom de la fonction (*emph*, *textbf*, *textcolor*, etc.) est appelé. Enfin, généralement, le mot à formater est placé entre accolades (`{}`).

Chaque document \LaTeX commence par un préambule. Celui-ci présente des informations telles que la taille des caractères, le type d'article, le format de mise en page, la police de caractères, l'utilisation d'en-têtes et de pieds de page, ainsi que l'utilisation de *packages* \LaTeX permettant différentes fonctionnalités de personnalisation du document. Il n'est pas nécessaire ni souhaitable d'apprendre l'ensemble des fonctions et des *packages* \LaTeX qui existent. Au contraire, il est souvent mieux de commencer par un gabarit de document qui convient au type de document que vous voulez créer et ensuite de rechercher en anglais sur Stack Overflow la manière d'ajouter des éléments de formatage que vous ne connaissez pas (par exemple, en recherchant `highlight latex text`). Des gabarits de documents \LaTeX sont disponibles sur le site Web d'Overleaf (2023).

Markdown fonctionne de manière similaire à \LaTeX , mais se démarque par sa plus grande flexibilité et sa syntaxe beaucoup plus légère. Par contre, il nécessite parfois l'utilisation de balises \LaTeX afin de réaliser certaines tâches, comme changer la couleur du texte. Tout document Markdown débute avec un court bloc de syntaxe **YAML** (acronyme de **Yet**

⁷*mot*

Another Markup Language) qui définit les paramètres généraux du document. Voici un bloc **YAML** typique pour un document Quarto :

Markdown fonctionne de manière similaire à \LaTeX , mais se démarque par sa plus grande flexibilité et sa syntaxe beaucoup plus légère. Par contre, il nécessite parfois l'utilisation de balises \LaTeX afin de réaliser certaines tâches, comme changer la couleur du texte. Tout document Markdown débute avec un court bloc de syntaxe **YAML** (acronyme de **Yet Another Markup Language**) qui définit les paramètres généraux du document. Voici un bloc **YAML** typique pour un document Quarto :

8

```
---
title: "Baliser les sciences sociales"
subtitle: "Langages et pratiques"
date: today
author:
  - Alexandre Fortier-Chouinard^[University of Toronto]
  - Maxime Blanchard^[McGill University]
  - Étienne Proulx^[McGill University]
format: pdf
toc: true
date-format: "MMMM D, YYYY"
bibliography: references.bib
---
```

Outre le titre, le sous-titre et le nom des auteurs, on trouve aussi dans l'en-tête YAML la présence d'une table des matières (`toc`), la date et son format, le format du document compilé — dans ce cas-ci, PDF — ainsi que le chemin d'arborescence afin d'accéder au document `BIBTEX` où sont enregistrées les références utilisées. Il est aussi possible d'y définir la taille de la police de caractères ou encore le gabarit Word servant à définir le format d'un document `DOCX` à produire. De manière particulièrement importante, c'est l'endroit où sont chargés les *packages* `LATEX` qui seront utilisés. En effet, la majorité des *packages* et fonctions `LATEX` sont utilisables dans Markdown, alors que l'inverse n'est pas vrai. Il est donc possible de personnaliser un document Markdown en utilisant des *packages* ayant été créés pour `LATEX`.

La syntaxe à utiliser au travers du texte est somme toute plutôt simple. Pour mettre un ou plusieurs **mots en gras**, il suffit de les entourer de deux astérisques (****mots en gras****); pour les mettre *en italique*, il faut les encadrer d'une seule astérisque (***en italique***). Pour définir un titre de section ou de sous-section, il suffit de mettre des # devant le titre en question. Plus vous ajoutez de #, plus le titre sera petit et plus il sera considéré à un niveau hiérarchique inférieur dans la structure du texte. La syntaxe Markdown est donc plus légère que celle de L^AT_EX, dans le but d'en rendre la lecture plus simple pour son utilisateur.

Bien que des gabarits Markdown soient disponibles, ceux-ci sont plus rares. Ils se trouvent pour la plupart sur GitHub et sont rendus disponibles par leur créateur. Cela étant dit, leur personnalisation peut s'avérer plutôt complexe. En somme, Markdown est particulièrement pratique pour les documents ne nécessitant pas de respecter un gabarit précis et réquérant simplement un document d'allure simple et professionnelle.

Pour sa part, BIB_TE_X a une syntaxe relativement simple. D'emblée, les références BIB_TE_X pour des articles et ouvrages scientifiques sont disponibles sur Google Scholar. Toutefois, pour citer des sites Web ou des articles de médias, la référence doit être écrite à la main selon un format précis. Une bibliographie sur BIB_TE_X peut ressembler à ceci :

```
@book{darwin03,
  address = {London},
  author = {Darwin, Charles},
  publisher = {John Murray},
  title = {{On the Origin of Species by Means of Natural Selection
or the Preservation of Favoured Races in the Struggle for Life}},
  year = {1859}
}
```

```
@article{goldfarb96,
  title={The Roots of SGML: A Personal Recollection},
```

```

author={Goldfarb, Charles F},
journal={Technical communication},
volume={46},
number={1},
pages={75},
year={1999},
publisher={Society for Technical Communication}
}

```

Un fichier `BIBTEX` ne contient rien de plus qu’une série de publications commençant chacune par la balise `@` suivie du type d’article — *article*, *book* pour un livre, *incollection* pour un chapitre de livre, *inproceedings* pour une présentation dans une conférence, *unpublished* pour un article non publié et *online* pour un site Web sont parmi les plus connus — et des informations sur la publication mises entre accolades. La première information entre accolades est le code de la référence, par exemple `goldfarb96`. Dans le fichier `LATEX`, l’auteur doit écrire `\cite{goldfarb96}` pour voir dans le document PDF compilé Goldfarb (1996); le lien est automatiquement cliquable et renvoie à la notice bibliographique correspondante. L’ordre des publications dans le document `BIBTEX` a peu d’importance, puisque `LATEX` réordonne par défaut la bibliographie en ordre alphabétique.

8.0.1 Environnements d’édition et de compilation

Contrairement à Microsoft Word et Apple Pages, il existe plusieurs options d’environnements d’édition et de compilation spécifiques à chaque langage. Ces environnements sont des plateformes et des logiciels conçus pour faciliter l’édition, la mise en forme et la compilation de documents dans des langages de balisage tels que `LATEX` et Markdown. Ils permettent également de rendre plus efficace et conviviale la production de documents tout en fournissant des fonctionnalités spécifiques aux besoins de chaque langage. Il existe une grande diversité d’environnements d’édition et de compilation, et le choix est libre pour le chercheur ou la chercheuse de

trouver celui qui convient le mieux à ses besoins ou aux besoins de son groupe de recherche. Les trois options discutées ici sont parmi les plus utilisées par les chercheurs en sciences sociales et peuvent être regroupées en deux catégories : les logiciels de bureau et les éditeurs en ligne.

D’abord, il existe plusieurs logiciels de bureau qui offrent un environnement d’édition et/ou de compilation pour les langages de balisage. Ces logiciels fournissent les programmes principaux, les extensions essentielles et des outils complémentaires de compilation et de visualisation afin de permettre la production de documents écrits en langages de balisage. Le logiciel RStudio, également abordé dans le chapitre Chapter 6, permet de produire des documents avec différents langages de balisage et programmation, ainsi que de naviguer entre eux, à partir d’une même fenêtre. Il suffit d’installer certains *packages* contenant les fichiers nécessaires à l’utilisation des langages de balisage. Par exemple, il est possible de produire des documents en \LaTeX en utilisant le code suivant dans la console pour installer le *package* nécessaire à l’utilisation de la distribution \LaTeX Tiny \TeX : `install.packages("tinytex")`. Suivant le même principe, il est possible de produire des documents en R Markdown sur RStudio en installant le *package* suivant : `install.packages("rmarkdown")`. Pour Quarto, le téléchargement se fait en ligne, directement à partir du site Web de Quarto (2023).

Pour l’écriture en \LaTeX , il est également nécessaire d’installer l’une des nombreuses distributions en ligne afin de pouvoir compiler ces documents dans un environnement local. Il existe des distributions telles que Mac \TeX pour Mac, Mik \TeX pour Windows et plusieurs autres (**distributions?**). Ces distributions se distinguent par les différents *packages* avec lesquelles elles sont compatibles.

Un autre environnement régulièrement utilisé pour travailler en langage de balisage est le logiciel de bureau VS Code. VS Code prend en compte un plus grand nombre de langages de programmation et est utilisé par les programmeurs de tous domaines, tandis qu’RStudio est surtout utile pour les chercheurs en sciences sociales qui travaillent surtout en R.

Lorsque vient le temps de collaborer à plusieurs sur un documents écrits en Markdown ou en \LaTeX , les logiciels de bureau évoqués précédemment nécessitent l'utilisation de GitHub et de Git. L'utilisation de ces éditeurs peut présenter un défi supplémentaire pour les équipes de recherche non initiées. Il existe ainsi des éditeurs en ligne qui permettent de collaborer en temps réel sans passer par Git et GitHub, de manière similaire à Google Docs¹. Le plus connu de ces logiciels est Overleaf, qui permet de produire des documents en langage \LaTeX . Puisqu'Overleaf permet d'avoir accès à ses documents \LaTeX à partir de n'importe quel navigateur, il n'y a pas de dépendance à un logiciel local sur un ordinateur, ce qui constitue un avantage important. La contrepartie de cet avantage est qu'en utilisant Overleaf, l'équipe de recherche est dépendante d'une connexion à Internet. En utilisant le package \LaTeX `rmarkdown`, Overleaf peut également inclure du code Markdown. Cependant, Overleaf ne permet malheureusement pas de créer des documents en format DOCX ou HTML, ce qui constitue une limite de l'application. Overleaf comporte un compteur de mots intégré, ce qui n'est pas le cas des autres logiciels et environnements présentés plus haut.

8.1 Conclusion

Somme toute, les langages de balisage permettent d'effectuer des tâches que vous ne pourriez pas normalement réaliser en utilisant un logiciel de traitement de texte classique. Ils facilitent la production de documents professionnels dans différents formats personnalisés, produits avec des processus automatisés, avec une grande qualité graphique. Les langages de balisage demandent un certain temps d'apprentissage, entre autres pour \LaTeX , mais peuvent ensuite être utilisés dans différents environnements de travail en ligne comme hors ligne.

¹VS Code possède également une extension, Live Share, qui permet de travailler en temps réel sur un même document.

8.2 Références

9 La gestion des références

9.1 Pourquoi citer ?

La citation des sources joue un rôle essentiel dans le milieu académique, offrant de nombreux avantages. Tout d'abord, elle nous permet de nous insérer dans le contexte de la recherche existante. Chaque scientifique s'appuie sur les travaux précédents de ses pairs, en utilisant leurs découvertes comme point de départ et en engageant un dialogue continu. Référencer d'autres articles nous offre également la possibilité d'accéder à des informations pertinentes pour notre propre recherche. De plus, la transparence et la fiabilité de la science sont cruciales. La possibilité de vérifier les méthodes et les résultats d'une étude est indispensable. Si des erreurs sont découvertes dans la méthodologie ou l'analyse des résultats, d'autres scientifiques peuvent les corriger grâce aux références fournies. Enfin, la citation des sources est une démonstration de transparence et d'intégrité, renforçant la crédibilité de notre travail. Cette pratique favorise l'honnêteté intellectuelle en reconnaissant la contribution des autres scientifiques à notre propre travail scientifique.

9.2 À quoi sert un logiciel de gestion bibliographique ?

Un outil de référence bibliographique est un logiciel conçu pour aider les scientifiques à gérer et organiser les références bibliographiques de manière efficace. Ces outils sont particulièrement utiles lors de la rédaction d'articles

9 La gestion des références

de recherche, de thèses, de mémoires ou d'autres travaux académiques. Voici quelques-unes des fonctions principales d'un tel outil :

1. Collecte de références : Les outils de référence bibliographique permettent aux personnes utilisatrices de collecter et d'importer des références bibliographiques à partir de bases de données, de catalogues de bibliothèques, de sites Web ou d'autres sources. Certains outils offrent même la possibilité d'extraire automatiquement les métadonnées à partir de documents PDF.
2. Organisation et classement : Les références collectées peuvent être organisées en différentes catégories et dossiers. Cela facilite la recherche ultérieure et permet de garder une vue d'ensemble claire de la bibliographie.
3. Citation et génération de bibliographies : L'un des avantages majeurs des outils de référence est leur capacité à générer automatiquement des citations et des bibliographies conformes à différents styles de citation (APA, MLA, Chicago, etc.). Ce processus permet de gagner énormément de temps en formatage. Les personnes utilisatrices peuvent insérer des références directement dans leurs documents sans avoir à se soucier des détails de formatage.
4. Collaboration : Certains outils offrent la possibilité de collaborer en ligne, ce qui donne l'occasion à plusieurs personnes de travailler sur une bibliographie commune. Cela peut être utile pour les projets de groupe ou de recherche partagée comme c'est le cas dans une chaire de recherche. En plus d'utiliser un même logiciel, l'utilisation d'un outil de référencement contribue à économiser du temps par la centralisation des données sur un même interface.
5. Recherche et exploration : De nombreux outils de référence bibliographique offrent des fonctionnalités de recherche avancée qui facilitent la découverte de nouvelles références liées à un sujet spécifique.

9.2 À quoi sert un logiciel de gestion bibliographique ?

6. Synchronisation et sauvegarde : Les références et les bibliographies peuvent être synchronisées sur plusieurs appareils, ce offre la possibilité aux personnes utilisatrices d'accéder à leurs références où qu'elles soient. Les sauvegardes régulières assurent que les données ne soient pas perdues en cas de problème technique.
7. Suivi de lecture : Certains outils permettent aux personnes utilisatrices de suivre les articles et les documents qu'elles ont lus, ce qui est particulièrement utile pour garder une trace de la littérature pertinente.
8. Importation et exportation : Les outils de référence bibliographique autorisent généralement l'importation et l'exportation des références dans différents formats, ce qui facilite le transfert de données.

En résumé, un outil de référence bibliographique simplifie grandement le processus de gestion des références bibliographiques, de bonnes pratiques de formatage et de création de bibliographies cohérentes, ce qui permet aux scientifiques de se concentrer davantage sur le contenu de leurs travaux plutôt que sur les détails de formatage. D'ailleurs, il existe plusieurs outils de référence bibliographique, dont : Endnote, Zotero et Mendeley.

Bien que chaque logiciel de référence présente ses propres caractéristiques distinctes, il est indéniable qu'ils partagent des similitudes notables dans leur objectif principal. C'est-à-dire qu'ils permettent tous d'économiser du temps et de rendre le travail d'équipe plus facile en centralisant les références. Afin de choisir le logiciel qui répond aux besoins de la personne utilisatrice, celle-ci doit se demander s'il est nécessaire de partager les résultats de ses recherches ainsi que de travailler en collaboration. De ce fait, s'il est nécessaire de partager ses résultats avec le restant de son équipe, la personne utilisatrice devrait se munir du même logiciel que ses collègues. Enfin, il est surtout important d'utiliser le logiciel que la personne préfère. Bien que Zotero, EndNote et Mendeley partagent des similitudes fondamentales, chacun possède des fonctionnalités spécifiques pouvant ainsi mieux répondre aux besoins individuels. Dans ce paysage

d'options, l'élément crucial demeure l'adéquation entre les fonctionnalités offertes par le logiciel et les objectifs de l'utilisateur, tout en prenant en considération les aspects de partage, de collaboration et de convivialité.

9.3 Pourquoi Zotero?

L'avantage de Zotero est qu'il est gratuit et libre d'accès. Son code est ouvert à tous et son Github compte plus de 13,000 commits. Il offre une grande gamme de fonctionnalités ainsi que la possibilité d'y ajouter des extensions, complétant ainsi son utilisation. Zotero est puissant mais reste facile à utiliser. Il est distribué sur plusieurs plateformes (Windows, Mac, Linux, iOS, Android), permettant ainsi la collaboration entre tous les membres d'une équipe de recherche utilisant une diversité de plateforme. Il est possible de synchroniser sa bibliothèque Zotero sur plusieurs appareils, soit en utilisant le service cloud payant de Zotero ou en installant son propre espace de stockage infonuagique. Zotero s'intègre parfaitement dans un projet de recherche utilisant LaTeX ou Quarto puisqu'il est possible de générer des fichiers .bib à partir des bibliothèques et les maintenir à jour automatiquement. Il s'intègre aussi aux logiciels de traitement de texte comme LibreOffice et Microsoft Office. Il est possible de générer des bibliographies et des citations dans plus de 9000 styles de citation différents et peut donc convenir à tous.

Un autre grand avantage de Zotero est la centralisation des sources bibliographique et de leurs fichiers. Il est possible d'ajouter des PDF à Zotero et de les synchroniser dans des groupes de travail. Cela permet de partager des documents facilement avec les autres membres de l'équipe de recherche. Plus besoin de dossiers partagés ou d'envoyer des documents par courriel ou sur des plateformes de partage de fichiers. Tout est centralisé dans Zotero. Cette centralisation permet d'accomplir des recherches du type `ctrl+f` à travers l'ensemble des sources contenues dans une bibliothèque. Vous écrivez une conclusion à propos des radis finlandais et vous désirez discuter des enjeux internationaux liés à son agriculture en citant une

9.4 Pourquoi BibLaTeX?

source que vous vous rappelez avoir consultée il y a 3 ans? Pas de problème, Zotero vous permet de retrouver cette source en quelques secondes avec une simple recherche.

Le désavantage de Zotero est qu'il peine à gérer d'immenses bibliothèques comportant plusieurs milliers de fichiers. Il est d'ailleurs nécessaire de payer pour avoir un large espace de stockage. En outre, le logiciel n'est pas parfait. Il faut parfois ajouter manuellement des informations manquantes que le logiciel n'a pas repéré directement à l'aide du connecteur intégré.

9.4 Pourquoi BibLaTeX?

BibLaTeX est une extension moderne pour le traitement des bibliographies dans LaTeX et Quarto généralement utilisé avec Biber, un programme moderne de traitement des données bibliographiques pour BibLaTeX. Biber offre des fonctionnalités avancées, telles que le tri avancé, la prise en charge de multiples bibliographies et la capacité de traiter des sources de données bibliographiques dans différents formats. BibLaTeX prend en charge de nombreuses langues, ce qui le rend adapté pour la rédaction d'articles ou de livres destinés à des publics internationaux. Bien que BibLaTeX soit principalement un paquet pour LaTeX, il est possible d'exporter des bibliothèques depuis des outils comme Zotero sous forme de fichiers .bib, qui peuvent ensuite être utilisés avec BibLaTeX. Betterbib-tex peut tenir vos fichiers .bib à jour automatiquement à partir de Zotero. Conservez seulement les références que vous utilisez et organisez votre fichier en ordre alphabétique pour favoriser la coopération et le partage de source.

Voici un exemple d'utilisation de BibLaTeX en LaTeX.

```
\usepackage[style=apa,url=false,isbn=false,doi=false,backend=biber,language=english,autolang=  
\addbibresource{references.bib}  
\parencite[page]{citationkey}
```

```
\printbibliography{}
```

Les inconvénients potentiels de BibLaTeX incluent une courbe d'apprentissage pour ceux qui sont habitués à BibTeX, ainsi que la nécessité d'une mise à jour régulière pour rester compatible avec les versions récentes de LaTeX. De plus, certains éditeurs académiques ou revues ont leurs propres styles de citation et n'acceptent pas les soumissions utilisant BibLaTeX, bien que cela devienne de moins en moins courant.

En conclusion, pour ceux qui cherchent à maximiser la flexibilité et la puissance de leurs outils de gestion de bibliographie dans LaTeX, BibLaTeX, en tandem avec Biber, offre une solution moderne et robuste.

9.5 Installation et configuration de Zotero

Dans cette section, vous serez amené notamment à Installer Zotero ainsi que Better Bibtex. Better Bibtex est une extension de Zotero servant à générer ainsi qu'à maintenir à jour des fichiers .bib comptatible avec BibLaTeX, à partir de Zotero.

9.5.1 Zotero

- Installer Zotero
- Installer Zotero Connector
- Une fois Zotero installé, vous avez l'option de créer un compte Zotero. L'identifiant que vous utiliserez sera celui que vous partagerez à vos collaborateurs pour créer et joindre des groupes.

9.5.1.1 Better Bibtex

- La prochaine étape sera d'installer Better BibTex. Pour ce faire, allez dans l'onglet tools > Add-ons ensuite cliquez sur l'icône de paramètre et faites Install Add-on From File. Sélectionnez le fichier .xpi que vous avez téléchargé.

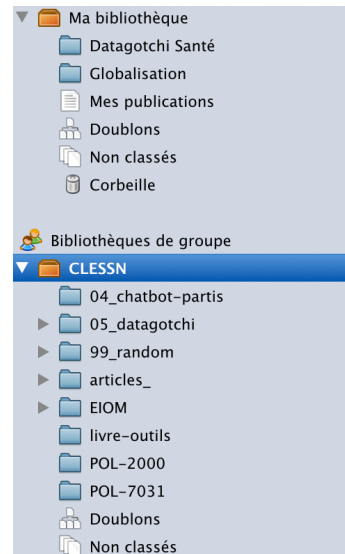
IMPORTANT

- Une fois l'add-on installé, allez dans les paramètres de Better Bibtex en allant dans l'onglet Zotero > Settings > Onglet Better Bibtex > Open Better Bibtex preferences...
- Il est important, lors de la collaboration, de s'assurer d'avoir les mêmes clés de citation que vos collègues. Betterbibtex peut s'assurer que vos clés respectent un format standard. Pour ce faire,
- Voici une suggestion de clé de citation. Il s'agit simplement du nom de l'auteur et de la date de publication en deux chiffres. Pour l'utiliser, collez ceci Dans la section Citation Key Format: `authEtal2.fold.lower.replace(find=".",replace=_) + len + shortyear | veryshorttitle + shortyear`
- Afin de vous assurer d'avoir les mêmes clé de citation, vous pouvez clic-droit sur vos références, allez dans les options Betterbibtex et cliquer sur "Refresh Citation Keys".

9.5.1.2 Génération du fichier .bib

Dans Zotero, vous devriez maintenant voir le groupe Zotero de votre équipe dans les Group Libraries.

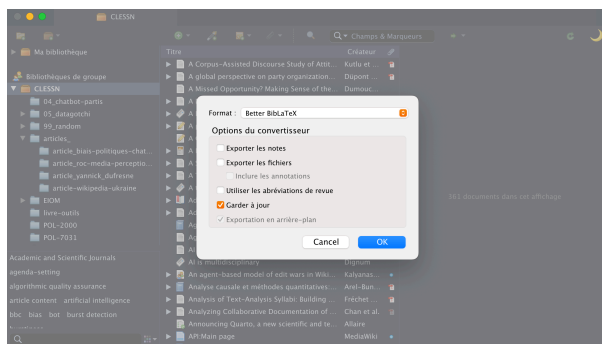
9 La gestion des références



Il est important de comprendre que tout changement que vous faites dans Zotero sera automatiquement synchronisé avec le groupe de votre équipe de travail. Ainsi, si vous supprimez une référence, elle sera supprimée pour tout le monde!

Clic-droit sur la collection livre-outils > Export Collection choisissez le format Better BibLaTeX et cochez la case ☒ Keep updated. Faites OK et sauvegardez le fichier dans le dossier .git du projet livre-outils. Ce dossier sera constamment mis à jour avec les changements que vous faites dans Zotero et sera synchronisé avec le projet Github quand vous ferez vos pull requests.

9.5 Installation et configuration de Zotero



9.5.1.3 Utilisation de Zotero lors de l'écriture

Lors de l'écriture, vous n'avez qu'à écrire @ dans votre éditeur pour faire sortir la palette de référencement.

9.5.1.4 Ajouter des références à Zotero

Il y a différentes façons d'ajouter des références à Zotero.

- Drag & drop à partir de votre librairie personnelle
- Drag & drop les pdf que vous avez sur votre ordinateur dans la collection livre-outils. Zotero va essayer de trouver les métadonnées automatiquement.
- Si il ne réussi pas, vous pourrez ajouter la références en cliquant sur la baguette magique en haut à gauche du symbole " + " vert. L'outil de baguette magique est utile si vous possédez le DOI ou le ISBN de l'article/livre que vous devez ajouter. Dans les rares cas où Zotero ne trouve rien à propos de votre référence, vous pourrez remplir les différents champs manuellement.
- Utiliser le connecteur à l'intérieur de votre fureteur. Zotero va aussi tenter de télécharger l'article directement et l'inclure dans la collection approprié.

9.6 Conclusion

En résumé, l'utilisation d'un outil de référencement bibliographique apporte son lot d'avantages. Que ce soit pour faciliter le travail de recherche en équipe, pour citer des études et faire une bibliographie plus rapidement ou pour éviter de perdre des sources intéressantes, nous vous encourageons fortement à prendre le temps d'apprendre à utiliser Zotero. D'ailleurs, même si vous ne faites pas encore de recherches exhaustives, Zotero peut être utilisé pour enregistrer les lectures de vos cours ou de vos séminaires. De cette manière, vous pratiquerez l'utilisation de cet outil et vous aurez en main les lectures que vous aurez trouvées intéressantes pour des travaux ultérieurs. Si vous voulez en apprendre encore davantage, il existe plusieurs tutoriels en ligne comprenant des fonctions plus poussées.

10 Une image vaut mille mots

Camille Tremblay-Antoine¹ Nadjim Fréchet²

10.1 Introduction

Une fois les données collectées, nettoyées, traitées et analysées, une partie centrale du travail d'un scientifique de données est de faire parler les résultats de ses tests empiriques. Il s'agit alors de trouver la meilleure manière de rendre l'information digeste pour les experts et initiés de votre discipline académique ou pour le grand public. La visualisation graphique des données est donc centrale afin de vulgariser les résultats d'une recherche empirique.

L'objectif de ce chapitre est d'apprendre aux codeurs débutants les rudiments de la visualisation graphique en R. Ce chapitre présentera plus particulièrement les packages R *ggplot2* et *dplyr* eux-mêmes téléchargeable à partir du package *tidyverse*. Si *dplyr* permet de préparer les données avant leur visualisation, *ggplot2* est un package dédié à la production de graphiques. Ce chapitre présente sa grammaire avec une série d'exemples (Wickham 2009; Wickham, Çetinkaya-Rundel, and Grolemund 2023).

Ce chapitre est plus technique que théorique et permet aux codeurs débutants d'en apprendre davantage sur la manière de construire des graphiques en R avec des données concrètes. La question centrale qui

¹Université Laval

²Université de Montréal

devrait vous guider lorsque vous créez des visualisations est la suivante: **Comment optimiser l'intelligibilité des données?** L'objectif d'un graphique n'est pas seulement d'illustrer les données. Un bon graphique devrait permettre de vulgariser une information ou de mettre en saillance un aspect particulier des données. L'objectif communicationnel devrait toujours être gardé en tête. Les graphiques en exemple dans ce chapitre sont construits avec les données de l'Étude Électorale Canadienne de 2019 qui sont facilement téléchargeables sur leur site³.

La première section de ce chapitre expose les options et packages également disponibles pour la construction de graphiques en R. La deuxième section de ce chapitre compare les avantages et inconvénients de l'utilisation de *ggplot2* par rapport aux autres packages de visualisation de données qui auront été présentés. La troisième section de ce chapitre montre des exemples de graphiques construits avec la grammaire de *ggplot2* en utilisant les données de l'Étude électorale canadienne de 2019. Les codes employés pour produire les graphiques en exemple sont disponibles dans l'annexe de ce livre. Ces codes reproductibles permettront aux codeurs débutants d'adapter ces derniers pour leurs propres projets.

10.2 Réflexion théorique

10.2.1 Les options disponibles

De nombreux *packages* ont été développés dans le langage R dans le but de visualiser des données graphiquement, il devient donc facile de s'y perdre. Heureusement, les options qui s'offrent à nous se précisent lorsque l'on s'intéresse à ce qui est le plus utilisé dans la communauté des codeurs de ce langage de programmation. Les *packages* les plus utilisés représentent des outils qui ont été substantiellement validés et améliorés par leurs développeurs, mais aussi par une importante communauté de codeurs en

³<http://www.ces-ec.ca/>

ligne et de chercheurs universitaires. Trois de ces options sont présentées dans ce chapitre: les graphiques du *Base R*, le *package Lattice* et le *package ggplot2*. Les avantages et inconvénients respectifs de ces trois approches pour la création de graphiques sont explicités dans les sections suivantes.

10.2.1.1 Avantages et inconvénients de Base R

Le *Base R* est le langage de base de R et il permet de faire de nombreuses manipulations statistiques sans avoir à installer de *packages* au préalable. Le *Base R* permet notamment de produire des graphiques rapidement. Cela peut être utile pour visualiser la distribution d'une variable ou pour regarder la relation entre deux d'entre elles, par exemple. Pour produire un graphique avec le langage de base R, il suffit de faire appel à la fonction `plot()`. Avec la fonction `plot()`, le codeur peut visualiser la distribution d'une variable seule en spécifiant l'axe des x dans cette dernière. Le codeur peut également visualiser la relation entre deux variables en spécifiant à l'intérieur de la fonction celles qui composeront les axes des x et des y du graphique. Les fonctions `barplot()`, `hist()` ou `boxplot()` disponibles dans le *Base R* permettent de spécifier le style de graphique souhaité, qu'on veuille représenter nos données sous forme de diagramme à barre, d'histogramme ou de diagramme en boîtes (Kabacoff 2022, 119–32).

```
# Exemple de graphique avec la fonction barplot() du BaseR

barplot(y, names.arg=x,
  main="Figure 1 - Proportion (%) de répondants par province\n",
  col = "blue",
  sub="\nSource: Étude Électorale Canadienne de 2019
```

Alors qu'un peu tout peut être fait avec le *Base R*, ce langage demeure élémentaire; il est difficile d'innover dans la visualisation ou même de produire des graphiques plus sophistiqués. Le *Base R* peut sembler plus simple pour l'exploration de données ou pour produire des graphiques de

base rapidement, mais ce langage devient rapidement complexe lorsqu'on cherche à améliorer l'esthétique de son graphique ou à visualiser des relations entre plusieurs variables, ce que *lattice* et *ggplot2* permettent plus facilement (Wickham 2009, 3–4).

10.2.1.2 Avantages et inconvénients de *lattice*

Développé par Deepayan Sarkar, *lattice* cherche à faciliter la visualisation de graphique en facettes. Plus précisément, ce *package* vise à améliorer les graphiques du *Base R* en fournissant de meilleures options de graphisme par défaut pour visualiser des relations multivariées. Ce *package* est donc intéressant pour les chercheurs et les codeurs voulant présenter graphiquement la relation entre plus de deux variables (Kabacoff 2022, 373–77; Sarkar 2008, 2023). Pour produire un graphique de base avec *Lattice*, le *package lattice* doit préalablement être installé dans la bibliothèque de *packages* du codeur et chargé dans sa session au début de son code (voir annexe). Par la suite, le codeur doit spécifier le type de graphique souhaité avec la fonction appropriée⁴. Une fois la fonction choisie, il doit spécifier par une formule les variables x et y ainsi que la troisième variable à contrôler et à visualiser en facettes (*graph_type(formula / variable en facettes, data=)*).

Si la Figure 1 produite à partir du *Base R* nous permet de visualiser le pourcentage de répondants par province dans l'Étude Électorale Canadienne de 2019, le *package lattice* nous permet de visualiser facilement ce même pourcentage de répondants en tenant compte du positionnement idéologique des Canadiens par province sur l'échelle gauche-droite, comme l'illustre la Figure 2 (0 étant la gauche et 10 la droite).

⁴Plusieurs options disponibles comme des histogrammes avec la fonction *histogram()* ou des graphiques de densité avec la fonction *densityplot()*.

10.2 Réflexion théorique

```
# Exemple de graphique avec la fonction histogram() du package lattice

histogram(~gaucheDroite | province, data = GraphiqueLattice, breaks = seq(0, 10,
  by = 1),
  main = "Figure 2 - Distribution des Canadiens\n par province sur l'échelle gauche-droit",
  xlab = "\nIdéologie gauche-droite",
  ylab = "Pourcentage (%)\n",
  col = "blue",
  sub="\nSource: Étude Électorale Canadienne de 2019
```

Cependant, le *package lattice* a pour désavantage d’avoir un modèle formel (une grammaire de graphique) moins compréhensible et intuitif que celui de *ggplot2* lorsque vient le temps d’améliorer l’esthétisme des graphiques. De plus, sa plus faible popularité fait en sorte que ce *package* demeure moins développé par la communauté de codeurs de R que ne l’est *ggplot2*. Nous examinons plus en détail la grammaire de graphique de ce dernier *package* ainsi que ses avantages et inconvénients dans la prochaine section (Kabacoff 2022, 373–77 et 390; Wickham 2009, 6).

10.2.1.3 Avantages et inconvénients de ggplot2

Développé principalement par Hadley Wickham, *ggplot2* est un *package R* faisant partie de la collection de *packages* de *tidyverse*. Ainsi, *Ggplot2* peut être utilisé avec les autres *packages* centraux de *tidyverse* ce qui limite de potentiels conflits entre les fonctions de *packages* qui puissent être incompatibles avec *ggplot2*. Par exemple, le *package dplyr* de *tidyverse* est très utile pour analyser, organiser et préparer vos données à visualiser avec *ggplot2* (Wickham et al. 2019; Wickham, Çetinkaya-Rundel, and Grolemund 2023, 30).

Le principal avantage de *ggplot2* reste sa grammaire qui permet à l’utilisateur de rendre ses graphiques beaucoup plus visuellement attrayants en facilitant la personnalisation esthétique. Ceci permet de

pousser l'esthétisme de vos graphiques à un très haut niveau par rapport aux autres *packages* de visualisation graphique disponibles en R. Les graphiques *ggplot2* se construisent couche par couche, soit par l'ajout des différents éléments du graphique au fur et à mesure dans le code du graphique à construire.

La première couche des graphiques *ggplot* est généralement celle des données et des variables à visualiser. Elle contient plusieurs éléments fondamentaux qui sont essentiels à chaque graphique. Le premier élément est la spécification de l'utilisation du *package ggplot2* qui se fait simplement en appelant la fonction *ggplot2()*. Dans cette fonction, il faut ensuite mentionner quelle est la base de données (*data=*) ainsi que la fonction qui sera utilisée pour positionner les données (*aes()*). Le positionnement le plus courant est de positionner des données *x* par rapport à des données *y*, ce qui se fait de la sorte: *aes(x=, y=)*.

La deuxième couche des graphiques *ggplot2* est celle du *geom*, qui spécifie le type de graphique souhaité. Les types de graphiques les plus couramment utilisés avec *ggplot2* sont les nuages de points (*geom_point()*), les diagrammes de lignes de tendances ou de séries chronologiques (*geom_line()*), les courbes de densité (*geom_density()*) ainsi que les graphiques à bandes (*geom_bar()*). Mais les possibilités sont infinies (ou presque!) avec *ggplot2* et bien plus de types de graphiques existent.

Les autres couches des graphiques *ggplot2* dépendent souvent du codeur et des étapes de construction de son graphique⁵ (Wickham 2009, 77 et 89-93). Le reste de ce chapitre présente la grammaire de *ggplot2* avec un exemple de construction de graphique à bande présenté couche par couche.

⁵Les étapes (couches) d'un graphique *ggplot2* ne sont pas nécessairement dans le même ordre d'un graphique à un autre.

10.3 Réflexion méthodologique

10.3.1 Comment utiliser ggplot2

```
# Première couche de l'exemple de graphique
# ggplot2 (base de données, variables et geom)

ggplot(data=GraphiqueExemple, aes(x=province, y=prop)) +
  geom_bar(stat="identity")
```

Tel que mentionné dans le dernier paragraphe, la première étape est de spécifier la base de données et les variables qu'on souhaite visualiser. Vous vous souviendrez qu'au début de la section, nous avons mentionné la collection *tidyverse*, et plus spécifiquement le *package dplyr* qui y est compris. Ce dernier a été utilisé pour nettoyer/calculer la proportion de répondants par province au préalable, ce qui nous permet de positionner directement la variable *prop* dans l'axe y.

10.3.2 Exemples et fonctionnalités

10.4 Trucs et astuces

10.5 Pour aller plus loin

10 Une image vaut mille mots

10.6 Références

11 À la quête de l'optimisation

Le monde de la recherche en sciences sociales numériques est en constante évolution, offrant de nouvelles opportunités mais aussi des défis uniques. Dans cette quête incessante pour optimiser notre efficacité et notre collaboration, l'utilisation des bons outils devient la clé de la réussite. Que vous soyez un chercheur en herbe ou un professionnel chevronné, la manière dont vous organisez vos méthodes de travail et gérez vos ressources peut déterminer la qualité et l'impact de vos résultats.

11.1 L'importance d'une méthode de travail efficace

Avant même de plonger dans les détails des méthodes de recherche et des analyses, il est crucial de poser les bases d'une méthode de travail efficace. Qu'il s'agisse de travailler en solitaire ou en équipe, l'ordre et la structure sont des éléments essentiels. Des dossiers bien organisés, une arborescence claire et un entreposage sécurisé deviennent les piliers sur lesquels repose votre productivité. Après tout, un environnement de travail organisé engendre des résultats ordonnés.

Ce chapitre vous emmènera à découvrir une gamme d'outils conçus pour répondre aux besoins spécifiques des chercheurs en sciences sociales numériques. Dans une quête pour maximiser votre temps, améliorer vos flux de travail et renforcer vos collaborations, nous explorerons trois types d'outils qui vous guideront dans cette quête d'optimisation :

11 À la quête de l'optimisation

1. **Logiciels de communication** : La communication transparente est le cœur d'une collaboration réussie. Nous explorerons des outils tels que Slack qui facilitent les échanges en temps réel, connectant les chercheurs, même à distance, pour un partage rapide d'idées et d'informations.
2. **Logiciels de gestion de versions décentralisé** : Nous plongerons dans le monde de Git et GitHub, des outils indispensables pour le suivi des versions et la collaboration efficace sur le code source.
3. **Outils d'entreposage de données** : Que vous traitiez des données sensibles ou non, la conservation sécurisée de vos informations est primordiale. Des plateformes telles que Dropbox et Amazon Web Services (AWS) offrent des espaces sécurisés pour entreposer et partager vos données avec votre équipe.

Chacun de ces outils est une pièce du puzzle, conçue pour vous aider à gagner du temps, à collaborer de manière plus fluide et à renforcer la qualité de votre recherche en sciences sociales numériques. Plongeons dans ces outils avec un désir commun d'optimisation et d'excellence dans notre travail.

11.2 Logiciel de gestion de communication (Slack)

Dans tout bon projet de recherche, la communication est primordiale. Que ce soit pour décrire les avancements, discuter des étapes à venir, entretenir un partenariat avec des partenaires ou simplement structurer ses pensées, la plateforme par laquelle vous communiquez vous accompagne à chacune des étapes du travail. Il est donc important de choisir un outil qui convient bien à vos projets et de prendre le temps de l'approprier et d'optimiser son utilisation.

Il y a tellement de plateformes différentes pour communiquer qu'il faut être prudent par rapport au nombre utilisé. Si vous ne faites pas un choix,

11.2 Logiciel de gestion de communication (Slack)

vous pouvez, sans vous en rendre compte, mêler Teams, courriels, Zoom et autres. Rapidement, vous perdez le contrôle de ce qui est dit. Nous vous proposons d’opter pour un logiciel de gestion de communication. Il existe plusieurs logiciels du genre, tels que Microsoft Teams, Slack, Google Workspace et Workplace. Toutes ces options peuvent vous permettre de collaborer efficacement en équipe. Dans le cadre de nos travaux, nous utilisons Slack. C’est donc principalement de cet outil que nous parlerons dans cette section, mais n’hésitez pas à vérifier quelle plateforme correspond le mieux à vos besoins.

11.2.1 Pourquoi utiliser une de ces plateformes

Peu importe votre niveau d’implication, la collaboration et la communication sont inévitables en recherche. La science n’est pas une discipline qui se développe en solitaire, elle nécessite des échanges et des débats. Les équipes de recherche sont souvent dispersées géographiquement. Même si vous travaillez actuellement seulement avec votre directeur, il est certain que plusieurs équipes de recherche dans votre département utilisent un tel logiciel. Un courriel peut faire l’affaire pour une discussion ponctuelle qui se règle rapidement. Cependant, dans une équipe de travail dynamique, où plusieurs membres participent à divers projets, les courriels deviennent rapidement chaotiques, il est difficile de retracer ce qui a été dit et de conserver les pièces jointes. Les discussions deviennent rapidement trop complexes pour le médium utilisé.

Les logiciels de gestion de communication ont été conçus spécifiquement pour répondre aux besoins des équipes collaboratives. Vous y trouverez leur facette la plus attrayante : une structure simple et adaptée. Les chaînes et fils de discussions permettent de garder des traces et de se retrouver facilement dans ce qui a été dit. Une autre force de ces logiciels est la centralisation des outils de travail. Sur Slack, comme sur Teams, vous pouvez faire des appels en visioconférence à l’endroit où vos conversations

11 À la quête de l'optimisation

écrites se trouvent. Il est aussi possible d'y télécharger l'application mobile, ce qui facilite l'accessibilité et la connexion des membres de l'équipe. Tout avoir structuré à son goût au même endroit et à portée de main, cela permet de structurer sa pensée plus efficacement, d'éviter les oublis et de réduire le stress.

11.2.2 Comment utiliser votre logiciel efficacement

Une fois que vous êtes convaincu d'aller de l'avant avec un de ces outils, vous devrez apprendre à bien vous en servir. Voici quelques trucs qui pourront vous aider à optimiser son utilisation. Les points ci-dessous font référence à Slack, mais peuvent très bien être adaptés à d'autres plateformes.

11.2.2.1 Structuration

Il est important de bien réfléchir à la structuration de vos chaînes. Si vous ne faites pas ce travail, les chaînes peuvent se multiplier rapidement et les conversations se mettent alors à s'entrecroiser, vous faisant ainsi perdre le fil. L'objectif de ces outils étant d'éviter ces problèmes, vous ne voulez pas perdre l'avantage comparatif que vous venez tout juste de gagner face aux courriels! La structuration des chaînes devrait être similaire à celle de votre équipe de recherche. Si vous utilisez Notion ou un autre logiciel du genre, la structure des deux outils devrait être la même. Nous vous proposons d'avoir une chaîne pour chacun des projets. Si le projet est trop gros et que la conversation devient chaotique, pensez à créer une sous-chaîne (un sous-projet) qui vous permettra d'aborder un sujet précis, sans mêler les discussions. Pour faciliter la structuration des chaînes, vous pouvez utiliser des préfixes, pour classer les chaînes par thème, ou autre typologie qui vous convient. Également, utilisez les Espaces d'équipe. Chaque équipe devrait avoir son propre espace, avec ses propres chaînes. Vous pouvez faire partie de plusieurs équipes et naviguer à travers les espaces. Si plusieurs équipes

11.2 Logiciel de gestion de communication (Slack)

partagent un même espace de travail, vous pourriez perdre le contrôle de sa structure.

11.2.2.2 Maintenance

Slack est un espace dynamique, tout comme votre équipe! La structure que vous avez choisie n'est pas permanente. Vous devriez rapidement vous questionner à savoir si elle convient toujours à vos activités. Votre espace d'équipe est comme votre réel lieu de travail, faites-y régulièrement le ménage pour vous assurer que tout est propre et en ordre. Archivez les chaînes qui ne sont plus pertinentes ou actives, puisque vous pourrez toujours les désarchiver quand cela sera nécessaire. Épinglez des messages importants et des documents utiles aux projets dans les chaînes appropriées. Faites le tour de ce qui est épinglé à l'occasion pour vérifier si c'est encore pertinent. Cela peut paraître énergivore, mais l'efficacité de votre travail d'équipe va en bénéficier. Également, rappelez aux membres de votre équipe d'utiliser les bonnes chaînes pour chacune des discussions. Il ne faut pas que les conversations se croisent à travers les chaînes. Chaque chaîne a son utilité et doit être utilisée en conséquence. Les appels d'équipe doivent aussi se faire dans les bonnes chaînes. Quand vous êtes en appel, utilisez le fil de discussion pour conserver des traces écrites des points abordés dans la réunion. Les fils de discussions sont en général un bon outil pour ne pas se perdre dans une discussion. Si l'usage des mauvaises chaînes est un problème récurrent, il est possible que la structure que vous employez est mal adaptée à vos travaux. Vous pouvez alors retourner à la planche à dessin. Assurez-vous que toute l'équipe comprenne bien comment utiliser Slack. Si ce n'est pas le cas, formez-les. Une structure adaptée et une équipe bien formée peuvent faire des miracles.

11 À la quête de l'optimisation

11.2.2.3 Collaboration

La grande majorité des conversations devraient se faire dans les chaînes. Les conversations privées ont leur utilité, vous vous en servirez. Il est parfois nécessaire d'avoir des discussions plus confidentielles et de parler rapidement à quelqu'un sur un sujet éphémère. Toutefois, par soucis de transparence et d'inclusion, toute discussion à propos d'un projet devrait se faire dans sa chaîne. Si vous jugez qu'un membre d'une chaîne ne devrait pas lire ce que vous avez à dire sur le projet, c'est qu'il ne devrait pas faire partie de la chaîne. Par rapport aux membres, trouvez le bon équilibre par rapport à qui devrait être dans quelle chaîne. L'objectif n'est pas d'exclure et de cacher du contenu, vous voulez une équipe transparente. Vous voulez que vos membres restent bien informés de l'avancement des projets sans les submerger d'information qui ne leur est pas utile. C'est à vous de trouver la formule gagnante. Invitez vos partenaires externes dans votre espace d'équipe. Créez des chaînes spécifiques aux partenaires pour que les conversations externes soient tout aussi organisées. N'invitez pas vos partenaires dans vos chaînes privées, question de confidentialité. Si un partenaire n'a pas l'habitude d'utiliser Slack ou l'outil que vous utilisez, proposez-lui de vous y joindre quand même. Moins vous utilisez les outils des autres, plus vous gardez centralisées vos communications et évitez de jongler avec plusieurs plateformes.

11.2.2.4 Optimisation personnelle

Une fois que la structure d'équipe est définie et que vos membres et vos partenaires sont à l'aise avec l'utilisation de la plateforme, il est temps d'organiser la structure de votre Slack personnel. Créez des sections pour trier les chaînes. La structure d'équipe est essentielle, mais une fois qu'elle est déterminée, chaque membre n'utilise pas forcément les chaînes de la même façon. Vous pouvez vous créer une section de favoris, ou encore différentes sections par rapport aux différents thèmes pour y faciliter la navigation. Également, ajustez vos paramètres de notifications. C'est à

11.2 Logiciel de gestion de communication (Slack)

vous de déterminer quelle chaînes méritent de produire des alertes, et à quels moments vous souhaitez les recevoir. Slack a plusieurs applications intégrées qui facilitent la compatibilité avec vos autres outils. Vous pouvez connecter votre calendrier, votre Notion et votre GitHub pour recevoir des alertes pertinentes. Allez explorer ces applications pour déterminer lesquelles vous conviennent.

Tel que mentionné précédemment, plusieurs logiciels peuvent convenir à vos besoins. Puisque nous utilisons Slack, voici quelques raisons qui pourraient vous convaincre d’opter pour cette option ou de vous en éloigner. Sachez que cette liste n’est pas du tout exhaustive, mais reflète simplement quelques-unes de nos observations par rapport à notre outil de travail.

- Avantages

L’utilisation de Slack est très intuitive. Nous l’utilisons régulièrement dans des cours, et les étudiants apprennent rapidement à l’utiliser. La distinction entre les chaînes publiques accessibles à tous les membres d’un espace d’équipe et les chaînes privées est claire et simple d’utilisation. Slack offre aussi une fonction de recherche, qui vous permet de retrouver des messages à travers les chaînes. L’intégration de applications qui font le pont avec d’autres outils est fort appréciée. Enfin, Slack est utilisé partout dans le monde par des équipes de toutes les tailles et dans tous les domaines. C’est un outil très présent en recherche académique qui facilite la collaboration et la multidisciplinarité. Les chances sont élevées que vos partenaires utilisent déjà l’outil, ou au minimum en aient déjà entendu parlé.

- Inconvénients

Si vous avez l’habitude d’utiliser les outils d’une suite, comme celles de Microsoft ou de Google, il est possible que vous trouviez l’intégration de ces outils à Slack moins pratique que si vous utilisiez les plateformes proposées par ces compagnies. Également, gardez en tête que la version gratuite de Slack a plusieurs limitations. Elle

11 À la quête de l'optimisation

implique notamment une limite de temps par rapport à l'archivage des messages, que vous ne pourrez pas retracer après 90 jours. Les coûts pour utiliser Slack à son plein potentiel peuvent être élevés, mais puisque ce genre d'outils est de plus en plus répandu, il est fort possible que son utilisation soit financée par votre département.

11.3 Logiciel de gestion de versions décentralisé

Lorsque l'on aborde le domaine de la recherche scientifique en sciences sociales numériques, la collaboration et la gestion efficace du code deviennent des éléments cruciaux pour progresser dans ses projets. Dans cette optique, les outils de gestion de versions décentralisés ont pris une place prépondérante. Parmi eux, Git et GitHub se démarquent tant par leur popularité que par leur efficacité.

11.3.1 Pourquoi choisir Git et GitHub?

11.3.1.1 Avantages

Git, développé par Linus Torvalds en 2005, s'est imposé comme le système de gestion de versions décentralisé de référence. Sa principale force réside dans sa capacité à suivre l'évolution d'un projet en enregistrant les modifications apportées au code source. Chaque modification est enregistrée sous forme de dépôts (*commits*), avec un message explicatif, permettant aux collaborateurs de comprendre facilement les évolutions du projet.

GitHub, lancé en 2008, est une plateforme qui utilise Git comme base pour l'entreposage et la gestion de projets. C'est une vitrine virtuelle où les développeurs peuvent héberger leurs dépôts Git et collaborer de manière transparente. L'aspect social de GitHub, avec ses fonctionnalités de suivi des projets, de gestion des problèmes et de demandes de fusion, en fait un lieu de choix pour les projets en code source ouvert et collaboratifs.

11.3 Logiciel de gestion de versions décentralisé

En sciences sociales numériques, où le partage et la collaboration sont essentiels, Git et GitHub offrent plusieurs avantages majeurs. Tout d’abord, ils permettent de suivre les modifications apportées au code, ce qui facilite la reproductibilité des résultats. Les chercheurs peuvent revenir à n’importe quelle version précédente du code, ce qui est particulièrement utile pour corriger des erreurs ou analyser l’impact de différentes approches.

De plus, Git et GitHub favorisent le travail collaboratif. Plusieurs chercheurs peuvent travailler sur le même projet simultanément, chacun dans sa branche de développement. Une fois les modifications effectuées, il est possible de fusionner les branches pour intégrer les changements. Cette approche évite les conflits majeurs et facilite la répartition des tâches au sein de l’équipe.

Enfin, l’aspect de code source ouvert de GitHub permet aux chercheurs en sciences sociales numériques de partager leurs codes avec la communauté académique et de bénéficier des contributions d’autres chercheurs. Cela favorise un environnement de partage des connaissances et de collaboration fructueuse.

11.3.1.2 Inconvénients

Cependant, Git et GitHub ne sont pas sans leurs défis. La courbe d’apprentissage peut être raide pour les débutants, car ces outils impliquent des concepts spécifiques tels que les branches, les conflits de fusion et les requêtes de tirage. De plus, bien que GitHub offre un niveau de gratuité pour les projets en code source ouvert, des frais peuvent être appliqués pour des fonctionnalités avancées ou pour des projets privés.

11.3.2 Comment les utiliser efficacement (en parallèle à Dropbox, etc.)

Pour utiliser Git et GitHub efficacement dans un contexte de recherche en sciences sociales numériques, il est recommandé de suivre quelques bonnes pratiques. Tout d'abord, il est important de structurer son dépôt Git de manière logique, en organisant les fichiers et les dossiers de manière cohérente. Les messages de commit doivent être descriptifs et clairs, pour permettre à tous les collaborateurs de comprendre les changements effectués.

Il est également conseillé de travailler sur des branches distinctes pour chaque fonctionnalité ou modification majeure. Cela facilite la gestion des changements et minimise les conflits lors de la fusion. Les chercheurs devraient également consulter régulièrement les projets et les problèmes sur GitHub pour encourager une communication ouverte et résoudre rapidement les problèmes.

L'utilisation de Git et de GitHub peut être complémentaire à d'autres outils d'entreposage, tels que Dropbox ou Google Drive. Ces derniers peuvent être utilisés pour entreposer des fichiers non liés au code, tels que des données brutes non sensibles ou des documents de recherche, tandis que Git et GitHub gèrent le code source et ses évolutions.

Bien qu'il existe plusieurs alternatives à l'utilisation combinée de Git et de GitHub sur le marché, ces deux plateformes liées continuent de dominer le domaine de la gestion de versions décentralisée. Parmi les alternatives notables, on peut citer Mercurial, Bitbucket, GitLab et SourceForge. Chacun de ces outils offre des fonctionnalités similaires à celles de Git et GitHub, mais il est important de comprendre pourquoi Git et GitHub restent les choix privilégiés pour les chercheurs en sciences sociales numériques.

11.3.3 Pourquoi prioriser Git et GitHub pour les chercheurs en sciences sociales

1. *Intégration et adoption répandue* : Git est devenu un standard de facto dans l'industrie du développement logiciel. Sa popularité et son adoption répandue signifient que de nombreuses ressources d'apprentissage, des tutoriels et des forums de support sont disponibles en ligne, ce qui facilite l'utilisation de cet outil pour les chercheurs en sciences sociales débutants. GitHub, en tant que plateforme principale de gestion des versions, bénéficie également d'une grande base d'utilisateurs et d'une communauté active, ce qui encourage la collaboration et le partage des connaissances.
2. *Facilité de collaboration* : Git et GitHub sont conçus pour faciliter la collaboration entre les individus et les équipes. Les chercheurs en sciences sociales travaillent souvent ensemble sur des projets de recherche, et la capacité de suivre les modifications, de gérer les conflits et de fusionner les contributions devient essentielle. L'interface conviviale de GitHub, avec des fonctionnalités telles que les demandes de fusion et les commentaires en ligne, simplifie grandement la collaboration.
3. *Visibilité et partage* : GitHub brille par sa fonctionnalité de projet open source, qui permet aux chercheurs en sciences sociales de partager leurs travaux avec la communauté mondiale. Les projets en code source ouvert sont visibles et accessibles à tous, favorisant ainsi la collaboration et l'examen par les pairs. Cela peut être particulièrement bénéfique pour les chercheurs souhaitant contribuer à des initiatives académiques et collaborer à des projets interdisciplinaires.
4. *Suivi des versions et recherche reproductible* : Les chercheurs en sciences sociales doivent s'assurer que leurs travaux sont reproductibles et vérifiables. Git permet de suivre les versions du code, ce qui signifie que les chercheurs peuvent retrouver facilement des versions antérieures pour reproduire des analyses spécifiques ou corriger des

11 À la quête de l'optimisation

erreurs. Cette fonctionnalité est cruciale pour maintenir l'intégrité des résultats de recherche.

5. *Infrastructure et sécurité* : GitHub offre une infrastructure robuste pour l'entreposage sécurisé des dépôts Git. Les chercheurs peuvent être assurés que leurs travaux sont sauvegardés et protégés contre les pertes de données accidentelles. De plus, les contrôles d'accès et les autorisations granulaires de GitHub permettent aux chercheurs de contrôler qui peut accéder et contribuer à leurs projets.

En somme, Git et GitHub offrent aux chercheurs en sciences sociales numériques un moyen puissant de gérer leur code, de collaborer efficacement et de contribuer à la communauté académique grâce à l'open source. Bien que leur apprentissage puisse représenter un défi initial, les avantages qu'ils apportent en termes de suivi des versions, de collaboration et de partage des connaissances en font des outils essentiels dans l'arsenal de tout chercheur moderne.

11.3.4 Pratiques à éviter sur GitHub pour les chercheurs en sciences sociales

Lorsque les chercheurs en sciences sociales utilisent GitHub pour partager leur code, collaborer sur des projets et contribuer à la communauté académique, il est essentiel de connaître les pratiques à éviter. En effet, certaines erreurs peuvent compromettre la sécurité, la confidentialité et l'efficacité de la recherche. Voici quelques éléments à éviter :

1. *Entreposer des informations sensibles* : Évitez d'entreposer des données sensibles ou confidentielles sur GitHub. Cela inclut les données de sondages, les informations personnelles identifiables et tout autre contenu pouvant porter atteinte à la vie privée des individus. Assurez-vous de supprimer ou de masquer soigneusement ces informations avant de les télécharger sur la plateforme.

11.3 Logiciel de gestion de versions décentralisé

2. *Inclure des mots de passe et clés d'accès* : Ne jamais inclure de mots de passe, de clés d'accès ou d'informations d'identification dans votre code source. Cela peut compromettre la sécurité de vos systèmes et de vos données. Utilisez plutôt des méthodes sécurisées pour gérer ces informations, telles que les variables d'environnement ou les fichiers de configuration externes.
3. *Entreposer des fichiers lourds* : Évitez d'entreposer des fichiers volumineux sur GitHub, notamment des fichiers binaires, des données brutes massives ou des ensembles de données volumineux. Ces fichiers peuvent ralentir les opérations de clonage et de fusion, ce qui affecte la performance globale du dépôt. Utilisez plutôt des services d'entreposage dédiés pour ces fichiers et fournissez des liens vers ces ressources dans votre dépôt.
4. *Inclure des identifiants personnels* : Évitez de publier vos propres identifiants personnels, tels que des numéros de sécurité sociale, des numéros de carte de crédit ou d'autres informations confidentielles. Ces informations pourraient être exploitées à des fins malveillantes si elles tombent entre de mauvaises mains.
5. *Ignorer les pratiques de branches et de fusion* : Évitez de fusionner directement du code dans la branche principale (habituellement appelée *main* ou *master*). Utilisez plutôt des branches distinctes pour les fonctionnalités et les corrections, et suivez les pratiques de fusion pour intégrer proprement les changements. Ignorer ces pratiques peut entraîner des conflits et une perte de trace des modifications.
6. *Ignorer les commentaires des collaborateurs* : Lorsque vous travaillez avec d'autres chercheurs, ne négligez pas les commentaires et les suggestions qu'ils fournissent. Les retours d'expérience et les idées des autres peuvent contribuer à améliorer la qualité de votre code et de vos analyses.
7. *Ne pas documenter* : Évitez de ne pas documenter votre code. Une documentation claire et détaillée est essentielle pour permettre à

11 À la quête de l'optimisation

d'autres chercheurs de comprendre vos méthodes et vos résultats. Utilisez des commentaires explicatifs et fournissez des explications sur la manière d'exécuter votre code.

En suivant ces conseils et en évitant ces erreurs courantes, les chercheurs en sciences sociales peuvent garantir la sécurité, la qualité et l'efficacité de leurs projets sur GitHub. La responsabilité de préserver la confidentialité des données et de créer un environnement de travail collaboratif et respectueux repose sur les épaules de chaque contributeur.

11.3.5 Exemple d'utilisation de Git et de GitHub pour un chercheur en sciences sociales

Dans le contexte de la recherche en sciences sociales numériques, la gestion efficace du code, la collaboration transparente et la préservation des données sensibles sont des impératifs. Imaginons que vous êtes un jeune chercheur en sciences sociales qui étudie l'impact des médias sur l'opinion publique. Vous utilisez le langage de programmation R pour analyser des données de médias et des données de sondage. Bien que vous travailliez seul, vous souhaitez rendre votre travail accessible à votre équipe pour validation et permettre à vos collègues de contribuer aux améliorations. Voici comment vous pouvez utiliser Git et GitHub pour gérer votre projet de manière structurée et collaborative.

11.3.5.1 Étape 1 : Création d'un répertoire local et initialisation de Git

Ouvrez votre terminal et naviguez vers le dossier où vous souhaitez enregistrer votre projet.

```
cd chemin/vers/votre/dossier
```

Créez un nouveau répertoire pour votre projet et accédez-y.

11.3 Logiciel de gestion de versions décentralisé

```
mkdir mon_projet
```

```
cd mon_projet
```

Initialisez Git dans ce répertoire.

```
git init
```

11.3.5.2 Étape 2 : Ajout de votre code et de vos fichiers

Ajoutez vos fichiers R contenant le code pour l'analyse des médias et des sondages dans le répertoire. Par exemple, vous pouvez avoir des fichiers *analyse_medias.R* et *analyse_sondages.R*.

Utilisez la commande `git status` pour vérifier l'état de vos fichiers.

```
git status
```

11.3.5.3 Étape 3 : Ajout, validation et commit de vos modifications

Ajoutez vos fichiers pour qu'ils soient prêts à être validés.

```
git add -A
```

Validez vos modifications avec un message descriptif.

```
git commit -m "Ajout du code d'analyse des médias et des sondages"
```

11 À la quête de l'optimisation

11.3.5.4 Étape 4 : Création du répertoire sur GitHub et du lien avec votre répertoire local

Allez sur GitHub et connectez-vous à votre compte. Créez un nouveau répertoire vide avec le nom *mon_projet*.

De retour dans votre terminal, ajoutez le lien GitHub à votre répertoire local.

```
git remote add origin https://github.com/votre-utilisateur/mon_projet.git
```

11.3.5.5 Étape 5 : Push de votre travail sur GitHub

Envoyez vos commits locaux vers GitHub.

```
git push -u origin master
```

11.3.5.6 Étape 6 : Collaboration avec vos collègues

Si vos collègues souhaitent contribuer à votre projet, ils peuvent *forker* votre répertoire sur GitHub, ce qui créera une copie dans leur propre compte.

Lorsqu'ils ont fait des modifications dans leur copie, ils peuvent soumettre une *pull request* pour vous demander de fusionner leurs modifications dans votre répertoire principal.

11.3.5.7 Étape 7 : Pull des modifications de vos collègues

Lorsque vos collègues ont soumis des modifications et vous ont demandé de les fusionner, vous pouvez mettre à jour votre répertoire local avec leurs changements.

```
git pull origin master
```

11.3.5.8 Étape 8 : Répéter le processus

Répétez les étapes 2 à 7 au fur et à mesure que vous développez votre projet, ajoutez du code, effectuez des analyses et collaborez avec vos collègues. Assurez-vous de valider et de pousser régulièrement vos modifications pour maintenir le dépôt à jour.

11.3.6 GitHub Desktop

Alors que le terminal reste une approche fondamentale pour maîtriser Git et GitHub, il existe des outils conviviaux tels que GitHub Desktop qui offrent une alternative intuitive. Cet outil simplifie le processus de gestion de versions décentralisée, en particulier pour ceux qui souhaitent commencer par une approche visuelle. Cependant, comprendre son fonctionnement et équilibrer les avantages et les inconvénients est essentiel.

GitHub Desktop fournit une vue claire de vos dépôts, de vos modifications, de vos branches et de vos demandes de fusion. Il élimine la nécessité de mémoriser les commandes en ligne de terminal, ce qui peut être un défi pour certains chercheurs. L'application simplifie également la résolution des conflits lors de la fusion des branches.

Toutefois, en utilisant GitHub Desktop, il est possible de perdre la compréhension des commandes Git en ligne de commande, ce qui pourrait devenir un inconvénient si vous devez travailler dans un environnement sans interface visuelle. De plus, GitHub Desktop est spécifiquement conçu pour interagir avec GitHub. Si vous devez travailler avec d'autres plateformes de gestion de versions, cela pourrait poser des problèmes.

La décision entre l'utilisation du terminal et de GitHub Desktop dépend de vos préférences et de vos besoins. Pour les chercheurs qui débutent,

11 À la quête de l'optimisation

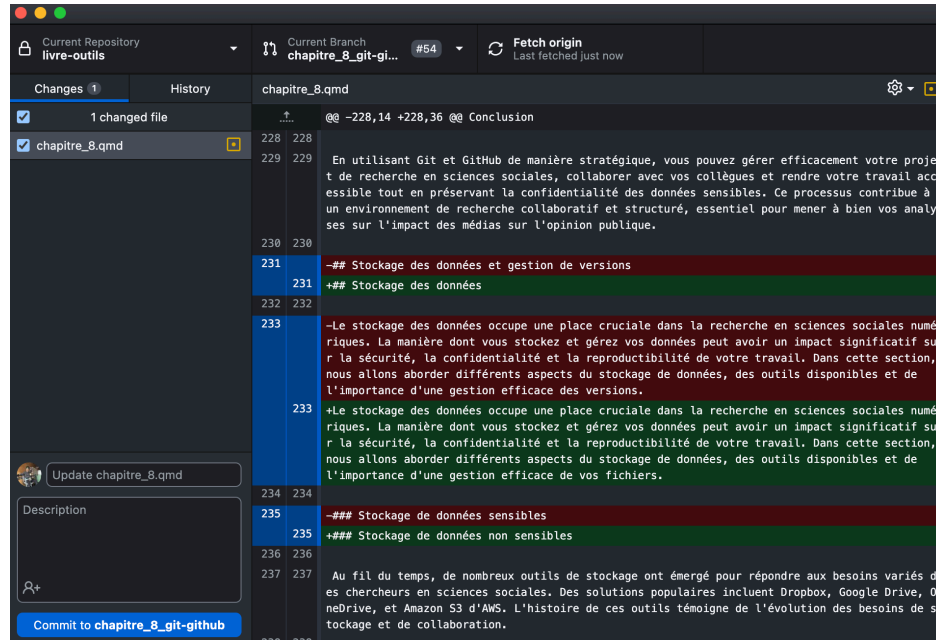


Figure 11.1: image

GitHub Desktop offre une transition en douceur vers les concepts de gestion de versions. Cependant, il est important de ne pas se limiter à une interface visuelle. Comprendre les commandes Git en ligne de commande reste essentiel pour résoudre des problèmes complexes, gérer des projets avancés et collaborer avec d'autres chercheurs qui utilisent des approches basées sur le terminal.

11.4 Conclusion

En utilisant Git et GitHub de manière stratégique, vous pouvez gérer efficacement votre projet de recherche en sciences sociales, collaborer avec vos collègues et rendre votre travail accessible tout en préservant la confidentialité des données sensibles. Ce processus contribue à un environnement de recherche collaboratif et structuré, essentiel pour mener à bien vos analyses sur l'impact des médias sur l'opinion publique.

11.5 Outils d'entreposage des données

L'entreposage des données occupe une place cruciale dans la recherche en sciences sociales numériques. La manière dont vous entreposez et gérez vos données peut avoir un impact significatif sur la sécurité, la confidentialité et la reproductibilité de votre travail. Dans cette section, nous allons aborder différents aspects de l'entreposage de données, des outils disponibles et de l'importance d'une gestion efficace de vos fichiers.

11.5.1 Entreposage de données non sensibles

Au fil du temps, de nombreux outils d'entreposage ont émergé pour répondre aux besoins variés des chercheurs en sciences sociales. Des solutions populaires incluent Dropbox, Google Drive, OneDrive et Amazon

11 À la quête de l'optimisation

S3 d’AWS. L’histoire de ces outils témoigne de l’évolution des besoins d’entreposage et de collaboration.

Lorsqu’il s’agit d’entreposer vos données de recherche, la règle d’or est de ne jamais perdre d’informations précieuses. Cette préoccupation prend toute son importance lorsqu’un chercheur en sciences sociales, seul ou en équipe restreinte, se lance dans un projet. Pour répondre à ce besoin, les services d’entreposage cloud tels que Dropbox, Google Drive et OneDrive se révèlent indispensables. Voici quelques avantages d’un entreposage sur le cloud pour la recherche :

1. *Sauvegarde automatique* : Les solutions cloud sauvegardent automatiquement vos fichiers, garantissant que vous ne perdrez jamais vos données en cas de panne d’ordinateur ou d’accident.
2. *Accessibilité universelle* : Vous pouvez accéder à vos fichiers à partir de n’importe quel appareil avec une connexion Internet, ce qui favorise la flexibilité dans la gestion de vos projets.
3. *Partage facilité* : Les services cloud permettent de partager facilement des fichiers et des dossiers avec des collègues, même en dehors de votre équipe de recherche. Cela favorise la collaboration et la communication.

Il est important de noter que le choix d’un service cloud dépend de vos besoins et de vos préférences. Considérez des facteurs tels que la capacité d’entreposage, les fonctionnalités de partage, la convivialité et la compatibilité avec vos outils de recherche existants.

Dropbox est connu pour sa simplicité d’utilisation et sa convivialité. Il peut être un choix approprié pour entreposer des fichiers non sensibles, partager des documents avec des collègues et faciliter la collaboration.

Pour utiliser Dropbox efficacement, organisez vos fichiers en arborescence logique. Créez des dossiers spécifiques pour chaque projet et partagez-les avec les membres de votre équipe. Pour éviter de pousser des fichiers

11.5 Outils d'entreposage des données

sensibles sur GitHub, ajoutez le nom de dossier à exclure dans un fichier *.gitignore*.

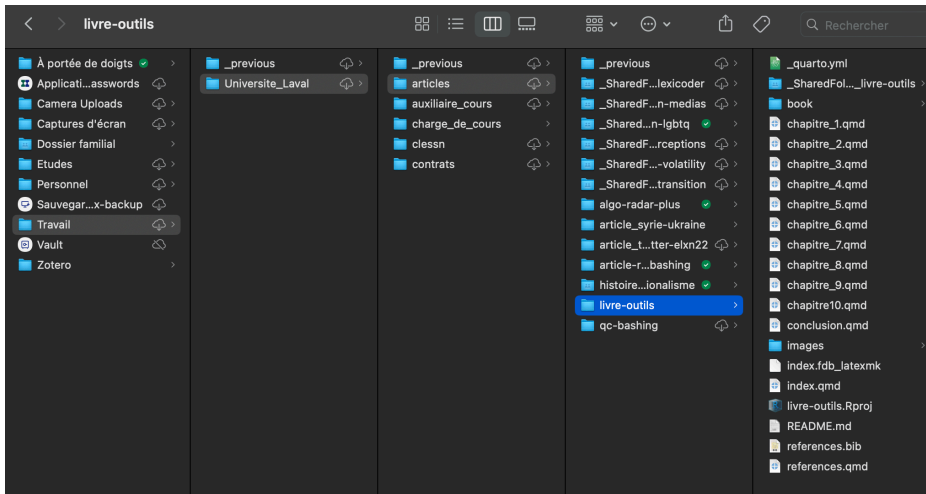


Figure 11.2: image1

Dropbox offre un suivi automatique des modifications, ce qui vous permet de remonter dans le temps pour restaurer des versions antérieures de vos fichiers. Cela garantit l'intégrité de vos données et vous permet de revenir à des versions précédentes si nécessaire. De plus, l'archivage de dossiers et de projets complets peut aider à conserver une vue chronologique de votre travail au fil du temps.

Il est également crucial de considérer la taille de vos données. Si vous traitez des fichiers volumineux tels que des images, des vidéos ou des ensembles de données massifs, il peut être judicieux d'utiliser un service cloud pour entreposer ces fichiers et les partager avec vos collaborateurs, plutôt que de les pousser sur des plateformes de gestion de versions comme GitHub.

Pour les données sensibles, les services cloud tels que Dropbox et Google Drive peuvent ne pas être suffisamment sécurisés. C'est là que des solu-

11 À la quête de l'optimisation

tions comme AWS entrent en jeu. Cependant, il est important de noter que l'utilisation d'AWS peut s'avérer complexe, en particulier pour un jeune chercheur travaillant en solo ou en petite équipe.

11.5.2 Entreposage de données sensibles

Lorsqu'il s'agit d'entreposer des données sensibles, tels que des données de sondage comportant des informations personnelles identifiables, la sécurité et la confidentialité sont essentielles. Comme abordé précédemment, GitHub n'est pas adapté à l'entreposage de telles données en raison de ses caractéristiques publiques et de son orientation vers le code source ouvert. Une solution courante est d'utiliser des services de cloud sécurisés, tels qu'AWS, qui offrent des mesures de sécurité robustes pour protéger vos données sensibles.

AWS regroupe un ensemble de services *cloud* proposés par Amazon. Il offre une vaste gamme de services, allant de l'entreposage et de la gestion des données à la computation et à l'analyse avancée. AWS est conçu pour offrir une infrastructure hautement évolutive et sécurisée, ce qui en fait un choix attrayant pour les chercheurs qui gèrent des données sensibles. L'outil présente de multiples avantages:

1. *Sécurité robuste* : AWS met l'accent sur la sécurité, avec des fonctionnalités telles que le chiffrement des données en transit et au repos, la gestion des accès basée sur les rôles et la conformité à des normes de sécurité strictes.
2. *Scalabilité* : AWS permet de faire évoluer vos ressources en fonction des besoins, garantissant des performances optimales même lorsque vos projets de recherche croissent en taille et en complexité.
3. *Flexibilité* : AWS propose une variété de services adaptés à différentes utilisations, allant de l'entreposage de données au calcul intensif pour l'analyse avancée.

11.5 Outils d'entreposage des données

4. *Collaboration simplifiée* : Bien que le coût d'entrée soit généralement bas, la possibilité de partager des ressources avec des collègues et de travailler en équipe rend AWS adapté à la collaboration.

AWS n'est pas le seul service cloud disponible. Microsoft Azure et Google Cloud Platform (GCP) sont des concurrents majeurs offrant des fonctionnalités similaires. Lorsque vous choisissez un fournisseur, prenez en compte les coûts, la convivialité et les fonctionnalités offertes. Le coût d'utilisation d'AWS peut varier en fonction des services utilisés, de la quantité de données entreposées et de la capacité de calcul requise. Lorsque vous travaillez seul, le coût peut sembler élevé par rapport à l'utilisation de solutions gratuites telles que Dropbox. Cependant, en équipe, la répartition des coûts peut rendre AWS plus abordable.

11.5.2.1 Exemple d'utilisation d'AWS pour entreposer et accéder à des données de sondages dans RStudio

Imaginez un jeune chercheur en sciences sociales qui travaille sur une analyse comparative de données de sondages recueillies sur plusieurs décennies. Pour maintenir la sécurité des données sensibles et faciliter l'accès pour les analyses dans RStudio, il décide d'utiliser AWS pour l'entreposage et la gestion de ses données.

11.5.2.1.1 Étape 1 : Création d'un compte AWS et configuration

Le chercheur crée un compte AWS et configure ses paramètres de sécurité, y compris la configuration de l'authentification à deux facteurs pour renforcer la sécurité de son compte.

11.5.2.1.2 Étape 2 : Création d'un espace d'entreposage S3

Le chercheur crée un compartiment Amazon S3 (Simple Storage Service) pour entreposer ses données de sondage. Il choisit une région AWS et

11 À la quête de l'optimisation

définit les paramètres de sécurité appropriés, tels que le chiffrement des données.

11.5.2.1.3 Étape 3 : Transfert des données vers Amazon S3

Le chercheur transfère les données de sondage dans son compartiment Amazon S3 à l'aide de l'interface en ligne AWS ou d'outils d'importation.

11.5.2.1.4 Étape 4 : Configuration des autorisations

Pour sécuriser davantage les données, le chercheur configure les autorisations d'accès aux données dans Amazon S3. Il attribue des rôles et des politiques d'accès spécifiques aux utilisateurs, garantissant que seules les personnes autorisées peuvent accéder aux données.

11.5.2.1.5 Étape 5 : Configuration d'accès dans RStudio

Le chercheur installe le package *aws.s3* dans RStudio pour accéder à ses données entreposées dans Amazon S3. Il configure également les informations d'identification AWS dans son environnement RStudio.

11.5.2.1.6 Étape 6 : Accès et analyse des données dans RStudio

À l'aide du package *aws.s3*, le chercheur peut maintenant accéder à ses données directement dans RStudio par quelques lignes de code. Il peut charger les données dans des structures de données R et effectuer des analyses statistiques, des visualisations et des croisements.

11.5.2.1.7 Étape 7 : Sécurité et conservation des données

Après avoir effectué ses analyses, le chercheur peut choisir de conserver les données de sondage dans Amazon S3 en utilisant les politiques de conservation appropriées. Il peut également archiver des copies de sauvegarde pour garantir l'intégrité des données à long terme.

Dropbox se concentre principalement sur l'entreposage et la collaboration de fichiers, alors que AWS offre une gamme de services *cloud*, y compris l'entreposage sécurisé de données sensibles et la mise en place d'infrastructures évolutives. GitHub, d'autre part, se concentre sur la gestion de versions et la collaboration de code source. Chaque outil a son propre domaine d'expertise et peut être utilisé de manière complémentaire pour différents aspects de la recherche.

11.5.3 Conclusion

L'entreposage des données est une étape cruciale dans la recherche en sciences sociales numériques. Choisissez des outils adaptés à la sensibilité des données, privilégiez des services sécurisés comme AWS pour les données sensibles, et utilisez Dropbox pour la collaboration et l'entreposage de fichiers non sensibles. Une gestion efficace des versions, de la structure des dossiers et de la sécurité garantira l'intégrité de vos données et facilitera la collaboration tout au long de vos projets de recherche.

12 Outils d'intelligence artificielle

Ce chapitre vise à initier les lecteurs à ce qu'est l'intelligence artificielle (IA), aux enjeux qu'elle engendre ainsi qu'à son potentiel pour les sciences sociales. Nous souhaitons d'abord et avant tout mener les lecteurs vers la réflexion. Dans ces pages, il y a beaucoup plus de questions que de réponses. Cet état de fait reflète bien l'état de la connaissance que nous avons sur l'IA. Nous sommes aussi bien loin d'être des spécialistes en la matière. Malgré cela, nous pensons qu'initier la réflexion et la discussion s'impose. Se poser des questions et tenter de trouver des réponses ne peut que générer des bénéfices. Par conséquent, si ce chapitre aura permis d'éclairer certains enjeux, ou s'il stimulera davantage de questionnement, alors nous aurons réussi notre objectif.

Nous débutons par définir et expliquer ce qu'est l'intelligence artificielle. Bien que ça ne soit pas une tâche facile, nous souhaitons simplement donner une idée générale de ce qu'il s'agit. Ensuite, nous souhaitons mettre l'accent sur l'évolution constante de ce champ de recherche, ainsi que les défis que ça pose. La section suivante se penche sur certains enjeux éthiques liés à l'utilisation de l'IA, notamment en ce qui concerne le plagiat. Cela nous mène à se questionner sur la place du chercheur, aujourd'hui et dans le futur, avec l'arrivée de ces nouvelles technologies. La dernière section se penchera sur la place de cette technologie en sciences sociales. Quel usage pourrait-on en faire, et surtout quels en sont les limites.

12.1 Définition et différents type d'IA

Qu'est-ce que l'intelligence artificielle (IA)? Est-ce quelque chose d'homogène, ou s'agit-il plutôt « des intelligences artificielles »? Dans un premier temps, afin de répondre à ces questions, il est important de préciser que l'intelligence artificielle est un champ d'études (Devedzic 2022). Par conséquent, il s'agit d'un ensemble d'objets, relativement vaste et en constante expansion, qui s'intéresse, à sa façon, à l'intelligence artificielle. Pour préciser ce propos, prenons l'exemple de la science politique. Malgré la formulation au singulier, la science politique est un grand ensemble de différents sous-champs d'études, qui ont chacun leur propre objet d'intérêt. La philosophie politique, les relations internationales, la politique comparée et l'étude de l'opinion publique, par exemple, sont tous des sous-champs qui s'intéressent, à leur façon, au phénomène politique. Dans le même sens, et compte tenu de cette pluralité de perspectives, il est important de noter qu'il n'y a pas de consensus dans la définition de l'IA (Wang 2019 ; König et al. 2022). De plus, la rapidité du développement de ce champ rend le traçage de frontières définitionnelles plutôt difficile : comment définir, d'une manière précise et consensuelle, quelque chose qui évolue constamment (Devedzic 2022 ; Bertolini 2020, 15)?

Malgré ces différents enjeux, il est tout de même possible de spécifier ce que nous entendons par intelligence artificielle. Deux définitions de l'IA nous seront utiles. La première vient de John McCarthy (2007, 2) : « Il s'agit de la science et de l'ingénierie qui consistent à créer des machines intelligentes¹, en particulier des programmes informatiques intelligents. Elle est liée à la tâche similaire consistant à utiliser des ordinateurs pour comprendre l'intelligence humaine, mais l'IA ne doit pas se limiter aux méthodes qui sont biologiquement observables. » [Traduction DeepL] La seconde définition provient de la compagnie IBM (2023) : « Dans sa forme la plus simple, l'intelligence artificielle est un domaine qui combine l'informatique

¹Allo

12.2 Utilisation du package OpenAI

et des ensembles de données robustes pour permettre la résolution de problèmes. Elle englobe également les sous-domaines de l'apprentissage automatique et de l'apprentissage profond, qui sont souvent mentionnés en conjonction avec l'intelligence artificielle. Ces disciplines sont composées d'algorithmes d'IA qui cherchent à créer des systèmes experts qui font des prédictions ou des classifications basées sur des données d'entrée. » [Traduction DeepL] À l'aide de ces extraits, on comprend que l'IA consiste à reproduire artificiellement certaines capacités cognitives humaines, afin de rendre les machines « intelligentes ». En d'autres termes, de leur donner la capacité de résoudre des problèmes par elles-mêmes.

12.2 Utilisation du package OpenAI

12.2.1 Installation et chargement du package

```
install.packages("openai") ## au besoin  
library(openai)
```

12.3 Configuration de l'API

Procurez vous une clé API sur le site d'OpenAI. Soyez conscient que vous aurez besoin d'une carte de crédit pour vous inscrire et que l'utilisation de l'API est payante. Renseignez-vous sur les modèles disponibles et leurs frais d'utilisation. En date de la publication du livre, le modèle de tarification d'OpenAI est de charger un prix spécifique par 1000 tokens. Le prix des Tokens en entrée est moins élevé que celui des tokens en sortie.

Lorsque vous aurez votre clé API, utilisez le package `usethis` pour la configurer dans votre environnement R.

```
install.packages("usethis") ## au besoin
usethis::edit_r_environ()
```

Ajoutez la ligne suivante à votre fichier `.Renviron`.

```
OPENAI_API_KEY=inserez-votre-cle-api-ici-sans-guillemets
```

12.4 Utilisation de l'API

La fonction principale du package `openai` est `create_chat_completion()`. Elle prend en entrée le modèle que vous souhaitez utiliser ainsi que le message que vous souhaitez envoyer au modèle en format `list`. Voici un modèle d'utilisation de la fonction:

```
chat_prompt <- create_chat_completion(
  model = "gpt-3.5-turbo",
  messages = list(
    list(
      "role" = "system",
      "content" = "You are a helpful assistant."
    ),
    list(
      "role" = "user",
      "content" = "Please do the following:"
    )
  )
)
```

Le résultat de votre requête sera contenu dans l'objet `chat_prompt` formaté en JSON. Vous pouvez accéder aux variables de la même façon qu'un dataframe normal. Le contenu de la réponse sera dans `chat_prompt$choices$content`.

Utiliser chatgpt de cette façon ouvre plein de possibilités. Appliquer des instructions sur un ensemble d'observations à l'aide de boucles, utiliser des fonctions pour générer des messages et les appliquer à travers d'autres API, analyser des sites webs en temps réel en scraping avec des paquets tels rvest, etc. Ce sera à vous de réfléchir aux possibilités que vous souhaitez explorer.

12.5 Notes

- Il est possible d'accéder aux statistiques d'utilisation de token dans `chat_prompt$usage$prompt_tokens` et `chat_prompt$usage$completion_tokens`. Vous pouvez donc calculer le coût de votre requête en fonction du modèle que vous utilisez.
- Ne pas oublier d'inclure `.Renviron` dans votre `gitignore` pour ne pas vous faire voler votre clé API.
- Il est possible de créer des images avec la fonction `create_image("Inserez votre texte ici")`
- Il est possible d'effectuer du speech-to-text avec la fonction `create_transcription()` et `create_translation()`
- Plus de documentation est disponible au <https://irudnyts.github.io/openai/>
- Plus de fonctionnalités sont disponibles en python mais le package R est suffisant pour la plupart des utilisations.

13 Serpents et échelles

Marc-Antoine Rancourt, Flavie Lachance, Justine Béchar, William Poirier

13.1 Introduction

Alors que les chapitres précédents se sont consacrés à la présentation théorique et pratique des sciences sociales numériques, le présent chapitre s'efforcera à aider le lecteur à faire sens de la grande quantité d'information que contient l'ouvrage et à commencer sa propre aventure numérique. Ce chapitre peut être vu comme le résultat de la rencontre entre un jeu éducatif et un guide d'apprentissage. À ce propos, le titre n'est pas anodin. Il est possible de visualiser l'apprentissage de nouveaux outils numériques à l'aide d'un jeu de serpents et échelles. Généralement, la progression du protagoniste se fait en toute tranquillité. À certains moments, l'aventure est corsée par des pièges ou des difficultés propres à sa progression – les serpents. Lorsque cela arrive, le protagoniste régresse ou cesse d'avancer. À d'autres moments, l'aventure est facilitée par des occasions positives particulières – les échelles.

En plus d'une représentation visuelle du tableau d'apprentissage de serpents et échelles qui permettra au lecteur de rendre l'acquisition de nouvelles connaissances plus agréable, ce chapitre offre des lectures, des exercices et des travaux pratiques pour les aventuriers numériques débutants, intermédiaires et avancés. Le corps du texte du présent chapitre est divisé en trois parties en lien avec les niveaux de connaissances des outils

13 Serpents et échelles

numériques – débutant, intermédiaire et avancé – et chacune d’entre elles comportent une première section concernant sur des lectures, une seconde portant sur des tutoriels à faire en ligne et une troisième sur des travaux pratiques à réaliser dans RStudio. De plus, chaque partie termine par un résumé des principaux « serpents » associés au niveau d’apprentissage qui lui est propre. Le chapitre se termine par une section qui aidera le lecteur à voler de ses propres ailes.

13.1.1 Datacamp

Tout au long du parcours, des exercices Datacamp vous seront proposés. Datacamp est une plateforme d’apprentissage en ligne proposant des exercices interactifs en ligne variés, axé principalement sur l’analyse et les données.

Utiliser Datacamp comporte de nombreux avantages. Premièrement, la plateforme est également facile à utiliser et permet un apprentissage ludique. La plateforme a également un forum où il est possible d’interagir avec les autres utilisateurs et poser des questions. Un autre avantage de Datacamp est son accessibilité. En plus des cours accessibles gratuitement, Datacamp offre des réductions sur les abonnements pour les étudiants et les enseignants. Il est également possible pour les enseignants d’obtenir gratuitement un compte Datacamp Entreprise à des fins d’utilisation éducative.

Toutefois, Datacamp comporte certains désavantages. En effet, la plateforme demeure un jeu et, mis à part le forum et les indices fournis dans les exercices, n’apporte pas beaucoup de soutien. Comme nous le verrons plus tard, il est important de mettre en pratique ses connaissances afin de consolider ses acquis.

Malgré tout, il s’agit d’une manière tout autant enrichissante que divertissante pour se familiariser avec le langage R.

13.2 Débutant

13.2.1 Environnements de programmation

13.2.2 Les alternatives à Word : les langages de balisage

- le chapitre 5 parle des langages de balisage (markdown, latex, etc.)
- <http://wcours.gel.ulaval.ca/2018/h/GEL1001/default/5chronologie/2017-01%20GEL-1001%20tutoriel%20LaTeX.pdf> (powerpoint sur l'utilisation de latex)
- <https://mirrors.ibiblio.org/CTAN/info/guide-latex-fr/guide-latex-fr.pdf> (latex)
- https://books.google.ca/books?hl=fr&lr=&id=9jb_DwAAQBAJ&oi=fnd&pg=PP1&dq=how+to+use+aDPButn0-xoiXKJkwWg&redir_esc=y#v=onepage&q=how%20to%20use%20markdown&f=false (utilisation de markown)
- https://books.google.ca/books?hl=fr&lr=&id=__FwPEAAAQBAJ&oi=fnd&pg=PP1&dq=how+to+use (utilisation de markdown)

13.2.3 Serpents

Beaucoup de nouvelles informations ont été présentées jusqu'à présent dans ce livre. Il est normal de se sentir dépassé et de ne pas tout comprendre. En fait, il aurait été surprenant qu'un lecteur qui débute l'aventure numérique ait tout compris. L'important est de garder une attitude propice à l'apprentissage et se rappeler que rien de ceci n'est inatteignable. C'est au tout début du parcours que se trouve le premier serpent : **croire qu'il sera trop difficile d'apprendre, que c'est un objectif impossible à atteindre**. Même les auteurs de ce livre ont, un jour, commencés par faire *Hello World!* dans la console de RStudio. Le premier serpent est souvent lié à un autre piège qui frappe les codeurs débutants : **la peur**

de demander de l'aide. Il faut garder à l'esprit qu'une grande quantité des utilisateurs des outils présentés dans le présent livre sont passés par l'incertitude du début et la crainte du jugement des autres. N'ayez pas peur de poser vos questions, c'est comme cela qu'on apprend.

Une autre catégorie de serpents pour les débutants concerne la pratique des connaissances nouvellement acquises. Les serpents de cette catégorie sont au nombre de trois. Tout d'abord, on retrouve **la croyance qu'il est possible d'apprendre sans pratiquer**. Bien que cela puisse être possible pour quelques personnes ayant une mémoire phénoménale, la réalité est qu'il sera difficile pour le lecteur moyen de retenir l'information contenue dans ce livre et dans les exercices sans pratiquer les nouvelles notions. Le second serpent de cette catégorie est lié à ce dernier point : DataCamp – où il y a des indices et du code déjà écrit – ne forme pas à lui seul des codeurs. Il faut faire attention à **ne pas rester pris dans une boucle infinie de tutoriels**. Faire des tests avec des projets personnels aide à assimiler les nouvelles connaissances en plus d'être plus intéressant. Le troisième serpent de cette catégorie est de **ne pas être constant dans ses apprentissages**. Avec les exercices comme Datacamp, il est facile d'apprendre très rapidement. Toutefois, les apprentissages peuvent se perdre aussi rapidement qu'elles ont été acquises. Il est donc important de suivre une certaine continuité et même parfois de refaire certains exercices afin de se rafraîchir la mémoire ou encore pour s'assurer de bien comprendre les connaissances de base.

Le dernier serpent pour débutants est le suivant : **ne pas construire des bases solides avant d'aller plus loin**. Plusieurs nouveaux codeurs, excités par les nouveaux outils qu'ils apprennent, oublient qu'il est primordial de bien comprendre les éléments de base de la programmation et de la gestion de données avant de se lancer dans des projets plus complexes. Bien qu'il ne soit pas requis de connaître la mécanique pour conduire une automobile, il est tout de même parfois utile – voir nécessaire – de comprendre comment entretenir celle-ci.

13.3 Intermédiaire

13.3.1 La gestion des références

- le chapitre 6 porte entièrement sur la gestion des références (zotero, biblatex, etc.).
- https://uottawa.libguides.com/comment_utiliser_zotero (site de l'université d'Ottawa. Beaucoup d'explications sur l'utilisation de Zotero + vidéo)
- https://www.bibl.ulaval.ca/fichiers_site/aide_recherche/zotero/guide-zotero.pdf (document de l'université laval)
- <http://svn.tug.org/pracjourn/2006-4/fenn/fenn.pdf> (explique comment utiliser Biblatex)

13.3.2 Visualisation graphique en R

- Ggplot2 : elegant graphics for data analysis <https://ulaval.on.worldcat.org/search/detail/951778044?query=ggplot2> (livre qui explique comment utiliser ggplot2)
- <https://link-springer-com.acces.bibl.ulaval.ca/book/10.1007/978-0-387-75969-2> (utilisation de Lattice)

13.3.3 Serpents :

À la suite des différents exercices et lectures complétés dans le cadre de cette familiarisation aux sciences sociales numériques, le lecteur doit s'assurer d'éviter certains pièges qui se dressent sur le chemin des chercheurs de niveau intermédiaire. Le premier d'entre eux est **vouloir apprendre plusieurs langages et n'en maîtriser aucun**. Plusieurs chercheurs, lorsqu'ils commencent à maîtriser de nouveaux outils, s'emballent et souhaitent en apprendre davantage. C'est une

13 Serpents et échelles

bonne chose, mais il faut faire attention à ne pas apprendre que quelques éléments de plusieurs langages de programmation, et plutôt en maîtriser un. Comme le dit un diction populaire, « qui trop embrasse mal étreint ».

Un second serpent auquel de jeunes chercheurs sont la proie est **coder en n'utilisant pas un style et une planification cohérente et constante**. En n'adoptant pas un style standard – ou en n'utilisant pas le plus souvent le même style – il peut devenir difficile pour les autres et pour soi-même de se retrouver dans le code. Cela peut causer d'importants problèmes de compréhension ou des problèmes techniques. Il est rare qu'un même code ne serve qu'une seule fois. Il est donc de viser à ce que le code qu'on produit soit compréhensible, transférable et – idéalement – optimisé. Un autre serpent s'inscrivant dans la lignée du précédent est **écrire du code mais ne pas le commenter**. Commenter son code contribue grandement à la transférabilité et la pérennité de son travail. Bien que la fonction d'une section de code peut sembler évidente pour son créateur le jour où elle est produite, elle ne le sera pas nécessairement pour d'autres ou pour lui-même dans le futur.

Le troisième serpent concerne l'utilisation des packages R. De nombreux packages R sont disponibles sur Internet. Dans certaines situations, l'utilisation de ceux-ci peut représenter un gain de temps et résoudre certains problèmes spécifiques. Toutefois, pour des tâches relativement simples, utiliser un package R risque d'ajouter une complexité inutile. En effet, comprendre un package R et l'adapter en fonction de son projet peut être long et laborieux. Il est donc souvent beaucoup plus efficace d'écrire son propre code plutôt que d'utiliser un package R.

Le dernier piège se dressant sur le chemin d'un chercheur de niveau intermédiaire est de **croire qu'il a suffisamment de connaissances et ne pas sortir de sa zone de confort**. L'apprentissage de techniques plus complexes demande de sortir de sa zone de confort et de se confronter à l'inconnu. Cela demande également d'accepter qu'on ne connaît pas tout et qu'il y aura des échecs et des frustrations. C'est ainsi qu'un chercheur

intermédiaire peut dépasser ses limites et devenir un chercheur de niveau avancé.

13.4 Avancé

13.4.1 Gestion de projet et de données

13.4.2 Outils d'intelligence artificielle

13.4.3 Snakes

L'un des pièges importants à éviter lorsque le chercheur se retrouve à un niveau avancé est la peur de partager son code. Estimant leur code comme étant une propriété intellectuelle, plusieurs chercheurs développent cette réticence et refusent de partager le fruit de leur labeur. Toutefois, partager son code comporte de nombreux avantages, non seulement pour les autres membres de la communauté, mais également pour le chercheur lui-même. D'un côté, cela permet de recevoir des rétroactions de la part d'autres chercheurs et développeurs. Cette collaboration peut donc grandement contribuer à l'amélioration de son code. De plus, partager son code représente une opportunité d'apprentissage pour les autres membres de la communauté qui peuvent s'en inspirer pour développer leurs compétences ou même le réutiliser dans leur propre projet. Cette transparence et cette collaboration sont donc avantageuses pour tous les partis.

Le deuxième serpent duquel le chercheur avancé doit se méfier est de laisser le parfait devenir l'ennemi du bien. Certains chercheurs ont parfois tendance à être perfectionnistes et à perdre du temps et de l'énergie sur des détails mineurs qui n'ont, en fin de compte, aucune retombée majeure sur la qualité globale du projet, comme chercher à optimiser son code de manière excessive. Se soucier de la qualité de son travail est essentiel, mais le chercheur avancé doit également apprendre à savoir quand s'arrêter.

13 Serpents et échelles

Après avoir consacré de nombreuses heures et travaillé d'arrache-pied pour acquérir des connaissances avancées en codage, le chercheur a de quoi être fier. Toutefois, il doit se méfier de l'ultime serpent : manquer d'empathie et de compréhension envers les nouveaux utilisateurs. Certains chercheurs de niveau avancé peuvent oublier qu'ils ont déjà été, eux aussi, des débutants. Il faut éviter de prendre pour acquis certaines connaissances de base qui peuvent sembler très simple pour un chercheur avancé, mais très complexe pour un débutant. Soutenir les nouveaux utilisateurs dans leur apprentissage avec patients et empathie permet une meilleure transmission des connaissances.

- La peur de partager son code/Douchebagisme
 - Laisser le parfait être l'ennemi du bien
 - Manquer d'empathie et de compréhension envers les nouveaux utilisateurs

13.5 Conclusion

References

- Adcock, Robert, and David Collier. 2001. "Measurement Validity: A Shared Standard for Qualitative and Quantitative Research." *American Political Science Review* 95 (3): 529–46. <https://doi.org/10.1017/S0003055401003100>.
- Allaire, JJ. 2022. *Announcing Quarto, a New Scientific and Technical Publishing System*. <https://posit.co/blog/announcing-quarto-a-new-scientific-and-technical-publishing-system/>.
- Bertolini, Andrea. 2020. "Artificial Intelligence and Civil Liability." Policy Department for Citizen's Rights and Constitutional Affairs. Brussel: European Union. [https://www.europarl.europa.eu/RegData/etudes/STUD/2020/621926/IPOL_STU\(2020\)621926_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/STUD/2020/621926/IPOL_STU(2020)621926_EN.pdf).
- Devedzic, Vladan. 2022. "Identity of AI." *Discover Artificial Intelligence* 2 (23). <https://doi.org/10.1007/s44163-022-00038-0>.
- Encyclopaedia Britannica. 2023. *LaTeX*. <https://www.britannica.com/technology/LaTeX-computer-programming-language>.
- Goldfarb, Charles F. 1996. *The Roots of SGML – A Personal Recollection*. <http://www.sgmlsource.com/history/roots.htm>.
- Hameed, Sharqa. 2023. *HTML History | Explained*. <https://linuxhint.com/html-history/>.
- IBM. 2023. "What Is Artificial Intelligence (AI)?" 2023. <https://www.ibm.com/topics/artificial-intelligence>.
- Kabacoff, Robert. 2022. *R in Action: Data Analysis and Graphics with R and Tidyverse*. Third edition. Shelter Island, NY: Manning Publications.
- König, Pascal D., Tobias D. Krafft, Wolfgang Schulz, and Katharina A. Zweig. 2022. "Essence of AI. What Is AI?" In *The Cambridge*

References

- Handbook of Artificial Intelligence*, edited by and Michel Cannarsa Cristina Poncibò, 18–34. Cambridge: Cambridge University Press. <https://doi.org/10.1017/9781009072168.005>.
- Markdown Guide. 2023. *Getting Started*. <https://www.markdownguide.org/getting-started/>.
- Morandat, Floréal, Brandon Hill, Leo Osvald, and Jan Vitek. 2012. “Evaluating the Design of the R Language.” In *ECOOP 2012 – Object-Oriented Programming: 26th European Conference, Beijing, China, June 11-16, 2012. Proceedings*, edited by James Noble, 7313:104–31. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-31057-7>.
- Muenchen, Robert A. 2011. *R for SAS and SPSS Users*. 2nd ed. Statistics and Computing. New York: Springer.
- Overleaf. 2023. *Modèles*. <https://fr.overleaf.com/latex/templates>.
- Quarto. 2023. *Get Started*. <https://quarto.org/docs/get-started/>.
- Sarkar, Deepayan. 2008. *Lattice: Multivariate Data Visualization with R*. New York, NY: Springer New York. <https://doi.org/10.1007/978-0-387-75969-2>.
- . 2023. “Trellis Graphics for R.” <https://cran.r-project.org/web/packages/lattice/lattice.pdf>.
- Stack Overflow. 2023. *Stack Overflow*. <https://stackoverflow.com/>.
- Tippmann, Sylvia. 2015. “Programming Tools: Adventures with R.” *Nature* 517 (7532): 109–10. <https://doi.org/10.1038/517109a>.
- Wang, Pei. 2019. “On Defining Artificial Intelligence.” *Journal of Artificial General Intelligence* 10 (2): 1–37. <https://doi.org/10.2478/jagi-2019-0002>.
- Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. New York, NY: Springer New York. <https://doi.org/10.1007/978-0-387-98141-3>.
- Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy McGowan, Romain François, Garrett Grolemond, et al. 2019. “Welcome to the Tidyverse.” *Journal of Open Source Software* 4 (43): 1686. <https://doi.org/10.21105/joss.01686>.
- Wickham, Hadley, Mine Çetinkaya-Rundel, and Garrett Grolemond. 2023.

References

- R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. Second edition. Beijing: O'Reilly.
- World Wide Web Consortium (W3C). 1998. *Extensible Markup Language (XML) 1.0*. <https://www.w3.org/TR/1998/REC-xml-19980210.html>.
- Xie, Yihui. 2023. *Markdown*. <https://github.com/rstudio/markdown>.

