

Introduction aux méthodes quantitatives en sciences sociales avec R

Philippe Apparicio et Jérémy Gelb

2020-09-17

Table des matières

Préface	13
Comment lire ce livre	13
Structure du livre	13
Remerciements	13
À propos des auteurs	15
Première partie Découverte de R	17
1 Prise en main de R	19
1.1 Histoire et philosophie de R	19
1.2 Environnement de travail	21
1.2.1 Installer R	22
1.2.2 L'environnement RStudio	22
1.2.3 Installer et charger un <i>package</i>	26
1.2.4 Obtenir de l'aide	27
1.3 Les bases du langage R	28
1.3.1 Hello World!	28
1.3.2 Objets et expressions	29
1.3.3 Fonctions et arguments	30
1.3.4 Principaux types de données	31
1.3.5 Opérateurs	32
1.3.6 Structures de données	34
1.4 Manipuler des données	41
1.4.1 Charger un <i>DataFrame</i> depuis un fichier	41
1.4.2 Manipuler un <i>DataFrame</i>	44
1.5 Conclusion et ressources pertinentes	60
Deuxième partie Analyses univariées	63
2 Statistiques descriptives univariées	65
2.1 Notion de variable	65
2.1.1 La variable : l'opérationnalisation d'un concept	65
2.1.2 Les types de variables	68
2.2 Les types de données	69
2.2.1 Données secondaires <i>versus</i> données primaires	69
2.2.2 Données transversales <i>versus</i> données longitudinales	70

2.2.3	Données spatiales versus données aspatiales	70
2.2.4	Données individuelles <i>versus</i> données agrégées	70
2.3	Statistique descriptive et statistique inférentielle	72
2.3.1	Population, échantillon et inférence	72
2.3.2	Deux grandes familles de méthodes statistiques	72
2.4	La notion de distribution	73
2.4.1	Définitions générales	73
2.4.2	Anatomie d'une distribution	75
2.4.3	Principales distributions	77
2.4.4	Conclusion sur les distributions	92
2.5	Statistiques descriptives sur des variables quantitatives	92
2.5.1	Les paramètres de tendance centrale	92
2.5.2	Les paramètres de position	93
2.5.3	Les paramètres de dispersion	95
2.5.4	Les paramètres de forme	98
2.5.5	La transformation des variables	114
2.5.6	Mise en œuvre dans R	116
2.6	Statistiques descriptives sur des variables qualitatives et semi-qualitatives	127
2.6.1	Les fréquences	127
2.6.2	Mise en œuvre dans R	128
2.7	Pour aller un peu plus loin : les statistiques descriptives pondérées	130
3	La magie des graphiques	133
3.1	Philosophie du ggplot2	135
3.1.1	Une grammaire	135
3.1.2	Les types de géométries	138
3.1.3	L'habillage	139
3.1.4	Utiliser des thèmes	142
3.1.5	Composer une figure avec plusieurs graphiques	146
3.1.6	La couleur	148
3.2	Principaux graphiques	151
3.2.1	Histogramme	151
3.2.2	Graphique de densité	156
3.2.3	NUAGE DE POINTS	156
3.2.4	Graphique en lignes	163
3.2.5	Boites à moustache	166
3.2.6	Graphique en violons	168
3.2.7	Graphique en barres	169
3.2.8	Graphique circulaire	173
3.3	Graphiques spéciaux	175
3.3.1	Graphique en radar	175
3.3.2	Diagramme d'accord	178
3.3.3	NUAGE DE MOTS	179
3.3.4	Treemaps	182
3.4	Les cartes	184
3.5	Exporter des graphiques	187
3.6	Conclusion sur les graphiques	189

Troisième partie Analyses bivariées	191
4 Analyses bivariées	193
4.1 Relation linéaire entre deux variables quantitatives	194
4.1.1 Bref retour sur le postulat de la relation linéaire	194
4.1.2 Covariance	196
4.1.3 Corrélation	197
4.1.4 Régression linéaire simple	213
4.2 Relation entre deux variables qualitatives	222
4.2.1 Construction de tableau de contingence	222
4.2.2 Test du χ^2	224
4.2.3 Mise en œuvre dans	226
4.2.4 Interprétation d'un tableau de contingence	231
4.3 Relation entre une variable quantitative et une variable qualitative à deux modalités	233
4.3.1 Test t et ses différentes variantes	234
4.3.2 Test non paramétrique de Wilcoxon	250
4.4 Relation entre une variable quantitative et une variable qualitative à plus de deux modalités	252
4.4.1 Analyse de variance	252
4.4.2 Test non paramétrique de Kruskal-Wallis	257
4.4.3 Mise en œuvre dans	258
4.4.4 Comment rapporter les résultats d'une ANOVA et du test de Kruskal-Wallis	266
Quatrième partie Modèles de régression	269
5 La régression linéaire multiple	271
6 Régressions linéaires généralisées (GLM)	273
6.1 Qu'est qu'un modèle GLM?	273
6.1.1 Principes de base des GLM	273
6.1.2 Reformulation d'un modèle OLS sous forme GLM	273
6.2 Les principaux modèles GLM utilisées	273
6.2.1 Les régressions logistiques	273
6.2.2 Les régressions de type Poisson	273
6.3 D'autres modèles GLM	273
7 Régressions à effets mixtes (GLMM) et régression multiniveaux	275
7.1 Principes de base des GLMM	275
7.2 Application à une variable continue	275
7.3 Régressions multiniveaux	275
8 Modèles généralisés additifs	277
8.1 Principes de base des GAM	277
8.2 Extensions des GAM	277
8.2.1 GAMM	277
8.2.2 GAMMAR	277
Cinquième partie Analyses exploratoires multivariées	279
9 Méthodes factorielles	281

9.1	Un petit historique	281
9.2	Analyses de composantes principales (ACP)	281
9.3	Analyses factorielles de correspondances (AFC)	281
9.4	Analyses factorielles de correspondances multiples (AFM)	281
9.5	Analyses factorielles de correspondances mixte	281
10	Méthodes de classification non-supervisées	283
10.1	Un aperçu sur la multitude des méthodes de classifications	283
10.2	Combinaisons méthodes factorielles et méthodes de classifications	283
10.3	Classification ascendantes hiérachiques	283
10.4	Nuées dynamiques	283
10.4.1	k-means	283
10.4.2	k-median	283
10.4.3	Les extensions en logique floues : c-means, c-median	283
11	Conclusion	285
11.1	Synthèse des méthodes abordées	285
11.2	Autres méthodes non abordées	285
11.2.1	Régressions par quantile	285
11.2.2	Régressions par panel	285
11.2.3	Régressions par Tobit	285
11.2.4	Analyses de survie	285
11.2.5	Équations structurelles	285
11.3	Deux écoles de statistique inférielle : fréquentiste et bayésienne	285
11.4	Éthique en méthodes quantitatives	285
12	Annexes	287
12.1	Tableau des valeurs critiques de χ^2	287
12.2	Tableau des valeurs critiques de F	287

Liste des tableaux

1.1	Opérateurs mathématiques	32
1.2	Opérateurs relationnels	33
1.3	Opérateurs logiques	33
1.4	Un premier DataFrame	39
1.5	Temps nécessaire pour lire les données en fonction du type de fichiers	45
1.6	Avantages et inconvénients du tidyverse	45
1.7	Ressources pertinente pour en apprendre plus sur R	61
2.1	Revenus moyens et médians des ménages en dollars, municipalités de l'île de Montréal, 2015	93
2.2	Stastistiques descriptives de l'exposition au bruit des cyclistes par minute dans trois villes (dB(A), Laeq 1min)	95
2.3	Calcul des mesures de dispersion sur des données fictives	97
2.4	Illustration de la sensibilité des mesures de dispersion à l'unité de mesure et aux valeurs extrêmes	98
2.5	Résumé de la sensibilité de la moyenne et des mesures de dispersion	99
2.6	Les différents tests d'hypothèse pour la normalité	104
2.7	Calculs des tests de normalité pour différentes distributions	104
2.8	Comparaison des LogLikelihood des trois distributions	110
2.9	Illustration des trois tranformations	116
2.10	Les différents types de fréquences sur une variable qualitative ou semi-qualitative	128
2.11	Les différents types de fréquences sur une variable semi-qualitative	128
2.12	Calcul de la moyenne pondérée	130
2.13	Statistiques de l'aire de jeux la plus proche par secteur de recensement pondérées par la population de moins de 10 ans	131
3.1	Principales géométries proposée par <code>ggplot2</code>	139
4.1	Intervalles pour l'interprétation du coefficient de corrélation habituellement utilisés en sciences sociales	199
4.2	Comparaison de différentes corrélations pour les variables densité de population et pourcentage de personnes ayant 65 ans et plus	203
4.3	Données fictives sur l'utilisation du vélo par municipalité	216
4.4	Valeurs observées, prédites et résidus	218
4.5	Calcul du coefficient de détermination	219
4.7	Mesures d'association entre deux variables qualitatives	231
4.8	Conventions pour l'interprétation du d de Cohen	242
4.9	Données fictives et calcul des trois variances (cas 1)	255
4.10	Données fictives et calcul des trois variances (cas 2)	255

12.1 Distribution des valeurs critiques du khi2	288
12.2 Distribution des valeurs critiques de F avec p=0,05	289

Table des figures

1	Cargo, le plus beau	14
1.1	Lieu de pélerinage de R	20
1.2	Nombre d'articles trouvés sur Google Scholar (Source : Robert A. Muenchen)	20
1.3	Métaphore sur les langages et programmes d'analyse statistique	22
1.4	Icône des encadrés dédiés aux packages	22
1.5	La console de base de R	23
1.6	Environnement de base de RStudio	23
1.7	RStudio avec le style pastel on dark	24
1.8	Les quatre fenêtres de RStudio	25
1.9	Les quatre fenêtres de RStudio	25
1.10	Description des packages	27
1.11	Arguments de la fonction round	31
1.12	Du vecteur à la matrice	37
1.13	Un array avec trois dimension	38
1.14	De la donnée au DataFrame	39
1.15	Répartition temporelle des accidents à vélo	53
1.16	Fusion de DataFrames	57
1.17	Jointure de DataFrames	59
2.1	Les types de variables	68
2.2	Exemple d'agrégation de données individuelles	71
2.3	Distribution théorique d'un lancé de dé	74
2.4	Distribution empirique d'un lancé de dé (n=10)	74
2.5	Distribution empirique d'un lancé de dé (n=10)	75
2.6	Distributions uniformes continues	76
2.7	18 distributions essentielles, design inspiré de Sean (2018)	78
2.8	La distribution binomiale	79
2.9	La distribution géométrique	80
2.10	La distribution binomiale négative	81
2.11	La distribution de poisson	82
2.12	La distribution de poisson avec excès de zéros	83
2.13	La distribution Gaussienne	83
2.14	La distribution skew-Gaussienne	84
2.15	La distribution log-gaussienne	85
2.16	La distribution de Student	86
2.17	La distribution de Cauchy	87
2.18	La distribution du Chi ²	87
2.19	La distribution exponentielle	88

2.20 La distribution de Gamma	89
2.21 La distribution de Beta	90
2.22 La distribution de Weibull	91
2.23 La distribution de Pareto	91
2.24 Exemples de cartographie avec une discréétisation selon les quantiles	94
2.25 Graphique en violon, boîte à moustaches et intervalle interquartile	95
2.26 Formes d'une distribution et les coefficients d'asymétrie et d'aplatissement	99
2.27 Asymétrie d'une distribution	100
2.28 Aplatissement d'une distribution	101
2.29 Distributions et courbe normale	103
2.30 Diagrammes quantile-quantile	103
2.31 Distribution empirique des temps de retard des bus à Toronto en janvier 2019	108
2.32 Comparaison des distributions ajustées aux données de retard des bus	109
2.33 Distribution empirique du nombre d'accidents par intersection impliquant un cycliste à Montréal en 2017 dans les quartiers centraux	112
2.34 Ajustement des distributions de poisson et poisson avec excès de zéros	113
2.35 Histogramme avec courbe normale	120
2.36 Histogramme des transformations	127
2.37 Différents graphiques pour représenter les fréquences absolues et relatives	129
2.38 Accessibilité aux aires de jeux par secteur de recensement, Communauté métropolitaine de Montréal, 2016	131
 3.1 Noyer dans le plastique selon Reuters Graphics	134
3.2 Changement climatique et vulnérabilité par Verisk Maplecroft	134
3.3 Trois composantes d'un graphique, adapté de @wickham2010layered	136
3.4 Base d'un graphique	137
3.5 Ajout des dimensions au graphique	137
3.6 Autre spécification des arguments mapping et data	138
3.7 Ajouter les annotations	138
3.8 Ajouter des annotations textuelles	140
3.9 Ajouter des annotations géométriques	141
3.10 Gérer l'ordre des annotations	141
3.11 Thème classique	142
3.12 Thème gris	143
3.13 Thème noir et blanc	143
3.14 Thème minimal	144
3.15 Thème tufte	145
3.16 Thème economist	145
3.17 Thème solarized	146
3.18 Graphique à facettes	147
3.19 Graphique à facette en une ligne	147
3.20 Figure avec plusieurs graphiques avec ggarrange	149
3.21 Couleurs de base	150
3.22 palette de couleurs de ColorBrewer	150
3.23 Histogrammes	152
3.24 Empiler les données d'un dataframe	152
3.25 Histogrammes en facettes	153
3.26 Histogrammes de densité	154
3.27 Histogrammes et courbe normale	155
3.28 Histogramme coloré	155

3.29 Graphiques de densité en facette	156
3.30 Graphiques de densité	157
3.31 Nuage de points simple	158
3.32 Nuage de points simple sans données aberrantes	159
3.33 Nuage de points simple avec transparence	159
3.34 Densité en deux dimensions par carrés	160
3.35 Densité en deux dimensions par hexagones	161
3.36 Densité en deux dimensions lissée	161
3.37 Nuage de points avec droite de régression	162
3.38 Nuage de points avec droite de régression exponentielle	163
3.39 Nuage de points avec droite de régression non linéaire	163
3.40 Graphique en ligne	164
3.41 Graphique en ligne avec barres d'erreur	165
3.42 Graphique en ligne avec marge d'erreur	166
3.43 Boîte à moustache	167
3.44 Boîte à moustache améliorée	167
3.45 Boîte à moustache avec observations	168
3.46 Graphique en violon	169
3.47 Graphique en violon et boîte à moustaches	170
3.48 Grands secteurs de Québec	170
3.49 Graphiques en barres simples	172
3.50 Graphique en barres empilées	173
3.51 Graphique en pointes de tarte	174
3.52 Graphique en anneau	175
3.53 Graphique en anneau	177
3.54 Diagramme d'accord	180
3.55 Nuage de mots pour le SAD de Montréal	182
3.56 Nuage de mots pour le SAD de Québec	183
3.57 Treemap	184
3.58 Carte thématique avec ggplot2	186
3.59 Carte thématique avec tmap	187
3.60 Combiner des cartes avec tmap	188
3.61 Exporter un graphique dans RStudio	188
4.1 Les principales méthodes bivariées	193
4.2 Relations linéaires et curvilinéaires entre deux variables continues	195
4.3 Exemples de relations curvilinéaires	196
4.4 Covariance et unités de mesure	197
4.5 Relations entre deux variables continues et coefficients de corrélation de Pearson	198
4.6 Coefficient de corrélation et proportion de la variance expliquée	199
4.7 Illustration de l'effet des valeurs extrêmes sur le coefficient de Pearson	200
4.8 Comparaison des coefficients de Pearson, Spearman et Kendall sur deux variables anormalement distribuées	202
4.9 Histogramme pour les valeurs de corrélation issues du Bootstrap	207
4.10 Matrice de corrélation avec corrrplot (chiffres)	212
4.11 Matrice de corrélation avec corrrplot (chiffres et ellipses)	213
4.12 Corrélations significatives obtenues aléatoirement	216
4.13 Relation linéaire entre l'utilisation du vélo et la distance au centre-ville	217
4.14 Droite de régression, valeurs observées, prédites et résidus	218
4.15 Taille des ensembles HLM selon la période de construction	233

4.16 Boîtes à moustaches sur des échantillons fictifs non appariés	237
4.17 Boites à moustaches sur des échantillons fictifs appariées	240
4.18 Pourcentage de locataires par secteur de recensement, RMR de Montréal, 2016	243
4.19 Graphiques de densité et en violons	259
4.20 QQ Plot pour les groupes	260
4.21 Graphique de densité et en violons	263
4.22 QQ Plot pour les groupes	264

Préface

Comment lire ce livre

Si vous googlez l'expression « comment lire un livre ? », vous trouverez une multitude de conseils et astuces. Pour ce livre, nous conseillons de le lire de gauche à droite et page par page. Plus sérieusement, il comprend plusieurs types de blocs de texte qui, on l'espère, faciliteront la lecture.



Bloc packages : habituellement localisé en début du chapitre, il comprend la liste des packages R utilisés pour un chapitre.



Bloc objectif : comprend une description des objectifs d'une section.



Bloc notes : comprend une information secondaire sur une notion, un élément, une idée abordée dans une section.



Bloc pour aller plus loin : peut comprendre des références ou des extensions d'une méthode statistique abordée dans une section.



Bloc astuce : décrit un élément qui vous facilera la vie : une propriété statistique, un *package*, une fonction, une syntaxe R.



Bloc attention : comprend une notion ou un élément important à bien maîtriser.

Structure du livre

À écrire plus tard.

Remerciements

Note au beau Cargo (chien Mira) qui nous supporte dans l'écriture du livre !



FIG. 1 : Cargo, le plus beau

À propos des auteurs

Philippe Apparicio (<http://www.ucs.inrs.ca/philippe-apparicio>) est professeur titulaire au Centre Urbanisation Culture Société de l'INRS (<http://www.ucs.inrs.ca/>). Il enseigne au programme de maîtrise en études urbaines (<http://www.ucs.inrs.ca/ucs/etudier/programmes/etudes-urbaines>) les cours *méthodes quantitatives appliquées aux études urbaines et analyses spatiales appliquées aux études urbaines*. Il a aussi créé et enseigné, il y a plusieurs années, le cours *systèmes d'information géographique appliqués aux études urbaines*. Durant les dernières années, il a offert plusieurs formations aux Écoles d'été du Centre interuniversitaire québécois de statistiques sociales (CIQSS, <https://www.ciqss.org/>). Titulaire de la Chaire de recherche du Canada (niveau 2) sur l'équité environnementale et la ville, il est le directeur du **laboratoire d'équité environnementale** (<http://laeq.ucs.inrs.ca>). Géographe de formation, ses intérêts de recherche actuels incluent la justice et l'équité environnementale, la pollution atmosphérique et le bruit et le vélo en ville. Il a publié une centaine d'articles dans différents domaines des études urbaines et de la géographie.

Jérémy Gelb est candidat au doctorat en études urbaines à l'INRS (sous la supervision de Philippe Apparicio) et membre du **laboratoire d'équité environnementale** (<http://laeq.ucs.inrs.ca>). Son sujet de thèse porte sur l'exposition des cyclistes aux pollutions atmosphériques et sonores en milieu urbain. Il utilise quotidiennement des systèmes d'information géographique (SIG) et est tombé dans la marmite de l'*open source* avec le triptyque QGIS, R et Python au début de sa maîtrise. Il a récemment développé deux packages R : **geocmeans** et **spNetwork**, permettant respectivement d'effectuer des analyses de classification floue non-supervisée pondérée spatialement et des estimations de densité par kernel sur réseau.

Philippe et Jérémy travaillent étroitement ensemble depuis déjà plusieurs années. Avec d'autres collègues, ils ont copubliés plusieurs articles (Apparicio and Gelb, 2020; Bureguya et al., 2020; Gelb and Apparicio, 2020; Apparicio et al., 2019; Delaunay et al., 2019; Gelb and Apparicio, 2019; Apparicio and Gelb, 2020; Bureguya et al., 2020; Apparicio et al., 2018a, 2017, 2016). Tous deux s'intéressent à l'exposition des cyclistes à la pollution atmosphérique et sonore dans plusieurs villes à travers le monde : Philippe ayant une préférence pour les collectes dans les villes du Sud Global (notamment indiennes, africaines et latino-américaines) et Jérémy dans les villes du Nord (européennes et nord-américaines).

Première partie

Découverte de R

Chapitre 1

Prise en main de R

Dans ce chapitre, nous reviendrons brièvement sur l'histoire de R et la philosophie qui entoure le logiciel. Nous donnerons quelques conseils pour son installation et la mise en place d'un environnement de développement. Nous présenterons les principaux objets qui sous-tendent le travail effectué avec R (dataframe, vecteur, matrice, etc.) et comment les manipuler avec des exemples appliqués. Enfin, nous terminerons cette section avec un tour d'horizon des capacités graphiques de R . Si vous maîtrisez déjà R , nullement besoin de lire ce chapitre!



Dans cette section, nous utiliserons principalement les *packages* suivants :

- Pour importer des fichiers externes :
 - * **foreign** pour entre autres les fichiers *dbase* et ceux des logiciels SPSS et Stata
 - * **sas7bdat** pour les fichiers du logiciel SAS
 - * **xlsx** pour les fichiers Excel
- Pour manipuler des chaînes de caractères et des dates :
 - * **stringr** pour les chaînes de caractères
 - * **lubridate** pour les dates
- Pour manipuler des données :
 - * **dplyr** du **tidyverse** propose une grammaire pour manipuler et structurer des données.

1.1 Histoire et philosophie de R

R est à la fois un langage de programmation et un logiciel libre (sous la licence publique générale GNU) dédié à l'analyse statistique et supporté par une fondation : *R foundation for Statistical computing*. Il est principalement écrit en C et Fortran.

R a été créé par Ross Ihaka et Robert Gentleman à l'Université d'Auckland en Nouvelle-Zélande. Si vous avez un jour l'occasion de passer dans le coin, une plaque est affichée dans le département de statistique de l'université, ça mérite le détour (figure 1.1). Une première version a été publiée en 1996, mais la première version stable ne date que de 2000, il s'agit donc d'un logiciel relativement récent si on le compare à ses concurrents SAS (1976), SPSS (1968) et Stata (1984).

R a cependant réussi à s'imposer tant dans la milieu de la recherche que dans le secteur privé. Pour s'en convaincre, il suffit de lire l'excellent article concernant la popularité des logiciels d'analyse de données tiré du site r4stats.com¹ (figure 1.2).

¹<http://r4stats.com/articles/popularity>

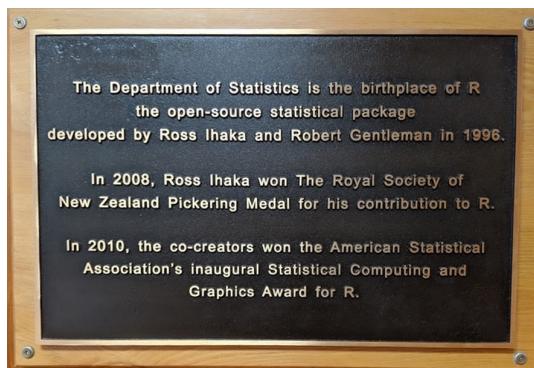


FIG. 1.1 : Lieu de pélerinage de R

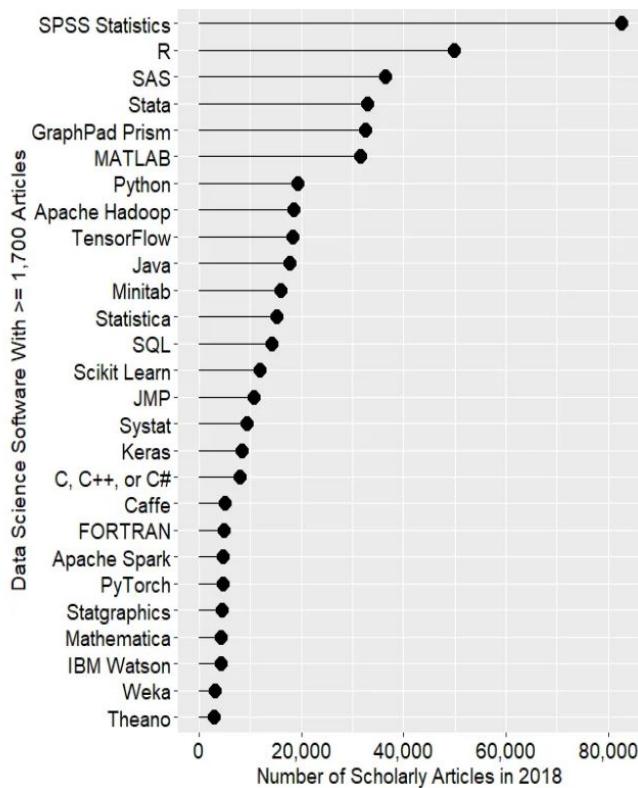


FIG. 1.2 : Nombre d'articles trouvés sur Google Scholar (Source : Robert A. Muenchen)

Les nombreux atouts de R justifient largement sa popularité sans cesse croissante :

- R est un logiciel à code source ouvert (*open source*) et ainsi accessible à tous gratuitement.
- Le développement du langage R est centralisé, mais la communauté peut créer et partager facilement des *packages*. Les nouvelles méthodes sont ainsi rapidement implémentées comparativement aux logiciels propriétaires.
- R est un logiciel multi-plateforme, fonctionnant sur Linux, Unix, Windows et Mac.
- Comparativement à ses concurrents, R dispose d'excellentes solutions pour manipuler des données et réaliser des graphiques.
- R dispose de nombreuses interfaces lui permettant de communiquer, notamment avec des systèmes de bases de données SQL et non SQL (MySQL, PostgreSQL, MongoDB, etc.), à des systèmes de *big data* (Spark, Hadoop), à des systèmes d'information géographique (QGIS, ArcGIS) et même à des services en ligne comme Microsoft Azure ou Amazon AWS.

- R est un langage de programmation à part entière, ce qui lui donne plus de flexibilité que ses concurrents dans le domaine privé (SPSS, SAS, STATA). Avec R, vous pouvez accomplir des tâches aussi variées que : monter un site web, créer un robot collectant des données en ligne, effectuer des analyses qualitatives, combiner des fichiers PDF, composer des diapositives pour une présentation ou même éditer un livre (comme celui-ci), mais aussi, réaliser des analyses statistiques.

Un des principaux attrait de R est la quantité astronomique de *packages* actuellement disponibles. Un *package* est un ensemble de nouvelles fonctionnalités développées par un ou plusieurs utilisateurs de R et mises à disposition de l'ensemble de la communauté. Par exemple, le *package ggplot2* est dédié à la réalisation de graphiques; les *packages* **data.table** et **plyr** permettent de manipuler des tableaux de données; le *package car* apporte de nombreux outils pour faciliter l'analyse de modèles de régressions, etc. Ce partage des *packages* rend accessible à tous des méthodes d'analyses complexes et récentes et favorise grandement la reproductibilité de la recherche. Cependant, ce fonctionnement implique quelques désavantages :

- il existe généralement plusieurs *packages* pour effectuer le même type d'analyse, ce qui peut devenir une source de confusion;
- certains *packages* cessent d'être mis à jour au fil des années, ce qui nécessite de leur trouver d'autres alternatives (et ainsi apprendre la syntaxe des nouveaux *packages*);
- il est impératif de s'assurer de la fiabilité des *packages* que vous souhaitez utiliser, car n'importe qui peut proposer un *package*.

Il nous semble important de relativiser d'emblée la portée du dernier point. Il est rarement nécessaire de lire et analyser le code source d'un *package* pour s'assurer de sa fiabilité. Nous ne sommes pas des spécialistes de tous les sujets et il peut être extrêmement ardu de comprendre la logique d'un code écrit par quelqu'un d'autre. Nous vous recommandons donc de privilégier l'utilisation de *packages* qui :

- ont fait l'objet d'une publication dans une revue à comité de lecture ou qui ont déjà été cités dans des études ayant fait l'objet d'une publication revue par les pairs;
- font partie de projets comme ROpenSci² prônant la vérification par les pairs ou subventionnés par des organisations comme R Consortium³.
- sont disponibles sur l'un des deux principaux répertoires de *packages* R CRAN⁴ ou Bioconductor⁵.

Toujours pour nuancer notre propos, il convient de distinguer *package* de *package!* Certains d'entre eux sont des ensembles très complexes de fonctions permettant de réaliser des analyses poussées alors que d'autres sont des projets plus modestes dont l'objectif principal est de simplifier le travail des utilisateurs. Ces derniers ressemblent à des petites boîtes à outils et font généralement moins l'objet d'une vérification intensive.

Pour conclure cette section, l'illustration partagée sur Twitter par Darren L Dahly résume avec humour la force du logiciel R et de sa communauté (1.3) : R apparaît clairement comme une communauté hétéroclète, mais diversifiée et adaptable.

Dans ce livre, nous détaillerons les **packages** utilisés dans chaque section avec un encadré spécifique, accompagné de l'icône suivant :

1.2 Environnement de travail

Dans cette section, nous vous proposons une visite de l'environnement de travail classique R .

²<https://ropensci.github.io/reproducibility-guide/sections/introduction/>

³<https://www.r-consortium.org/>

⁴<https://cran.r-project.org/>

⁵<https://www.bioconductor.org/>

If statistics programs/languages were cars...



FIG. 1.3 : Métaphore sur les langages et programmes d'analyse statistique



FIG. 1.4 : Icône des encadrés dédiés aux packages

1.2.1 Installer R

La première étape pour travailler avec R est bien sûr de l'installer. Pour ce faire, il suffit de visiter le site web de CRAN⁶ et de télécharger la dernière version de R en fonction de votre système d'exploitation : Windows, Linux ou Mac. Une fois installé, si vous démarrez R immédiatement, vous aurez alors accès à une console, plutôt rudimentaire, attendant sagement vos instructions (figure 1.5).

Notez que vous pouvez aussi télécharger des versions plus anciennes de R en allant sur ce lien⁷. Ceci peut être intéressant lorsque vous voulez reproduire des résultats d'une autre étude ou que certains *packages* ne sont plus disponibles dans les nouvelles versions.

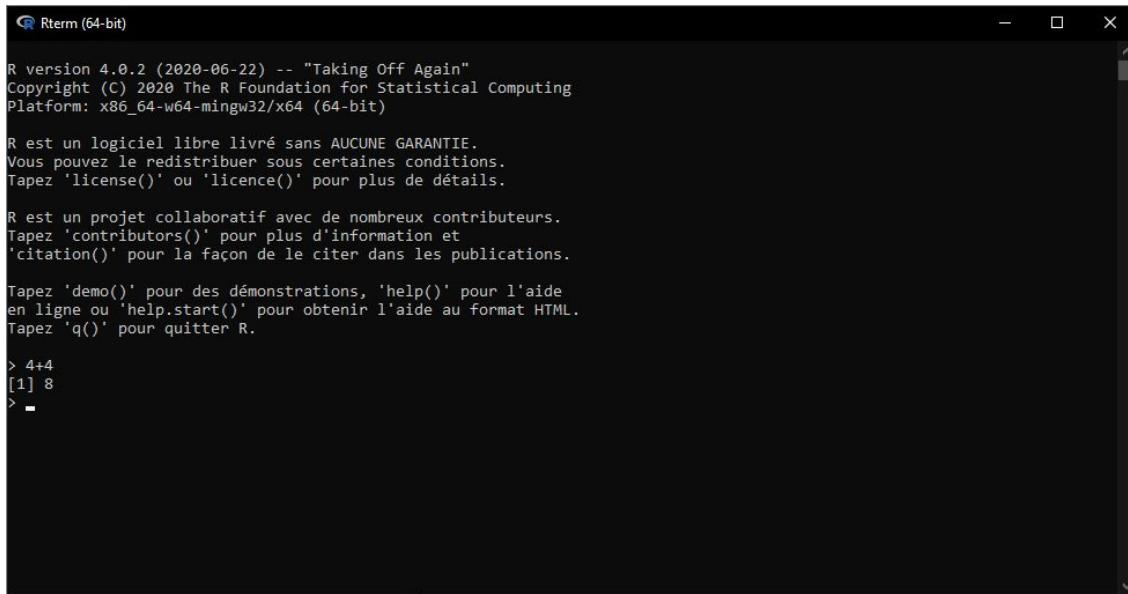
1.2.2 L'environnement RStudio

Rares sont les utilisateurs de R qui préfèrent travailler directement avec la console classique. Nous vous recommandons vivement d'utiliser RStudio, soit un environnement de développement dédié à R, offrant une intégration très intéressante d'une console, d'un éditeur de texte, d'une fenêtre de visualisation des données, d'une autre pour les graphiques, d'un accès à la documentation, etc. En d'autres termes, si R est un vélo minimaliste, RStudio permet d'y rajouter des freins, des vitesses, un porte-bagage, des gardes-boues et une selle confortable. Vous pouvez télécharger⁸ et installer RStudio sur Windows, Linux et Mac. La version de base est gratuite, mais l'entreprise qui développe ce logiciel propose aussi des versions commerciales du logiciel qui assurent essentiellement un support technique. Il existe d'autres environnements de développement pour travailler avec R (VisualStudio, Jupyter, Tinn-R, Radian, RIDE, etc.), mais RStudio offre à ce jour la meilleure option en terme de facilité d'installation, de prise en main et de fonctionnalités proposées (voir l'interface de RStudio à la figure 1.6).

⁶<https://cran.r-project.org/>

⁷<https://cran.r-project.org/bin/windows/base.old/>

⁸<https://rstudio.com/products/rstudio/download>



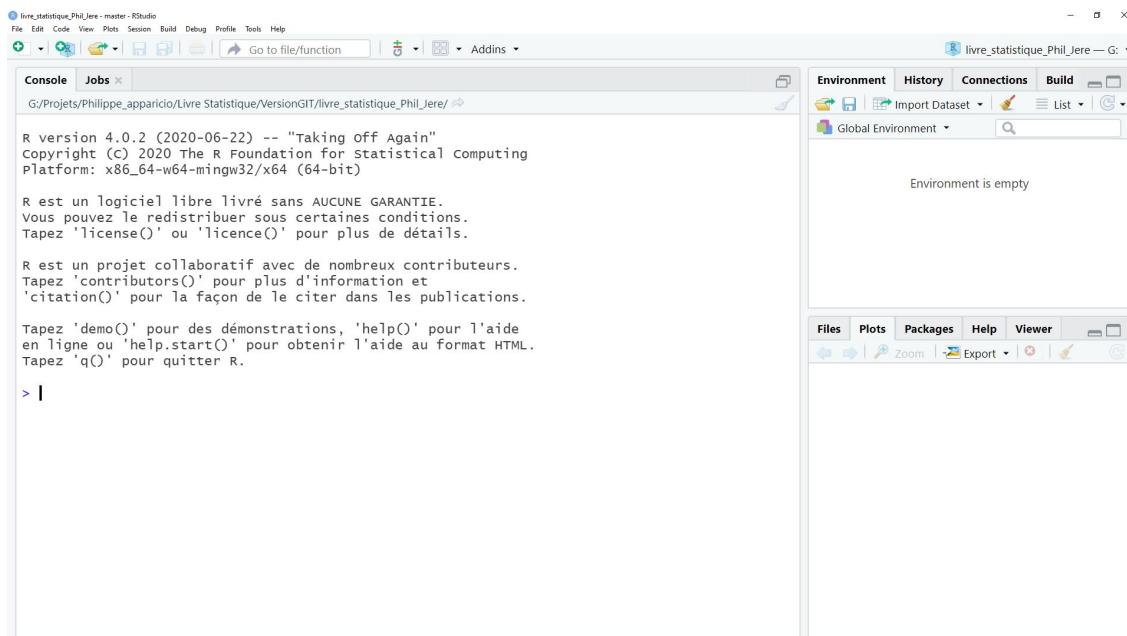
```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> 4+4
[1] 8
> -
```

FIG. 1.5 : La console de base de R**FIG. 1.6 :** Environnement de base de RStudio

Avant d'aller plus loin, notez que :

- La console actuellement ouverte dans RStudio vous informe de la version de R que vous utilisez. Vous pouvez en effet avoir plusieurs versions de R installées sur votre ordinateur et passer de l'une à l'autre avec RStudio. Pour cela, naviguez dans l'onglet *Tools / Global Options* et dans le volet *General*, vous pouvez sélectionner la version de R que vous souhaitez utiliser.
- L'aspect de RStudio peut être modifié en navigant dans l'onglet *Tools / Global Options* et dans le volet *Appearance*. Nous avons une préférence pour le mode sombre avec le style *pastel on dark*, mais libre à chacun de choisir le style qui lui convient.

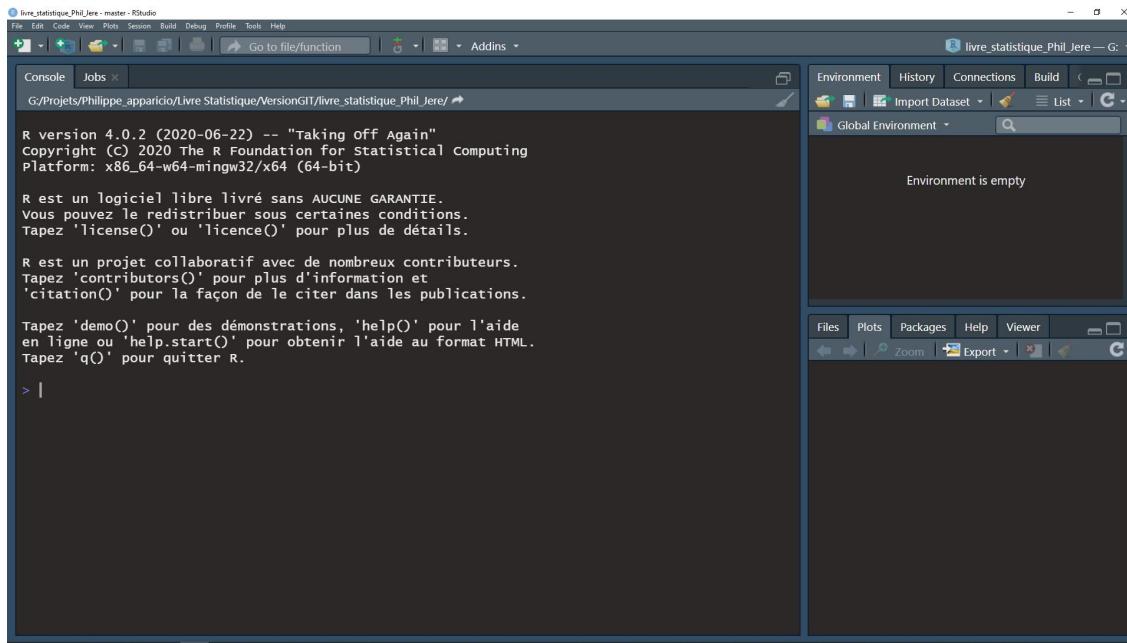


FIG. 1.7 : RStudio avec le style pastel on dark

Une fois ces détails réglés, vous pouvez ouvrir votre première feuille de code en allant dans l'onglet *File / New File/ R Script*, votre environnement est maintenant découpé en quatre fenêtres (figure 1.8) :

1. L'éditeur de code, vous permettant d'écrire le script que vous voulez exécuter et permettant de garder une trace de votre travail. Ce script peut être enregistré sur votre ordinateur avec l'extension **.R**, mais ce n'est qu'un simple fichier texte.
2. La console vous permettant d'exécuter votre code R et de voir les résultats s'afficher au fur et à mesure.
3. La fenêtre d'environnement vous montrant les objets, fonctions et jeux de données actuellement disponibles dans votre session (chargés dans la mémoire vive).
4. La fenêtre de l'aide, des graphiques et de l'explorateur de fichiers. Vous pouvez accéder ici à la documentation de R et des *packages* que vous utilisez, aux sorties graphiques que vous produisez et aux dossier de votre environnement de travail.

Prenons un bref exemple, tapez la syntaxe suivante dans l'éditeur de code (fenêtre 1 à la figure 1.8) :

```
ma_somme <- 4+4
```

Sélectionnez ensuite cette syntaxe (mettre en surbrillance avec votre souris), quand vous utilisez le raccourci *Ctrl+Enter* ou cliquez sur le bouton *Run* (avec la flèche verte), cette syntaxe est envoyée à la console qui l'exécute immédiatement. Notez que rien ne se passe tant que le code n'est pas envoyé à

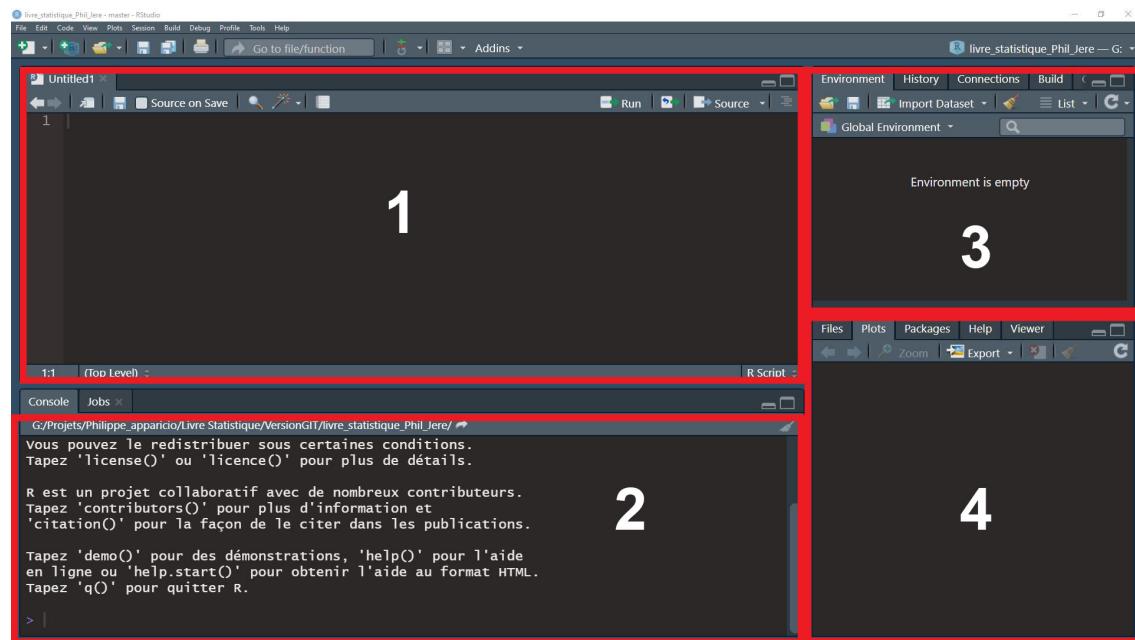


FIG. 1.8 : Les quatre fenêtres de RStudio

la console. Il s'agit donc de deux étapes distinctes : écrire son code, puis l'envoyer à la console. Vous constaterez également qu'un objet `ma_somme` est apparu dans votre environnement et que sa valeur est bien 8. Votre console se "souvient" de cette valeur, elle est actuellement stockée dans votre mémoire vive sous le nom de `ma_somme` (figure 1.9).

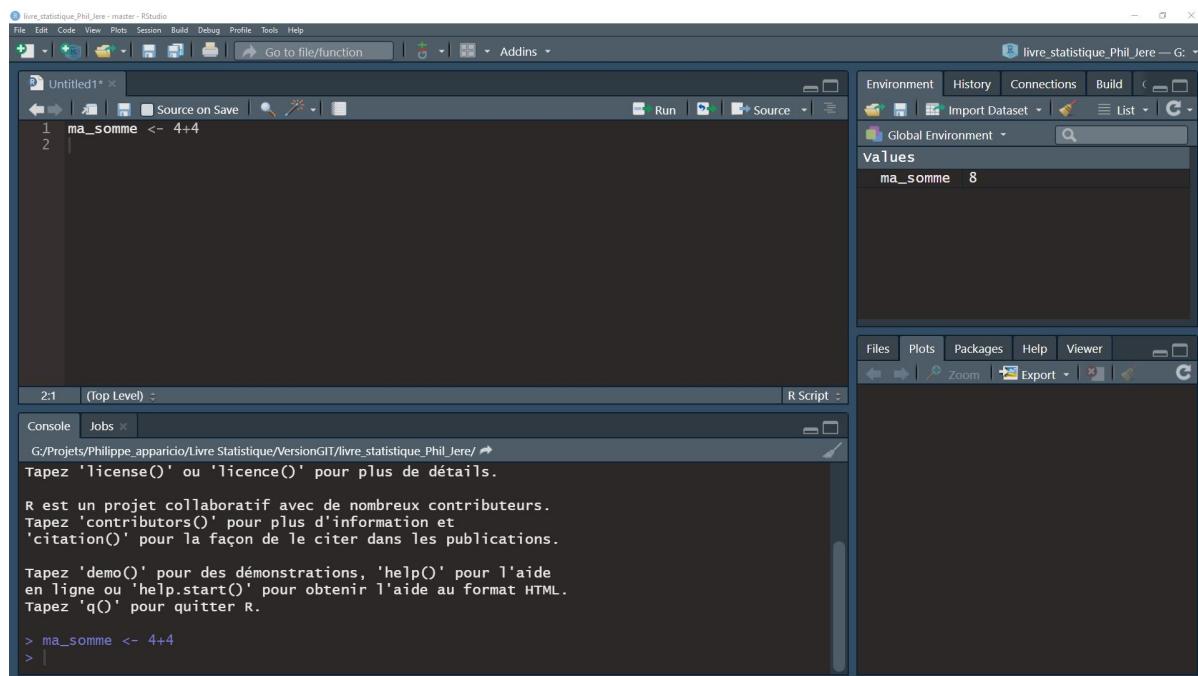


FIG. 1.9 : Les quatre fenêtres de RStudio

Pour conclure cette section, nous vous invitons à enregistrer votre première syntaxe R (*File / Save As*) dans un fichier `.R` que vous pouvez appeler par exemple "mon_premier_script.R". Fermez ensuite RStudio, redémarrez-le et ouvrez (*File / Open File*) votre fichier "mon_premier_script.R". Vous pouvez constater

que votre code est toujours présent, mais que votre environnement est vide tant que vous n'exécutez pas votre syntaxe. En effet, lorsque vous fermez RStudio, l'environnement est vidé pour libérer de la mémoire vive. Ceci peut poser problème lorsque certains codes sont très longs à exécuter, nous verrons donc plus tard comment enregistrer l'environnement en cours pour le recharger par la suite.

1.2.3 Installer et charger un *package*

Dans la section sur la Philosophie de R , nous avons souligné la place centrale jouée par les *packages*. Voyons ensemble comment installer un *package* intitulé **lubridate**, qui nous permettra plus tard de manipuler des données temporelles.

1.2.3.1 Installer un *package* depuis CRAN

Pour installer un *package*, vous devez être connecté à internet; en effet, R va accéder au répertoire de *packages CRAN* pour télécharger le *package* et l'installer sur votre machine. Cette opération est réalisée avec la fonction `install.packages`.

```
install.packages("lubridate")
```

Notez qu'une fois que le *package* est installé, vous n'aurez plus besoin de le refaire. Le *package* est disponible localement sur votre ordinateur, à moins de le désinstaller explicitement avec la fonction `remove.packages`.

1.2.3.2 Installer un *package* depuis GitHub

CRAN est le répertoire officiel des *packages* de R . Vous pouvez cependant télécharger des *packages* provenant d'autres sources. Très souvent, les *packages* sont disponibles sur le site web GitHub⁹ et l'on peut même y trouver des versions en développement avec des fonctionnalités encore non intégrées dans la version sur CRAN. Reprenons le cas de **lubridate**, sur GitHub, il est disponible à la page suivante¹⁰. Pour l'installer nous devons d'abord installer un autre *package* appelé **devtools** (depuis CRAN).

```
install.packages("devtools")
```

Maintenant que nous disposons de **devtools**, nous pouvons utiliser la fonction d'installation `devtools::install_github` pour directement télécharger **lubridate** depuis GitHub.

```
devtools::install_github("tidyverse/lubridate")
```

1.2.3.3 Charger un *package*

Maintenant que **lubridate** est installé, nous pouvons le charger dans notre session actuelle de R et accéder aux fonctions qu'il propose. Pour cela, suffit d'utiliser la fonction `library`. Notez que conventionnellement, l'appel des *packages* se fait au tout début du script que vous rédigez. Rien ne vous empêche de le faire au fur et à mesure de votre code mais vous perdez alors en lisibilité.

⁹<https://github.com/>

¹⁰<https://github.com/tidyverse/lubridate>

```
library(lubridate)
```

Si vous obtenez un message d'erreur du type :

```
Error in library(mon_package) : aucun package nommé 'mon_package' n'est trouvé
```

C'est que le *package* que vous tentez de charger n'est pas encore installé sur votre ordinateur. Dans ce cas, réessayer de l'installer avec la fonction `install.packages`. Si le problème persiste, vérifiez que vous n'avez pas fait de faute de frappe dans le nom du *package*. Vous pouvez également redémarrer RStudio et réessayer d'installer le *package*.

1.2.4 Obtenir de l'aide

Lorsque vous installez des *packages* dans R, vous téléchargez aussi leur documentation. Tous les *packages* de CRAN disposent d'une documentation, mais ceci n'est pas forcément vrai pour GitHub. Dans RStudio, vous pouvez accéder à la documentation des *packages* dans l'onglet **Packages** (figure 1.10). Vous pouvez utiliser la barre de recherche pour retrouver rapidement un *package* installé. Si vous cliquez sur le nom du *package*, vous accédez directement à sa documentation dans cette fenêtre.

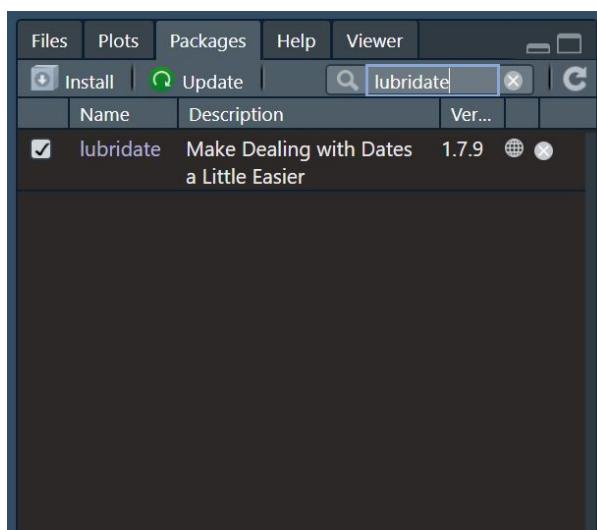


FIG. 1.10 : Description des packages

Vous pouvez également accéder à ces informations en utilisant la syntaxe suivante dans votre console :

```
help(package = 'lubridate')
```

Souvent, vous aurez besoin d'accéder à la documentation d'une fonction spécifique d'un *package*. Affichons la documentation de la fonction `now` de **lubridate** :

```
help(now, package = 'lubridate')
```

ou plus simplement :

```
?lubridate:::now
```

Vous pouvez aussi utiliser le raccourci suivant :

```
?now
```

Si vous connaissez le nom d'une fonction, mais vous ne vous souvenez plus à quel *package* elle appartient, lancez une recherche en utilisant un double point d'interrogation :

```
??now
```

Vous allez ainsi découvrir que la fonction `now` n'existe pas que dans **`lubridate`**, ce qui souligne l'importance de bien connaître les *packages* que l'on installe et que l'on charge dans notre session !

Maintenant que nous avons fait le tour de l'environnement de travail, nous allons pouvoir entamer les choses sérieuses avec les bases du langage R .

1.3 Les bases du langage R

R est un langage de programmation. Il vous permet de communiquer avec votre ordinateur pour lui donner des tâches à accomplir. Dans cette section, nous aborderons les bases du langage. Ce type de section introductory à R est présente dans tous les manuels sur R; elle est donc incontournable. À la première lecture, elle vous semblera probablement aride, et ce, d'autant plus que nous ne réalisons pas d'analyse à proprement parler. Gardez en tête que l'analyse de données requiert au préalable une phase de structuration de ces dernières, opération qui nécessite la maîtrise des notions abordées dans cette section. Nous vous recommandons une première lecture de ce chapitre pour comprendre quelles manipulations vous pouvez effectuer avec R , la lecture des chapitres suivants dédiés aux statistiques, puis de consulter à nouveau cette section au besoin. Notez aussi que la maîtrise des différents objets et opérations de base de R ne s'acquiert qu'en pratiquant. Vous gagnerez cette expertise au fil de vos prochains codes R , période durant laquelle vous pourrez consulter ce chapitre tel un guide de références des différents objets et notions fondamentales de R .

1.3.1 Hello World!

Une introduction à un langage de programmation se doit de commencer par le rite de passage **Hello World**. Il s'agit d'une forme de tradition consistant à montrer aux nouveaux utilisateurs comment afficher le message "Hello World" à l'écran avec le langage en question.

En C, cela donne :

```
#include <stdio.h>

main()
{
    printf("hello, world\n");
}
```

En COBOL :

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.

ENVIRONMENT DIVISION.
```

```
DATA DIVISION.
```

```
PROCEDURE DIVISION.
```

```
    DISPLAY "Hello, world!".
    STOP RUN.
```

et plus simplement en R :

```
print("Hello World")
```

```
## [1] "Hello World"
```

Bravo! Vous venez officiellement de faire votre premier pas dans R!

1.3.2 Objets et expressions

Dans R, nous passons notre temps à manipuler des **objets** à l'aide d'**expressions**. Prenons un exemple concret, si vous tapez la syntaxe `4 + 3`, vous manipulez deux objets (4 et 3) au travers d'une expression indiquant que vous souhaitez obtenir la somme des deux objets.

```
4 + 3
```

```
## [1] 7
```

Cette expression est correcte, R comprend vos indications et effectue le calcul.

Il est possible d'enregistrer le résultat d'une expression et de la conserver dans un nouvel objet. On appelle cette opération déclarer une variable.

```
ma_somme <- 4 + 3
```

Concrètement, nous venons de demander à R d'enregistrer le résultat de `4 + 3` dans un espace spécifique de notre mémoire vive. Si vous regardez dans votre fenêtre **Environment**, vous verrez en effet qu'un objet appelé `ma_somme` est actuellement en mémoire et a pour valeur 7.

Notez ici que le nom des variables ne peut être composé que de lettres, de chiffres, de points(.) et de tiret bas (_) et doit commencer par une lettre. R est sensible à la case, en d'autre termes, les variables `Ma_somme`, `ma_sommE`, `ma_SOMME`, et `MA_SOMME` renvoient toutes à un objet différent. Attention donc aux fautes de frappes. Si vous déclarez une variable en utilisant le nom d'une variable existante, la première est écrasée par la seconde :

```
age <- 35
age
```

```
## [1] 35
```

```
age <- 45
age
```

```
## [1] 45
```

Attention donc aux noms de variables que vous utilisez et réutilisez.

Réutilisons notre objet `ma_somme` dans une nouvelle expression :

```
ma_somme2 <- ma_somme + ma_somme
```

Avec cette nouvelle expression, nous indiquons à R que nous souhaitons déclarer une nouvelle variable appelée `ma_somme2`, et que cette variable aura pour valeur `ma_somme + ma_somme`, soit $7 + 7$. Sans surprise, `ma_somme2` a pour valeur 14.

Notez que la mémoire vive (l'environnement) est vidée lorsque vous fermez R . En d'autres termes, R perd complètement la mémoire lorsque vous le fermez. Vous pouvez bien sûr recréer vos objets en relançant les mêmes syntaxes. C'est pourquoi vous devez conserver vos feuilles de codes et ne pas seulement travailler dans la console. La console ne garde aucune trace de votre travail. Pensez donc à bien enregistrer votre code!

Nous verrons dans un autre chapitre comment sauvegarder des objets et les recharger dans une session ultérieure de R (LIEN SECTION). Ce type d'opération est pertinent quand le temps de calcul nécessaire à la production de certains objets est très long.

1.3.3 Fonctions et arguments

Dans R , nous manipulons le plus souvent nos objets avec des **fonctions**. Une fonction est elle-même un objet, mais qui a la particularité de pouvoir effectuer des opérations sur d'autres objets. Par exemple, déclarons l'objet `taille` avec une valeur de 175.897 :

```
taille <- 175.897
```

Nous allons utiliser la fonction `round` dont l'objectif est d'arrondir un nombre à virgule pour obtenir un nombre entier.

```
round(taille)
```

```
## [1] 176
```

Pour effectuer leurs opérations, les fonctions ont généralement besoin d'**arguments**. Ici, `taille` est un argument passé à la fonction `round`. Si nous regardons la documentation de `round` avec `help(round)`, nous constatons que cette fonction prend en réalité deux arguments : `x` et `digits`. Le premier est le nombre que nous souhaitons arrondir et le second le nombre de décimales à conserver. On peut lire dans la documentation que la valeur par défaut de `digits` est 0, ce qui explique que `round(taille)` a produit le résultat de 176.

Réutilisons maintenant la fonction `round` mais en gardant une décimale :

```
round(taille, digits = 1)
```

```
## [1] 175.9
```

Il est aussi possible que certaines fonctions ne requièrent pas d'arguments. Par exemple, la fonction `now` va indiquer la date précise (avec l'heure) et n'a besoin d'aucun argument pour le faire :

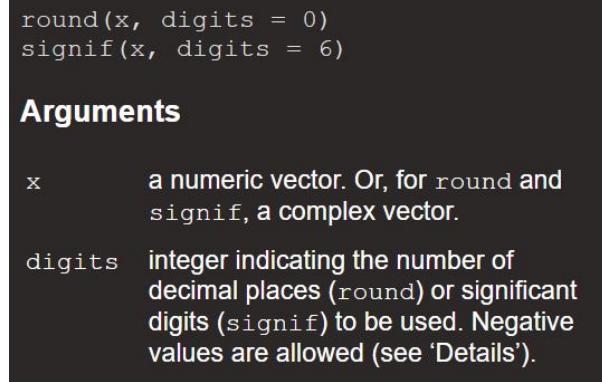


FIG. 1.11 : Arguments de la fonction `round`

```
now()
```

```
## [1] "2020-09-17 21:12:00 EDT"
```

Par contre, si nous essayons de lancer la fonction `round` sans argument, nous obtiendrons une erreur :

```
round()
```

Erreur : 0 arguments passed to 'round' which requires 1 or 2 arguments

Le message est très clair, `round` a besoin d'au moins un argument pour fonctionner. Si au lieu d'un nombre, nous avions donné du texte à la fonction `round`, nous aurions aussi obtenu une erreur :

```
round("Hello World")
```

Error in round("Hello World") : non-numeric argument to mathematical function

À nouveau le message est très explicite : nous avons passé un argument non-numérique à une fonction mathématique. Lisez toujours vos messages d'erreurs qui vous permettront d'identifier des coquilles et de corriger votre code !

Une fonction essentielle est la fonction `print` qui permet d'afficher la valeur d'une variable.

```
print(ma_somme)
```

```
## [1] 7
```

1.3.4 Principaux types de données

Depuis le début de ce chapitre, nous avons déclaré plusieurs variables et essentiellement des données numériques. Dans R, il existe trois principaux types de données de base :

- Les données numériques, qui peuvent être des nombres entiers (appelés *integers*), ou des nombres décimaux (appelés *floats*), 15 et 15.3.
- Les données textuelles, qui sont des chaînes de caractères (appelées *strings*) et déclarées entre guillemets "abcdefg"
- Les données booléennes (*booleans*) qui représentent les concepts de vrai (`TRUE`) ou de faux (`FALSE`).

Déclarons une variable pour chacun de ces types :

```
age <- 35
taille <- 175.5
adresse <- '4225 rue de la gauchetiere'
proprietaire <- TRUE
```

Notez également qu'il existe des types pour représenter l'absence de données :

- pour représenter un objet vide, on utilisera l'objet `NULL`,
- pour représenter une données manquante, on utilisera l'objet `NA`,
- pour représenter un texte vide, on utilisera une chaîne de caractère de longueur 0 `""`.

```
age2 <- NULL
taille2 <- NA
adresse2 <- ''
```

1.3.5 Opérateurs

Nous avons vu que les fonctions nous permettent de manipuler des objets. Nous pouvons également effectuer un grand nombre d'opérations avec des opérateurs.

1.3.5.1 Opérateurs mathématiques

Les opérateurs mathématiques permettent d'effectuer du calcul avec des données de type numérique.

1.3.5.2 Opérateurs relationnels

Les opérateurs relationnels permettent de vérifier des conditions dans R . Ils renvoient un booléen, `TRUE` si la condition est vérifiée et `FALSE` si ce n'est pas le cas.

1.3.5.3 Opérateurs logiques

Les opérateurs logiques permettent de combiner plusieurs conditions :

- L'opérateur **ET** permet de vérifier que deux conditions (l'une ET l'autre) sont `TRUE`. Si l'une des deux est `FALSE`, il renvoie `FALSE`.
- L'opérateur **OU** permet de vérifier que l'une des deux conditions est `TRUE` (l'une OU l'autre). Si les deux sont `FALSE`, alors il renvoie `FALSE`.

TAB. 1.1 : Opérateurs mathématiques

Opérateur	Description	Syntaxe	Résultat
+	Addition	4 + 4	8,0
-	Soustraction	4 - 3	1,0
*	Multiplication	4 * 3	12,0
/	Division	12 / 4	3,0
^	Exponentiel	4 ^ 3	64,0
**	Exponentiel	4 ** 3	64,0

TAB. 1.2 : Opérateurs relationnels

Opérateur	Description	Syntaxe	Résultat
<code>==</code>	Égalité	<code>4 == 4</code>	TRUE
<code>!=</code>	Différence	<code>4 != 4</code>	FALSE
<code>></code>	Est supérieur	<code>5 > 4</code>	TRUE
<code><</code>	Est inférieur	<code>5 < 4</code>	FALSE
<code>>=</code>	Est supérieur ou égal	<code>5 >= 4</code>	TRUE
<code><=</code>	Est inférieur ou égal	<code>5 <= 4</code>	FALSE

- L'opérateur NOT permet d'inverser une condition. Ainsi NOT TRUE est FALSE et NOT FALSE est TRUE.

Prenons le temps pour un rapide exemple :

```
a <- 4
b <- 10
c <- -5

# produit TRUE car a est bien plus petit que b et c est bien plus petit que a
a < b & c < a
```

```
## [1] TRUE
```

```
# produit FALSE car si a est bien plus petit que b,
# b est en revanche plus grand que c
a < b & b < c
```

```
## [1] FALSE
```

```
# produit TRUE car la seconde condition est inversée
a < b & ! b < c
```

```
## [1] TRUE
```

```
# produit TRUE car au moins une des deux conditions est juste
a < b | b < c
```

```
## [1] TRUE
```

Notez que l'opérateur ET est prioritaire sur l'opérateur OU et que les parenthèses sont prioritaires sur tous les opérateurs :

TAB. 1.3 : Opérateurs logiques

Opérateur	Description	Syntaxe	Résultat
<code>&</code>	ET	TRUE & FALSE	FALSE
<code> </code>	OU	TRUE FALSE	TRUE
<code>!</code>	NOT	! TRUE	FALSE

```
# produit TRUE car on va commencer par tester a < b ET b < c ce qui donne FALSE
# on obtient ensuite
# FALSE | a > c
# enfin, a est bien supérieur à c, donc l'une des deux conditions est vraie
a < b & b < c | a > c
```

```
## [1] TRUE
```

Notez qu'en arrière-plan, les opérateurs sont en réalité des fonctions déguisées. Il est donc possible de définir de nouveau comportements pour les opérateurs. Il est par exemple possible d'additionner ou comparer des objets spéciaux comme des dates, des géométries, des graphes, etc.

1.3.6 Structures de données

Jusqu'ici, nous avons travaillé avec des objets ne comprenant qu'une seule valeur. Lors d'une analyse statistique, nous allons travailler avec des volumes de données bien plus conséquents. Pour stocker plusieurs valeurs, nous allons travailler avec les structures de données que sont les vecteurs, les matrices, les *dataframes* et les listes.

1.3.6.1 Vecteurs

Les vecteurs sont la brique élémentaire de R. Ils permettent de stocker une série de valeur du même type dans une seule variable. Pour déclarer un vecteur, on utilise la fonction *c()* :

```
ages <- c(35,45,72,56,62)
tailles <- c(175.5,180.3,168.2,172.8,167.6)
adresses <- c('4225 rue de la gauchetiere',
             '4223 rue de la gauchetiere',
             '4221 rue de la gauchetiere',
             '4219 rue de la gauchetiere',
             '4217 rue de la gauchetiere')
proprietaires <- c(TRUE,TRUE,FALSE,TRUE,TRUE)
```

Nous venons ainsi de déclarer quatre nouvelles variables étant chacune un vecteur de longueur cinq (comprenant chacun cinq valeurs). Ces vecteurs représentent, par exemple, les réponses de plusieurs répondants à un questionnaire.



Il existe dans R une subtilité à l'origine de nombreux malentendus : la distinctions entre un vecteur de type texte et un vecteur de type facteur. Dans l'exemple précédent, le vecteur *adresses* est un vecteur de type texte. Chaque nouvelle valeur ajoutée dans le vecteur peut être n'importe quelle nouvelle adresse. Déclarons un nouveau vecteur qui contiendrait cette fois-ci la couleur des yeux de personnes ayant répondu au questionnaire.

```
couleurs_yeux <- c('marron','marron','bleu','bleu','marron','vert')
```

Contrairement aux adresses, il y a un nombre limité de couleurs que nous pouvons mettre dans ce vecteur. Il serait intéressant de fixer les valeurs possibles du vecteur pour s'assurer que de nouvelles ne soient pas ajoutées par erreur. Pour cela, nous pouvons convertir ce vecteur texte en vecteur de type facteur avec la fonction *as.factor*.

```
couleurs_yeux_facteur <- as.factor(couleurs_yeux)
```

Notez que à présent, nous pouvons ajouter une nouvelle couleur dans le 1er vecteur, mais pas dans le second.

```
couleurs_yeux[7] <- "rouge"
couleurs_yeux_facteur[7] <- "rouge"
```

```
## Warning in `[<-.factor`(`*tmp*`, 7, value = "rouge"): invalid factor level, NA
## generated
```

Le message d'erreur nous informe que nous avons tenté d'introduire une valeur invalide dans le facteur.

Les facteurs peuvent sembler restrictifs et très régulièrement, on préfère travailler avec de simples vecteurs de type texte plutôt que des facteurs. Cependant, de nombreuses fonctions d'analyse nécessitent d'utiliser des facteurs car ils assurent une certaine cohérence dans les données. Il est donc essentiel de savoir passer du texte au facteur avec la fonction `as.factor`. À l'inverse, il est parfois nécessaire de revenir à une variable de type texte avec la fonction `as.character`.

Notez que des vecteurs numériques peuvent aussi être convertis en facteurs :

```
tailles_facteur <- as.factor(tailles)
```

Cependant, si vous souhaitez reconvertis ce facteur en format numérique, il faudra passer dans un premier temps par le format texte :

```
as.numeric(tailles_facteur)
```

```
## [1] 4 5 2 3 1
```

Comme vous pouvez le voir, convertir un facteur en valeur numérique renvoie des nombres entiers. Ceci est dû au fait que les valeurs dans un facteur sont recodées sous forme de nombres entiers, chaque nombre correspondant à une des valeurs originales (appelées niveaux). Si on convertit un facteur en valeurs numériques, on obtient donc ces nombres entiers.

```
as.numeric(as.character(tailles_facteur))
```

```
## [1] 175.5 180.3 168.2 172.8 167.6
```

Moralité de l'histoire, ne confondez pas les données de type texte et de type facteur. Dans le doute, vous pouvez demander à R quel est le type d'un vecteur avec la fonction `class`.

```
class(tailles)
```

```
## [1] "numeric"
```

```
class(tailles_facteur)
```

```
## [1] "factor"
```

```
class(couleurs_yeux)
```

```
## [1] "character"
```

```
class(couleurs_yeux_facteur)
## [1] "factor"
```

Quasiment toutes les fonctions utilisent des vecteurs. Par exemple, on pourrait calculer la moyenne du vecteur *ages* en utilisant la fonction *mean* présente de base dans R .

```
mean(ages)
```

```
## [1] 54
```

Quand nous disons que le vecteur est la brique élémentaire de R , ce n'est pas juste une façon de parler. Toutes les variables que nous avons déclarés dans les sections précédentes sont aussi des vecteurs, mais de longueur 1 !

1.3.6.2 Matrices

Il est possible de combiner des vecteurs pour former des matrices. Une matrice est un tableau en deux dimensions (colonnes et lignes) généralement utilisé pour représenter certaines structures de données comme des images (pixels), effectuer du calcul matriciel ou plus simplement présenter des matrices de corrélations. Vous aurez rarement à travailler directement avec des matrices, mais il est bon de savoir ce qu'elles sont. Créons deux matrices à partir de nos précédents vecteurs.

```
matrice1 <- cbind(ages,tailles)
# afficher la matrice 1
print(matrice1)
```

```
##      ages tailles
## [1,]    35   175.5
## [2,]    45   180.3
## [3,]    72   168.2
## [4,]    56   172.8
## [5,]    62   167.6
```

```
# afficher les dimensions de la matrice 1
print(dim(matrice1))
```

```
## [1] 5 2
```

```
matrice2 <- rbind(ages, tailles)
# afficher la matrice 2
print(matrice2)
```

```
##           [,1]  [,2]  [,3]  [,4]  [,5]
## ages      35.0  45.0  72.0  56.0  62.0
## tailles   175.5 180.3 168.2 172.8 167.6
```

```
# afficher les dimensions de la matrice 2
print(dim(matrice2))
```

```
## [1] 2 5
```

Comme vous pouvez le constater, la fonction `cbind` permet de concaténer des vecteurs comme s'ils étaient les colonnes d'une matrice, alors que `rbind` les combine comme s'ils étaient des lignes d'une matrice. La figure 1.12 présente graphiquement le passage du vecteur à la matrice.

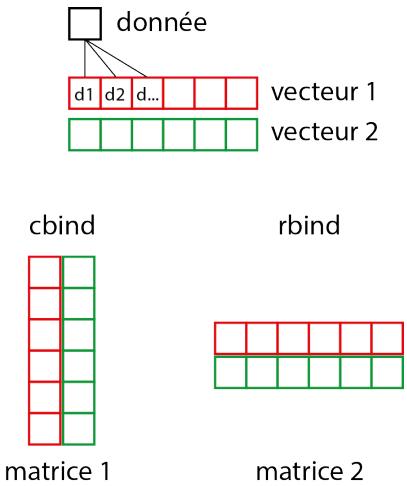


FIG. 1.12 : Du vecteur à la matrice

Notez que vous pouvez transposer une matrice avec la fonction `t`. Si nous essayons maintenant de comparer la matrice 1 et la matrice 2 nous allons avoir une erreur car elles n'ont pas les mêmes dimensions.

```
matrice1 == matrice2
```

Error in matrice1 == matrice2 : non-conformable arrays

En revanche, on pourrait transposer la matrice 1 et refaire cette comparaison :

```
t(matrice1) == matrice2
```

```
##      [,1] [,2] [,3] [,4] [,5]
## ages   TRUE TRUE TRUE TRUE TRUE
## tailles TRUE TRUE TRUE TRUE TRUE
```

Le résultat souligne bien que l'on a les mêmes valeurs dans les deux matrices. Il est aussi possible de construire des matrices directement avec la fonction `matrix`, ce que nous montrons dans la prochaine section.

1.3.6.3 Arrays

S'il est rare de travailler directement avec des matrices, il est encore plus rare de travailler avec des *arrays*. Un *array* est une matrice spéciale qui peut avoir plus que deux dimensions. Un cas simple serait un *array* en trois dimensions : lignes, colonnes, profondeur, que l'on pourrait se représenter comme un cube

divisé en sous cubes. Au delà de trois dimensions, il devient difficile de se les représenter. Cette structure de données peut être utilisée pour représenter les différentes bandes spectrales d'une image satellitaire. Les lignes et les colonnes délimiteraient les pixels de l'image, la profondeur quant à elle délimiterait les différents bandes composant l'image (figure 1.12).

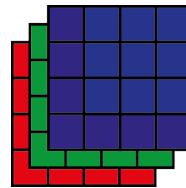


FIG. 1.13 : Un array avec trois dimension

Créons un array en combinant trois matrices avec la fonction `array`. Chacune de ces matrices sera composée respectivement de 1, de 2 et de 3 et aura une dimension de 5×5 . L'array final aura donc des dimensions de $5 \times 5 \times 3$.

```
mat1 <- matrix(1, nrow = 5, ncol = 5)
mat2 <- matrix(2, nrow = 5, ncol = 5)
mat3 <- matrix(3, nrow = 5, ncol = 5)

mon_array <- array(c(mat1, mat2, mat3), dim = c(5,5,3))

print(mon_array)
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    1    1
## [2,]    1    1    1    1    1
## [3,]    1    1    1    1    1
## [4,]    1    1    1    1    1
## [5,]    1    1    1    1    1
##
## , , 2
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2    2    2    2    2
## [2,]    2    2    2    2    2
## [3,]    2    2    2    2    2
## [4,]    2    2    2    2    2
## [5,]    2    2    2    2    2
##
## , , 3
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    3    3    3    3    3
## [2,]    3    3    3    3    3
## [3,]    3    3    3    3    3
## [4,]    3    3    3    3    3
## [5,]    3    3    3    3    3
```

1.3.6.4 DataFrames

S'il est rare de manipuler des matrices et des *arrays*, le *DataFrame* (tableau de données en français) est la structure de données avec laquelle vous travaillerez le plus souvent. Dans cette structure, chaque ligne du tableau représente un individu et chaque colonne représente une caractéristique de ces individus. Ces colonnes ont des noms, ce qui permet facilement d'accéder à leurs valeurs. Créons ensemble un *DataFrame* à partir de nos quatres vecteurs et de la fonction `data.frame`.

```
df <- data.frame(
  "age" = ages,
  "taille" = tailles,
  "adresse" = addresses,
  "proprietaire" = proprietaires
)
```

Dans Rstudio, vous pouvez visualiser votre tableau de données avec la fonction `View(df)`. Comme vous pouvez le constater, chaque vecteur est devenu une colonne de votre tableau de données *df*. La figure 1.14 résume ce passage d'une simple donnée à un DataFrame en passant par un vecteur.

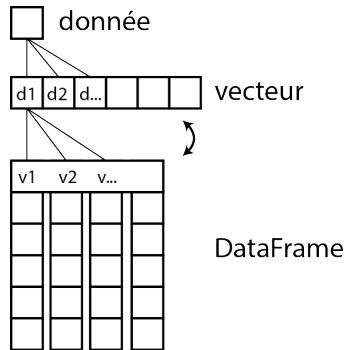


FIG. 1.14 : De la donnée au DataFrame

Plusieurs fonctions de base de R fournissent des informations importantes sur un *DataFrame* :

- `names` renvoie les noms des colonnes du DataFrame;
- `nrow` renvoie le nombre de lignes;
- `ncol` renvoie le nombre de colonnes.

```
names(df)
```

```
## [1] "age"          "taille"        "adresse"       "proprietaire"
```

TAB. 1.4 : Un premier DataFrame

age	taille	adresse	proprietaire
35	175,5	4225 rue de la gauchetiere	TRUE
45	180,3	4223 rue de la gauchetiere	TRUE
72	168,2	4221 rue de la gauchetiere	FALSE
56	172,8	4219 rue de la gauchetiere	TRUE
62	167,6	4217 rue de la gauchetiere	TRUE

```
nrow(df)
```

```
## [1] 5
```

```
ncol(df)
```

```
## [1] 4
```

Vous pouvez accéder à chaque colonne de *df* en utilisant le symbole \$ ou [[*"nom_de_la_colonne"*]]. Recalculons ainsi la moyenne des âges :

```
mean(df$age)
```

```
## [1] 54
```

```
mean(df[["age"]])
```

```
## [1] 54
```

1.3.6.5 Listes

La dernière structure de données à connaître est la liste. Elle ressemble à un vecteur, au sens où elle permet de stocker un ensemble d'objets les uns à la suite des autres. Cependant, une liste peut contenir n'importe quel type d'objets. Vous pouvez ainsi construire des listes de matrices, des listes d'*arrays*, des listes mixant des vecteurs, des graphiques, des *DataFrames*, des listes de listes...

Créons ensemble une liste qui va contenir des vecteurs et des matrices à l'aide de la fonction *list*.

```
ma_liste <- list(c(1,2,3,4),
                  matrix(1, ncol = 3, nrow = 5),
                  matrix(5, ncol = 3, nrow = 7),
                  'A'
                  )
```

Il est possible d'accéder aux éléments de la liste par leur position dans cette dernière en utilisant les doubles crochets [[]]:

```
print(ma_liste[[1]])
```

```
## [1] 1 2 3 4
```

```
print(ma_liste[[4]])
```

```
## [1] "A"
```

Il est aussi possible de donner des noms aux éléments de la liste et d'utiliser le symbole \$ pour y accéder. Créons une nouvelle liste de vecteurs et donnons leurs des noms avec la fonction *names*.

```

liste2 <- list(c(35,45,72,56,62),
               c(175.5,180.3,168.2,172.8,167.6),
               c(TRUE,TRUE,FALSE,TRUE,TRUE))
)
names(liste2) <- c("age",'taille','proprietaire')

print(liste2$age)

```

```
## [1] 35 45 72 56 62
```

Si vous avez bien suivi, vous devez avoir compris qu'un *DataFrame* n'est en fait rien d'autre qu'une liste de vecteurs avec des noms !

Bravo ! Vous venez de faire le tour des bases du langage R . Nous allons pouvoir passer à la suite et apprendre à manipuler des données dans des *DataFrames* !

1.4 Manipuler des données

Dans cette section, vous apprendrez à charger et manipuler des *DataFrames* en vue d'effectuer des opérations classiques de gestion de données.

1.4.1 Charger un *DataFrame* depuis un fichier

Il sera rarement nécessaire de créer vos *DataFrames* manuellement comme réalisé dans la section précédente. Le plus souvent, vous disposerez de fichiers contenant vos données et utiliserez des fonctions pour les importer dans R sous forme d'un *DataFrame*. Les formats à importer les plus répandus sont :

- *.csv*, soit un fichier texte dont chaque ligne représente une ligne du tableau de données dont les colonnes sont séparées par un délimiteur (généralement une virgule ou un point-virgule).
- *.dbf*, ou fichier *dBase*, souvent associés à des fichiers d'information géographique au format *ShapeFile*.
- *.xls* et *.xlsx*, soit des fichiers générés par Excel.
- *.json*, soit un fichier texte utilisant la norme d'écriture propre au langage JavaScript.

Plus rarement, il se peut que vous aillez à charger des fichiers provenant de logiciels propriétaires :

- *.sas7bdat* (SAS),
- *.sav* (SPSS) et
- *.dta* (STATA).

Pour lire la plupart de ces fichiers, nous allons utiliser le package **foreign** dédié à l'importation d'une multitude de formats. Commencez donc par l'installer (`install.packages("foreign")`). Nous allons charger cinq fois le même jeu de données enregistré dans des formats différents (*csv*, *dbf*, *dta*, *sas7bdat* et *xlsx*). Aussi, nous mesurerons le temps nécessaire pour importer chacun de ces fichiers avec la fonction `system.time`.

1.4.1.1 Lire un fichier *csv*

Pour le format *csv*, il n'y a pas besoin d'utiliser un package puisque R dispose d'une fonction de base pour lire ce format.

```
t1 <- Sys.time()
df1 <- read.csv("data/priseenmain/SR_MTL_2016.csv",
                 header = TRUE, sep = ",", dec = ".",
                 stringsAsFactors = FALSE)
t2 <- Sys.time()
d1 <- as.numeric(difftime(t2,t1,units="secs"))

cat('le dataframe df1 a ',nrow(df1),' observations',
    'et ',ncol(df1),"colonnes\n")
```

le dataframe df1 a 951 observations et 48 colonnes

Rien de bien compliqué! Notez tout de même que :

- Lorsque vous chargez un fichier *csv*, vous devez connaître le **séparateur**, soit le caractère utilisé pour délimiter les colonnes. Dans le cas présent, il s'agit d'une virgule (spécifiez avec l'argument *sep* = ","), mais il pourrait tout aussi bien être un point virgule (*sep* = ";") une tabulation (*sep* = " "), etc.
- Vous devez également spécifier le caractère utilisé comme séparateur de décimales. Le plus souvent, ce sera le point (*dec* = ".") , mais certains logiciels avec des paramètres régionaux de langue française (notamment Excel) exportent des fichiers *csv* avec des virgules comme séparateur de décimales (utilisez alors *dec* = ",").
- L'argument *header* indique si la première ligne (l'entête) du fichier comprend ou non les noms des colonnes du jeu de données (avec les valeurs *TRUE* ou *FALSE*). Il arrive que certains fichiers *csv* soient fournis sans entête et que les noms et descriptions des colonnes soient fournis dans un autre fichier.
- L'argument *stringsAsFactors* permet d'indiquer à R que les colonnes comportant du texte doivent être chargées comme des vecteurs de type texte et nom de type facteur.

1.4.1.2 Lire un fichier *dbase*

Pour lire un fichier *dbase* (.dbf), nous utilisons la fonction *read.dbf* du package **foreign** installé précédemment :

```
library(foreign)

t1 <- Sys.time()
df2 <- read.dbf("data/priseenmain/SR_MTL_2016.dbf")
t2 <- Sys.time()
d2 <- as.numeric(difftime(t2,t1,units="secs"))

cat('le dataframe df2 a ',nrow(df2),' observations',
    'et ',ncol(df2),"colonnes\n")
```

le dataframe df2 a 951 observations et 48 colonnes

Comme vous pouvez le constater, nous obtenons les mêmes résultats qu'avec le fichier *csv*.

1.4.1.3 Lire un fichier *dta* (Stata)

Si vous travaillez avec des collègues utilisant le logiciel Stata, il se peut que ces derniers vous partagent des fichiers *dta*. Toujours en utilisant le package **foreign**, vous serez en mesure de les charger directement dans R .

```
t1 <- Sys.time()
df3 <- read.dta("data/priseenmain/SR_MTL_2016.dta")
t2 <- Sys.time()
d3 <- as.numeric(difftime(t2,t1,units="secs"))

cat('le dataframe df3 a ',nrow(df3),' observations',
  'et ',ncol(df3),"colonnes\n", sep = "")

## le dataframe df3 a 951 observationset 48colonnes
```

1.4.1.4 Lire un fichier *sav* (SPSS)

SPSS est encore utilisé dans le milieu académique, surtout au premier cycle, bien que de moins en moins. Pour importer un fichier *sav*, vous pourrez utiliser la fonction `read.spss` package **foreign**.

```
t1 <- Sys.time()
df4 <- as.data.frame(read.spss("data/priseenmain/SR_MTL_2016.sav"))
t2 <- Sys.time()
d4 <- as.numeric(difftime(t2,t1,units="secs"))

cat('le dataframe df4 a ',nrow(df4),' observations',
  'et ',ncol(df4),"colonnes\n", sep = "")

## le dataframe df4 a 951 observationset 48colonnes
```

1.4.1.5 Lire un fichier *sas7bdat* (SAS)

SAS est encore utilisé dans les milieux académiques, gouvernementaux et privés. Pour importer un fichier *sas7bdat*, vous pourrez utiliser le package **sas7bdat** que vous devrez préalablement installer (`install.packages("sas7bdat")`) et charger (`library(sas7bdat)`).

```
library(sas7bdat)

t1 <- Sys.time()
df5 <- read.sas7bdat("data/priseenmain/SR_MTL_2016.sas7bdat")
t2 <- Sys.time()
d5 <- as.numeric(difftime(t2,t1,units="secs"))

cat('le dataframe df5 a ',nrow(df5),' observations',
  'et ',ncol(df5),"colonnes\n", sep = "")

## le dataframe df5 a 951 observationset 48colonnes
```

1.4.1.6 Lire un fichier *xlsx* (Excel)

Lire un fichier Excel dans R n'est pas toujours une tâche facile. Généralement, nous recommandons d'exporter les fichiers en question au format *csv* dans un premier temps, puis de le lire avec la fonction `read.csv` dans un second temps (LIEN SECTION). Il est néanmoins possible de lire directement un fichier *xlsx* avec le package **xlsx**. Ce dernier requiert que le logiciel JAVA soit installé sur votre ordinateur (Windows, Mac ou Linux). Si vous utilisez la version 64 bit de R , vous devrez télécharger et installer

la version 64 bit de JAVA. Une fois que ce logiciel tiers est installé, il ne vous restera plus qu'à installer (`install.packages("xlsx")`) et charger (`library(xlsx)`) le package `xlsx`.

```
library(xlsx)

t1 <- Sys.time()
df6 <- read.xlsx(file="data/priseenmain/SR_MTL_2016.xlsx",
                  sheetIndex = 1,
                  as.data.frame = TRUE)
t2 <- Sys.time()
d6 <- as.numeric(difftime(t2,t1,units="secs"))

cat('le dataframe df6 a ',nrow(df6),' observations',
    'et ',ncol(df6),"colonnes\n", sep = "")

## le dataframe df6 a 951 observationset 48colonnes
```

Il est possible d'accélérer significativement la vitesse de lecture d'un fichier `xlsx` en utilisant la fonction `read.xlsx2`. Il faut cependant indiquer à cette dernière le type de données de chaque colonne. Dans le cas présent, les cinq premières colonnes contiennent des données au format texte (`character`), alors que les 43 autres sont des données numériques (`numeric`). Nous utilisons la fonction `rep` afin de ne pas avoir à écrire plusieurs fois `character` et `numeric`.

```
library(xlsx)

t1 <- Sys.time()
df7 <- read.xlsx2(file="data/priseenmain/SR_MTL_2016.xlsx",
                  sheetIndex = 1,
                  as.data.frame = TRUE,
                  colClasses = c(rep("character",5),rep("numeric",43)))
)
t2 <- Sys.time()
d7 <- as.numeric(difftime(t2,t1,units="secs"))

cat('le dataframe df6 a ',nrow(df7),' observations',
    'et ',ncol(df7),"colonnes\n", sep = "")

## le dataframe df6 a 951 observationset 48colonnes
```

Si l'on compare les temps d'exécution (tableau 1.5), on constate que la lecture des fichiers `xlsx` peut être extrêmement longue si l'on ne spécifie pas le type des colonnes. Ceci peut devenir problématique pour des fichiers volumineux. Notez également que la lecture des fichiers `csv` devient de plus en plus laborieuse à mesure que la taille du fichier `csv` augmente. Si vous devez un jour charger des fichiers `csv` de plusieurs gigaoctets, nous vous recommandons vivement d'utiliser la fonction `fread` du package `data.table` qui est beaucoup plus rapide.

1.4.2 Manipuler un *DataFrame*

Une fois le *DataFrame* chargé, voyons comment il est possible de le manipuler.

TAB. 1.5 : Temps nécessaire pour lire les données en fonction du type de fichiers

Durée (s)	fonction
0,09	read.csv
0,12	read.dbf
0,03	read.spss
0,06	read.dta
1,35	read.sas7bdat
33,82	read.xlsx
1,12	read.xlsx2

1.4.2.1 Un petit mot sur le tidyverse

Tidyverse est un ensemble de *packages* conçus pour faciliter la structuration et la manipulation des données dans R. Avant d'aller plus loin, il est important d'aborder brièvement un débat actuel dans la Communauté R. Entre 2010 et 2020, l'utilisation du **tidyverse** s'est peu à peu répandue. Développé et maintenu par Hadley Wickham, **tidyverse** introduit une philosophie et une grammaire spécifiques qui diffèrent du langage R traditionnel. Une partie de la communauté a pour ainsi dire complètement embrassé le **tidyverse** et de nombreux *packages* en dehors du **tidyverse** ont adopté sa grammaire et sa philosophie. À l'inverse, une autre partie de la communauté est contre cette évolution (voir l'article du blogue suivant¹¹). Les arguments pour et contre **tidyverse** sont résumés dans le tableau suivant.

Le dernier point est probablement le plus problématique. Dans sa volonté d'évoluer au mieux et sans restriction, le *package* **tidyverse** n'offre aucune garantie de rétro-compatibilité. En d'autre termes, des changements importants peuvent être introduits d'une version à l'autre rendant potentiellement obsolète votre propre code. Nous n'avons pas d'opinion tranchée sur le sujet : **tidyverse** est un outil très intéressant dans de nombreux cas ; nous évitons simplement de l'utiliser systématiquement et préférons charger directement des sous-packages (comme **dplyr** ou **ggplot2**) du **tidyverse**. Notez que le *package* **data.table** offre une alternative au **tidyverse** dans la manipulation de données. Au prix d'une syntaxe généralement un peu plus complexe, le *package* **data.table** offre une vitesse de calcul bien supérieure au **tidyverse** et assure une bonne rétro-compatibilité.

1.4.2.2 Gérer les colonnes d'un DataFrame

Repartons du *DataFrame* que nous avions chargé précédemment grâce à un fichier *csv*.

```
df <- read.csv(file="data/priseenmain/SR_MTL_2016.csv",
               header = TRUE, sep = ",", dec = ".",
               stringsAsFactors = FALSE)
```

¹¹<https://blog.ephorie.de/why-i-dont-use-the-tidyverse>

TAB. 1.6 : Avantages et inconvénients du tidyverse

Avantage du tidyverse	Problème posé par le tidyverse
Simplicité d'écriture et d'apprentissage Ajout de l'opérateur La meilleure librairie pour réaliser des graphiques : ggplot2	Nouvelle syntaxe à apprendre Certaines fonctions de base sont remplacées par tidyverse lors de son chargement, pouvant créer des erreurs.
Crée un écosystème cohérent Package en développement et de plus en plus utilisé	Ajoute une dépendance dans le code Philosophie d'évolution agressive, aucune assurance de rétro-compatibilité

1.4.2.2.1 Sélectionner une colonne

Pour rappel, il est possible d'accéder aux colonnes dans ce *DataFrame* en utilisant le symbole dollar `$ma_colonne` ou les doubles crochets `["ma_colonne"]`.

```
# Calcul de la superficie totale de l'île de Montréal
sum(df$KM2)
```

```
## [1] 4680.543
```

```
sum(df[["KM2"]])
```

```
## [1] 4680.543
```

1.4.2.2.2 Sélectionner plusieurs colonnes

Il est possible de sélectionner plusieurs colonnes d'un *DataFrame* et filtrer ainsi les colonnes inutiles. Pour cela, on peut utiliser un vecteur contenant soit la position de la colonne (1 pour la première colonne, 2 pour la seconde et ainsi de suite), soit les noms des colonnes.

```
# Conserver les 5 premières colonnes
df2 <- df[1:5]

# Conserver les colonnes 1,5,10 et 15
df3 <- df[c(1,5,10,15)]

# Cela peut aussi être utilisé pour changer l'ordre des champs
df3 <- df[c(10,15,1,5)]

# Conserver les colonnes 1 à 5, 7 à 12, 17 et 22
df4 <- df[c(1:5,7:12,17,22)]

# Conserver les colonnes avec leurs noms
df5 <- df[c("SRIDU","KM2","Pop2016","MaisonIndi","LoyerMed")]
```

1.4.2.2.3 Supprimer des colonnes

Il est parfois plus intéressant et rapide de directement supprimer des colonnes plutôt que de recréer un nouveau *DataFrame*. Pour ce faire, on attribue la valeur `NULL` à ces colonnes.

```
# Supprimer les colonnes 2, 3 et 5
df3[c(2,3,5)] <- list(NULL)

# Supprimer une colonne avec son nom
df4$OID <- NULL

# Supprimer plusieurs colonnes par leur nom
df5[c("SRIDU","LoyerMed")] <- list(NULL)
```

Notez que si vous supprimez une colonne, vous ne pouvez pas revenir en arrière. Il faudra recharger votre jeu de données ou éventuellement relancer les calculs qui avaient produit cette colonne.

1.4.2.2.4 Renommer des colonnes

Il est possible de changer le nom d'un colonne. Cette opération est importante pour faciliter la lecture du *DataFrame* ou encore s'assurer que l'exportation du *DataFrame* dans un format ne posera pas de problème.

```
# Voici les noms des colonnes
names(df5)

## [1] "KM2"          "Pop2016"       "MaisonIndi"

# Renommer toutes les colonnes
names(df5) <- c('superficie_km2', 'population_2016', 'maison_individuelle_prt')
names(df5)

## [1] "superficie_km2"           "population_2016"
## [3] "maison_individuelle_prt"

# Renommer avec dplyr
library(dplyr)
df4 <- rename(df4, "population_2016" = "Pop2016",
               "prs_moins_14ans_prt" = "A014",
               "prs_15_64_ans_prt" = "A1564",
               "prs_65plus_ans_prt" = "A65plus"
               )
```

1.4.2.3 Calculer de nouvelles variables

Il est possible d'utiliser les colonnes de type numérique pour calculer de nouvelles colonnes en utilisant les opérateurs mathématiques vus dans la section @ref(sect01_35). Prenons un exemple concret : calculons la densité de population par secteur de recensement dans notre *DataFrame* et affichons un résumé de cette nouvelle variable.

```
# Calcul de la densité
df$pop_density_2016 <- df$Pop2016 / df$KM2

# Statistiques descriptives
summary(df$pop_density_2016)

##      Min.   1st Qu.   Median   Mean   3rd Qu.   Max.
##    17.45   1946.96  3700.50  5465.03  7918.39 48811.79
```

Nous pouvons aussi calculer le ratio entre le nombre de maisons et le nombre d'appartements.

```
# Calcul du ratio
df$total_maison <- (df$MaisonIndi + df$MaisJumule + df$MaisRangee + df$AutreMais)
df$total_apt <- (df$AppDuplex + df$App5Moins + df$App5Plus)
df$ratio_maison_apt <- df$total_maison / df$total_apt
```

Retenez ici que R va appliquer le calcul à chaque ligne de votre jeu de données et stocker le résultat dans une nouvelle colonne. Cette opération est du calcul vectoriel : toute la colonne est calculée en une seule fois. R est d'ailleurs optimisé pour le calcul vectoriel.

1.4.2.4 Fonctions mathématiques

R propose un ensemble de fonctions de base pour effectuer du calcul. Voici une liste non-exhaustive des principales fonctions :

- `abs` calcule les valeurs absolues des valeurs d'un vecteur
- `sqrt` calcule les racines carrées des valeurs d'un vecteur
- `log` calcule les logarithmes des valeurs d'un vecteur
- `exp` calcule les exponentiels des valeurs d'un vecteur
- `factorial` calcule la factorielle des valeurs d'un vecteur
- `round` arrondit les valeurs d'un vecteur
- `ceiling, floor` arrondit à l'unité supérieure ou inférieure les valeurs d'un vecteur
- `sin,asin,cos,acos,tan,atan` sont des fonctions de trigonométrie classiques
- `cumsum` calcule la somme cumulative des valeurs d'un vecteur.

Ces fonctions sont des fonctions vectorielles puisqu'elles s'appliquent à tous les éléments d'un vecteur. Si votre vecteur en entrée comprend cinq valeurs, le vecteur en sortie comprendra aussi cinq valeurs.

À l'inverse, les fonctions suivantes s'appliquent directement à l'ensemble d'un vecteur et ne vont renvoyer qu'une seule valeur :

- `sum` calcule la somme des valeurs d'un vecteur
- `prod` calcule le produit des valeurs d'un vecteur
- `min, max` renvoient les valeurs maximale et minimale d'un vecteur
- `mean, median` renvoient la moyenne et la médiane d'un vecteur
- `quantile` renvoie les percentiles d'un vecteur.

1.4.2.5 Fonctions pour manipuler du texte

En plus des données numériques, vous aurez à travailler avec des données textuelles. Le `tidyverse` avec le package `stringr` offre des fonctions très intéressantes pour manipuler ce type de données. Pour un aperçu de toutes les fonctions offertes par `stringr`, référez-vous à sa Cheat Sheet¹². Commençons avec un `DataFrame` assez simple comprenant des adresses et des noms de personnes.

```
library(stringr)

df <- data.frame(
  noms = c("Jérémie Toutanplace", "constant Tinople", "dino Resto", "Luce tancil"),
  adresses = c('15 rue Levy', '413 Blvd Saint-Laurent', '3606 rue Duké', '2457 route St Marys')
)
```

1.4.2.5.1 Majuscules et minuscules

Pour harmoniser ce `dataframe`, nous allons dans un premier temps mettre des majuscules au premier caractère des prénoms et noms des individus avec la fonction `str_to_title`.

```
df$noms_corr <- str_to_title(df$noms)
print(df$noms_corr)

## [1] "Jérémie Toutanplace" "Constant Tinople"    "Dino Resto"
## [4] "Luce Tancil"
```

¹²<https://github.com/rstudio/cheatsheets/blob/master/strings.pdf>

On pourrait également tout mettre en minuscule ou tout en majuscule.

```
df$noms_min <- tolower(df$noms)
df$noms_maj <- toupper(df$noms)
print(df$noms_min)

## [1] "jérémÿ toutanplace" "constant tinople"    "dino resto"
## [4] "luce tancil"

print(df$noms_maj)

## [1] "JÉRÉMY TOUTANPLACE" "CONSTANT TINOPLE"    "DINO RESTO"
## [4] "LUCE TANCIL"
```

1.4.2.5.2 Remplacer du texte

Dans les adresses, nous avons des caractères accentués. Ce type de caractères pose régulièrement des problèmes d'encodage et nous pourrions décider de les remplacer par des caractères simples avec la fonction `str_replace_all`.

```
df$adresses_1 <- str_replace_all(df$adresses, 'é', 'e')
print(df$adresses_1)

## [1] "15 rue Levy"                 "413 Blvd Saint-Laurent" "3606 rue Duke"
## [4] "2457 route St Marys"
```

Nous pouvons utiliser la même fonction pour remplacer les *St* par *Saint* et les *Blvd* par *Boulevard*.

```
df$adresses_2 <- str_replace_all(df$adresses_1, ' St ', ' Saint ')
df$adresses_3 <- str_replace_all(df$adresses_2, ' Blvd ', ' Boulevard ')
print(df$adresses_3)

## [1] "15 rue Levy"                 "413 Boulevard Saint-Laurent"
## [3] "3606 rue Duke"               "2457 route Saint Marys"
```

1.4.2.5.3 Découper du texte

Il est parfois nécessaire de découper du texte pour en extraire des éléments. On doit alors choisir un caractère de découpage. Dans notre exemple, on pourrait vouloir extraire les numéros civiques des adresses, en utilisant le premier espace comme caractère de découpage, en utilisant la fonction `str_split_fixed`.

```
df$num_civique <- str_split_fixed(df$adresses_3, ' ', n=2) [,1]
print(df$num_civique)

## [1] "15"   "413"  "3606" "2457"
```

Pour être exact, sachez que pour notre exemple, la fonction `str_split_fixed` renvoie deux colonnes de texte : une avec le texte avant le premier espace (donc le numéro civique) et une avec le reste du texte. Le nombre de colonnes est contrôlé par l'argument `n`. Si `n = 1`, la fonction ne fait aucun découpage, avec `n`

= 2 la fonction va découper en deux parties le texte avec la première occurrence du délimiteur, et ainsi de suite. En ajoutant [,1] à la fin, nous indiquons que nous souhaitons seulement garder la première des deux colonnes.

1.4.2.5.4 Coller du texte

À l'inverse du découpage, il est parfois nécessaire de concaténer des éléments de texte, ce qu'il est possible de faire avec la fonction `paste`.

```
df$texte_complet <- paste(df$noms_corr, df$adresses_3, sep = " : ")
print(df$texte_complet)
```

```
## [1] "Jérémie Toutanplace : 15 rue Levy"
## [2] "Constant Tinople : 413 Boulevard Saint-Laurent"
## [3] "Dino Resto : 3606 rue Duke"
## [4] "Luce Tancil : 2457 route Saint Marys"
```

Le paramètre `sep` permet de choisir le ou les caractères à intercaler entre les éléments à concaténer. Notez qu'il est possible de concaténer plus que deux éléments.

```
df$ville <- c('Montreal','Montreal','Montreal','Montreal')
paste(df$noms_corr, df$adresses_3, df$ville, sep = ", ")
```

```
## [1] "Jérémie Toutanplace, 15 rue Levy, Montreal"
## [2] "Constant Tinople, 413 Boulevard Saint-Laurent, Montreal"
## [3] "Dino Resto, 3606 rue Duke, Montreal"
## [4] "Luce Tancil, 2457 route Saint Marys, Montreal"
```

1.4.2.6 Manipuler des colonnes de type date

Nous avons vu que les principaux types de données dans R sont le numérique, le texte, le booléen et le facteur. Il existe d'autres types, introduits par différent *packages*. Nous abordons ici les types date et temps (*date and time*). Pour les manipuler, nous privilégions l'utilisation du *package* **lubridate** du **tidyverse**. Pour illustrer le tout, nous l'appliquerons avec un jeu de données ouvertes de la ville de Montréal représentant les accidents de la route incluant au moins un vélo après le premier janvier 2017.

```
accidents_df <- read.csv(file="data/priseenmain/accidents.csv", sep = ", ")
names(accidents_df)
```

```
## [1] "HEURE_ACCDN"          "DT_ACCDN"           "NB_VICTIMES_TOTAL"
```

Nous disposons de trois colonnes représentant respectivement l'heure, la date et le nombre de victimes impliquées dans l'accident.

1.4.2.6.1 Du texte à la date

Actuellement, les colonnes *HEURE_ACCDN* et *DT_ACCDN* sont au format texte. Nous pouvons afficher quelques lignes du jeu de données avec la fonction `head` pour visualiser comment elles ont été saisies.

```
head(accidents_df, n = 5)

##           HEURE_ACCDN DT_ACCDN NB_VICTIMES_TOTAL
## 1 16:00:00-16:59:00 2017/11/02                  0
## 2 06:00:00-06:59:00 2017/01/16                  1
## 3 18:00:00-18:59:00 2017/04/18                  0
## 4 11:00:00-11:59:00 2017/05/28                  1
## 5 15:00:00-15:59:00 2017/05/28                  1
```

Un peu de ménage s'impose : les heures sont indiquées comme des périodes d'une heure. Nous utilisons la fonction `str_split_fixed` du package **stringr** pour ne garder que la première partie de l'heure (avant le tiret). Nous allons ensuite concaténer l'heure et la date avec la fonction `paste`, puis nous convertirons ce résultat en un objet *date-time*.

```
library(lubridate)

# Étape 1 : découper la colonne Heure_ACCDN
accidents_df$heure <- str_split_fixed(accidents_df$HEURE_ACCDN, "-", n=2) [,1]

# Étape 2 : concaténer l'heure et la date
accidents_df$date_heure <- paste(accidents_df$DT_ACCDN,
                                    accidents_df$heure,
                                    sep = ' ')

# Étape 3 : convertir au format datetime
accidents_df$datetime <- as_datetime(accidents_df$date_heure,
                                       format = "%Y/%m/%d %H:%M:%S")
```

Pour effectuer la conversion, nous avons utilisé la fonction `as_datetime` du package **lubridate**. Elle prend comme paramètre un vecteur de texte et une indication du format de ce vecteur de texte. Il existe de nombreuses façons de spécifier une date et une heure et l'argument *format* permet de spécifier quelle nomenclature est utilisée. Dans cet exemple, la date est structurée comme suit : année/mois/jour heure:minute:seconde, ce qui se traduit par le format `%Y/%m/%d %H:%M:%S`.

- `%Y` signifie une année indiquée avec quatre caractères : 2017
- `%m` signifie un mois, indiqué avec deux caractères : 01, 02, 03, ... 12
- `%d` signifie un jour, indiqué avec deux caractères : 01, 02, 03, ... 31
- `%H` signifie une heure, au format 24 heures avec deux caractères : 00, 02, ... 23
- `%M` signifie des minutes indiquées avec deux caractères : 00, 02, ... 59
- `%S` signifie des secondes, indiquées avec deux caractères : 00, 02, ... 59

Notez que les caractères séparant les années, jours, heures, etc. sont aussi à indiquer dans le format. Dans notre exemple, nous utilisons des / pour séparer les éléments de la date, des : pour l'heure, et un espace pour séparer la date et l'heure.

Il existe d'autres nomenclatures pour spécifier un format *datetime* : par exemple, des mois renseignés par leur nom, l'indication AM-PM, etc. Vous pouvez vous référer à la documentation de la fonction `strptime` (`help(strptime)`) pour explorer les différentes nomenclatures et choisir celle qui vous convient. Bien évidemment, il est **nécessaire** que toutes les dates de votre colonne soient renseignées dans le même format. Sinon, la fonction renverra des valeurs NA aux endroits où elle a échoué à lire le format.

Après toutes ces opérations, rejettons un oeil à notre *DataFrame*.

```
head(accidents_df, n = 5)

##           HEURE_ACCDN DT_ACCDN NB_VICTIMES_TOTAL      heure      date_heure
## 1 16:00:00-16:59:00 2017/11/02                      0 16:00:00 2017/11/02 16:00:00
## 2 06:00:00-06:59:00 2017/01/16                      1 06:00:00 2017/01/16 06:00:00
## 3 18:00:00-18:59:00 2017/04/18                      0 18:00:00 2017/04/18 18:00:00
## 4 11:00:00-11:59:00 2017/05/28                      1 11:00:00 2017/05/28 11:00:00
## 5 15:00:00-15:59:00 2017/05/28                      1 15:00:00 2017/05/28 15:00:00
##           datetime
## 1 2017-11-02 16:00:00
## 2 2017-01-16 06:00:00
## 3 2017-04-18 18:00:00
## 4 2017-05-28 11:00:00
## 5 2017-05-28 15:00:00
```

1.4.2.6.2 Extraire des informations d'une date

À partir de la nouvelle colonne `datetime`, nous sommes en mesure d'extraire des informations intéressantes comme :

- le nom du jour de la semaine avec la fonction `weekdays`

```
accidents_df$jour <- weekdays(accidents_df$datetime)
```

- la période de la journée avec les fonctions `am` et `pm`

```
accidents_df$AM <- am(accidents_df$datetime)
accidents_df$PM <- pm(accidents_df$datetime)
```

```
head(accidents_df[c("jour", "AM", "PM")], n=5)
```

```
##     jour    AM    PM
## 1 jeudi FALSE  TRUE
## 2 lundi  TRUE FALSE
## 3 mardi FALSE  TRUE
## 4 dimanche  TRUE FALSE
## 5 dimanche FALSE  TRUE
```

Il est aussi possible d'accéder aux sous-éléments d'un `datetime` comme l'année, le mois, le jour, l'heure, la minute, la seconde avec les fonctions `year()`, `month()`, `day()`, `hour()`, `minute()` et `second()`.

1.4.2.6.3 Calculer une durée entre deux `datetime`

Une autre utilisation intéressante du format `datetime` est de calculer des différences de temps. Par exemple, nous pourrions utiliser le nombre de minutes écoulées depuis 07 :00 le matin comme une variable dans une analyse visant à déterminer le moment critique des accidents durant l'heure de pointe du matin. Pour cela, nous devons créer un `datetime` de référence en concaténant la date de chaque observation, et le temps 07 :00 :00 qui sera notre point de départ.

```
accidents_df$date_heure_07 <- paste(accidents_df$DT_ACCDN,
                                         '07:00:00',
                                         sep = ' ')
accidents_df$ref_datetime <- as_datetime(accidents_df$date_heure_07,
                                           format = "%Y/%m/%d %H:%M:%S")
```

Il ne nous reste plus qu'à calculer la différence de temps entre la colonne *datetime* et notre temps de référence *ref_datetime*.

```
accidents_df$diff_time <- difftime(accidents_df$datetime,
                                       accidents_df$ref_datetime,
                                       units = 'min')
```

Notez qu'ici la colonne *diff_time* est d'un type spécial : une différence temporelle (*difftime*). Il faut encore la convertir au format numérique pour pouvoir l'utiliser avec la fonction *as.numeric*.

Par curiosité, réalisons rapidement un histogramme avec la fonction *hist* pour analyser rapidement cette variable d'écart de temps !

```
accidents_df$diff_time_num <- as.numeric(accidents_df$diff_time)
hist(accidents_df$diff_time_num, breaks = 50)
```

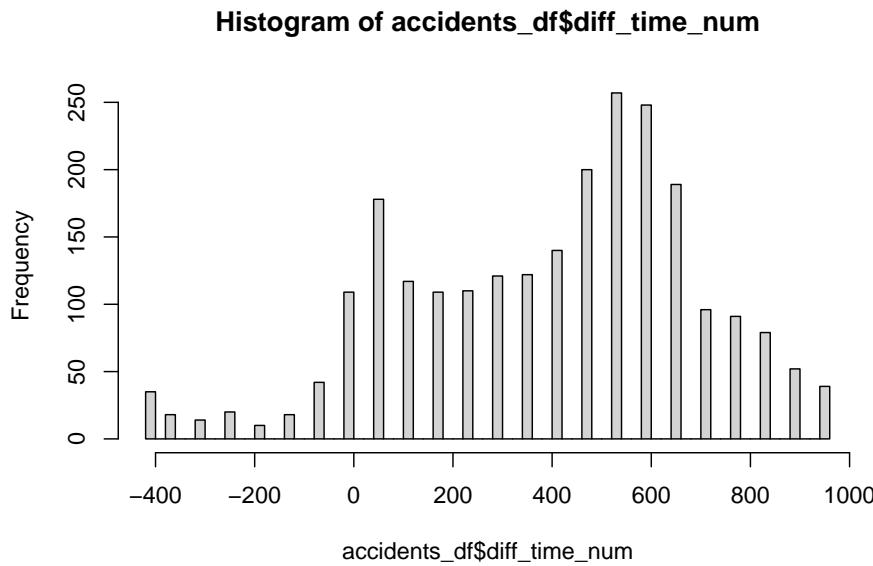


FIG. 1.15 : Répartition temporelle des accidents à vélo

On observe clairement deux pics, un premier entre 0 et 100 (donc entre 07:00 08:30 environ) et un second plus important entre 550 et 650 (entre 16:00 et 17:30 environ), ce qui correspond sans surprise aux heures de pointe. Il est intéressant de noter que plus d'accidents se produisent à l'heure de pointe du soir qu'à celle du matin.

1.4.2.7 Recoder des variables

Recoder des variables signifie changer la valeur d'une variable selon une condition afin d'obtenir une nouvelle variable. Si nous reprenons notre exemple précédent avec les accidents à vélo, nous pourrions

créer une nouvelle colonne nous indiquant si l'accident a eu lieu en heure de pointe ou hors heure de pointe. On obtiendrait ainsi une nouvelle variable avec seulement deux catégories plutôt que la variable numérique originale. Nous pourrions aussi définir trois catégories avec l'heure de pointe du matin, l'heure de pointe du soir, le reste de la journée et la nuit.

1.4.2.7.1 Le cas binaire avec `ifelse`

Si l'on ne souhaite créer que deux catégories, le plus simple est d'utiliser la fonction `ifelse`. Cette fonction va évaluer une condition (section @ref(sect01_35)) pour chaque ligne d'un *DataFrame* et produire un nouveau vecteur. Créons donc une variable binaire indiquant si un accident a eu lieu durant les heures de pointe ou hors heures de pointe. Nous devons alors évaluer les conditions suivantes :

Est-ce que l'accident a eu lieu entre 07 :00 (0) ET 09 :00 (120), OU est ce que l'accident a eu lieu entre 16 :30 (570) ET 18 :30 (690)?

```
Cond1 <- accidents_df$diff_time_num >= 0 & accidents_df$diff_time_num <= 120
Cond2 <- accidents_df$diff_time_num >= 570 & accidents_df$diff_time_num <= 690

accidents_df$moment_bin <- ifelse(Cond1 | Cond2,
                                    "en heures de pointe",
                                    "hors heures de pointe")
```

Comme vous pouvez le constater, la fonction `ifelse` nécessite trois arguments :

- Une condition, pouvant être TRUE ou FALSE,
- La valeur à renvoyer si la condition est TRUE
- La valeur à renvoyer si la condition est FALSE

Avec la fonction `table`, nous pouvons rapidement compter les effectifs des deux catégories ainsi créées :

```
table(accidents_df$moment_bin)

##
##   en heures de pointe hors heures de pointe
##                 841                  1573
```

Les heures de pointes représentent quatre heures de la journée, ce qui nous laisse neuf heures hors heure de pointe entre 07 :00 et 20 :00.

```
# Ratio d'accidents en heures de pointe
(841 / 2414) / (4 / 13)
```

```
## [1] 1.132249
```

```
# Ratio d'accidents hors heure de pointe
(1573 / 2414) / (9 / 13)
```

```
## [1] 0.9412225
```

En rapportant les accidents aux durées des deux périodes, on observe une nette sur-représentation des accidents impliquant un vélo pendant les heures de pointe d'environ 13% comparativement à la période hors des heures de pointe.

1.4.2.7.2 Le cas multiple avec la fonction `case_when`

Lorsque l'on souhaite créer plus que deux catégories, il est possible soit d'enchaîner plusieurs fonctions `ifelse` (ce qui produit un code plus long et moins lisible), soit d'utiliser la fonction `case_when` provenant du package `dplyr` du `tidyverse`. Reprenons notre exemple et créons quatre catégories :

- En heures de pointe du matin
- En heures de pointe du soir
- Le reste de la journée (entre 07 :00 et 20 :00)
- La nuit (entre 21 :00 et 07 :00)

```
library(dplyr)

accidents_df$moment_multi <- case_when(
  accidents_df$diff_time_num >= 0 & accidents_df$diff_time_num <= 120 ~ "pointe matin",
  accidents_df$diff_time_num >= 570 & accidents_df$diff_time_num <= 690 ~ "pointe soir",
  accidents_df$diff_time_num > 690 & accidents_df$diff_time_num < 780 ~ "journee",
  accidents_df$diff_time_num > 120 & accidents_df$diff_time_num < 570 ~ "journee",
  accidents_df$diff_time_num < 0 | accidents_df$diff_time_num >= 780 ~ "nuit"
)

table(accidents_df$moment_multi)

##          journee      nuit pointe matin  pointe soir
##            1155           418        404         437
```

La syntaxe de cette fonction est un peu particulière. Elle accepte un nombre illimité d'arguments. Chaque argument est composé d'une condition et d'une valeur à renvoyer si la condition est vraie ; ces deux éléments étant reliés par le symbole `~`. Notez que toutes les évaluations sont effectuées dans l'ordre des arguments. En d'autres termes, la fonction va d'abord tester la première condition et assigner ces valeurs, puis recommencer pour les prochaines conditions. Ainsi, si une observation (ligne du tableau de données) obtient `TRUE` à plusieurs conditions, elle obtiendra la valeur de la dernière condition qu'elle a validée.

1.4.2.8 Sous-sélection d'un `DataFrame`

Dans cette section, nous verrons comment extraire des sous-parties d'un `DataFrame`. Il est possible de sous-sélectionner des lignes et des colonnes en se basant sur des conditions ou leurs index. Pour cela, nous allons utiliser un jeu de données fourni avec R : le jeu de données `iris` décrivant des fleurs du même nom.

```
data("iris")
dim(iris)

## [1] 150    5
```

1.4.2.8.1 Sous-sélection des lignes

Sous-sélectionner des lignes par index est relativement simple. Admettons que nous souhaitons sélectionner les lignes 1 à 5, 10 à 25, 37 et 58.

```
sub_iris <- iris[c(1:5, 10:25, 37, 58),]
nrow(sub_iris)
```

```
## [1] 23
```

Sous-sélectionner des lignes avec une condition peut être effectué soit avec une syntaxe similaire, soit en utilisant la fonction `subset`. Sélectionnons toutes les fleurs de l'espèce Virginica.

```
iris_virginical <- iris[iris$Species == "virginica",]
iris_virginica2 <- subset(iris, iris$Species == "virginica")

# Vérifions que les deux dataframes ont le même nombre de lignes
nrow(iris_virginical) == nrow(iris_virginica2)
```

```
## [1] TRUE
```

Vous pouvez utiliser dans les deux cas tous les opérateurs vus dans les sections @ref(sect01_352) et @ref(sect01_353). L'enjeu est d'arriver à un vecteur booléen final permettant d'identifier les observations à conserver.

1.4.2.8.2 Sous-sélectionner des colonnes

Nous avons déjà vu comment sélectionner des colonnes en utilisant leur nom ou leur index dans la section @ref(sect01_4221). Ajoutons ici un cas particulier où nous souhaiterions sélectionner des colonnes selon une condition. Par exemple, nous pourrions vouloir conserver que les colonnes comprenant le mot *Length*. Pour cela, nous utiliserons la fonction `grepl`, permettant de déterminer si des caractères sont présents dans une chaîne de caractères.

```
nom_cols <- names(iris)
print(nom_cols)

## [1] "Sepal.Length" "Sepal.Width"   "Petal.Length" "Petal.Width"   "Species"

test_nom <- grepl("Length", nom_cols, fixed = TRUE)
ok_nom <- nom_cols[test_nom]

iris_2 <- iris(ok_nom)
print(names(iris_2))

## [1] "Sepal.Length" "Petal.Length"
```

Il est possible d'obtenir ce résultat en une seule ligne de code, mais elle est un peu moins lisible.

```
iris2 <- iris[names(iris)[grepl("Length", names(iris), fixed = TRUE)]]
```

1.4.2.8.3 Sélectionner des colonnes et des lignes

Nous avons vu qu'avec les crochets, nous pouvons extraire les colonnes et les lignes d'un *DataFrame*. Il est possible de combiner les deux opérations en même temps. Pour cela, il faut indiquer en premier les indices ou la condition permettant de sélectionner une ligne, puis les indices ou la condition pour

sélectionner les colonnes : [index_lignes , index_colonnes]. Sélectionnons les trois premières colonnes et les cinq premières lignes du jeu de données iris :

```
iris_5x3 <- iris[c(1,2,3,4,5),c(1,2,3)]
print(iris_5x3)
```

```
##   Sepal.Length Sepal.Width Petal.Length
## 1          5.1         3.5          1.4
## 2          4.9         3.0          1.4
## 3          4.7         3.2          1.3
## 4          4.6         3.1          1.5
## 5          5.0         3.6          1.4
```

Combinons nos deux exemples précédents pour sélectionner uniquement les lignes avec des fleurs de l'espèce virginica, et les colonnes avec le mot Length.

```
iris_virginica3 <- iris[iris$Species == "virginica",
                      names(iris)[grep("Length", names(iris), fixed = TRUE)]]
head(iris_virginica3, n=5)
```

```
##   Sepal.Length Petal.Length
## 101        6.3        6.0
## 102        5.8        5.1
## 103        7.1        5.9
## 104        6.3        5.6
## 105        6.5        5.8
```

1.4.2.9 Fusionner des *DataFrames*

Terminons cette section avec la fusion de *DataFrames* qu'il est possible de réaliser de deux façons : par ajout ou par jointure.

1.4.2.9.1 Fusionner des *DataFrame* par ajout

Ajouter deux *DataFrames* peut se faire en fonction de leurs colonnes, ou en fonction de leurs lignes. Dans ces deux cas, on utilisera respectivement les fonctions `cbind` et `rbind`. La figure 1.16 résume graphiquement le fonctionnement des deux fonctions.

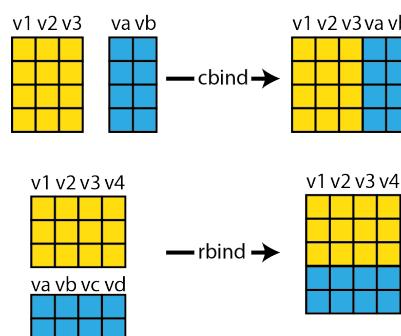


FIG. 1.16 : Fusion de *DataFrames*

Pour que `cbind` fonctionne, il faut que les deux *DataFrames* aient le même nombre de lignes. Pour `rbind`, les deux *DataFrames* doivent avoir le même nombre de colonnes.

Prenons à nouveau comme exemple le jeu de données *iris*. Nous allons commencer par le séparer en trois sous-jeux de données comprenant chacun une espèce d'*iris*. Puis, nous fusionnerons deux d'entre eux avec la fonction `rbind`.

```
iris1 <- subset(iris, iris$Species == "virginica")
iris2 <- subset(iris, iris$Species == "versicolor")
iris3 <- subset(iris, iris$Species == "setosa")

iris_comb <- rbind(iris2,iris3)
```

Nous pourrions aussi extraire dans les deux *DataFrames* les colonnes comprenant le mot *Length* et le mot *Width*, puis les fusionner.

```
iris_l <- iris[names(iris)[grepl("Length",names(iris), fixed = TRUE)]]
iris_w <- iris[names(iris)[grepl("Width",names(iris), fixed = TRUE)]]

iris_comb <- cbind(iris_l,iris_w)
names(iris_comb)

## [1] "Sepal.Length" "Petal.Length" "Sepal.Width"   "Petal.Width"
```

1.4.2.9.2 Joindre des *DataFrame*

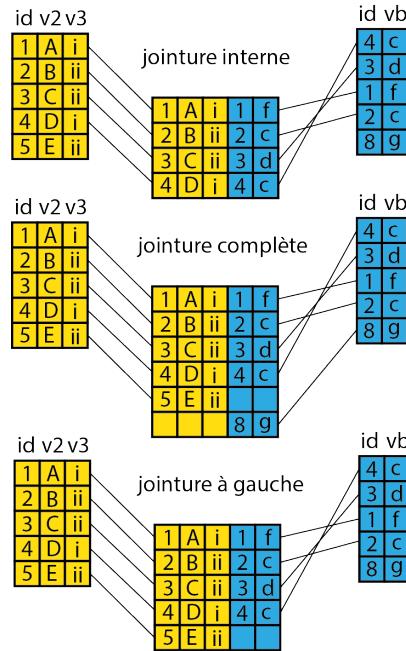
Une jointure est une opération un peu plus complexe qu'un simple ajout. L'idée est d'associer des informations de plusieurs *DataFrames* en utilisant une colonne (appelée une clef) présente dans les deux jeux de données. On distingue plusieurs types de jointure :

- Les jointures internes permettant de combiner les éléments communs entre un *DataFrame* A et B
- La jointure complète permettant de combiner les éléments présents dans A ou B
- La jointure à gauche, permettant de ne conserver que les éléments présents dans A même s'ils ne trouvent pas leur correspondance dans B.

Ces trois jointures sont présentées à la figure 1.16; dans ces trois cas, la colonne commune se nomme *id*.

Vous noterez que les deux dernières jointures peuvent produire des valeurs manquantes. Pour réaliser ces opérations, il est possible d'utiliser la fonction `merge`. Prenons un exemple simple à partir d'un petit jeu de données.

```
auteurs <- data.frame(
  name = c("Tukey", "Venables", "Tierney", "Ripley", "McNeil", "Apparicio"),
  nationality = c("US", "Australia", "US", "UK", "Australia", "Canada"),
  retired = c("yes", rep("no", 5)))
livres <- data.frame(
  aut = c("Tukey", "Venables", "Tierney", "Ripley", "Ripley", "McNeil", "Wickham"),
  title = c("Exploratory Data Analysis",
            "Modern Applied Statistics ...",
            "LISP-STAT",
            "Spatial Statistics", "Stochastic Simulation",
            "Interactive Data Analysis", "R for Data Science"))
```

**FIG. 1.17 :** Jointure de DataFrames

Nous avons donc deux *DataFrames*, le premier décrivant des auteurs et le second des livres. Effectuons une première jointure interne afin de savoir pour chaque livre la nationalité de son auteur et si ce dernier est à la retraite.

```
df1 <- merge(livres, auteurs, #les deux DataFrames
              by.x = "aut", by.y = 'name', #les noms des colonnes de jointures
              all.x = FALSE, all.y = FALSE)
```

```
print(df1)
```

```
##          aut                  title nationality retired
## 1  McNeil  Interactive Data Analysis    Australia     no
## 2  Ripley      Spatial Statistics        UK       no
## 3  Ripley  Stochastic Simulation        UK       no
## 4 Tierney        LISP-STAT        US       no
## 5  Tukey  Exploratory Data Analysis        US      yes
## 6 Venables Modern Applied Statistics ...    Australia     no
```

Cette jointure est interne car les deux paramètres `all.x` et `all.y` ont pour valeur `FALSE`. Ainsi, nous indiquons à la fonction que nous ne souhaitons ni garder tous les éléments du premier *DataFrame* ni tous les éléments du second, mais uniquement les éléments présents dans les deux. Vous noterez ainsi que le livre “R for Data Science” n'est pas présent dans le jeu de données final car son auteur “Wickham” ne fait pas partie du *DataFrame* `auteurs`. De même, l'auteur “Apparicio” n'apparaît pas dans la jointure, car aucun livre dans le *DataFrame* `books` n'a été écrit par cet auteur.

Pour conserver tous les livres, nous pouvons effectuer une jointure à gauche en renseignant `all.x = TRUE`. Nous allons ainsi forcer la fonction à garder tous les livres et mettre des valeurs vides aux informations manquantes des auteurs.

```
df2 <- merge(livres, auteurs, #les deux DataFrames
             by.x = "aut", by.y = 'name', #les noms des colonnes de jointures
             all.x = TRUE, all.y = FALSE)

print(df2)
```

	aut	title	nationality	retired
## 1	McNeil	Interactive Data Analysis	Australia	no
## 2	Ripley	Spatial Statistics	UK	no
## 3	Ripley	Stochastic Simulation	UK	no
## 4	Tierney	LISP-STAT	US	no
## 5	Tukey	Exploratory Data Analysis	US	yes
## 6	Venables	Modern Applied Statistics ...	Australia	no
## 7	Wickham	R for Data Science	<NA>	<NA>

Et pour garder tous les livres et tous les auteurs, nous pouvons faire une jointure complète en indiquant `all.x = TRUE` et `all.y = TRUE`.

```
df3 <- merge(livres, auteurs, #les deux DataFrames
             by.x = "aut", by.y = 'name', #les noms des colonnes de jointures
             all.x = TRUE, all.y = TRUE)

print(df3)
```

	aut	title	nationality	retired
## 1	Apparicio	<NA>	Canada	no
## 2	McNeil	Interactive Data Analysis	Australia	no
## 3	Ripley	Spatial Statistics	UK	no
## 4	Ripley	Stochastic Simulation	UK	no
## 5	Tierney	LISP-STAT	US	no
## 6	Tukey	Exploratory Data Analysis	US	yes
## 7	Venables	Modern Applied Statistics ...	Australia	no
## 8	Wickham	R for Data Science	<NA>	<NA>

1.5 Conclusion et ressources pertinentes

Voilà qui conclut ce chapitre sur les bases du langage R . Vous avez maintenant les connaissances nécessaires pour commencer à travailler. N'hésitez pas à revenir sur les différentes sous-sections au besoin! Pour aller plus loin dans l'apprentissage du langage, vous pouvez également vous plonger dans le chapitre R AVANCÉ. Cependant, nous vous recommandons de faire vos premiers pas avec cette base avant de vous lancer dans cette partie davantage orientée programmation. Quelques ressources pertinentes qui pourraient vous être utiles sont aussi reportées au tableau ci-dessous.

TAB. 1.7 : Ressources pertinente pour en apprendre plus sur R

Ressource	Description	Url
Rbloggers	Un recueil de nombreux blogues sur R : parfait pour être tenu au courant des nouveautés et faire des découvertes	https://www.r-bloggers.com
CRAN packages by date	Les derniers packages publiés sur CRAN : cela permet de garder un oeil sur les nouvelles fonctionnalités de ses packages préférés	https://cran.r-project.org/web/packages
Introduction à R et au TidyVerse	Une excellente ressource en français pour en apprendre plus sur le tidyverse	https://juba.github.io/tidyverse
Numyard	Une chaîne YouTube pour revoir les bases de R en vidéo	https://www.youtube.com/user/TheLearnR
Cheatsheets	Des feuilles de triche résumant les fonctionnalités de nombreux packages	https://rstudio.com/resources/cheatsheets

Deuxième partie

Analyses univariées

Chapitre 2

Statistiques descriptives univariées

Dans ce chapitre, nous décrirons la notion de variable, permettant l'opérationnalisation d'un concept. Comprendre les différents types de variables est essentiel en statistiques. En effet, en fonction du type de variable à l'étude, les tests d'hypothèse et les méthodes de statistique inférentielle que l'on pourra appliquer seront différents. Nous distinguerons ainsi cinq types de variables : nominale, ordinaire, discrète, continue et semi-quantitative. Aussi, nous abordons un concept central de la statistique : les distributions. Nous présenterons ensuite les différentes statistiques descriptives univariées qui peuvent s'appliquer à ces types de variables.



Dans cette section, nous utiliserons principalement les packages suivants (À MODIFIER PLUS TARD) :

- Pour créer des graphiques :
 - * **ggplot2**, le seul, l'unique
 - * **ggbubr** pour combiner des graphiques et réaliser des diagrammes
- Pour créer des distribution :
 - * **fitdistrplus** pour générer différentes distributions
 - * **actuar** pour la fonction de densité de Pareto
 - * **gamlss.dist** pour des distributions de Poisson
- Pour les statistiques descriptives :
 - **stats** pour les statistiques descriptives
 - **nortest** pour le test de Kolmogorov-Smirnov
 - **DescTools** pour les tests de Lilliefors, Shapiro-Wilk, Anderson-Darling et Jarque-Bera
 - Autres *packages* :
 - **Hmisc** et **Weighted.Desc.Stat** pour les statistiques descriptives pondérées
 - **foreign** pour importer des fichiers externes

2.1 Notion de variable

2.1.1 La variable : l'opérationnalisation d'un concept

Une variable permet d'opérationnaliser un concept, soit une « idée générale et abstraite que se fait l'esprit humain d'un objet de pensée concret ou abstrait, et qui lui permet de rattacher à ce même objet les diverses

perceptions qu'il en a, et d'en organiser les connaissances» (Larousse¹). Pour valider un modèle théorique, il convient alors d'opérationnaliser ses différentes concepts et d'établir les relations qu'ils partagent. L'opérationnalisation d'un concept nécessite soit de mesurer (dans un intervalle de valeurs, c'est-à-dire de manière quantitative), soit de qualifier (avec plusieurs catégories, c'est-à-dire de manière qualitative) un phénomène.

Selon Statistique Canada², «une variable est une caractéristique d'une unité statistique que l'on observe et pour laquelle une valeur numérique ou une catégorie d'une classification peut être attribuée». Il convient alors de bien saisir à quelle unité statistique (ou unité d'observation) s'applique les valeurs d'une variable : des personnes, des ménages, des municipalités, etc.

Prenons deux exemples concrets tirées du Recensement de 2016 de Statistique Canada :

- Le concept **famille de recensement** est défini comme étant «un couple marié et les enfants, le cas échéant, du couple et/ou de l'un ou l'autre des conjoints; un couple en union libre et les enfants, le cas échéant, du couple et/ou de l'un ou l'autre des partenaires; ou un parent seul, peu importe son état matrimonial, habitant avec au moins un enfant dans le même logement et cet ou ces enfants. Tous les membres d'une famille de recensement particulière habitent le même logement. Un couple peut être de sexe opposé ou de même sexe. Les enfants peuvent être des enfants naturels, par le mariage, par l'union libre ou par adoption, peu importe leur âge ou leur état matrimonial, du moment qu'ils habitent dans le logement sans leur propre conjoint marié, partenaire en union libre ou enfant. Les petits-enfants habitant avec leurs grands-parents, alors qu'aucun des parents n'est présent, constituent également une famille de recensement» (Statistique Canada³). À partir de cette définition, les familles de recensement peuvent être qualifiées selon plusieurs modalités : couples mariés sans enfant, couples mariés avec enfants, couples en union libre sans enfant, couples en union libre avec enfant, famille monoparentale (avec un parent de sexe féminin), famille monoparentale (avec un parent de sexe masculin).
- Le concept de **revenu d'emploi** est défini comme étant «tous les revenus reçus sous forme de traitements, salaires et commissions d'un travail rémunéré ou le revenu net d'un travail autonome dans une entreprise agricole ou non agricole non constituée en société et/ou dans l'exercice d'une profession au cours de la période de référence. Pour le Recensement de 2016, la période de référence est l'année civile 2015 pour toutes les variables de revenu» (Statistique Canada⁴). Il est donc mesurée en dollars pour chaque individu de 15 ans et plus. Pour l'ensemble de la population de 15 ans et plus, il peut ensuite être classé en déciles de revenu d'emploi, soit en dix groupes (Statistique Canada⁵).

Maîtriser la définition des variables que vous utilisez : un enjeu crucial!

Nous avons vu qu'une variable est l'opérationnalisation d'un concept. Par conséquent, ne pas maîtriser la définition d'une variable revient à ne pas bien saisir le concept sous-jacent qu'elle tente de mesurer. Si vous exploitez des données secondaires – par exemple, issues d'un recensement de population ou d'une enquête longitudinale ou transversale –, il faut impérativement lire les définitions des variables que vous souhaiteriez utiliser. Ne pas le faire risque d'aboutir à :

- Une mauvaise opérationnalisation de votre modèle théorique, même si votre analyse est bien menée statistiquement parlant. Autrement dit, vous risquez de ne pas sélectionner les bonnes variables. Prenons un exemple concret. Vous avez construit un modèle théorique dans lequel vous souhaitez inclure un concept sur la langue des personnes. Dans le recensement canadien de 2016, plusieurs variables relatives à la langue sont disponibles : connaissance des langues officielles⁶, langue parlée

¹<https://www.larousse.fr/dictionnaires/francais/concept/17875?q=concept#17749>

²<https://www.statcan.gc.ca/fra/concepts/variable>

³<https://www12.statcan.gc.ca/census-recensement/2016/ref/dict/fam004-fra.cfm>

⁴<https://www12.statcan.gc.ca/census-recensement/2016/ref/dict/pop027-fra.cfm>

⁵<https://www12.statcan.gc.ca/census-recensement/2016/ref/dict/pop204-fra.cfm%22%7D>

à la maison⁷, langue maternelle⁸, première langue officielle parlée⁹, connaissance des langues non officielles¹⁰ et langue de travail¹¹ (Statistique Canada, 2019¹²). La sélection de l'une de ces variables doit être faite de manière rigoureuse, c'est-à-dire en lien avec votre cadre théorique et suite à une bonne compréhension des définitions des variables. Dans une étude sur le marché du travail, on sélectionnerait probablement la variable sur *la connaissance des langues officielles du Canada*, afin d'évaluer son effet sur l'employabilité, toutes choses étant égales par ailleurs. Dans une autre étude portant sur la réussite ou la performance scolaire, il est probable qu'on utilise plutôt la *langue maternelle*.

- Une mauvaise interprétation et discussion de vos résultats en lien avec votre cadre théorique.
- Une mauvaise identification des pistes de recherche.

Finalement, la définition d'une variable peut évoluer à travers plusieurs recensements de population : la société évolue, les variables aussi ! Par conséquent, si vous comptez utiliser plusieurs années de recensement dans une même étude, assurez-vous que les définitions des variables soient similaires d'un jeu de données à l'autre et qu'elles mesurent ainsi la même chose.

Comprendre les variables utilisées dans un article scientifique : un exercice indispensable dans l'élaboration d'une revue de littérature

Une lecture rigoureuse d'un article scientifique suppose, entre autres, de bien comprendre les concepts et variables mobilisés. Il convient alors de lire attentivement la section méthodologique (pas uniquement la section des résultats ou pire le résumé), sans quoi vous risquez d'aboutir à une revue de littérature approximative. Ayez aussi un **regard critique** sur les variables visant à opérationnaliser les concepts clés de l'étude. Certains concepts sont très difficiles à traduire en variables ; leurs opérationnalisations (mesures) peuvent ainsi faire l'objet de vifs débats parmi les chercheurs. Très succinctement, c'est notamment le cas du concept de capital social. D'une part, les définitions et ancrages sont biens différents selon Bourdieu (sociologue, ancrage au niveau des individus) et Putman (politologue, ancrage au niveau des collectivités) ; d'autre part, aucun consensus ne semble clairement se dégager quant à la définition de variables permettant de le mesurer efficacement (de manière quantitative).

Variable de substitution (*proxy variable* en anglais)

On fait la moins pire des recherches ! En effet, les données disponibles sont parfois imparfaites pour répondre avec précision à une question de recherche ; on peut toujours les exploiter, tout en signalant honnêtement leurs faiblesses et limites, et ce, tant pour les données que les variables utilisées.

- Des bases de données peuvent être en effet imparfaites. Par exemple, en criminologie, des chercheur.e.s exploitant des données policières signalent habituellement la limite du **chiffre noir** : les données policières comprennent uniquement les crimes et délits découverts par la police et occultent ainsi les crimes non-découverts ; ils ne peuvent ainsi refléter la criminalité réelle sur un territoire donné.
- Des variables peuvent aussi être imparfaites. Dans un jeu de données, il est fréquent qu'une variable opérationnalisant un concept précis ne soit pas disponible ou qu'elle n'ait tout simplement pas été mesurée. On cherchera alors une variable de substitution (*proxy*) pour la remplacer. Prenons un exemple concret portant sur l'exposition des cyclistes à la pollution atmosphérique ou au bruit environnemental. L'un des principaux facteurs d'exposition à ces pollutions est le trafic routier : plus ce dernier est élevé, plus le cycliste risque de rouler dans un environnement bruyant et pollué. Toutefois, il est rare de disposer de mesures du trafic en temps réel qui nécessitent des comptages de véhicules pendant le trajet des cyclistes (par exemple, à partir de vidéos captées par une caméra fixée sur le guidon). Pour pallier à l'absence de mesures directes, plusieurs auteurs utilisent des variables de substitution de la densité du trafic, comme la typologie des types d'axes (primaire, secondaire, tertiaire, rue locale, etc.), supposant ainsi qu'un axe primaire supporte un volume de véhicules supérieur à un axe secondaire.

2.1.2 Les types de variables

On distingue habituellement les variables qualitatives (nominale ou ordinaire) des variables quantitatives (discrète ou continue). Tel qu'illustré à la figure 2.1, l'opérationnalisation du concept en variable est réalisée par différents mécanismes visant à qualifier, classer, compter ou mesurer afin de caractériser les unités statistiques (observations) d'une population ou d'un échantillon.

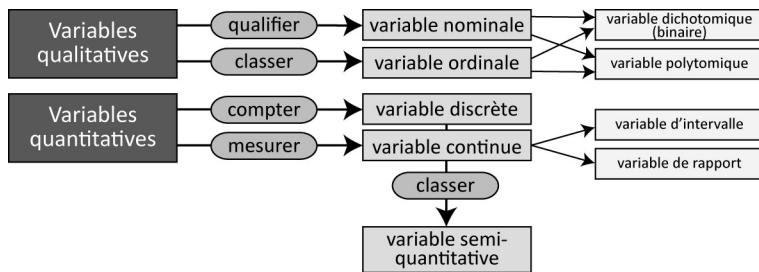


FIG. 2.1 : Les types de variables

2.1.2.1 Les variables qualitatives

Une variable nominale permet de **qualifier** des observations (individus) à partir de plusieurs catégories dénommées modalités. Par exemple, la variable *couleur des yeux* pourrait comprendre les modalités *bleu*, *marron*, *vert* tandis que les *types de familles* compendrait les modalités *couple marié*, *couple en union libre* et *famille monoparentale*.

Une variable ordinaire permet de classer des observations à partir de plusieurs modalités hiérarchisées. L'exemple le plus connu est certainement l'échelle de Likert, très utilisée dans les sondages évaluant le degré d'accord d'une personne à une affirmation avec les modalités suivantes : *tout à fait d'accord, d'accord, ni en désaccord ni d'accord, pas d'accord et pas du tout d'accord*. Une multitude de variantes sont toutefois possibles pour classer la fréquence d'un phénomène (*Très souvent, souvent, parfois, rarement, jamais*), l'importance accordée à un phénomène (*Pas du tout important, peu important, plus ou moins important, important, très important*) ou la proximité perçue d'un lieu (*très éloigné, loin, plus ou moins proche, proche, très proche*).

En fonction du nombre de modalités qu'elle comprend, une variable qualitative (nominale ou ordinale) est soit **dichtomique (binaire)** (deux modalités), soit **polytomique** (plus de deux modalités). Par exemple, dans le recensement canadien, le *sexe* est une variable binaire (avec les modalités *sexe masculin*, *sexe féminin*), tandis que le *genre* est une variable polytomique (avec les modalités *genre masculin*, *genre féminin* et *diverses identités de genre*).



Les variables nominales et ordinaires sont habituellement encodées avec des valeurs numériques entières (par exemple, 1 pour *couple marié*, 2 pour *couple en union libre* et 3 pour *famille monoparentale*). Toutefois, aucune opération arithmétique (moyenne ou écart-type par exemple) n'est possible sur ces valeurs. Dans R, on utilisera un facteur pour attribuer un intitulé à chacune des valeurs numériques de la variable qualitative :

```
df$Famille <- factor(df$Famille, c(1,2,3), labels = c("couple marié","couple en union libre", "famille monoparentale"))
```

On calculera toutefois les fréquences des différentes modalités pour une variable nominale ou ordinale. Il est aussi possible de calculer la médiane sur une variable ordinaire.

2.1.2.2 Les variables quantitatives

Une variable discrète permet de **compter** un phénomène dans un ensemble fini de valeurs, comme le nombre d'accidents impliquant un cycliste à une intersection sur une période de cinq ans ou encore le nombre de vélos en libre service disponibles à une station. Il existe ainsi une variable binaire sous-jacente : la présence ou non d'un accident à l'intersection ou d'un vélo ou non à la station pour laquelle on opère un comptage. Habituellement, une variable discrète ne peut prendre que des valeurs entières (sans décimales), comme le nombre de personnes fréquentant un parc.

Une variable continue permet de **mesurer** un phénomène avec un nombre infini de valeurs réelles (avec décimales) dans un intervalle donné. Par exemple, une variable relative à la distance de dépassement d'un cycliste par un véhicule motorisé pourrait varier de 0 à 5 mètres ($X \in [0, 5]$); toutefois cette distance peut être de 0,759421 ou de 4,785612 mètres. Le nombre de décimales de la valeur réelle dépendra de la précision et de la fiabilité de la mesure. Pour un capteur de distance de dépassement, le nombre de décimales dépendra de la précision du lidar ou du sonar de l'appareil; aussi, l'utilisation de trois décimales – soit une précision au millimètre – est largement suffisant pour mesurer la distance de dépassement. Une variable continue est soit une variable d'intervalle, soit une variable de rapport. Les **variables d'intervalle** ont une échelle relative, c'est-à-dire que les intervalles entre les valeurs de la variables ne sont pas constants; elles n'ont pas de vrai zéro. Ces valeurs peuvent être manipulées uniquement par addition et soustraction et non par multiplication et division. La variable d'intervalle la plus connue est certainement celle de la température. S'il fait 10 degrés Celsius à Montréal et 30°C à Mumbai (soit 50 et 86 degrés en Fahrenheit), on peut affirmer qu'il y a 20°C ou 36°F d'écart entre les deux villes, mais on ne peut pas affirmer qu'il fait trois fois plus chaud à Mumbai. Presque toutes les mesures statistiques sur une variable d'intervalle peuvent être calculées, exceptés le coefficient de variation et la moyenne géométrique puisqu'il n'y a pas de vrai zéro et d'intervalles constants entre les valeurs. À l'inverse, les **variables de rapport** ont une échelle absolue, c'est-à-dire que les intervalles entre les valeurs sont constants et elles ont un vrai zéro. Elles peuvent ainsi être manipulées par addition, soustraction, multiplication et division. Par exemple, le prix d'un produit exprimé dans une unité monétaire ou la distance exprimée dans le système métrique sont des variables de rapport. Un vélo dont le prix affiché est de 1000\$ est bien deux fois plus cher qu'un autre à 500\$, une piste cyclable hors rue à 25 mètres du tronçon routier le plus proche est bien quatre fois plus proche qu'une autre à 100 mètres.

Une variable semi-quantitative, appelée aussi variable quantitative ordonnée, est une variable discrète ou continue dont les valeurs ont été regroupées en classes hiérarchisées. Par exemple, l'âge est une variable continue pouvant être transformée avec les groupes d'âge ordonnés suivants : *moins 25 ans, 25 à 44 ans, 45 à 64 ans et 65 ans et plus*.

2.2 Les types de données

Différents types de données sont utilisés en sciences sociales. L'objectif ici n'est pas de les décrire en détail, mais plutôt de donner quelques courtes définitions. En fonction de votre question de recherche et des bases des données disponibles ou non, il s'agira de sélectionner le ou les types de données les plus appropriés à votre sujet.

2.2.1 Données secondaires *versus* données primaires

Les **données secondaires** sont des données qui existent déjà au début de votre projet de recherche : pas besoin de les collecter, il suffit de les exploiter! Une multitude de données de recensements ou d'enquêtes de Statistique Canada sont disponibles et largement exploitées en sciences sociales (par exemple, l'enquête nationale auprès des ménages – ENM, l'enquête sur la dynamique du marché du travail et du revenu – EDTR, l'enquête longitudinale auprès des immigrants – ELIC, etc.).



Au Canada, les chercheurs (étudiants et professeurs) ont accès aux microdonnées des enquêtes de Statistique Canada dans les Centres de données de recherche (CDR). Vous pouvez consulter le moteur de recherche du (RCCDR¹³) afin d'explorer les différentes enquêtes disponibles.

Au Québec, l'accès à ces enquêtes est possible dans les différentes antennes du Centre interuniversitaire québécois de statistiques sociales de Statistique Canada (CIQSS¹⁴).

Par opposition, les **données primaires** n'existent pas quand vous démarrez votre projet : vous devez les collecter spécifiquement pour votre étude ! Par exemple, une chercheure souhaitant analyser l'exposition des cyclistes au bruit et à la pollution dans une ville donnée devra réaliser une collecte de données avec idéalement plusieurs participants (équipés de différents capteurs), et ce, sur plusieurs jours. Une collecte de données primaires est peut aussi être réalisée avec une enquête par sondage. Brièvement, réaliser une collecte de données primaires nécessite différentes phases complexes comme la définition de la méthode de collecte, de la population à l'étude, l'estimation de la taille de l'échantillon, la validation des outils de collecte avec une phase de test, la réalisation de la collecte, la structuration, la gestion et l'exploitation de données collectées. Finalement, dans le milieu académique, une collecte de données primaires auprès d'individus doit être approuvée par le comité d'éthique de la recherche de l'université à laquelle est affilié le responsable du projet de recherche (qu'il soit professeur, chercheur ou étudiant).

2.2.2 Données transversales *versus* données longitudinales

Les **données transversales** sont des mesures pour une période relativement courte. L'exemple classique est un jeu de données constitué des variables extraites d'un recensement de population pour une année donnée (comme celui 2016 de Statistique Canada).

Les **données longitudinales**, appelées aussi données par panel, sont des mesures répétées pour plusieurs observations au cours du temps (N observations pour T dates). Par exemple, des observations pourraient être des pays, les dates pourraient être différentes années (de 1990 à 2019) pour lesquelles différentes variables seraient disponibles (population totale, taux d'urbanisation, produit intérieur brut par habitant, émissions de gaz à effet de serre par habitant, etc.).

2.2.3 Données spatiales *versus* données aspatiales

Les observations des **données spatiales** sont des unités spatiales géoréférencées (points, lignes, polygones ou encore pixels d'une image). Elles peuvent être par exemple :

- des points (x, y) ou (*lat-long*) représentant des entreprises avec plusieurs variables (adresse, date de création, nombre d'employés, secteurs d'activité, etc.);
- les lignes représentant des tronçons de rues pour lesquels plusieurs variables sont disponibles (types d'axe, longueur en mètres, nombre de voies, débit journalier moyen annuel, etc.);
- des polygones délimitant des régions ou des arrondissements pour lesquels une multitude de variables sociodémographiques et socioéconomiques sont disponibles.

À l'inverse, aucune information spatiale n'est disponible pour des **données aspatiales**.

2.2.4 Données individuelles *versus* données agrégées

Comme son nom l'indique, pour des **données individuelles**, chaque observation correspond à un individu. Les microdonnées de recensement ou d'enquêtes, par exemple, sont des données individuelles pour lesquelles toute une série de variables est disponible. Une étude analysant les caractéristiques de chaque arbre d'un quartier nécessite aussi des données individuelles : l'information doit être disponible

pour chaque arbre. Pour les microdonnées des recensements canadiens, «chaque enregistrement au niveau de la personne comprend des identifiants (comme les identifiants du ménage et de la famille), des variables géographiques et des variables directes et dérivées tirées du questionnaire» (Statistique Canada¹⁵). Comme signalé plus haut, ces microdonnées de recensement ou d'enquêtes sont uniquement accessibles dans les Centres de données de recherche (CDR).

Les données individuelles peuvent être **agrégées** à un niveau supérieur. Prenons le cas de microdonnées d'un recensement. Les informations disponibles pour chaque individu sont agrégées par territoire géographique (province, région économique, division de recensement, subdivision de recensement, région et agglomération de recensement, secteurs de recensement, aires de diffusion, etc.) en fonction du lieu d'habitation des individus. Des sommaires statistiques – basés sur la moyenne, la médiane, la somme ou la proportion de chacune des variables mesurées au niveau individuel (âge, sexe, situation familiale, revenu, etc.) – sont alors construits pour ces différents découpages géographiques (Statistique Canada¹⁶).

L'agrégation n'est pas nécessairement géographique. En éducation, il est fréquent de travailler avec des données concernant les élèves, mais agrégées au niveau des écoles. La figure 2.2 donne un exemple simple d'agrégation de données individuelles.

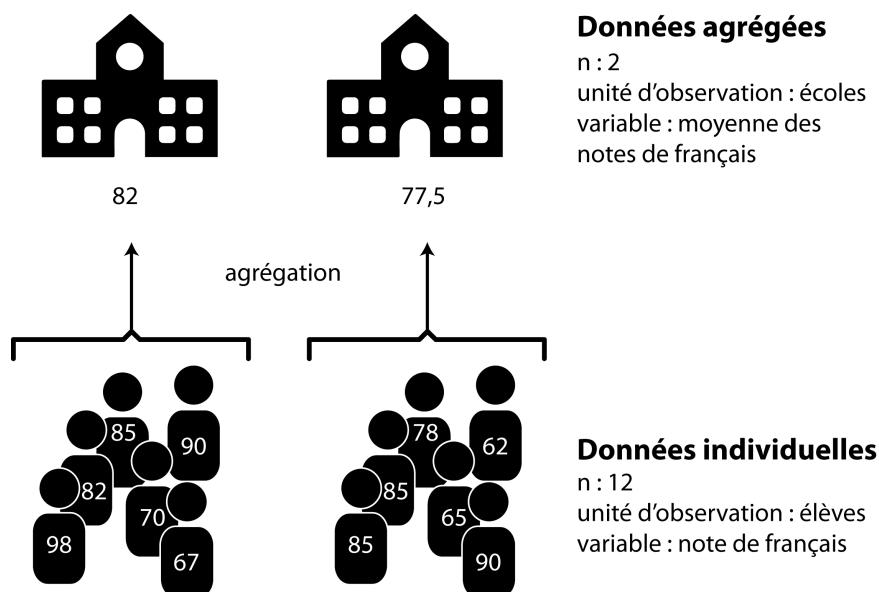


FIG. 2.2 : Exemple d'agrégation de données individuelles

Pour le cas de l'agrégation géographique, il convient alors de bien comprendre la hiérarchie des régions géographiques délimitées par l'organisme ou l'agence ayant la responsabilité de produire, gérer et diffuser les données des recensements et des enquêtes, puis de sélectionner le découpage géographique qui répond le mieux à votre question de recherche.



Pour le recensement de 2016 de Statistique Canada vous pourrez consulter :

- la hiérarchie des régions géographiques normalisées pour la diffusion¹⁷
- le glossaire illustré¹⁸ des régions géographiques
- les différents profils du recensement de 2016¹⁹ à télécharger pour les différentes régions géographiques.

¹⁵<https://www150.statcan.gc.ca/n1/pub/12-002-x/2012001/article/11642-fra.htm>

¹⁶<https://www.statcan.gc.ca/fra/idd/trousse/section5#a4>



Bien entendu, les différents types de données abordés ci-dessus ne sont pas exclusifs. Par exemple, des données pour des régions administratives extraites de plusieurs recensements sont en fait des données secondaires, spatiales, agrégées et longitudinales.

Une collecte de données sur la pollution atmosphérique et sonore réalisée à vélo (avec différents capteurs et un GPS) sont des données spatiales primaires.

2.3 Statistique descriptive et statistique inférentielle

2.3.1 Population, échantillon et inférence

Les notions de **population** et d'**échantillon** sont essentielles en statistique puisqu'elles sont le socle de l'inférence statistique. Un échantillon est un **sous-ensemble représentatif** d'une population donnée. Prenons un exemple concret. Une chercheure veut comprendre la mobilité des étudiants d'une université. Bien entendu, elle ne pourra interroger l'ensemble des étudiants de son université. Elle devra alors s'assurer d'obtenir un échantillon de taille suffisante et représentatif de la population étudiante. Une fois les données collectées (avec un sondage par exemple), elle pourra utiliser des techniques inférentielles pour analyser la mobilité des étudiants interrogés. Si son échantillon est représentatif, les résultats obtenus pourront être inférés – c'est-à-dire généralisés, extrapolés – à l'ensemble de la population.



Les méthodes d'échantillonnage

Nous n'abordons pas ici les méthodes d'échantillonnage. Sachez toutefois qu'il existe plusieurs méthodes probabilistes pour constituer un échantillon, notamment de manière aléatoire, systématique, stratifiée, par grappes (voir par exemple cette publique de Statistique Canada²⁰).

Autre exemple, une autre chercheure souhaite comprendre les facteurs influençant le sentiment de sécurité des cyclistes dans un quartier. De nouveau, elle ne pourra pas enquêter tous les cyclistes du quartier et devra constituer un échantillon représentatif. Par la suite, la mise en œuvre de techniques inférentielles lui permettra d'identifier les caractéristiques individuelles (âge, sexe, habiletés à vélo, etc.) et de l'environnement urbain (types de voies empruntés, niveaux de trafic, de pollution, de bruit, etc.) ayant des effets significatifs sur le sentiment de sécurité. Si l'échantillon est représentatif, les résultats pourront être généralisés à l'ensemble des cyclistes du quartier.

2.3.2 Deux grandes familles de méthodes statistiques

On distingue deux grandes familles de méthodes statistiques :

- « **La statistique descriptive et exploratoire** : elle permet, par des résumés et des graphiques plus ou moins élaborés, de décrire des ensembles de données statistiques, d'établir des relations entre les variables sans faire jouer de rôle privilégié à une variable particulière. Les conclusions ne portent dans cette phase de travail que sur les données étudiées, sans être inférées à une population plus large. L'analyse exploratoire s'appuie essentiellement sur des notions élémentaires telles que des indicateurs de moyenne et de dispersion, sur des représentations graphiques. [...]
- **La statistique inférentielle et confirmatoire** : elle permet de valider ou d'infirmer, à partir de tests statistiques ou de modèles probabilistes, des hypothèses formulées a priori (ou après une phase exploratoire), et d'extrapoler, c'est-à-dire d'étendre certaines propriétés d'un échantillon à une population plus large. Les conclusions obtenues à partir des données vont au-delà de ces données. La statistique confirmatoire fait surtout appel aux méthodes dites explicatives et prévisionnelles, destinées comme leurs noms l'indiquent, à expliquer puis à prévoir, suivant des règles de décision,

une variable privilégiée à l'aide d'une ou plusieurs variables explicatives (régressions multiples et logistiques, analyse de variance, analyse discriminante, segmentation, etc.)» (Lebart et al., 1995, p. 209).

2.4 La notion de distribution



Dans cette section, nous abordons un concept central de la statistique : les distributions. Prenez le temps de lire cette section à tête reposée et assurez-vous de bien comprendre chaque idée avant de passer à la suivante. N'hésitez pas à y revenir plusieurs fois si nécessaire, car la compréhension de ces concepts est essentielle pour utiliser adéquatement les méthodes que nous abordons dans ce livre.

2.4.1 Définitions générales

En statistique, on s'intéresse aux résultats d'expériences. Lancer un dé, mesurer la pollution atmosphérique, compter le nombre de collisions à une intersection, demander à une personne d'évaluer son sentiment de sécurité sur une échelle de 1 à 10 sont autant d'expériences pouvant produire des résultats.

Une distribution est une fonction permettant d'associer pour chaque résultat possible d'une expérience la probabilité d'obtenir ce résultat. En d'autres termes, il s'agit d'une fonction indiquant par exemple que pour l'expérience : «mesurer la concentration d'ozone à Montréal à 13h en été», la probabilité de mesurer une valeur inférieure à $15 \mu\text{g}/\text{m}^3$ est de seulement 2%.

Les distributions sont toujours définies dans un intervalle en dehors duquel elles sont indéfinies ; les valeurs dans cet intervalle sont appelées **l'espace d'échantillonnage**. Il s'agit donc des valeurs possibles que peut produire l'expérience. La somme des probabilités de l'ensemble des valeurs de l'espace d'échantillonnage est 1 (100%). Intuitivement, cela signifie que si l'on réalise l'expérience, on est obligé d'obtenir un résultat, et que cette probabilité totale est répartie entre tous les résultats possibles de l'expérience. En langage mathématique, on dit que l'intégrale des fonctions de distribution est 1 dans leur intervalle de définition.

Prenons un exemple concret avec l'expérience suivante : tirer à pile ou face avec une pièce de monnaie non truquée. Si l'on souhaite décrire la probabilité d'obtenir pile ou face, on peut utiliser une distribution qui aura comme espace d'échantillonnage [pile ; face] et ces deux valeurs auront chacune comme probabilité 0,5. Il est facile d'étendre cet exemple au cas d'un dé à six faces. La distribution de probabilité décrivant l'expérience «lancer le dé» a pour espace d'échantillonnage [1,2,3,4,5,6], chacune de ces valeurs étant associée à la probabilité 1/6.

Les deux distributions précédentes appartiennent à la famille des distributions **discrettes**. Elles servent à décrire des expériences dont le nombre de valeurs possibles est fini. Par opposition, la seconde famille de distributions regroupe les distributions **continues**, décrivant des expériences dont le nombre de résultats possibles est infini. Par exemple, mesurer la taille d'une personne adulte sélectionnée au hasard peut produire un nombre infini de valeurs comprises entre 50 cm et 280 cm. Les distributions sont utiles pour décrire les résultats attendus d'une expérience. Reprenons notre exemple du dé. Nous savons que chaque face a une chance sur six d'être tirée au hasard. Nous pouvons représenter cette distribution avec un graphique (figure 2.3).

Nous avons donc sous les yeux un modèle statistique décrivant le comportement attendu d'un dé, nous l'appelons la distribution **théorique**. Cependant, si nous effectuons l'expérience 10 fois (nous collectons donc un échantillon), nous obtiendrons une distribution différente de cette distribution théorique (figure 2.4).

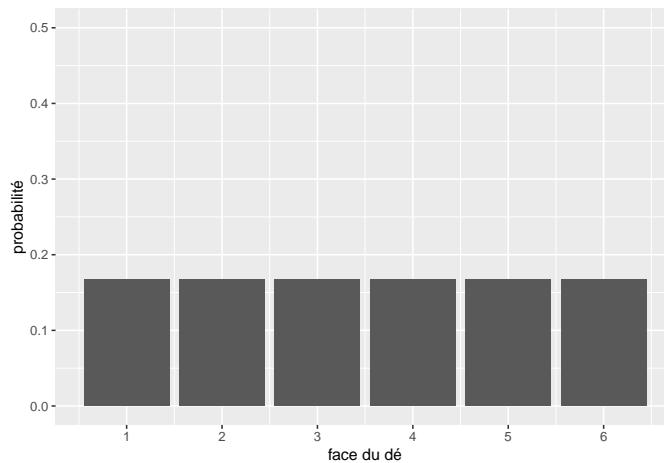


FIG. 2.3 : Distribution théorique d'un lancé de dé

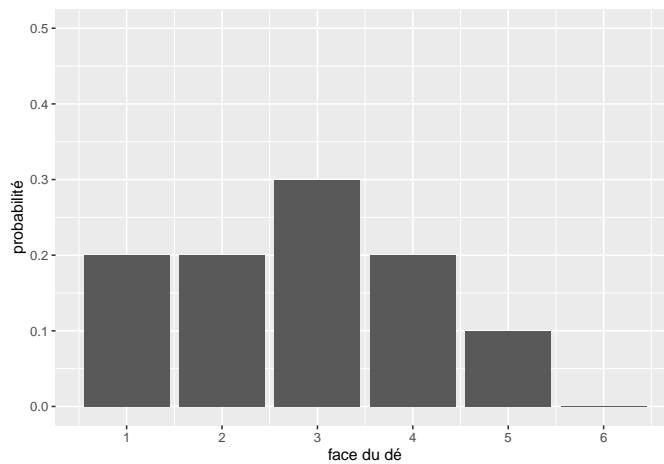


FIG. 2.4 : Distribution empirique d'un lancé de dé (n=10)

Nous appelons cette distribution la distribution **empirique**. Chaque échantillon aura sa propre distribution empirique. Cependant, comme le prédit la loi des grands nombres (ou théorème de Bernoulli) : si une expérience est répétée un grand nombre de fois, la probabilité empirique d'un résultat se rapproche de la probabilité théorique à mesure que le nombre de répétitions augmente. Pour nous en convaincre, collectons trois échantillons de lancer de dé de respectivement 30, 100 et 1000 observations (figure 2.5).

On constate bien qu'au fur et à mesure que la taille de l'échantillon augmente, on tend vers la distribution théorique. Ces dernières sont donc utilisées pour modéliser des phénomènes réels et sont à la base de presque tous les tests statistiques d'inférence fréquentiste ou bayésienne.

En pratique, la question que l'on se pose le plus souvent est : quelle distribution théorique peut le mieux décrire le phénomène empirique à l'étude ? Pour répondre à cette question, deux approches sont possibles :

- Considérant la littérature existante sur le sujet, les connaissances accumulées et la nature de la variable étudiée, il est possible de sélectionner des distributions théoriques pouvant vraisemblablement correspondre à la variable.
- Comparer visuellement ou à l'aide de tests statistiques la distribution empirique de la variable et diverses distributions théoriques pour trouver la plus adaptée.

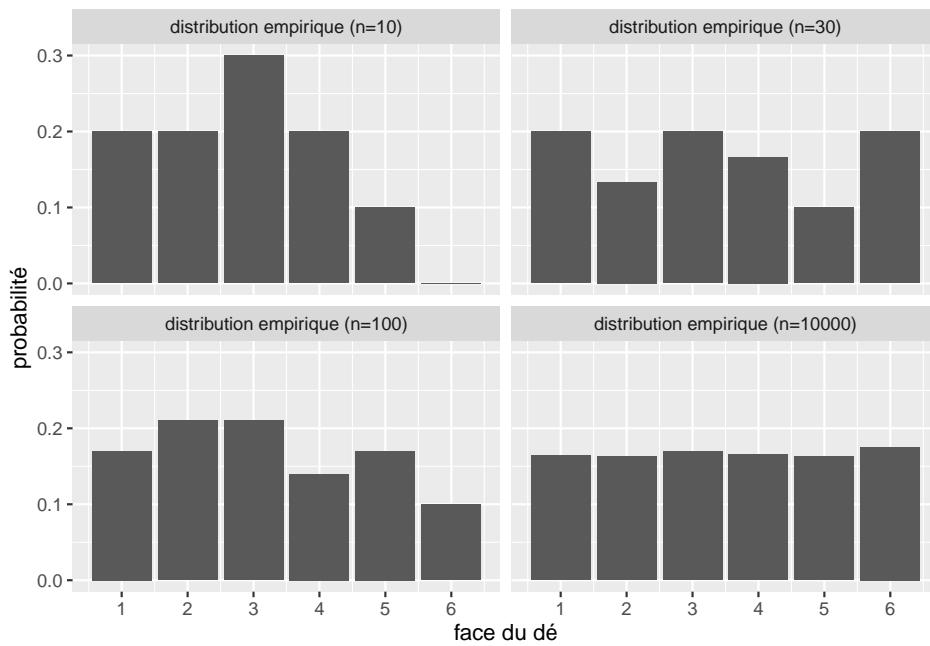


FIG. 2.5 : Distribution empirique d'un lancé de dé (n=10)

Idéalement, le choix d'une distribution théorique devrait reposer sur ces deux méthodes combinées.

2.4.2 Anatomie d'une distribution

Puisqu'une distribution est une fonction, il est possible de la représenter à l'aide d'une formule mathématique (appelée **fonction de masse** pour les distributions discrètes et **fonction de densité** pour les distributions continues). Prenons un premier exemple concret avec la distribution théorique associée au lancer de pièce de monnaie : la distribution de **Bernoulli**. Sa formule est la suivante :

$$f(x; p) = \begin{cases} q = 1 - p & \text{si } x = 0 \\ p & \text{si } x = 1 \end{cases} \quad (2.1)$$

avec p la probabilité d'obtenir $x = 1$ (pile), et $1-p$ la probabilité d'avoir $x = 0$ (face). La distribution de Bernoulli ne dépend que d'un paramètre : p . Avec différentes valeurs de p , on peut obtenir différentes formes pour la distribution de Bernoulli. Si $p = 1/2$, la distribution de Bernoulli décrit parfaitement l'expérience : obtenir pile à un lancer de pièce de monnaie. Si $p = 1/6$, elle décrit alors l'expérience : obtenir 4 (tout comme n'importe quelle valeur de 1 à 6) à un lancer de dé. Pour un exemple plus appliqué, la distribution de Bernoulli est utilisée en analyse spatiale pour étudier la concentration d'accidents de la route ou de crimes en milieu urbain. En chaque endroit du territoire, il est possible de calculer la probabilité qu'un tel événement ait lieu ou non en se basant sur les données observées et cette distribution. La distribution continue la plus simple à décrire est certainement la distribution **uniforme**. Il s'agit d'une distribution un peu spéciale puisqu'elle attribue la même probabilité à toutes ses valeurs dans son espace d'échantillonnage. Elle est définie sur l'intervalle $[-\infty; +\infty]$ et a la fonction de densité suivante :

$$f(x; a; b) = \begin{cases} \frac{1}{b-a} & \text{si } a \leq x \leq b \\ 0 & \text{sinon} \end{cases} \quad (2.2)$$

La fonction uniforme a donc deux paramètres, a et b , représentant respectivement les valeurs maximale et minimale au-delà desquelles les valeurs ont une probabilité 0 d'être obtenues. Pour avoir une meilleure

intuition de ce que décrit une fonction de densité, il est intéressant de la représenter avec un graphique (figure 2.6). Notez que sur ce graphique, l'axe des ordonnées n'indique pas précisément la probabilité associée à chaque valeur car celle-ci serait infinidécimale. Il sert uniquement à représenter la densité de la fonction de distribution.

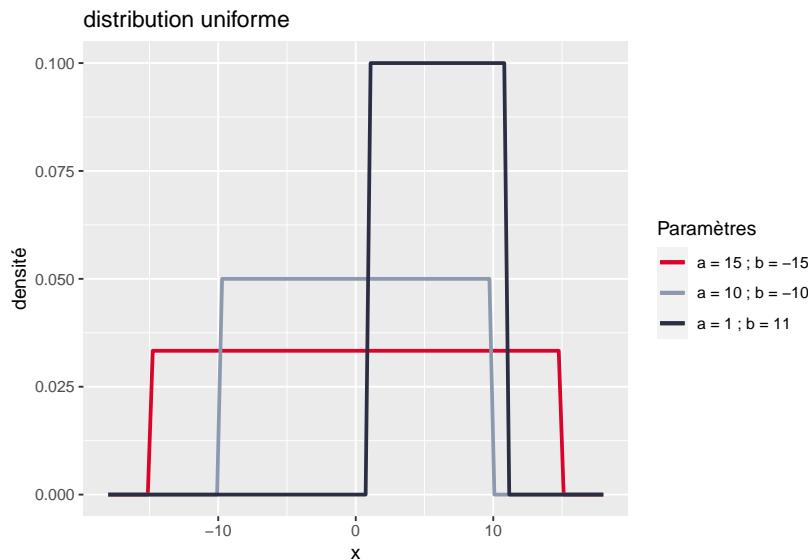


FIG. 2.6 : Distributions uniformes continues

On observe clairement que toutes les valeurs de x entre a et b ont la même probabilité pour chacune de trois distributions uniformes présentées dans le graphique. Plus l'étendue est grande ($a - b$), plus l'espace d'échantillonnage est grand et plus la probabilité totale est répartie dans cet espace. Cette distribution serait donc idéale pour décrire un phénomène pour lequel chaque valeur a autant de chance de se produire qu'une autre. Prenons pour exemple un cas fictif avec un jeu de hasard qui vous proposerait la situation suivante : en tirant sur la manette d'une machine à sous, un nombre est tiré aléatoirement entre -60 et +50. Si le nombre est négatif, vous perdez de l'argent et inversement si le nombre est positif. Nous pouvons représenter cette situation avec une distribution uniforme continue et l'utiliser pour calculer quelques informations essentielles :

1. Selon cette distribution, quelle est la probabilité de gagner de l'argent lors d'un tirage ($x > 0$)?
2. Quelle est la probabilité de perdre de l'argent? ($x < 0$)?
3. Si je perds moins de 30\$ au premier tirage, quelle est la probabilité que ai-je d'au moins récupérer ma mise au second tirage ($x > 30$)?

Il est assez facile de calculer ces probabilités en utilisant la fonction `punif` dans R. Concrètement, cela permet de calculer l'intégrale de la fonction de masse sur un intervalle donné.

```
# Probabilité d'obtenir une valeur supérieure ou égale à 0
punif(0,min = -60, max = 50)
```

```
## [1] 0.5454545
```

```
# Probabilité d'obtenir une valeur inférieure à 0
punif(0,min = -60, max = 50, lower.tail = F)
```

```
## [1] 0.4545455
```

```
# Probabilité d'obtenir une valeur supérieure à 30
punif(30, min = -60, max = 50,lower.tail = F)
```

```
## [1] 0.1818182
```

Les paramètres permettent donc d'ajuster la fonction de masse ou de densité d'une distribution afin de lui permettre de prendre des formes différentes. Certains paramètres vont changer la localisation de la distribution (la déplacer vers la droite ou la gauche de l'axe des X), d'autres son degré de dispersion (distribution pointue ou aplatie) ou encore sa forme (symétrie). Les différents paramètres d'une distribution correspondent donc à sa carte d'identité et donnent une idée précise sur sa nature.

2.4.3 Principales distributions

Il existe un très grand nombre de distributions théoriques et parmi elles, de nombreuses sont en fait des cas spéciaux d'autres distributions. Pour un petit aperçu du bestiaire, vous pouvez faire un saut à la page Univariate Distribution Relationships²¹, qui liste près de 80 distributions.

Nous nous concentrons ici sur une sélection de 18 distributions très répandues en sciences sociales. La figure ?? présente graphiquement leurs fonctions de masse et de densité présentées dans cette section. Notez que ces graphiques correspondent tous à une forme possible de chaque distribution. En modifiant leurs paramètres, il serait possible de produire une figure très différente. Les distributions discrètes sont représentées avec des graphiques en barres, et les distributions continues avec des graphiques de densité.

2.4.3.1 La distribution uniforme discrète

Nous avons déjà abordé cette distribution dans les exemples précédents. Elle permet de décrire un phénomène dont tous les résultats possibles ont exactement la même probabilité de se produire. L'exemple classique est bien sûr un lancer de dé.

2.4.3.2 La distribution de Bernoulli

La distribution de Bernoulli permet de décrire une expérience pour laquelle deux résultats sont possibles. Son espace d'échantillonnage est donc $[0; 1]$. Sa fonction de masse est la suivante :

$$f(x; p) = \begin{cases} q = 1 - p & \text{si } x = 0 \\ p & \text{si } x = 1 \end{cases} \quad (2.3)$$

avec p , la probabilité d'obtenir $x = 1$ (réussite) et donc $1-p$, la probabilité d'avoir $x = 0$ (échec). La distribution de Bernoulli ne dépend que d'un paramètre : p contrôlant la probabilité de réussite de l'expérience. Notez que si $p = 1/2$, alors la distribution de Bernoulli est également une distribution uniforme. Un exemple d'application de la distribution de Bernoulli en études urbaines serait la modélisation de la survie d'un cycliste (1 pour survie, 0 pour décès) lors d'une collision avec une voiture selon une vitesse donnée.

2.4.3.3 La distribution binomiale

La distribution binomiale est utilisée pour caractériser une somme de distributions de Bernoulli. Un exemple simple serait l'accumulation des lancers d'une pièce de monnaie. Si l'on compte le nombre de fois

²¹ <http://www.math.wm.edu/~leemis/chart/UDR/UDR.html>

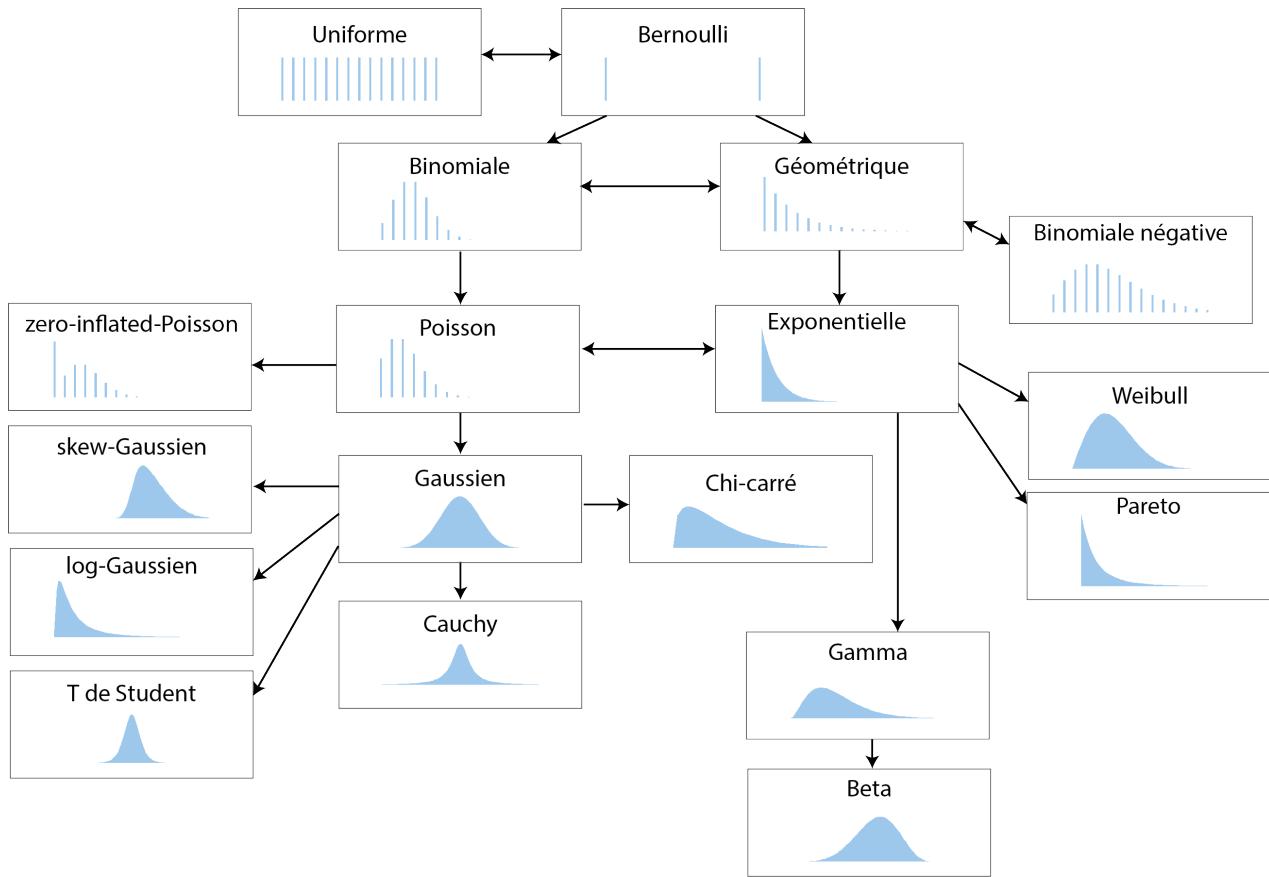


FIG. 2.7 : 18 distributions essentielles, design inspiré de Sean (2018)

où l'on fait pile, cette expérience est décrite par une distribution binomiale. Son espace d'échantillonnage est donc $[0; +\infty[$ (limité aux nombres entiers). Sa fonction de masse est la suivante :

$$f(x; n) = \binom{n}{x} p^x (1-p)^{n-x} \quad (2.4)$$

avec x le nombre de tirages réussis sur n essais avec une probabilité p de réussite à chaque tirage. Pour reprendre l'exemple précédent concernant les accidents de la route, une distribution binomiale permettrait de représenter la distribution du nombre de cyclistes survivants sur dix accidents impliquant une voiture à une intersection.

2.4.3.4 La distribution géométrique

La distribution géométrique permet de représenter le nombre de tirages nécessaires avec une distribution de Bernoulli avant d'obtenir une réussite. Par exemple, avec un lancer de dé, l'idée serait de compter le nombre de lancers nécessaires avant de tomber sur un 6. Son espace d'échantillonnage est donc $[1; +\infty[$ (limité aux nombres entiers). Sa distribution de masse est la suivante :

$$f(x; p) = (1-p)^x p \quad (2.5)$$

avec x le nombre de tentatives avant d'obtenir une réussite, $f(x)$ la probabilité que le premier succès n'arrive qu'après x tentatives et p la probabilité de réussite à chaque tentative. Cette distribution est

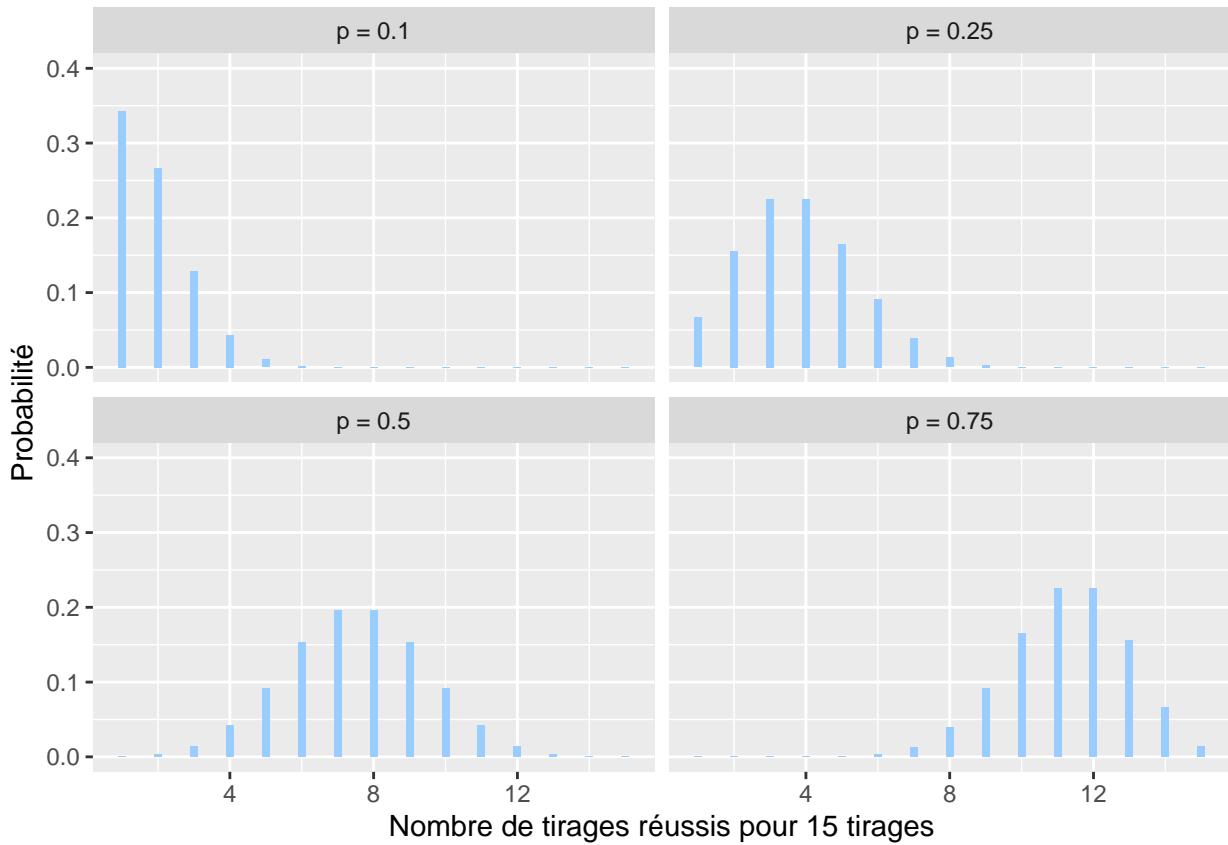


FIG. 2.8 : La distribution binomiale

notamment utilisée en marketing pour modéliser le nombre d'appels nécessaires avant de réussir une vente.

2.4.3.5 La distribution binomiale négative

La distribution binomiale négative est proche de la distribution géométrique. Elle permet de représenter le nombre de tentatives nécessaires afin d'obtenir un nombre n de réussites $[1; +\infty[$ (limité aux nombres entiers positifs). Sa formule est la suivante :

$$f(x; n; p) = \binom{x+n-1}{n} p^n (1-p)^x \quad (2.6)$$

avec x le nombre de tentatives avant d'obtenir n réussites et p la probabilité d'obtenir une réussite à chaque tentative. Cette distribution pourrait être utilisée pour modéliser le nombre de questionnaires x à envoyer pour une enquête si l'on espère au moins n réponses, sachant que la probabilité d'une réponse est p .

2.4.3.6 La distribution de poisson

La distribution de poisson est utilisée pour modéliser des comptages. Son espace d'échantillonnage est donc $[0; +\infty[$ (limité aux nombres entiers positifs). Par exemple, il est possible de compter à une intersection le nombre de collisions entre des automobilistes et des cyclistes sur une période donnée. Cet exemple devrait vous faire penser à la distribution binomiale vue plus haut. En effet, il serait possible de noter chaque rencontre entre une voiture et un cycliste et de considérer que leur collision est une

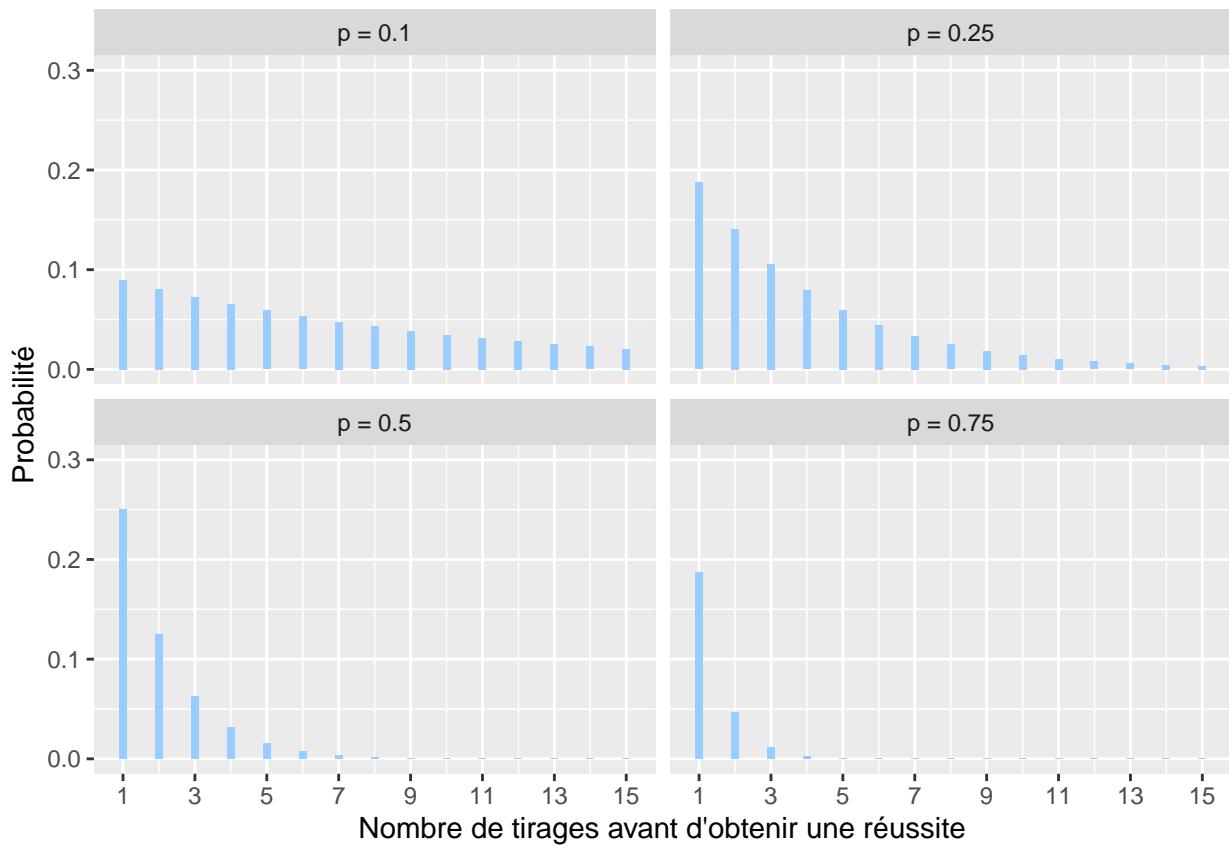


FIG. 2.9 : La distribution géométrique

«réussite» (0 : pas d'accidents, 1 : accident). Cependant, ce type de données serait fastidieux à collecter comparativement au simple comptage des accidents. La distribution de poisson à une fonction de densité avec un seul paramètre λ (lambda) et est décrite par la formule suivante :

$$f(x; \lambda) = \frac{\lambda^x}{x!} e^{-\lambda} \quad (2.7)$$

avec x le nombre de cas, $f(x)$ la probabilité d'obtenir x sachant λ . λ peut être vue comme le taux moyen d'occurrences (nombre d'événements divisé par la durée totale de l'expérience). Il permet à la fois de caractériser le centre et la dispersion de la distribution. Notez également que plus le paramètre λ augmente, plus la distribution de poisson tend vers une distribution normale.

2.4.3.7 La distribution de poisson avec excès de zéros

Il arrive régulièrement qu'une variable de comptage mesurée produise un très grand nombre de zéros. Prenons pour exemple le nombre de seringues de drogue injectable par tronçon de rue ramassées sur une période d'un mois. À l'échelle de toute une ville, un très grand nombre de tronçons n'auront tout simplement aucune seringue et dans ce contexte, la distribution classique de poisson n'est pas adaptée. On lui préfère alors sa version avec une inflation de zéros qui inclut un paramètre contrôlant la forte présence de zéros. Sa fonction de densité est la suivante :

$$f(x; \lambda; p) = (1 - p) \frac{\lambda^x}{x!} e^{-\lambda} \quad (2.8)$$

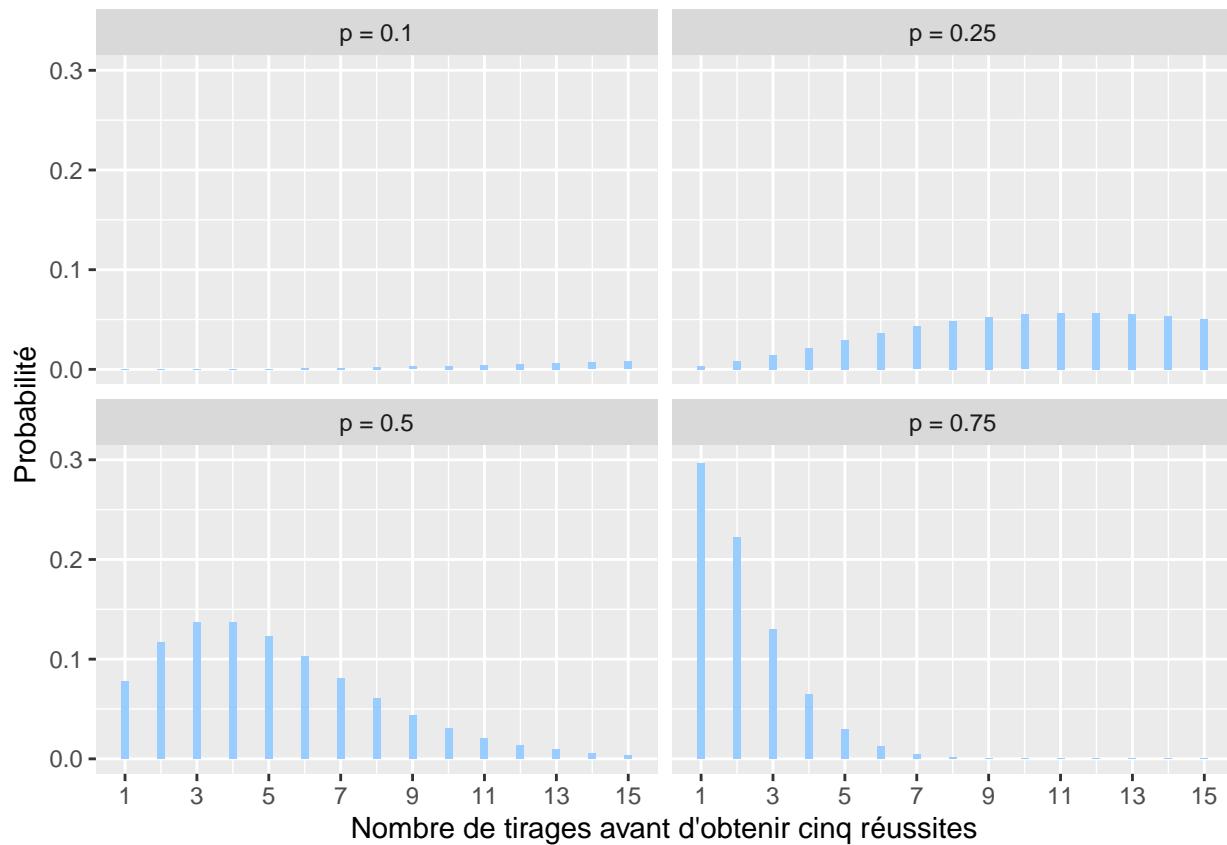


FIG. 2.10 : La distribution binomiale négative

Plus exactement, la distribution de poisson avec excès de zéro (zero-inflated en anglais) est une combinaison de deux processus générant des zéros. En effet, un zéro peut être produit par la distribution de poisson originale (aussi appelé vrai zéro) ou alors par le processus menant à la surreprésentation des 0 dans le jeu de données, capturée par la probabilité p (faux zéro). p est donc le paramètre contrôlant la probabilité d'obtenir un zéro, indépendamment du phénomène étudié.

```
## Loading required package: stats4
## Loading required package: splines
##
## Attaching package: 'VGAM'
## The following object is masked from 'package:tidyverse':
##     fill
```

2.4.3.8 La distribution gaussienne

Plus communément appelée la distribution normale, la distribution gaussienne est utilisée pour représenter des variables continues centrées sur leur moyenne. Son espace d'échantillonnage est $]-\infty; +\infty[$. Cette distribution joue un rôle central en statistique. Le théorème central limite stipule que la somme d'un grand nombre de distributions tend généralement vers une distribution normale. Autrement dit, lorsque nous répétons une même expérience et que nous conservons les résultats de ces expériences, la distribution du résultat de ces expériences tend vers la normalité. Ceci s'explique par le

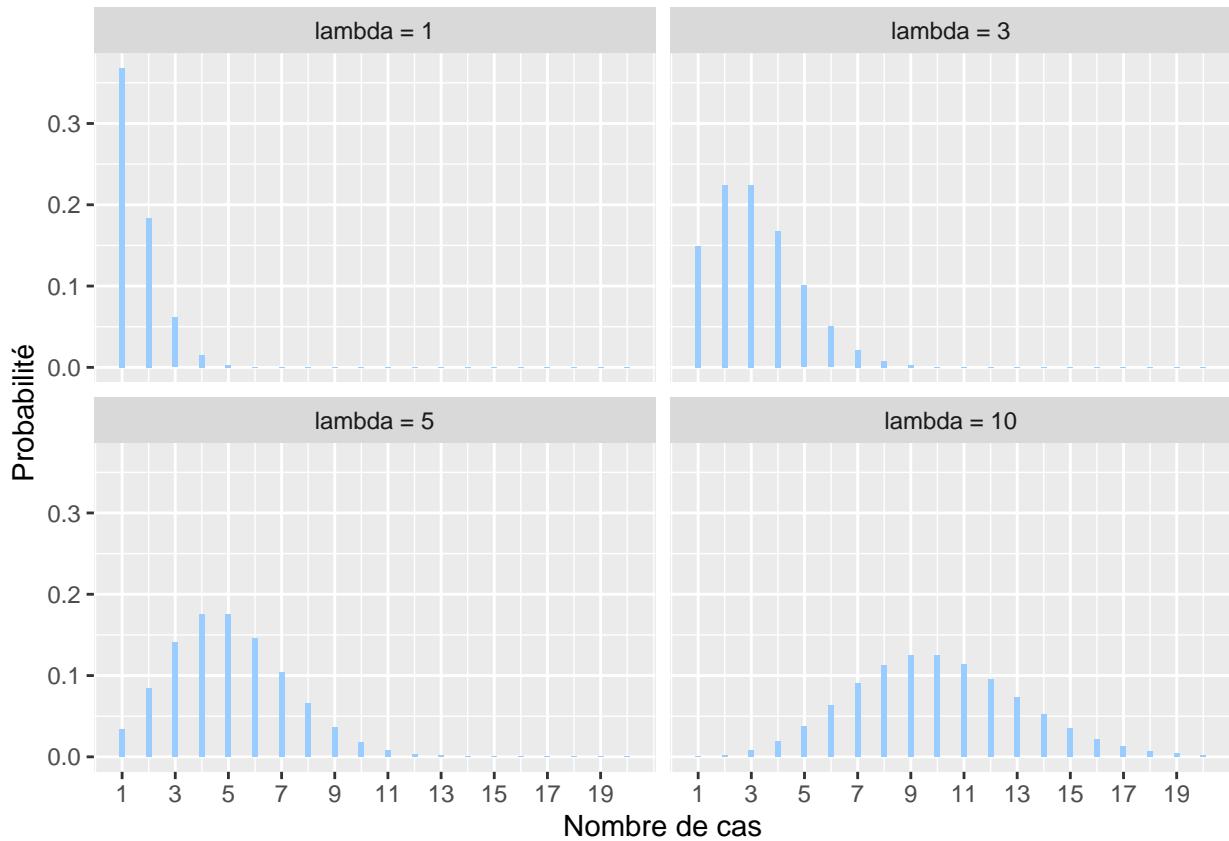


FIG. 2.11 : La distribution de poisson

fait qu'en moyenne, chaque répétition de l'expérience produit le même résultat, mais qu'un ensemble de petits facteurs aléatoires viennent rajouter de la variabilité dans les données collectées. Prenons un exemple concret, si l'on plante une centaine d'arbres simultanément dans un parc avec un degré d'ensoleillement identique et qu'on leur apporte les mêmes soins pendant dix ans, la distribution de leurs tailles suivra une distribution normale. Un ensemble de facteurs aléatoires (composition du sol, exposition au vent, aléas génétiques, passage de nuages, etc.) auront affecté différemment chaque arbre, ajoutant ainsi un peu de hasard dans leurs tailles finales. Ces dernières seront cependant davantage affectées par des paramètres centraux (espèces, ensoleillement, arrosage, etc.), et seront donc centrées autour d'une moyenne. La fonction de densité de la distribution normale est la suivante :

$$f(x; \mu; \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \quad (2.9)$$

avec x une valeur dont on souhaite connaître la probabilité, $f(x)$ sa probabilité, μ (mu) la moyenne de la distribution normale (paramètre de localisation) et σ (sigma) son écart-type (paramètre de dispersion). La courbe normale suit une forme de cloche. Notez que :

- 68,2% de la masse de la distribution normale est comprise dans l'intervalle $[\mu - \sigma \leq x \leq \mu + \sigma]$
- 95,4% dans l'intervalle $[\mu - 2\sigma \leq x \leq \mu + 2\sigma]$
- 99,7% dans l'intervalle $[\mu - 3\sigma \leq x \leq \mu + 3\sigma]$

Autrement dit, dans le cas d'une distribution normale, il est très invraisemblable d'observer des données situées à plus de trois écarts types de la moyenne. Notez ici que lorsque $\mu = 0$ et $\sigma = 0$, on obtient la loi normale générale (ou centrée-réduite) (section 2.5.5.2).

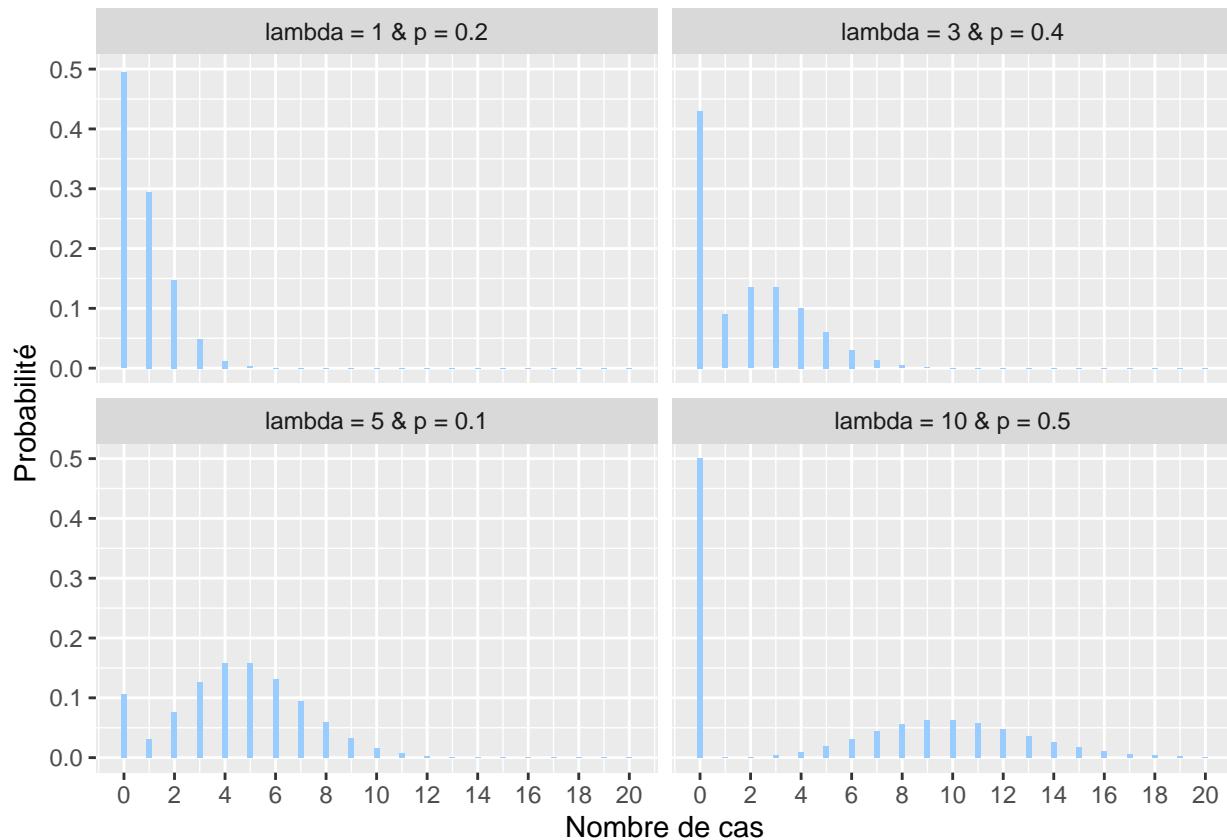


FIG. 2.12 : La distribution de poisson avec excès de zéros

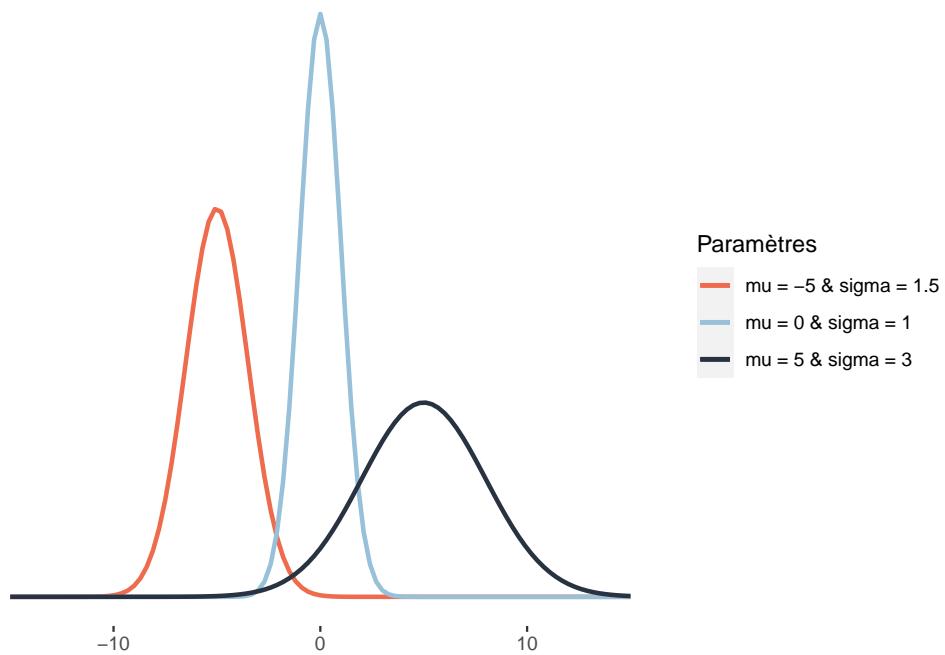


FIG. 2.13 : La distribution Gaussienne

2.4.3.9 La distribution gaussienne asymétrique

La distribution normale asymétrique (skew-normal) est une extension de la distribution gaussienne permettant de modifier la forme de la distribution normale pour qu'elle ne soit plus symétrique. Son espace d'échantillonnage est donc $]-\infty; +\infty[$. Sa fonction de densité est la suivante :

$$f(x; \xi; \omega; \alpha) = \frac{2}{\omega\sqrt{2\pi}} e^{-\frac{(x-\xi)^2}{2\omega^2}} \int_{-\infty}^{\alpha\left(\frac{x-\xi}{\omega}\right)} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \quad (2.10)$$

avec ξ (xi) le paramètre de localisation, ω (omega) le paramètre de dispersion (ou d'échelle) et α (alpha) le paramètre de forme (contrôlant le degré de symétrie). Si $\alpha = 0$, alors la distribution skew-normal est une simple distribution normale. Ce type de distribution est très utile lorsque que l'on souhaite modéliser une variable pour laquelle on sait que des valeurs plus extrêmes s'observeront d'un côté ou de l'autre de la distribution. Les revenus totaux annuels des personnes ou des ménages sont de très bons exemples puisqu'ils sont distribués généralement avec une asymétrie positive : bien qu'une moyenne existe, il y a généralement plus de personnes ou de ménages avec des revenus très faibles, que de personnes ou de ménages avec des revenus très élevés.

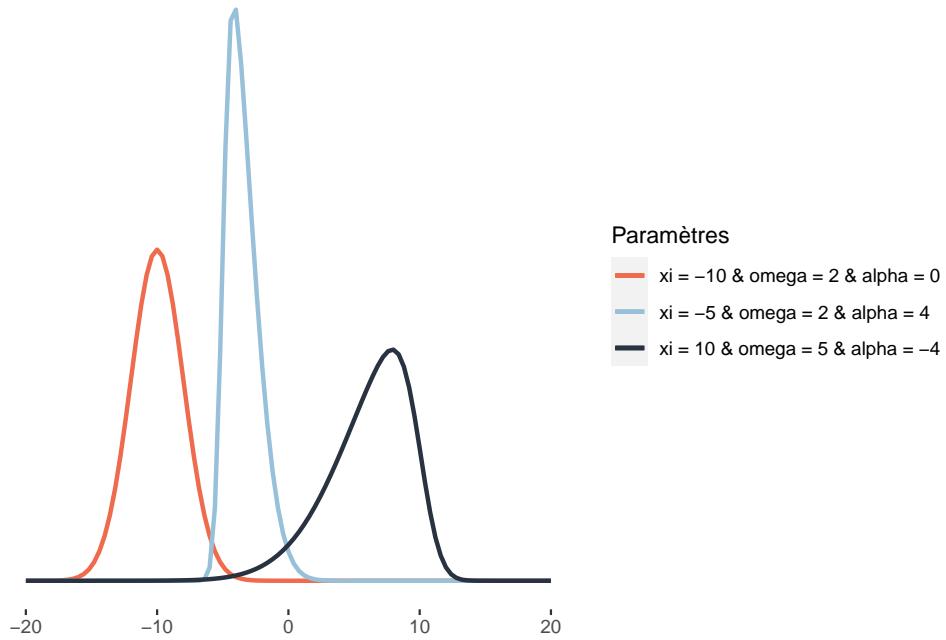


FIG. 2.14 : La distribution skew-Gaussienne

2.4.3.10 La distribution log-normale

Au même titre que la distribution skew-normal, la distribution log-normal est une version asymétrique de la distribution normale. Son espace d'échantillonnage est $]0; +\infty[$. Cela signifie que cette distribution ne peut décrire que des données continues et positives. Sa fonction de densité est la suivante :

$$f(x; \mu; \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\left(\frac{(\ln x - \mu)^2}{2\sigma^2}\right)} \quad (2.11)$$

À la différence la distribution skew-normal, la distribution log-normal ne peut avoir qu'une asymétrie positive (étirée vers la droite). Elle est cependant intéressante puisqu'elle ne compte que deux paramètres (μ et σ) ce qui la rend plus facile à ajuster. À nouveau, une distribution log-normal pourrait être utilisée

pour décrire les revenus totaux annuels des individus ou des ménages ou les revenus d'emploi. Elle est aussi utilisée en économie sur les marchés financiers pour représenter les cours des actions et des biens (ces derniers ne pouvant pas être inférieurs à 0).

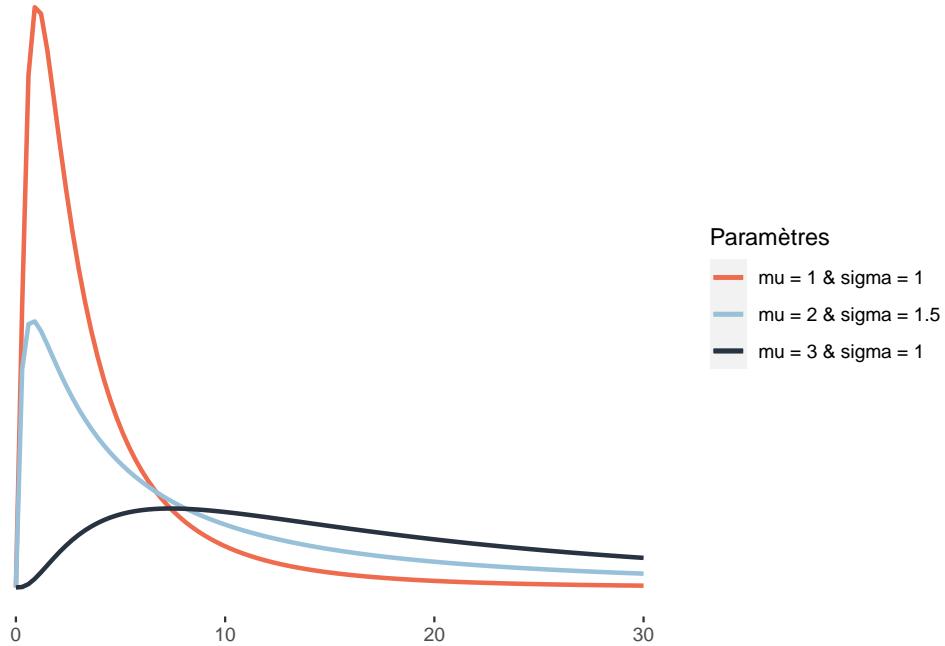


FIG. 2.15 : La distribution log-gaussienne

2.4.3.11 La distribution de Student

La distribution de Student joue un rôle important en statistique, elle est par exemple utilisée lors du test t pour calculer le degré de significativité du test. Comme la distribution gaussienne, la distribution de Student a une forme de cloche, est centrée sur sa moyenne et définie sur $]-\infty ; +\infty[$. Elle a cependant des « queues plus lourdes » (*heavy tails* en anglais). Entendez par-là que les valeurs extrêmes ont une plus grande probabilité d'occurrence dans une distribution de Student que dans une distribution gaussienne. Sa fonction de densité est la suivante :

$$p(x; \nu; \hat{\mu}; \hat{\sigma}) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2}) \sqrt{\pi\nu} \hat{\sigma}} \left(1 + \frac{1}{\nu} \left(\frac{x - \hat{\mu}}{\hat{\sigma}} \right)^2 \right)^{-\frac{\nu+1}{2}} \quad (2.12)$$

avec μ le paramètre de localisation, σ le paramètre de dispersion (qui n'est cependant pas un écart-type comme pour la distribution normale) et ν le nombre de degré de liberté. Plus ν est grand, plus la distribution de Student tend vers une distribution normale. Γ représente la fonction mathématique gamma (à ne pas confondre avec la distribution de Gamma). Un exemple d'application en études urbaines serait l'exposition au bruit environnemental de cyclistes. Cette distribution s'approcherait certainement d'une distribution normale, mais les cyclistes croisent régulièrement des secteurs peu bruyants (parcs, rues résidentielles, etc.) et des secteurs très bruyants (artères majeures, zones industrielles, etc.), ce qui conduit vers une distribution de Student.

2.4.3.12 La distribution de Cauchy

La distribution de Cauchy est également une distribution symétrique définie sur l'intervalle $]-\infty ; +\infty[$. Elle a comme particularité d'avoir des queues potentiellement plus lourdes que la distribution de

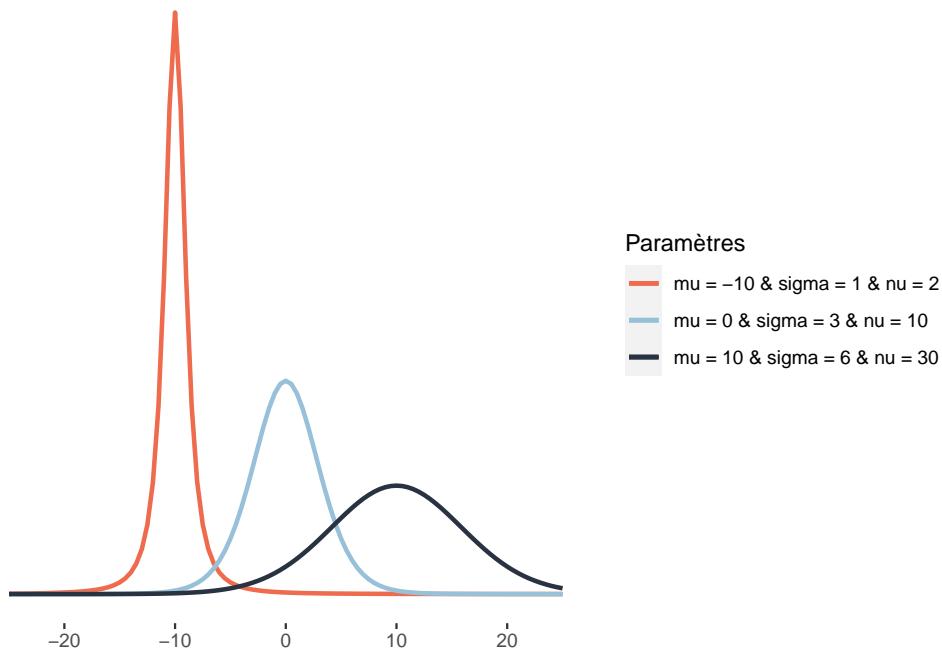


FIG. 2.16 : La distribution de Student

Student. Elle est notamment utilisée pour modéliser des phénomènes extrêmes comme les précipitations maximales annuelles, les niveaux d'inondations maximaux annuels ou les *values at risk* pour les portefeuilles financiers. Il est également intéressant de noter que le quotient de deux variables indépendantes normalement distribuées suit une distribution de Cauchy. Sa fonction de densité est la suivante :

$$\frac{1}{\pi\gamma} \left[\frac{\gamma^2}{(x - x_0)^2 + \gamma^2} \right] \quad (2.13)$$

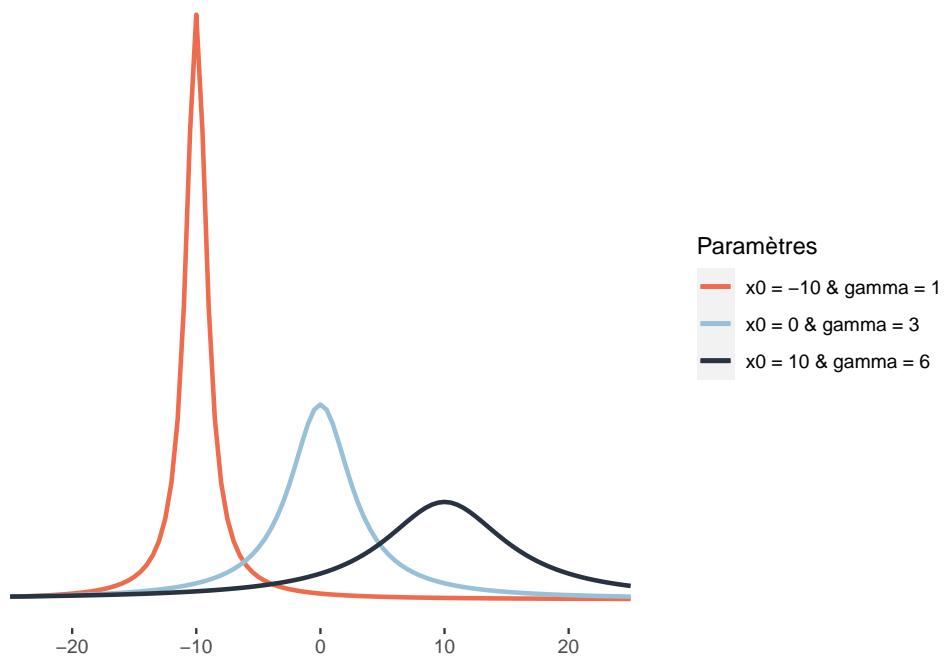
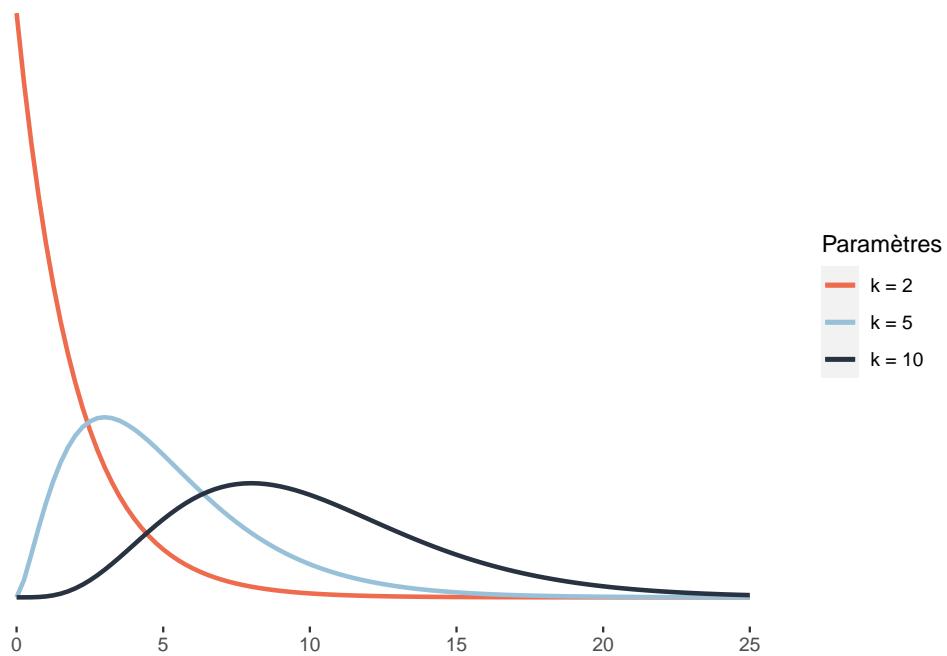
Elle dépend donc de deux paramètres : x_0 , le paramètre de localisation indiquant le pic de la distribution et γ , un paramètre de dispersion.

2.4.3.13 La distribution du Chi-carré

La distribution du Chi² est utilisée dans de nombreux tests statistiques. Spécifiquement, le test du Chi² de Pearson est utilisé pour comparer les écarts au carré entre des fréquences attendues et observées de deux variables qualitatives. La distribution du Chi² décrit donc les sommes des carrés d'un nombre k de variables indépendantes normalement distribuées. Il est assez rare de modéliser un phénomène à l'aide d'une distribution du Chi², mais son omniprésence dans les tests statistiques justifie qu'elle soit mentionnée ici. Cette distribution est définie sur l'intervalle $[0; +\infty[$ et a pour fonction de densité :

$$f(x; k) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2} \quad (2.14)$$

Cette fonction n'a qu'un paramètre k , représentant donc le nombre de variables au carré sommées pour obtenir la distribution du Chi²

**FIG. 2.17 :** La distribution de Cauchy**FIG. 2.18 :** La distribution du Chi²

2.4.3.14 La distribution exponentielle

La distribution exponentielle est une version continue de la distribution géométrique. Pour cette dernière, on s'intéresserait au nombre de tentatives nécessaires pour obtenir un résultat positif, soit une dimension discrète. Pour la distribution exponentielle, cette dimension discrète est remplacée par une dimension continue. L'exemple le plus intuitif est sûrement le cas du temps. Dans ce cas, la distribution exponentielle servirait à décrire le temps d'attente nécessaire pour qu'un évènement se produise. Il pourrait aussi s'agir d'une force que l'on applique jusqu'à ce qu'un matériau cède. Cette distribution est donc définie sur l'intervalle $[0; +\infty[$ et a pour fonction de densité :

$$f(x; \lambda) = \lambda e^{-\lambda x} \quad (2.15)$$

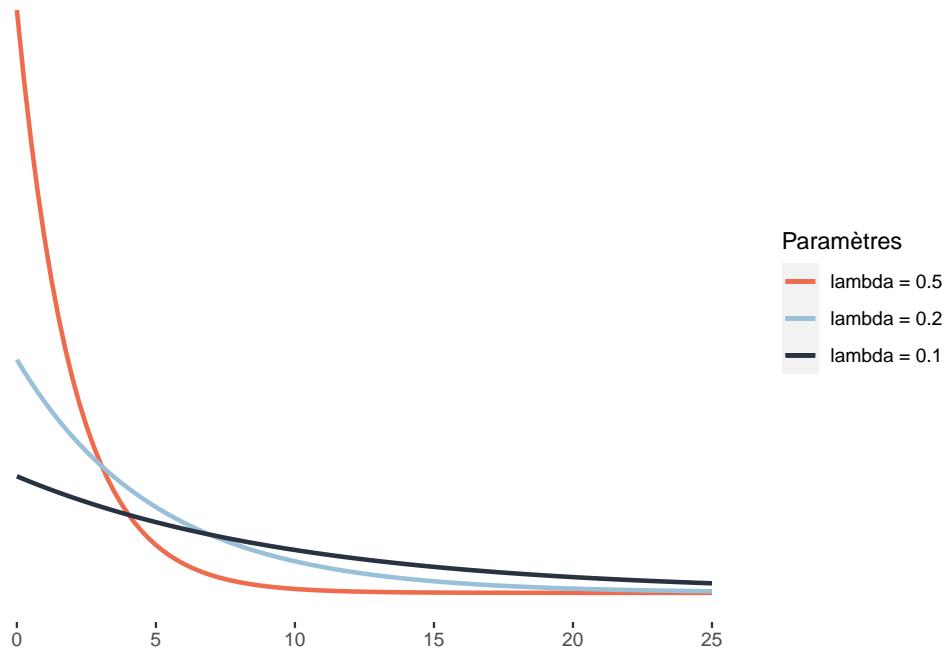


FIG. 2.19 : La distribution exponentielle

2.4.3.15 La distribution de Gamma

La distribution de Gamma est une généralisation d'un grand nombre de distributions. Elle regroupe ainsi la distribution exponentielle et du Chi2. En d'autres termes, les distributions du chi2 et exponentielles sont des cas particuliers de la distribution de Gamma. Cette distribution est définie sur l'intervalle $]0; +\infty[$ (notez que le 0 est exclu) et sa fonction de densité est la suivante :

$$f(x; \alpha; \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} \quad (2.16)$$

Elle comprend donc deux paramètres : α et β . Le premier est le paramètre de forme et le second un paramètre d'échelle (à l'inverse d'un paramètre de dispersion, plus sa valeur est petite, plus la distribution sera dispersée). Notez que cette distribution ne dispose pas d'un paramètre de localisation. Du fait de sa flexibilité, cette distribution est largement utilisée, que ce soit dans la modélisation des temps d'attente avant un évènement, la taille des réclamations d'assurance, les quantités de précipitations, etc.

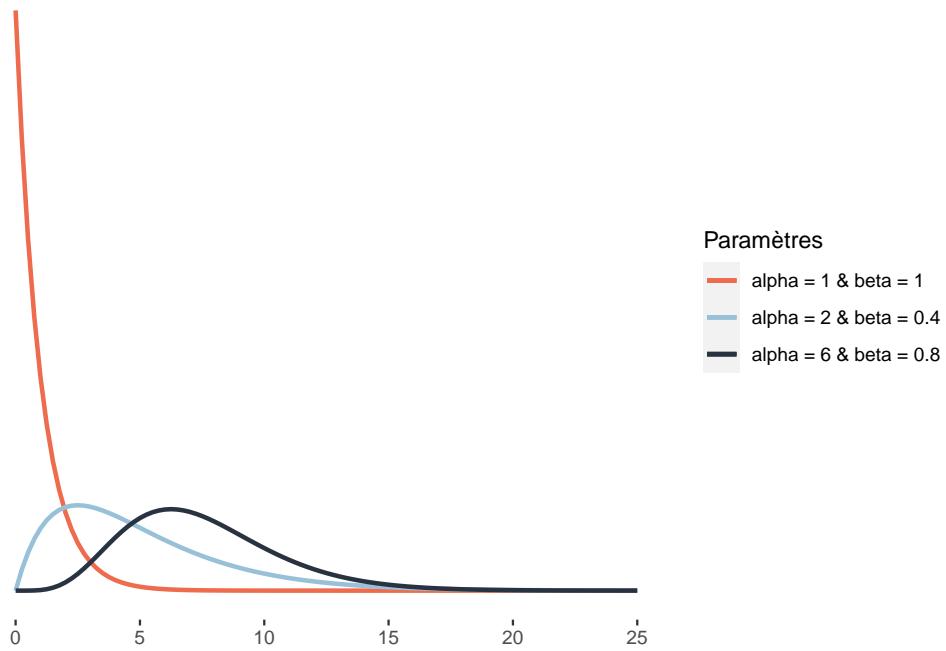


FIG. 2.20 : La distribution de Gamma

2.4.3.16 La distribution de Beta

La distribution de Beta est définie sur l'intervalle $[0; 1]$, elle est donc énormément utilisée pour représenter des variables étant des proportions ou des probabilités. Elle a aussi une utilité pratique en statistique, car en combinaison avec d'autres distributions, elle permet de modéliser leurs paramètres de probabilité (distribution beta-binomial, beta-negative-binomial, etc.). Un autre usage plus rare, mais intéressant est la modélisation de la fraction du temps représentée par une tâche dans le temps nécessaire à la réalisation de deux tâches de façon séquentielle. Ceci est dû au fait que la distribution d'une distribution gamma $g1$ divisée par la somme de $g1$ et d'une autre distribution gamma $g2$, suit une distribution beta. Un exemple concret serait par exemple la fraction du temps effectué à pied dans un déplacement multimodal. La distribution de beta a la fonction de densité suivante :

$$f(x; \alpha; \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad (2.17)$$

Elle a donc deux paramètres α et β contrôlant tous les deux la forme de la distribution. Cette caractéristique lui permet d'avoir une très grande flexibilité et même d'adopter des formes bimodales. B correspondant à la fonction mathématique Beta, à ne pas confondre avec la distribution de Beta et le paramètre Beta (β) de cette même distribution.

2.4.3.17 La distribution de Weibull

La distribution de Weibull est directement liée à la distribution exponentielle, cette dernière étant en fait un cas particulier de distribution de Weibull. Elle sert donc à représenter une quantité x (souvent le temps) à accumuler pour qu'un événement se produise. La distribution de Weibull est définie sur l'intervalle $[0; +\infty[$ et a la fonction de densité suivante :

$$f(x; \lambda) = \frac{k}{\lambda} \left(\frac{x}{\lambda} \right)^{k-1} e^{-(\frac{x}{\lambda})^k} \quad (2.18)$$

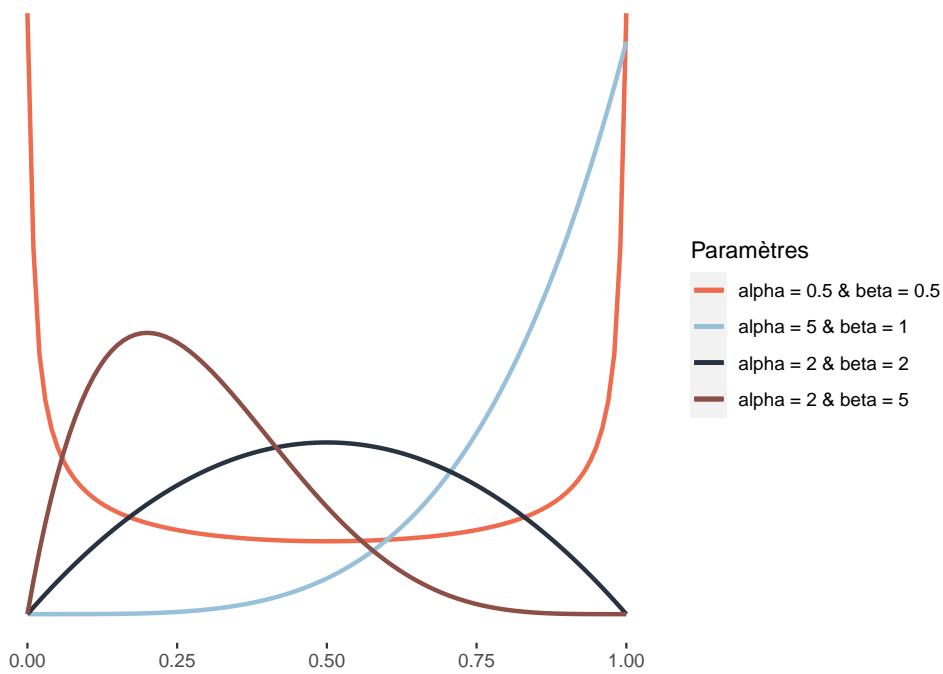


FIG. 2.21 : La distribution de Beta

λ est le paramètre de dispersion (analogie à celui d'une distribution exponentielle classique) et k le paramètre de forme. Pour bien comprendre le rôle de k , prenons un exemple : la propagation d'un champignon d'un arbre à son voisin. Si $k < 1$, cela signifie que la probabilité que l'événement modélisé se produise diminue avec le temps. En d'autres termes, dans de nombreux cas la contamination se fait rapidement. Si $k = 1$, alors la probabilité que l'événement se produise reste stable dans le temps. Si $k > 1$, alors la probabilité que l'événement se produise augmente avec le temps, ce qui signifie une augmentation des risques de contamination à mesure que les deux arbres restent à proximité. La distribution de Weibull est très utilisée en analyse de survie, en météorologie, en ingénierie des matériaux et dans la théorie des valeurs extrêmes.

2.4.3.18 La distribution de Pareto

La distribution de Pareto est à la distribution exponentielle ce que la distribution log-normal est à la distribution gaussienne : la distribution de l'exponentiel (e) de cette distribution originale. Elle est définie sur l'intervalle $[x_m; +\infty[$ avec la fonction de densité suivante :

$$f(x; x_m; k) = \left(\frac{x_m}{x}\right)^k \quad (2.19)$$

Elle comprend donc deux paramètres, x_m étant un paramètre de localisation (décalant la distribution vers la droite ou vers la gauche) et k un paramètre de forme. Plus k augmente, plus la probabilité prédictive par la distribution décroît rapidement.

Originellement, le mathématicien Pareto a utilisé cette fonction pour décrire la répartition du capital parmi la population puisqu'une grande partie du capital est détenue par une petite fraction de la population. Elle peut également être utilisée pour décrire la répartition de la taille des villes (Reed, 2002), la popularité des hommes sur tinder²² ou la taille des fichiers échangés sur internet (Reed and Jorgensen, 2004). Pour ces

²²<https://medium.com/@worstonlinedater/tinder-experiments-ii-guys-unless-you-are-really-hot-you-are-probably-better-off-not-wasting-your-2ddf370a6e9a>

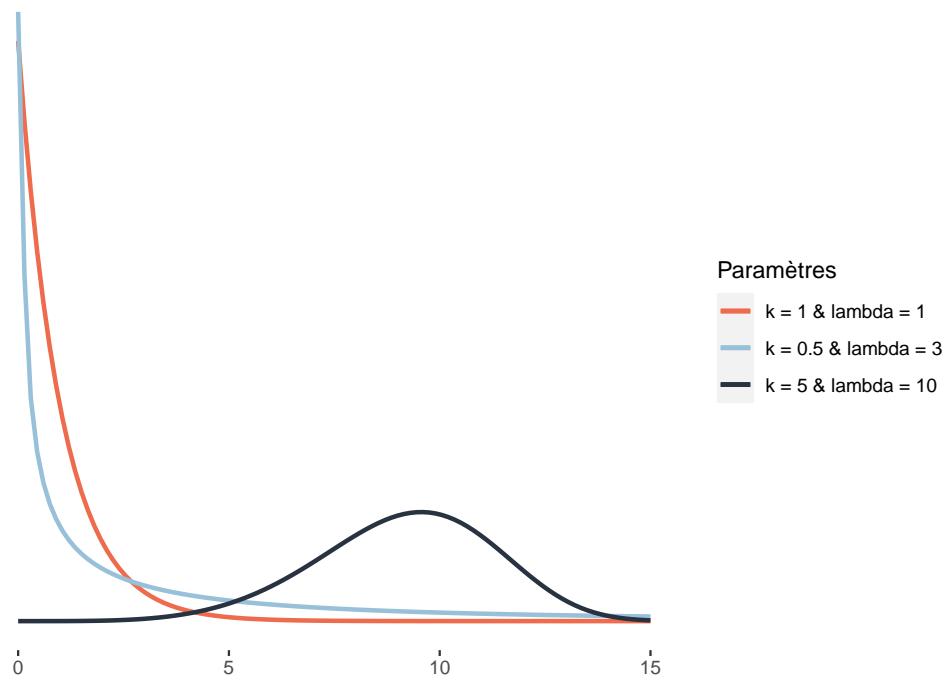


FIG. 2.22 : La distribution de Weibull

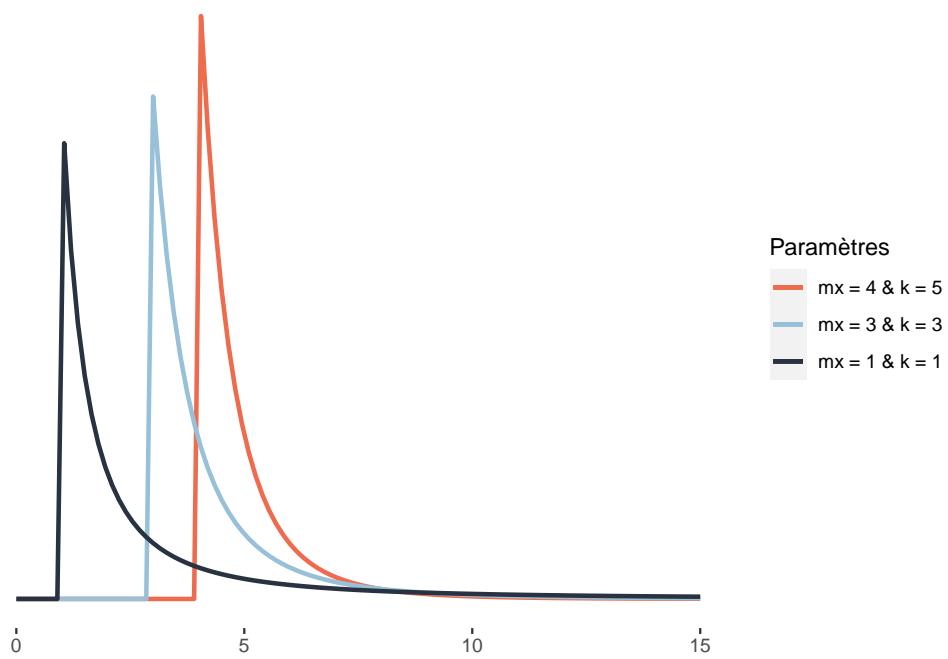


FIG. 2.23 : La distribution de Pareto

trois exemples, nous avons une situation avec : de nombreuses petites villes, profils peu attractifs, petits fichiers échangés et à l'inverse très peu de grandes villes, profils très attractifs, gros fichiers échangés.

2.4.3.19 Cas particuliers

Sachez également qu'il existe des formes «plus exotiques» de distributions que nous n'abordons pas ici, mais auxquelles vous pourriez être confrontés un jour :

- Les distributions sphériques, servant à décrire des données dont le 0 est équivalent à la valeur maximale. Par exemple, des angles puisque 0 et 360 degrés sont identiques.
- Les mixtures de distributions, décrivant des combinaisons de distributions. Par exemple, la distribution de la taille de tous les humains est en réalité un mixte entre deux distributions gaussiennes, une pour chaque sexe, puisque ces deux sous-distributions n'ont pas la même moyenne ni le même écart-type.
- Les distributions multivariées, permettant de décrire des phénomènes multidimensionnels. Par exemple, la réussite des élèves en français et en mathématique pourrait être modélisée comme une distribution gaussienne bivariée plutôt que deux distributions distinctes.
- Les distributions censurées décrivant des variables pour lesquels des valeurs sont possibles au-delà d'une certaine limite mais que l'on est incapable de mesurer. Un bon exemple serait la mesure de la pollution sonore avec un capteur incapable de détecter des niveaux sonores en dessous de 55 décibels. Il arrive parfois en ville que les niveaux sonores soient si faibles, mais les données collectées ne le montrent pas. Dans ce contexte, il est important d'utiliser des versions censurées des distributions présentées précédemment. Les observations au-delà de la limite sont conservées dans l'analyse, mais nous ne disposons que d'une information partielle à leur égard.
- Les distributions tronquées, souvent confondues avec les distributions censurées, décrivent des situations où des données qui au-delà d'une certaine limite sont retirées simplement de l'analyse.

2.4.4 Conclusion sur les distributions

Voilà qui conclut cette exploration des principales distributions à connaître. L'idée n'est bien sûr pas de toutes les retenir par cœur (et encore moins les formules mathématiques), mais plutôt de se rappeler dans quels contextes elles peuvent être utiles; et de revenir au besoin sur ce chapitre. Vous aurez certainement besoin de le relire avant d'aborder le chapitre portant sur les modèles linéaires généralisés (GLM). Wikipédia dispose d'informations très détaillées sur chaque distribution si vous avez besoin d'informations complémentaires. Pour un tour d'horizon plus exhaustif des distributions, vous pouvez aussi faire un tour sur les projets probonto²³ et the ultimate probability distribution explorer²⁴.

2.5 Statistiques descriptives sur des variables quantitatives

2.5.1 Les paramètres de tendance centrale

Trois mesures de tendance centrale permettent de résumer rapidement une variable quantitative :

- la **moyenne arithmétique** est simplement la somme des données d'une variable divisée par le nombre d'observations (n), soit $\frac{\sum_{i=1}^n x_i}{n}$ notée μ (prononcez *mu*) pour des données pour une population et \bar{x} (prononcez *x barre*) pour un échantillon.

²³<https://sites.google.com/site/probonto/screenshots>

²⁴<https://blog.wolfram.com/2013/02/01/the-ultimate-univariate-probability-distribution-explorer/>

- la **médiane** est la valeur qui coupe la distribution d'une variable d'une population ou d'un échantillon en deux parties égales. Autrement dit, 50% des valeurs des observations lui sont supérieures et 50% lui sont inférieures.
- le **mode** est la valeur la plus fréquente parmi un ensemble d'observations pour une variable. Il s'applique ainsi à des variables discrètes (avec un nombre fini de valeurs discrètes dans un intervalle donné) et non à des variables continues (avec un nombre infini de valeurs réelles dans un intervalle donné). Prenons deux variables, l'une discrète relative au nombre d'accidents par intersection (avec $X \in [0, 20]$) et l'autre continue relative à la distance de dépassement (en mètres) d'un cycliste par un véhicule motorisé (avec $X \in [0, 5]$). Pour la première, le mode – la valeur la plus fréquente – est certainement 0. Pour la seconde, identifier le mode n'est pas pertinent puisqu'il peut y avoir un nombre infini de valeurs entre 0 et 5 mètres.

Il convient de ne pas confondre moyenne et médiane ! Dans le tableau 2.1, nous avons reporté les valeurs moyennes et médianes des revenus des ménages pour les municipalités de l'île de Montréal en 2015. Par exemple, les 8685 ménages résidant à Westmount disposaient en moyenne d'un revenu de 295099\$; la moitié de ces 8685 ménages avaient un revenu inférieur à 100153\$ et l'autre moitié un revenu supérieur à cette valeur (médiane). Cela démontre clairement que la moyenne peut être grandement affectée par des valeurs extrêmes (faibles ou fortes); autrement dit, plus l'écart entre les valeurs de la moyenne et la médiane est importante, plus les données de la variable sont inégalement réparties. À Westmount, soit la municipalité la plus nantie de l'île de Montréal, les valeurs extrêmes sont des ménages avec des revenus très élevés tirant fortement la moyenne vers le haut. À l'inverse, le faible écart entre les valeurs moyenne et médiane dans la municipalité de Montréal-Est (58594\$ versus 50318\$) soulignent que les revenus des ménages sont plus également répartis. Cela explique que pour comparer les revenus totaux ou d'emploi entre différents groupes (selon le sexe, le groupe d'âge, le niveau d'éducation, la municipalité ou région métropolitaine, etc.), on privilie habuellement l'utilisation des revenus médians.

2.5.2 Les paramètres de position

Les paramètres de position permettent de diviser une distribution en n parties égales.

- Les **quartiles** qui divisent une distribution en quatre parties (25%) :
 - Q1 (25%), soit le quartile inférieur ou premier quartile;
 - Q2 (50%), soit la médiane;
 - Q3 (75%), soit le quartile supérieur ou troisième quartile.

TAB. 2.1 : Revenus moyens et médians des ménages en dollars, municipalités de l'île de Montréal, 2015

Municipalité	Nombre de ménages	Revenu moyen	Revenu médian
Baie-D'Urfé	1 330	171 390	118 784
Beaconsfield	6 660	187 173	123 392
Côte-Saint-Luc	13 490	94 570	58 935
Dollard-Des Ormeaux	17 210	102 104	78 981
Dorval	8 390	89 952	64 689
Hampstead	2 470	250 497	122 496
Kirkland	6 685	144 676	115 381
Montréal	779 805	69 047	50 227
Montréal-Est	1 730	58 594	50 318
Montréal-Ouest	1 850	159 374	115 029
Mont-Royal	7 370	205 309	109 540
Pointe-Claire	12 380	100 294	80 242
Sainte-Anne-de-Bellevue	1 960	102 969	67 200
Senneville	345	203 790	116 224
Westmount	8 685	295 099	100 153

- Les **quintiles** qui divisent une distribution en cinq parties égales (20%).
- Les **déciles** (de D1 à D9) qui divisent une distribution en dix parties égales (10%).
- Les **centiles** (de C1 à C99) qui divisent une distribution en cent parties égales (1%).

En cartographie, les quartiles et les quintiles sont souvent utilisés pour discréteriser une variable quantitative (continue ou discrète) en quatre ou cinq classes et plus rarement, en huit ou dix classes. Avec les quartiles, les bornes des classes qui comprendront chacune 25% des unités spatiales seront ainsi définies comme suit : [Min à Q1], [Q1 à Q2], [Q2 à Q3] et [Q3 à Max]. La méthode de discréterisation selon les quartiles ou quintiles permet alors de repérer, en un coup d'œil, à quelle tranche de 25% ou 20% des données appartient chacune des unités spatiales. Cette méthode de discréterisation est aussi utile pour comparer plusieurs cartes et vérifier si deux phénomènes sont ou non colocalisés ([Pumain and Béguin, 1994](#)). En guise d'exemple, les pourcentages de personnes à faible revenu et de locataires par secteur de recensement ont clairement des distributions spatiales très semblables dans la région métropolitaine de Montréal en 2016 (figure 2.24).

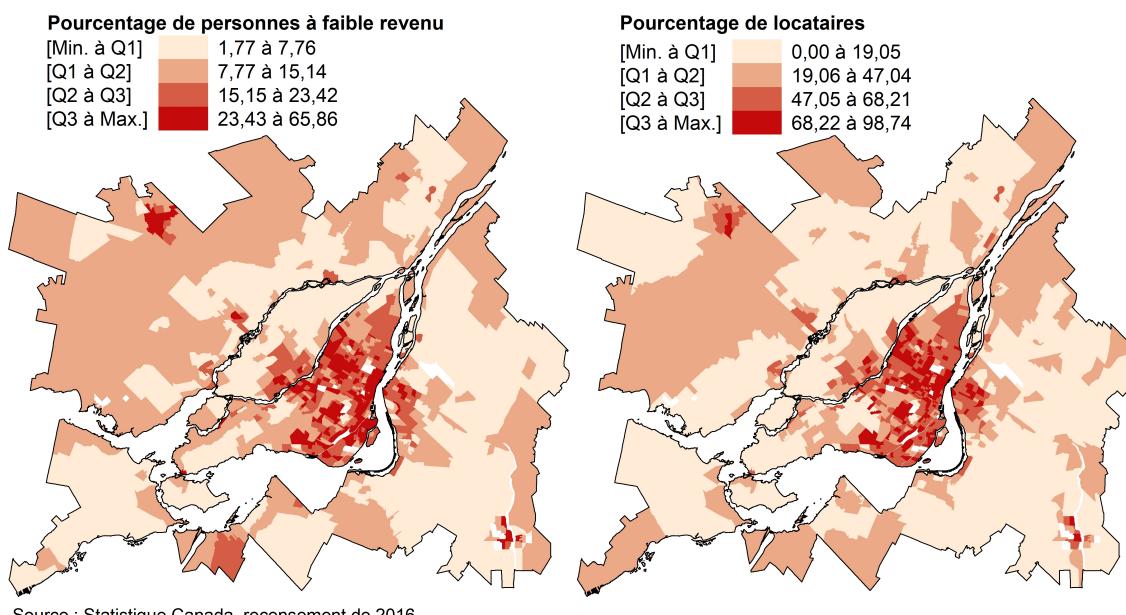


FIG. 2.24 : Exemples de cartographie avec une discréterisation selon les quantiles

Une lecture attentive des valeurs des centiles permet de repérer la présence de valeurs extrêmes voire aberrantes dans un jeu de données. Il n'est donc pas rare de les voir reportées dans un tableau de statistiques descriptives d'un article scientifique, et ce, afin de décrire succinctement les variables à l'étude. Par exemple, dans une étude récente comparant les niveaux d'exposition au bruit des cyclistes dans trois villes ([Apparicio and Gelb, 2020](#)), les auteurs reportent à la fois les valeurs moyennes et celles de plusieurs centiles. Globalement, la lecture des valeurs moyennes permet de constater que, sur la base des données collectées, les cyclistes sont plus exposés au bruit à Paris qu'à Montréal et Copenhague (73,4 dB(A) contre 70,7 et 68,4, tableau 2.2). Compte tenu de l'échelle logarithmique du bruit, la différence de 5 dB(A) entre les valeurs moyennes du bruit de Copenhague et de Paris peut être considérée comme une multiplication de l'énergie sonore par plus de 3. Pour Paris, l'analyse des quartiles montre que durant 25% du temps des trajets à vélo (plus de 63 heures de collecte), les participants ont été exposés à des niveaux de bruit soit inférieurs à 69,1 dB(A) (premier quartile), soit supérieurs à 74 dB(A). Quant à l'analyse des centiles, elle permet de constater que durant 5% et 10% du temps, les participants étaient exposés à des niveaux de bruit très élevés, dépassant 77 dB(A) (C90=76 et C90=77,2).

TAB. 2.2 : Statistiques descriptives de l'exposition au bruit des cyclistes par minute dans trois villes (dB(A), Laeq 1min)

Statistiques	Copenhague	Montréal	Paris
N	6 212,0	4 723,0	3 793,0
Moyenne de bruit	68,4	70,7	73,4
Centiles			
1	57,5	59,2	62,3
5	59,1	61,1	65,0
10	60,3	62,3	66,5
25 (premier quartile)	62,7	64,5	69,1
50 (médiane)	66,0	67,7	71,6
75 (troisième quartile)	69,2	71,0	74,0
90	71,9	73,7	76,0
95	73,3	75,2	77,2
99	76,5	78,9	81,0

2.5.3 Les paramètres de dispersion

Cinq principales mesures de dispersion permettent d'évaluer la variabilité des valeurs d'une variable quantitative : l'étendue, l'écart interquartile, la variance, l'écart-type et le coefficient de variation. Notez d'emblée que cette dernière mesure ne s'applique pas à des variables d'intervalle (section 2.1.2.2).

- **L'étendue** est la différence entre les valeurs minimale et maximale d'une variable, soit l'intervalle des valeurs dans lequel elle a été mesurée. Il convient d'analyser avec prudence cette mesure puisqu'elle inclut dans son calcul des valeurs potentiellement extrêmes voire aberrantes (faibles ou fortes).
- **L'intervalle ou écart interquartile** est la différence entre les troisième et premier quartiles ($Q_3 - Q_1$). Il représente ainsi une mesure de la dispersion des valeurs de 50% des observations centrales de la distribution. Plus la valeur de l'écart interquartile est élevée, plus la dispersion des 50% des observations centrales est forte. Contrairement à l'étendue, cette mesure élimine l'influence des valeurs extrêmes puisqu'elle ne tient pas compte des 25% des observations les plus faibles [Min à Q_1] et des 25% des observations les plus fortes [Q_3 à Max]. Graphiquement, l'intervalle interquartile est représenté à l'aide d'une boîte à moustaches (*boxplot* en anglais) : plus l'intervalle interquartile sera grand, plus la boîte sera allongée (figure 2.25)

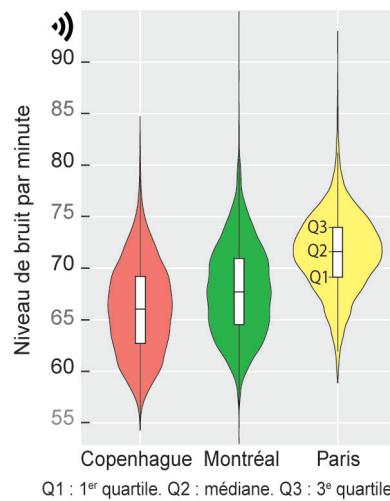


FIG. 2.25 : Graphique en violon, boîte à moustaches et intervalle interquartile

- **La variance** est la somme des déviations à la moyenne au carré (numérateur) divisée par le nombre d'observations pour une population (σ^2) ou divisée par le nombre d'observations moins une (s^2) pour un échantillon (eq. (2.20)). Puisque les déviations à la moyenne sont mises au carré, la valeur de la variance (tout comme celle de l'écart-type) sera toujours positive. Plus sa valeur est élevée, plus les observations sont dispersées autour de la moyenne. La variance représente ainsi l'écart au carré moyen des observations à la moyenne.

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n} \text{ ou } s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1} \quad (2.20)$$

- **L'écart-type** est la racine carrée de la variance (eq. (2.21)). Rappelez-vous que la variance est calculée à partir des déviations à la moyenne mises au carré. Étant donné que l'écart-type est la racine carrée de la variance, il est donc évalué dans les mêmes unités que la variable, contrairement à la variance. Bien entendu, comme pour la variance, plus la valeur de l'écart-type est élevée, plus la distribution des observations autour de la moyenne est dispersée.

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}} \text{ ou } s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (2.21)$$



Les formules des variances et des écart-types pour une population et un échantillon sont très similaires : seul le dénominateur change avec n versus $n - 1$ observations. Par conséquent, plus le nombre d'observations de votre jeu de données sera important, plus l'écart entre ces deux mesures de dispersion pour une population et un échantillon sera minime.

Comme dans la plupart des logiciels de statistique, les fonctions de base `var` et `sd` de R calculent la variance et l'écart-type pour un échantillon ($n - 1$ au dénominateur). Si vous souhaitez les calculer pour une population, adaptez la syntaxe ci-dessous dans laquelle `df$var1` représente la variable intitulée `var1` présente dans un *dataframe* nommé `df`.

```
var.p <- mean((df$var1 - mean(df$var1))^2)
sd.p <- sqrt(mean((df$var1 - mean(df$var1))^2))
```

- **Le coefficient de variation (CV)** est le rapport entre l'écart-type et la moyenne, représentant ainsi une standardisation de l'écart-type ou, en d'autres termes, une mesure de dispersion relative (eq. (2.22)). L'écart-type étant exprimé dans l'unité de mesure de la variable, il ne peut pas être utilisé pour comparer les dispersions de variables exprimées des unités de mesure différentes (par exemple, en pourcentage, en kilomètres, en dollars, etc.). Pour y remédier, on utilisera le coefficient de variation : une variable est plus dispersée qu'une autre si la valeur de son CV est plus élevée. Certains préfèreront multiplier la valeur du CV par 100 : l'écart-type est alors exprimé en pourcentage de la moyenne.

$$CV = \frac{\sigma}{\mu} \text{ ou } CV = \frac{s^2}{\bar{x}} \quad (2.22)$$

Illustrons comment calculer les cinq mesures de dispersion précédemment décrites à partir de valeurs fictives pour huit observations (colonne intitulée x_i au tableau 2.3). Les différentes statistiques reportées dans ce tableau sont calculées comme suit :

- **La moyenne** est la somme divisée par le nombre d'observations, soit $248/8 = 31$.
- **L'étendue** est la différence entre les valeurs maximale et minimale, soit $40 - 22 = 30$.

- Les quartiles coupent la distribution en quatre parties égales. Avec huit observations triées par ordre croissant, le **premier quartile** est égale à la valeur de la 2^e observation (soit 25), la **médiane** à celle de la 4^e (30), le **troisième quartile** à celle de la 6^e (35).
- L'**écart interquartile** est la différence entre Q3 et Q1, soit $35 - 25 = 10$.
- La seconde colonne du tableau est l'écart à la moyenne ($x_i - \bar{x}$), soit $22 - 31 = -9$ pour l'observation 1 ; la somme de ces écarts est toujours égale à 0. La troisième colonne est cette déviation mise au carré ($(x_i - \bar{x})^2$), soit $-9^2 = 81$, toujours pour l'observation 1. La somme de ces déviations à la moyenne au carré (268) représente le numérateur de la variance (eq. (2.20)). En divisant cette somme par le nombre d'observations, on obtient la **variance pour une population** ($268/8 = 33,5$) tandis que la **variance d'un échantillon** est égale à $268/(8 - 1) = 38,29$.
- L'**écart-type** est la racine carrée de la variance (eq. (2.21)), soit $\sigma = \sqrt{33,5} = 5,79$ et $s = \sqrt{38,29} = 6,19$.
- Finalement, les valeurs des coefficients de variation (eq. (2.22)) sont de $5,79/31 = 0,19$ pour une population et $6,19/31 = 0,20$ pour un échantillon.

Le tableau 2.4 vise à démontrer à partir de trois variables comment certaines mesures de dispersion sont sensibles à l'unité de mesure et/ou aux valeurs extrêmes.

Concernant l'**unité de mesure**, nous avons créé deux variables *A* et *B*, avec *B* étant simplement *A* multiplié par 10. Pour *A*, les valeurs de la moyenne, l'étendue et l'intervalle interquartile sont respectivement de 31, 18 et 10. Sans surprise, celles de *B* sont multipliées par 10 (310, 180, 100). La variance étant la moyenne des déviations à la moyenne au carré, elle est égale à 33,50 pour *A* et donc à $33,50 \times 10^2 = 3350$ pour *B* ; l'écart-type de *B* est égal à celui de *A* multiplié par 10. Cela démontre que l'étendue, l'intervalle interquartile, la variance et l'écart-type sont des mesures de dispersion dépendantes de l'unité de mesure. Par contre, le coefficient de variation (CV) étant le rapport de l'écart-type avec la moyenne, il a la même

TAB. 2.3 : Calcul des mesures de dispersion sur des données fictives

Observation	x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$
1	22,00	-9	81,0
2	25,00	-6	36,0
3	27,00	-4	16,0
4	30,00	-1	1,0
5	32,00	1	1,0
6	35,00	4	16,0
7	37,00	6	36,0
8	40,00	9	81,0
Statistique			
N	8,00		
Somme	248,00	0	268,0
Moyenne (\bar{x} ou μ)	31,00	0	33,5
Étendue	18,00		
Premier quartile	25,00		
Troisième quartile	35,00		
Intervalle interquartile	10,00		
Variance (population, σ^2)	33,50		
Écart-type (population, σ)	5,79		
Variance (échantillon, s^2)	38,29		
Écart-type (échantillon, s)	6,19		
Coefficient de variation (σ/μ)	0,19		
Coefficient de variation (s/\bar{x})	0,20		

valeur pour A et B , ce qui démontre que CV est bien une mesure de dispersion relative permettant de comparer des variables exprimées dans des unités de mesure différentes.

Concernant la **sensibilité aux valeurs extrêmes**, nous avons créé la variable C pour laquelle seule la huitième observation a une valeur différente (40 pour A et 105 pour B). Cette valeur de 105 pourrait être soit une valeur extrême positive mesurée, soit une valeur aberrante (par exemple, si l'unité de mesure était un pourcentage variant de 0 à 100%). Cette valeur a un impact important sur la moyenne (31 contre 39,12) et l'étendue (18 contre 83) et corollairement sur la variance (33,50 contre 641,86), l'écart-type (5,79 contre 25,33) et le coefficient de variation (0,19 contre 0,65). Par contre, l'intervalle interquartile étant calculé sur 50% des observations centrales ($Q3 - Q1$), il n'est pas affecté par cette valeur extrême.

2.5.4 Les paramètres de forme

2.5.4.1 Vérifier la normalité d'une variable quantitative



De nombreuses méthodes statistiques qui seront abordées dans les chapitres suivants – entre autres, la corrélation de Pearson, les test t et l'analyse de variance, les régressions simple et multiple – requièrent que la variable quantitative suive une **distribution normale** (nommée aussi **distribution gaussienne**).

Dans cette sous-section, nous décrirons trois démarches pour vérifier si la distribution d'une variable est normale : les coefficients d'asymétrie et d'aplatissement (*skewness* et *kurtosis* en anglais), les graphiques (histogramme avec courbe normale, diagramme quantile-quantile), les tests de normalité (tests de Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors, Anderson-Darling et Jarque-Bera).

Il est vivement recommandé de réaliser les trois démarches !

Une distribution est normale quand elle est symétrique et mésokurtique (figure 2.26).

2.5.4.1.1 Vérifier la normalité avec les coefficients d'asymétrie et d'aplatissement

Une **distribution est dite symétrique** quand la moyenne arithmétique est au centre de la distribution, c'est-à-dire que les observations sont bien réparties de part et d'autre de la moyenne qui sera alors égale à la médiane et au mode (on utilisera uniquement le mode pour une variable discrète et non pour une

TAB. 2.4 : Illustration de la sensibilité des mesures de dispersion à l'unité de mesure et aux valeurs extrêmes

Observation	A	B	C
1	22,00	220,00	22,00
2	25,00	250,00	25,00
3	27,00	270,00	27,00
4	30,00	300,00	30,00
5	32,00	320,00	32,00
6	35,00	350,00	35,00
7	37,00	370,00	37,00
8	40,00	400,00	105,00
Statistique			
Moyenne (μ)	31,00	310,00	39,12
Étendue	18,00	180,00	83,00
Intervalle interquartile	10,00	100,00	10,00
Variance (population, σ^2)	33,50	3 350,00	641,86
Écart-type (population, σ)	5,79	57,88	25,33
Coefficient de variation (σ/μ)	0,19	0,19	0,65

TAB. 2.5 : Résumé de la sensibilité de la moyenne et des mesures de dispersion

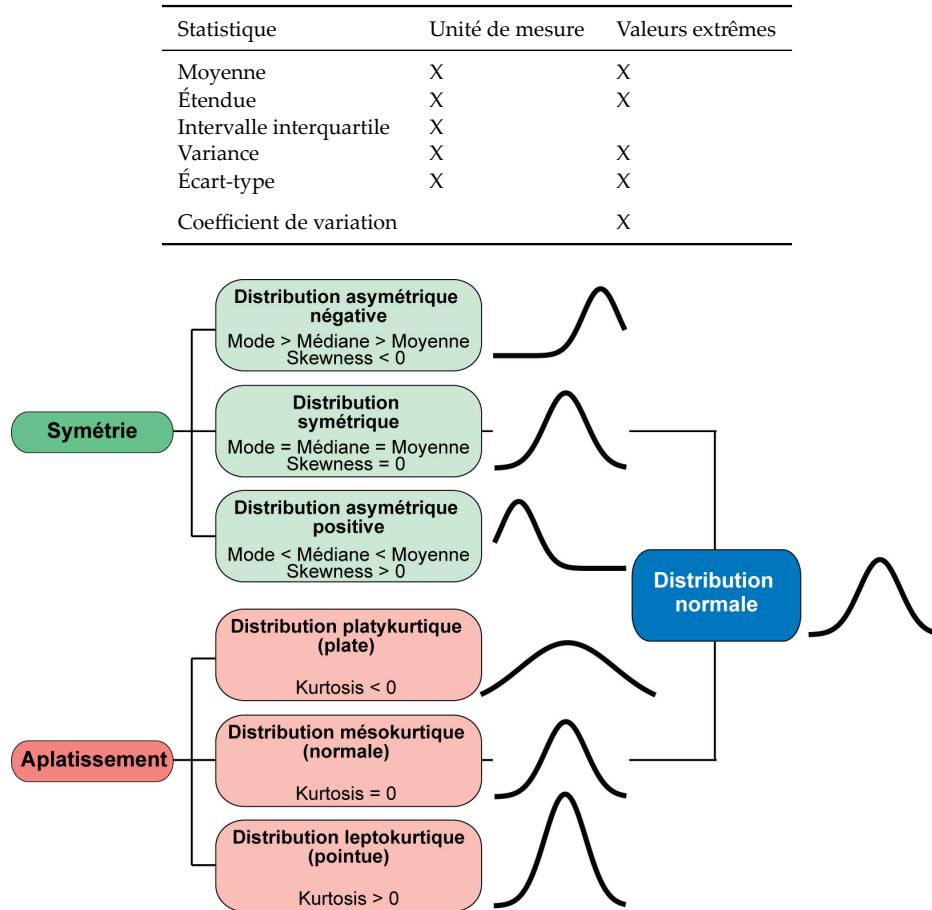


FIG. 2.26 : Formes d'une distribution et les coefficients d'asymétrie et d'aplatissement

variable continue). Pour évaluer l'asymétrie, on utilise habituellement le coefficient d'asymétrie (*skewness* en anglais).

Sachez toutefois qu'il existe trois façons (formules) pour le calculer (Joanes and Gill, 1998) : g_1 est la formule classique (eq. (2.23), disponible dans R avec la fonction `skewness` du package **moments**), G_1 est une version ajustée (eq. (2.24), utilisée dans les logiciels SAS et SPSS notamment) et b_1 est une autre version ajustée (eq. (2.25), utilisée par les logiciels MINITAB et BMDP). Nous verrons qu'avec les packages **DescTools** ou **e1071**, il est possible de calculer ces trois méthodes. Aussi, pour des grands échantillons ($n > 100$), il y a très peu de différences entre les résultats produits par ces trois formules (Joanes and Gill, 1998). Quelle que soit la formule utilisée, le coefficient d'assymétrie s'interprète comme suit (figure 2.27) :

- quand la valeur du *skewness* est négative, la **distribution est asymétrique négative**. La distribution est alors tirée à gauche par des valeurs extrêmes faibles, mais peu nombreuses. On emploie souvent l'expression *la queue de distribution* est étirée vers la gauche. La moyenne est alors inférieure à la médiane.
- quand la valeur du *skewness* est égale à 0, la **distribution est symétrique** (la médiane sera égale à la moyenne). Pour une variable discrète, les valeurs du mode, de la moyenne et de la médiane seront égales.
- quand la valeur du *skewness* est positive, la **distribution est symétrique positive**. La distribution est alors tirée à droite par des valeurs extrêmes fortes, mais peu nombreuses. La queue de distribution est alors étirée vers la droite. La moyenne est alors supérieure à la médiane. En sciences sociales,

les variables de revenu (totaux ou d'emploi, des individus ou des ménages) ont souvent des distributions asymétriques positives : la moyenne est affectée par quelques observations avec des valeurs de revenu très élevées et est ainsi supérieure à la médiane. En études urbaines, la densité de population pour des unités géographiques d'une métropole donnée (secteur de recensement par exemple) a aussi souvent une distribution asymétrique positive : quelques secteurs de recensement au centre de la métropole sont caractérisés par des valeurs de densité très élevées qui tirent la distribution vers la droite.

$$g_1 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left[\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right]^{\frac{3}{2}}} \quad (2.23)$$

$$G_1 = \frac{\sqrt{n(n-1)}}{n-2} g_1 \quad (2.24)$$

$$b_1 = \left(\frac{n-1}{n} \right)^{\frac{3}{2}} g_1 \quad (2.25)$$

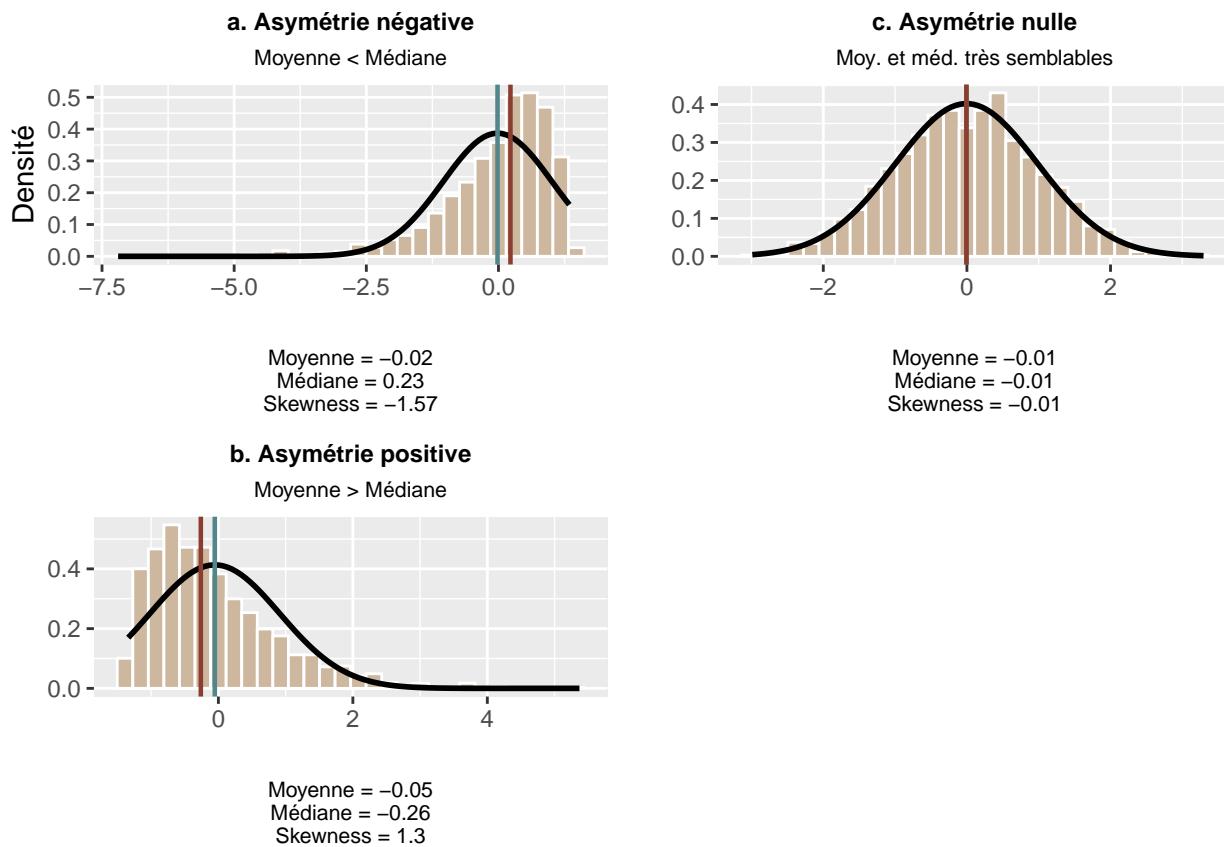


FIG. 2.27 : Asymétrie d'une distribution

Pour évaluer l'aplatissement d'une distribution, on utilisera le coefficient d'aplatissement (*kurtosis* en anglais). Là encore, il existe trois formules pour le calculer (eq. (2.26), (2.27), (2.28)) qui renverront des valeurs très semblables pour de grands échantillons (Joanes and Gill, 1998). Cette mesure s'interprète comme suit (figure 2.27) :

- quand la valeur du *kurtosis* est négative, la **distribution est platikurtique**. La distribution est dite plate, c'est-à-dire que la valeur de l'écart-type est importante (comparativement à une distribution

normale), signalant une grande dispersion des valeurs de part et d'autre la moyenne.

- quand la valeur du *kurtosis* est égale à 0, la **distribution est mésokurtique**, ce qui est typique d'une distribution normale.
- quand la valeur du *kurtosis* est positive, la **distribution est leptokurtique**, signalant que l'écart-type (la dispersion des valeurs) est plutôt faible. Autrement dit, la dispersion des valeurs autour de la moyenne est faible.

$$g_2 = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left(\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^2} - 3 \quad (2.26)$$

$$G_2 = \frac{n-1}{(n-2)(n-3)} \{(n+1)g_2 + 6\} \quad (2.27)$$

$$b_2 = (g_2 + 3)(1 - 1/n)^2 - 3 \quad (2.28)$$

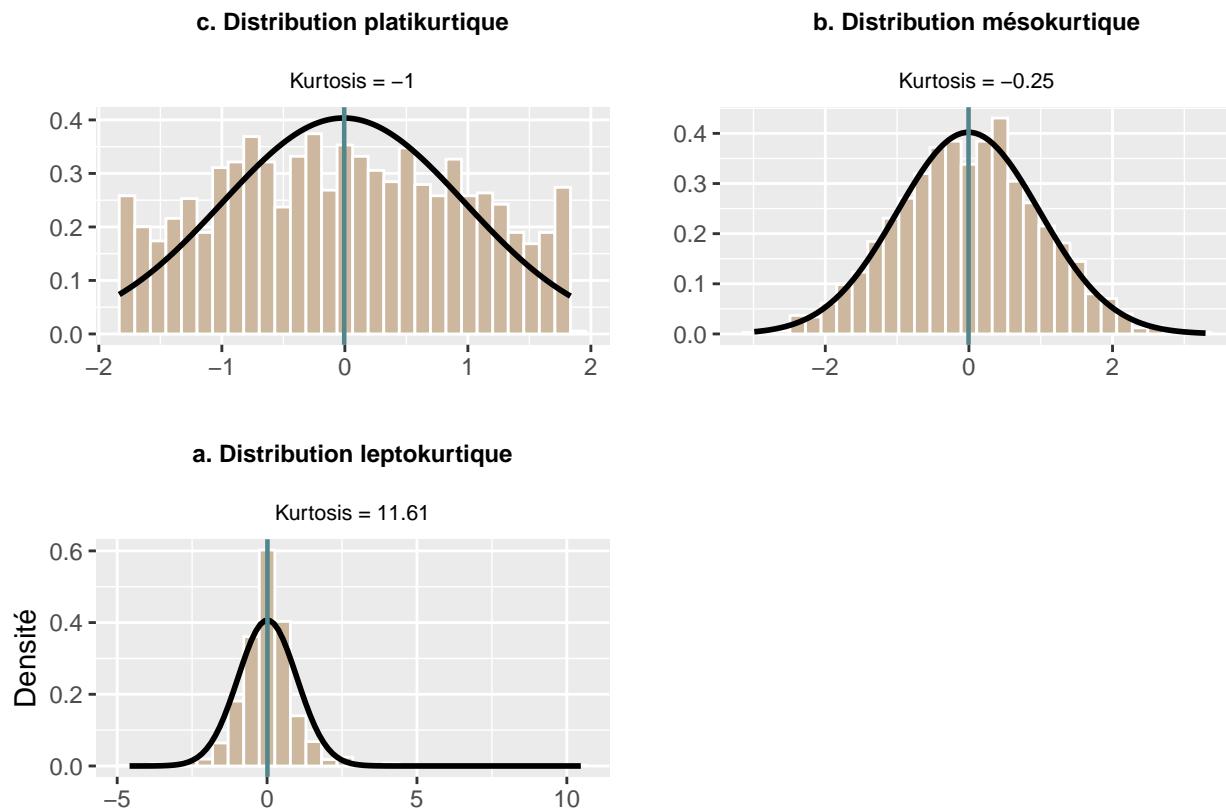


FIG. 2.28 : Applatissement d'une distribution



Regardez attentivement les équations (2.26), (2.27), (2.28); vous remarquez que pour g_2 et b_2 , il y a une soustraction de -3 et une addition $+6$ pour G_2 . On parle alors de *kurtosis* normalisé (*excess kurtosis* en anglais). Pour une distribution normale, il prendra la valeur de 0, comparativement à la valeur de 3 pour un *kurtosis* non normalisé. Par conséquent, avant de calculer du *kurtosis*, il convient de s'assurer que la fonction que vous utilisez implémente une méthode de calcul normalisée (donnant une valeur de 0 pour une distribution normale). Par exemple, la fonction *Kurt* du package **DescTools** calcule les trois formules normalisées tandis que la fonction *kurtosis* du package **moments** renvoie un *kurtosis* non normalisé.

```

library(DescTools)
library(moments)
#Générer une variable normalement distribuée avec 1000 observations
Normale <- rnorm(1500,0,1)
round(DescTools:::Kurt(Normale),3)

## [1] -0.146

round(moments::kurtosis(Normale),3)

## [1] 2.858

```

2.5.4.1.2 Vérifier la normalité avec des graphiques

Les graphiques sont un excellent moyen de vérifier visuellement si une distribution est normale ou pas. Bien entendu, les histogrammes, que nous avons déjà largement utilisés, sont un incontournable; à titre de rappel, ils permettent de représenter la forme de la distribution des données (figure 2.29). Un autre type de graphique intéressant est le **diagramme quantile-quantile** (*Q-Q plot* en anglais) qui permet de comparer la distribution d'une variable avec une distribution gaussienne (normale). Trois éléments composent ce graphique tel qu'illustré à la figure 2.30 :

- les points, représentant les observations de la variable
- la distribution gaussienne (normale), représentée par une ligne
- l'intervalle de confiance à 5% de la distribution normale (en orange sur la figure).

Quand la variable est normale distribuée, les points seront situés le long de la ligne. Plus les points localisés en dehors de l'intervalle de confiance (bande orange) seront nombreux, plus la variable sera alors anormalement distribuée.

2.5.4.1.3 Vérifier la normalité avec des tests de normalité

Cinq principaux tests d'hypothèse permettent de vérifier la normalité d'une variable : les tests de **Kolmogorov-Smirnov** (KS), **Lilliefors** (LF), **Shapiro-Wilk** (SW), **Anderson-Darling**, et de **Jarque-Bera** (JB); sachez toutefois qu'il y en a d'autres non discutés ici (tests de D'Agostino-Pearson, Cramer-von Mises, de Ryan-Joiner, Shapiro-Francia, etc.). Pour les formules et une description détaillée de ces tests, vous pouvez consulter Razali et al. (2011) ou Yap et Sim (2011). **Quel test choisir?** Plusieurs auteurs ont comparé ces différents tests à partir de plusieurs échantillons, et ce, en faisant varier la forme de la distribution et le nombre d'observations (Razali et al., 2011; Yap and Sim, 2011). Selon Razali et al. (2011), le meilleur test semble être celui de Shapiro-Wilk, puis ceux de Anderson-Darling, Lilliefors et Kolmogorov-Smirnov. Yap et Sim (2011) concluent aussi que le Shapiro-Wilk semble être le plus performant.

Quoi qu'il en soit, ces cinq tests postulent que la variable suit une distribution gaussienne (hypothèse nulle, h_0). Cela signifie que si la valeur de P associée à la valeur de chacun des tests est supérieure au seuil alpha choisi (habituellement $\alpha = 0,05$), la distribution est normale. À l'inverse, si $P < 0,05$, on choisit l'hypothèse alternative (h_1), c'est-à-dire que la distribution est anormale.

Dans le tableau ci-dessous sont reportées les valeurs des différents tests pour les cinq types de distribution générées à la figure 2.29. Sans surprise, pour l'ensemble des tests, la valeur de P est inférieur à 0,05 pour la distribution normale.

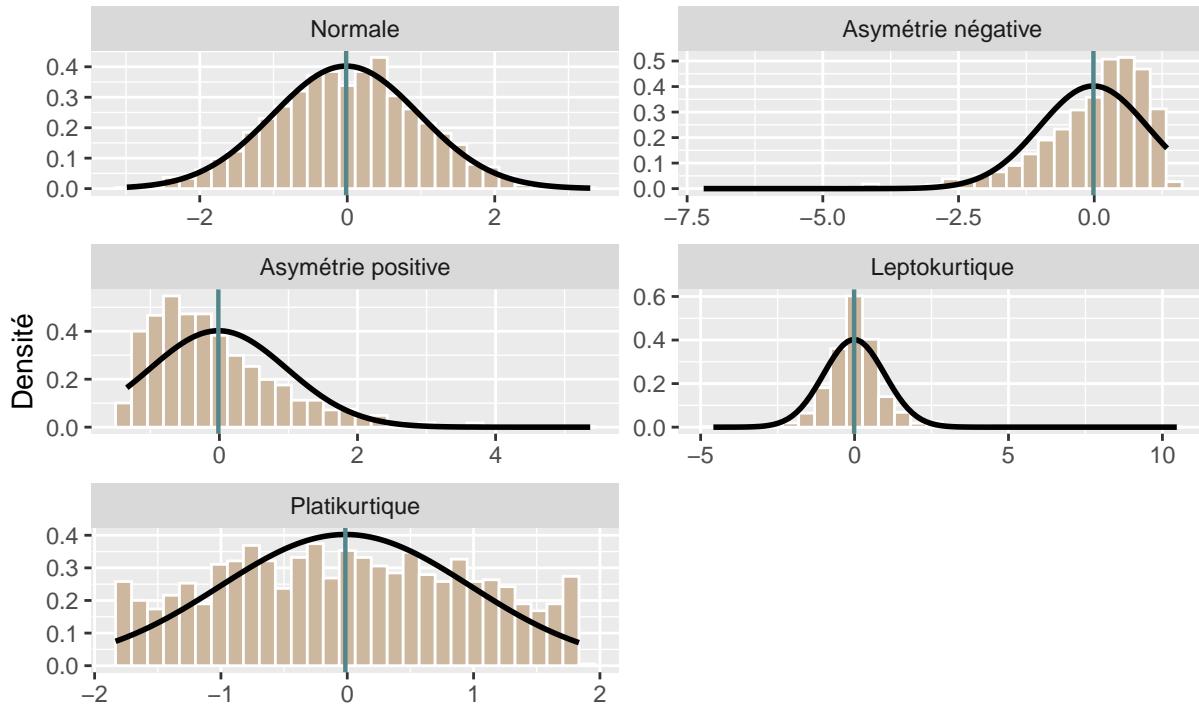


FIG. 2.29 : Distributions et courbe normale

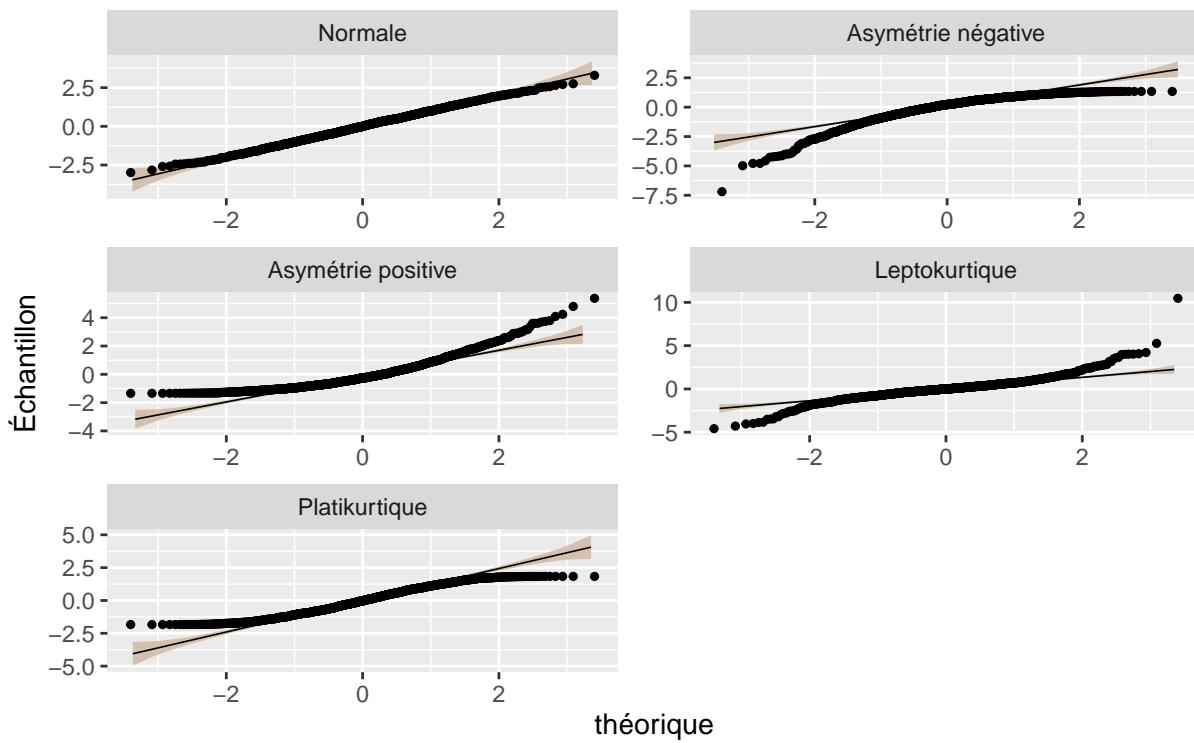


FIG. 2.30 : Diagrammes quantile-quantile

TAB. 2.6 : Les différents tests d'hypothèse pour la normalité

Test	Propriétés et interprétation	Fonction R
Kolmogorov-Smirnov	Plus sa valeur est proche de zéro, plus la distribution est normale. L'avantage de ce test est qu'il peut être utilisé pour vérifier si une variable suit la distribution de n'importe quelle loi (autre que la loi normale).	ks.test du package stats
Lilliefors	Ce test est une adaptation du test de Kolmogorov-Smirnov. Plus sa valeur est proche de zéro, plus la distribution est normale.	lillie.test du package nortest
Shapiro-Wilk	Si la valeur de la statistique de Shapiro-Wilk est proche de 1, alors la distribution est normale; et anormale quand elle est inférieure à 1.	shapiro.test du package stats
Anderson-Darling	Ce test est une modification du test de Cramer-von Mises (CVM). Il peut être aussi utilisé pour tester d'autres distributions (uniforme, log-normale, exponentielle, Weibull, distribution de pareto généralisée, logistique, etc.).	ad.test du package stats
Jarque-Bera	Basé sur un test du type multiplicateur de Lagrange, il utilise dans son calcul les valeurs du <i>Skewness</i> et du <i>Kurtosis</i> . Plus sa valeur s'approche de 0, plus la distribution est normale. Ce test est surtout utilisé pour vérifier si les résidus d'un modèle de régression linéaire sont normalement distribués, nous y reviendrons dans le chapitre sur la régression multiple. Il s'écrit $JB = \frac{1}{6} \left(g_{-1}^2 + \frac{g_{-2}^2}{4} \right)$ avec g_{-1} et g_{-2} qui sont respectivement les valeurs du <i>skewness</i> et du <i>kurtosis</i> de la variable (voir plus haut les équations 2.23 et 2.26).	JarqueBeraTest du package DescTools

TAB. 2.7 : Calculs des tests de normalité pour différentes distributions

	Normale	Asymétrie négative	Asymétrie positive	Leptokurtique	Platikurtique
Skewness	0.123	1.364	-1.047	-0.416	-0.001
Kurtosis	-0.224	2.072	1.011	2.8	-1.007
Kolmogorov-Smirnov (KS)	0.033	0.12	0.109	0.096	0.049
Lilliefors (LF)	0.033	0.12	0.109	0.096	0.049
Shapiro-Wilk (SW)	0.997	0.888	0.922	0.943	0.971
Anderson-Darling (AD)	0.494	14.4	10.387	8.526	2.782
Jarque-Bera (JB)	2.481	386.341	128.422	468.751	15.525
KS (valeur p)	0.66	0	0	0	0.18
LF (valeur p)	0.218	0	0	0	0.006
SW (valeur p)	0.432	0	0	0	0
AD (valeur p)	0.215	0	0	0	0
JB (valeur p)	0.289	0	0	0	0



Attention! La plupart des auteurs s'entendent sur le fait que ces tests sont très restrictifs : plus la taille de votre échantillon (n) est importante, plus les tests risquent de vous signaler que vos distributions sont anormales (à la lecture des valeurs de P).

Certains conseillent même de ne pas les utiliser quand $n > 200$ et de vous fier uniquement aux graphiques (histogramme et diagramme Q-Q) !



Bref, vérifier la normalité d'une variable n'est pas une tâche si simple. De nouveau, nous vous conseillons vivement de :

- construire les graphiques pour analyser visuellement la forme de la distribution (histogramme avec courbe normale et diagramme Q-Q)
- calculer le *skewness* et le *kurtosis*,
- calculer plusieurs tests (minimamente Shapiro-Wilk et Kolmogorov-Smirnov)
- accorder une importance particulière aux graphiques lorsque vous traitez des grands échantillons ($n > 200$).

2.5.4.2 Vérifier d'autres formes de distributions

Comme nous l'avons vu, la distribution normale n'est que l'une des multiples distributions existantes. Dans de nombreuses situations, elle ne sera pas adaptée pour décrire vos variables. La démarche à adopter pour trouver une distribution adaptée est la suivante :

1. Définissez la nature de votre variable, identifier si elle est discrète ou continue et l'intervalle dans lequel elle est définie. Une variable dont les valeurs sont positives ou négatives ne pourra pas être décrite avec une distribution Gamma par exemple (à moins de la décaler).
2. Explorez votre variable, affichez son histogramme et son graphique de densité pour avoir une vue générale de sa morphologie.
3. Présélectionnez un ensemble de distributions candidates compte tenu des observations précédentes. Vous pouvez également vous reporter à la littérature existante sur votre sujet d'étude pour inclure d'autres distributions. Soyez flexible ! Une variable strictement positive pourrait tout de même avoir une forme normale. De même, une variable décrivant des comptages suffisamment grands pourrait être mieux décrite par une distribution normale qu'une distribution de poisson.
4. Tentez d'ajuster chacune des distributions retenues à vos données et comparez les qualités d'ajustements pour retenir la plus adaptée.

Pour ajuster une distribution à un jeu de données, il faut trouver les valeurs des paramètres de cette distribution qui lui permettront d'adopter une forme la plus proche possible des données. On appelle cette opération **ajuster un modèle**, puisque la distribution théorique est utilisée pour modéliser les données. L'ajustement des paramètres est un problème d'optimisation que plusieurs algorithmes sont capables de résoudre (*gradient descent*, *Newton-Raphson method*, *Fisher scoring*, etc.). Dans R, le package **fitdistrplus** permet d'ajuster pratiquement n'importe quelle distribution à des données en offrant plusieurs stratégies d'optimisation grâce à la fonction **fitdist**. Il suffit de disposer d'une fonction représentant la distribution de densité ou de masse de la distribution en question, généralement noté **dnomde la distribution** (**dnorm**, **dgamma**, **dpoisson**, etc.) dans R. Notez que certains packages comme **VGAM** ou **gamlss.dist** ajoutent un grand nombre de fonctions de densité et de masse à celles déjà disponibles de base dans R.

Pour comparer l'ajustement de plusieurs distributions théoriques à des données, trois approches doivent être combinées :

- Observer graphiquement l'ajustement de la courbe théorique à l'histogramme des données. Cela permet d'éliminer au premier coup d'œil les distributions qui ne correspondent pas.

- Comparer les *loglikelihood*. Le *loglikelihood* est un score d'ajustement des distributions aux données. Pour faire simple, plus le *loglikelihood* est grand, plus la distribution théorique est proche des données. Référez-vous à l'encadré suivant pour une description plus en profondeur du *loglikelihood*.
- Utiliser le test de Kolmogorov-Smirnov pour déterminer si une distribution particulière est mieux ajustée pour les données.



Qu'est-ce-que le loglikelihood ?

Le *loglikelihood* est une mesure de l'ajustement d'un modèle à des données. Il est utilisé à peu près partout en statistique. Comprendre sa signification est donc un exercice important pour développer une meilleure intuition du fonctionnement général de nombreuses méthodes. Si les concepts de fonction de densité et de fonction de masse vous semblent encore flous, reportez-vous à la section 2.4 sur les distributions dans un premier temps.

Admettons que nous disposons d'une variable continue v que nous avons tenté de modéliser avec une distribution d (il peut s'agir de n'importe quelle distribution). d a une fonction de densité avec laquelle il est possible de calculer pour chacune des valeurs de v sa probabilité d'être observée selon le modèle d .

Prenons un exemple concret dans R. Admettons que nous avons une variable comprenant 10 valeurs (oui, c'est un petit échantillon, mais c'est pour faire un exemple simple).

```
v <- c(5,8,7,8,10,4,7,6,9,7)
moyenne <- mean(v)
ecart_type <- sd(v)
```

En calculant sa moyenne et son écart type, nous obtenons les paramètres d'une distribution normale que nous pouvons utiliser pour représenter les données observées. En utilisant la fonction `dnorm` (la fonction de densité de la distribution normale), nous pouvons calculer la probabilité d'observer chacune des valeurs de v selon cette distribution normale.

```
probas <- dnorm(v, moyenne, ecart_type)
df <- data.frame(valeur = v,
                  proba = probas)
print(df)
```

	valeur	proba
## 1	5	0.11203710
## 2	8	0.19624888
## 3	7	0.22228296
## 4	8	0.19624888
## 5	10	0.06009897
## 6	4	0.04985613
## 7	7	0.22228296
## 8	6	0.18439864
## 9	9	0.12689976
## 10	7	0.22228296

On observe ainsi que les valeurs 7 et 8 sont très probables selon le modèle alors que la valeur 10 est très improbable.

Le *likelihood* est simplement le produit de toutes ces probabilités. Il s'agit donc de **la probabilité conjointe** d'avoir observé toutes les valeurs de v **sous l'hypothèse** que d est la distribution produisant ces valeurs. Si d décrit efficacement v , alors le *likelihood* est plus grand que si d ne décrit pas efficacement v . Il s'agit d'une forme de raisonnement par l'absurde : après avoir observé v , on calcule la probabilité d'avoir observé v (*likelihood*) si

notre modèle d était vrai. Si cette probabilité est très basse, alors c'est que notre modèle est mauvais puisqu'on a bien observé v .

```
likelihood_norm <- prod(probas)
print(likelihood_norm)
```

```
## [1] 3.322759e-09
```

Cependant, multiplier un grand nombre de valeurs inférieures à zéro tend à produire des chiffres infiniment petits et donc à complexifier grandement le calcul. On préfère donc utiliser le *loglikelihood*. L'idée étant de transformer les probabilités obtenues avec la fonction *log* puis d'additionner leurs résultats, puisque $\log(xy) = \log(x) + \log(y)$.

```
loglikelihood_norm <- sum(log(probas))
print(loglikelihood_norm)
```

```
## [1] -19.52247
```

Comparons ce *loglikelihood* à celui d'un second modèle dans lequel nous utilisons toujours la distribution normale, mais avec une moyenne différente (faussée en rajoutant +3) :

```
probas2 <- dnorm(v, moyenne+3, ecart_type)
loglikelihood_norm2 <- sum(log(probas2))
print(loglikelihood_norm2)
```

```
## [1] -33.53631
```

Ce second *loglikelihood* est plus faible, indiquant clairement que le premier modèle est plus adapté aux données.

Passons à la pratique avec deux exemples.

2.5.4.2.1 Temps de retard des bus de la ville de Toronto

Analysons les temps de retard pris par les bus de la ville de Toronto lorsqu'un évènement perturbe la circulation. Ce jeu de données est disponible sur le site de l'Open Data²⁵ de la ville de Toronto. Compte tenu de la grande quantité d'observations, nous avons fait le choix de nous concentrer sur les évènements ayant eu lieu durant le mois de janvier 2019. Puisque la variable étudiée est une durée exprimée en minutes, elle est strictement positive (supérieure à 0), car un bus avec zéro minute de retard est à l'heure. Nous considérons également qu'un bus ayant plus de 150 minutes de retard (2h30) n'est tout simplement pas passé (personne ne risque d'attendre 2h30 pour prendre son bus). Commençons par charger les données et observer leur distribution empirique.

```
library(ggplot2)
# charger le jeu de données
data_trt_bus <- read.csv('data/univariee/bus-delay-2019_janv.csv', sep =';')
# retirer les observations aberrantes
data_trt_bus <- subset(data_trt_bus, data_trt_bus$Min.Delay > 0 &
                        data_trt_bus$Min.Delay < 150)
# représenter la distribution empirique du jeu de données
ggplot(data = data_trt_bus) +
```

²⁵ <https://open.toronto.ca/catalogue/?search=bus%20delay&sort=score%20desc>

```
geom_histogram(aes(x=Min.Delay, y = ..density..), bins = 40) +
  geom_density(aes(x=Min.Delay), color = 'blue', bw = 2, size = 0.8) +
  labs(x = 'temps de retard (min)',
       y = '')
```

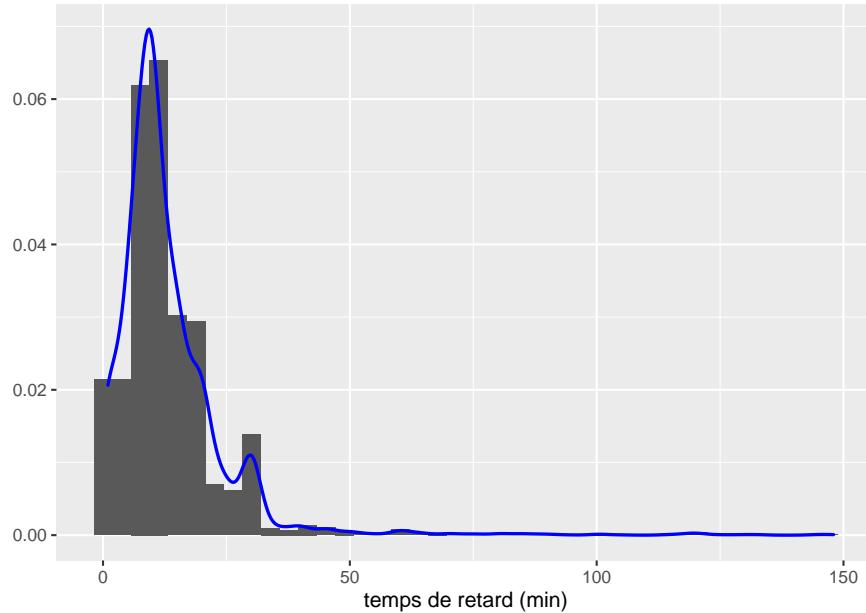


FIG. 2.31 : Distribution empirique des temps de retard des bus à Toronto en janvier 2019

Compte tenu de la forme de la distribution empirique et de sa nature, quatre distributions sont envisageables :

- La distribution Gamma, strictement positive et asymétrique, elle est aussi une généralisation de la distribution exponentielle utilisée pour modéliser des temps d'attente. Pour des raisons similaires, on peut aussi retenir la distribution de Weibull et la distribution log-normale. Nous écartons ici la distribution skew-normale puisque le jeu de données n'a clairement pas une forme normale au départ.
- La distribution de Pareto, strictement positive et permettant de représenter ici le fait que la plupart des retards durent moins de 10 minutes, mais que quelques retards sont également beaucoup plus longs.

Commençons par ajuster les quatre distributions avec la fonction `fitdist` du package **fitdistrplus** et représentions-les graphiquement pour éliminer les moins bons candidats. Nous utilisons également le package **actuar** pour la fonction de densité de Pareto (`dpareto`).

```
library(fitdistrplus)
library(actuar)
library(ggpubr)
# ajustement des modèles
model_gamma <- fitdist(data_trt_bus$Min.Delay, distr = "gamma")
model_weibull <- fitdist(data_trt_bus$Min.Delay, distr = "weibull")
model_lognorm <- fitdist(data_trt_bus$Min.Delay, distr = "lnorm")
model_pareto <- fitdist(data_trt_bus$Min.Delay, distr = "pareto",
                        start = list(shape = 1, scale = 1),
                        method = "mse") # différentes méthodes d'optimisations
```

```
# réalisation des graphiques
plot1 <- ggplot(data = data_trt_bus) +
  geom_histogram(aes(x=Min.Delay, y = ..density..), bins = 40) +
  stat_function(fun = dgamma, color = 'red', size = 0.8,
                args = as.list(model_gamma$estimate)) +
  labs(x = 'temps de retard (min)',
       y = '',
       subtitle = "modèle Gamma")
plot2 <- ggplot(data = data_trt_bus) +
  geom_histogram(aes(x=Min.Delay, y = ..density..), bins = 40) +
  stat_function(fun = dweibull, color = 'red', size = 0.8,
                args = as.list(model_weibull$estimate)) +
  labs(x = 'temps de retard (min)',
       y = '',
       subtitle = "modèle Weibull")
plot3 <- ggplot(data = data_trt_bus) +
  geom_histogram(aes(x=Min.Delay, y = ..density..), bins = 40) +
  stat_function(fun = dlnorm, color = 'red', size = 0.8,
                args = as.list(model_lognorm$estimate)) +
  labs(x = 'temps de retard (min)',
       y = '',
       subtitle = "modèle log-normal")
plot4 <- ggplot(data = data_trt_bus) +
  geom_histogram(aes(x=Min.Delay, y = ..density..), bins = 40) +
  stat_function(fun = dpareto, color = 'red', size = 0.8,
                args = as.list(model_pareto$estimate)) +
  labs(x = 'temps de retard (min)',
       y = '',
       subtitle = "Modèle Pareto")
ggarrange(plotlist = list(plot1, plot2, plot3, plot4),
          ncol = 2, nrow = 2)
```

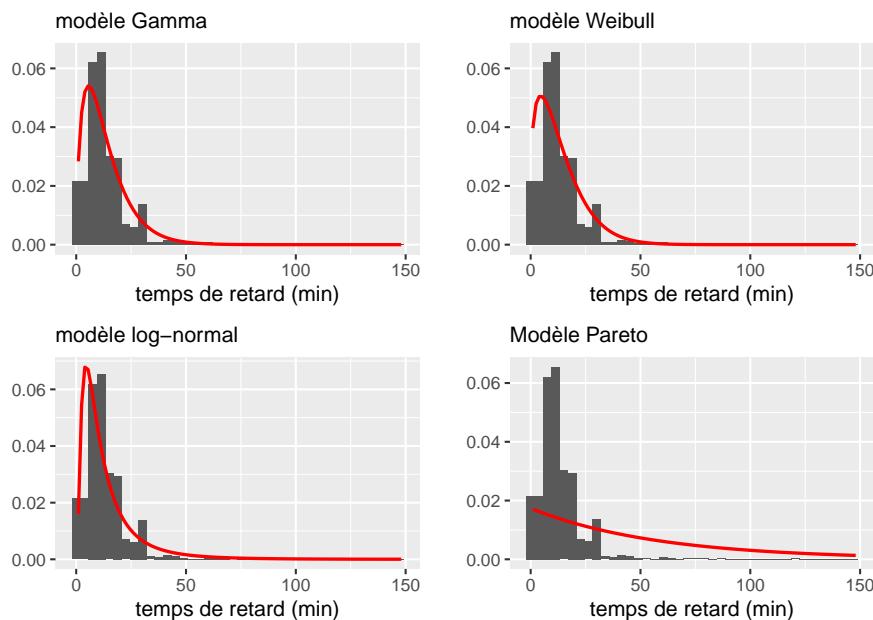


FIG. 2.32 : Comparaison des distributions ajustées aux données de retard des bus

Visuellement, on constate que la distribution de Pareto est un mauvais choix. Pour les trois autres

distributions, la comparaison des *loglikelihood* s'impose.

```
df <- data.frame(model = c("Gamma", "Weibull",
                           "log-normal"),
                  loglikelihood = c(model_gamma$loglik,
                                    model_weibull$loglik,
                                    model_lognorm$loglik))

show_table(df,
           col.names = c("Distributon", "LogLikelihood"),
           caption = 'Comparaison des LogLikekelihood des trois distributions',
           )
```

Le plus grand *logLikelihood* est obtenu par la distribution de Gamma qui s'ajuste donc le mieux à nos données. Pour finir, nous pouvons tester formellement avec le test de Kolmogorov-Smirnov si nos données proviennent bien de cette distribution de Gamma.

```
params <- as.list(model_gamma$estimate)
ks.test(data_trt_bus$Min.Delay,
        y = pgamma, shape = params$shape, rate = params$rate)
```

```
##  
## One-sample Kolmogorov-Smirnov test  
##  
## data: data_trt_bus$Min.Delay  
## D = 0.099912, p-value < 2.2e-16  
## alternative hypothesis: two-sided
```

La valeur de *p* est inférieure à 0,05, on ne peut donc pas accepter l'hypothèse que notre jeu de données suit effectivement un loi de Gamma. Considérant le nombre d'observations et le fait que de nombreux temps d'attente sont identiques (ce à quoi le test est très sensible), ce résultat n'est pas surprenant. La distribution de Gamma reste cependant la distribution qui représente le mieux nos données. Nous pouvons estimer grâce à cette distribution la probabilité qu'un bus ait un retard de plus de 10 minutes de la façon suivante :

```
pgamma(10, shape = params$shape, rate = params$rate, lower.tail = F)
```

```
## [1] 0.5409424
```

ce qui correspond à 54% de chance.

Pour moins de 10 minutes :

```
pgamma(10, shape = params$shape, rate = params$rate, lower.tail = T)
```

```
## [1] 0.4590576
```

TAB. 2.8 : Comparaison des LogLikekelihood des trois distributions

Distributon	LogLikelihood
Gamma	-23 062,56
Weibull	-23 195,54
log-normal	-23 375,74

soit 46%.

Un dernier exemple avec la probabilité qu'un retard dépasse 45 minutes :

```
pgamma(45, shape = params$shape, rate = params$rate, lower.tail = F)
```

```
## [1] 0.01348194
```

Soit seulement 1,3%.

Par conséquent, si un matin à Toronto votre bus a plus de 45 minutes de retard, bravo vous êtes tombé sur une des très rares occasions où un tel retard se produit

2.5.4.2.2 Les accidents de vélo à Montréal

Le second jeu de données représente le nombre d'accidents de la route impliquant un vélo sur les intersections dans les quartiers centraux de Montréal. Le jeu de données complet est disponible sur le site des données ouvertes²⁶ de la ville de Montréal. Puisque ces données correspondent à des comptages, la première distribution à envisager est la distribution de poisson. Cependant, puisque nous aurons également un grand nombre d'intersections sans accident, il serait judicieux de tester la distribution de poisson avec excès de zéro.

```
library(ggplot2)
# charger le jeu de données
data_accidents <- read.csv('data/univariee/accidents_mtl.csv', sep = ',', )
counts <- data.frame(table(data_accidents$nb_accident))
names(counts) <- c("nb_accident", 'fréquence')
counts$nb_accident <- as.numeric(as.character(counts$nb_accident))
counts$prop <- counts$fréquence / sum(counts$fréquence)
# représenter la distribution empirique du jeu de donnée
ggplot(data = counts) +
  geom_bar(aes(x=nb_accident, weight = fréquence), width = 0.5) +
  labs(x = "nombre d'accidents",
       y = 'fréquence')
```

Nous avons effectivement de nombreux zéros ici, essayons d'ajuster nos deux distributions à ce jeu de données. Dans le graphique suivant, les barres grises représentent la distribution empirique du jeu de données et les barres rouges les distributions théoriques ajustées. Nous utilisons ici le package **gamlss.dist** pour avoir la fonction de masse d'une distribution de poisson avec excès de zéros.

```
library(gamlss.dist)
#ajuster le modèle de poisson
model_poisson <- fitdist(data_accidents$nb_accident, distr = "pois")
#ajuster le modèle de poisson avec excès de zéros
model_poissonzi <- fitdist(data_accidents$nb_accident, "ZIP",
  start = list(mu = 4, sigma = 0.15), # valeurs pour faciliter la convergence
  optim.method = "L-BFGS-B", # méthode d'optimisation recommandée dans la doc
  lower = c(0.00001, 0.00001),# valeurs minimales des deux paramètres
  upper = c(Inf, 1)# valeurs maximales des deux paramètres
)
dfpoisson <- data.frame(x=c(0:10),
```

²⁶ <http://donnees.ville.montreal.qc.ca/dataset/collisions-routieres>

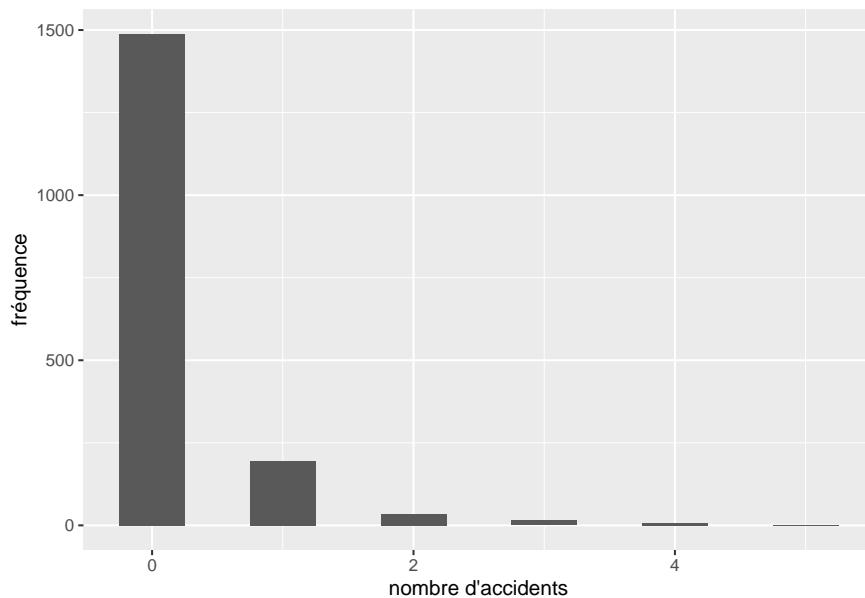


FIG. 2.33 : Distribution empirique du nombre d'accidents par intersection impliquant un cycliste à Montréal en 2017 dans les quartiers centraux

```

y=dpois(0:10, model_poisson$estimate)
)
plot1 <- ggplot() +
  geom_bar(aes(x=nb_accident, weight = prop), width = 0.6, data = counts)+ 
  geom_bar(aes(x=x, weight = y), width = 0.15, data = dfpoisson, fill = "red")+
  scale_x_continuous(limits = c(-0.5,7), breaks = c(0:7))+ 
  labs(subtitle = "modèle poisson",
       x = "nombre d'accidents",
       y = "")
dfpoissonzi <- data.frame(x=c(0:10),
                           y=dZIP(0:10, model_poissonzi$estimate[[1]],
                                   model_poissonzi$estimate[[2]]))
)
plot2 <- ggplot() +
  geom_bar(aes(x=nb_accident, weight = prop), width = 0.6, data = counts)+ 
  geom_bar(aes(x=x, weight = y), width = 0.15, data = dfpoissonzi, fill = "red")+
  scale_x_continuous(limits = c(-0.5,7), breaks = c(0:7))+ 
  labs(subtitle = "modèle poisson avec excès de zéro",
       x = "nombre d'accident",
       y = "")
ggarrange(plotlist = list(plot1,plot2), ncol = 2)

```

Visuellement, le modèle avec excès de zéro semble s'imposer. Nous pouvons vérifier cette impression avec la comparaison des *loglikelihood*.

```
print(model_poisson$loglik)
```

```
## [1] -989.83
```

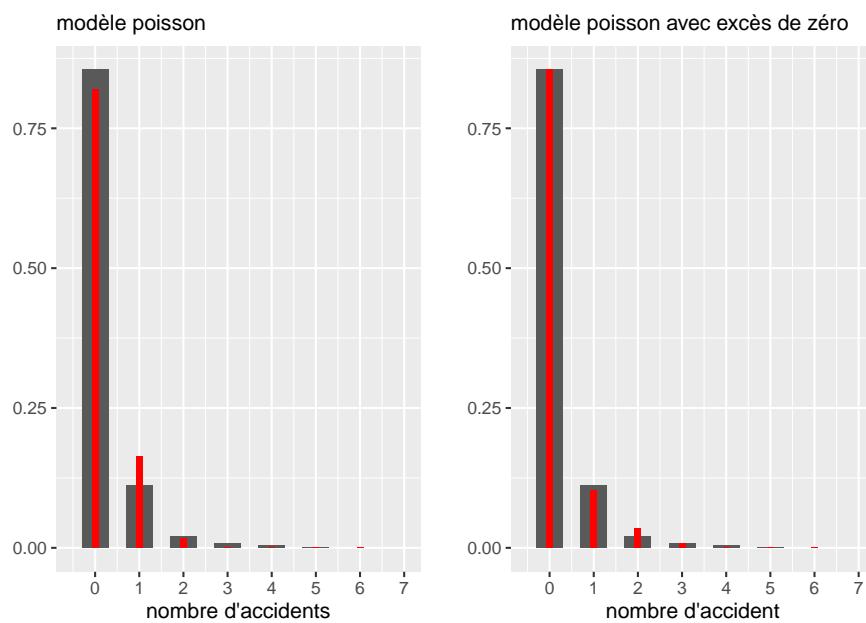


FIG. 2.34 : Ajustement des distributions de poisson et poisson avec excès de zéros

```
print(model_poissonzi$loglik)
```

```
## [1] -931.8778
```

```
#afficher les paramètres ajustés
model_poissonzi$estimate
```

```
##      mu      sigma
## 0.6690301 0.7022605
```

Nous avons donc la confirmation que le modèle de poisson avec excès de zéros est mieux ajusté. Nous apprenons donc que 70% ($\sigma = 0,70$) des intersections sont en fait exclues du phénomène étudié (probablement parce que très peu de cyclistes les utilisent ou parce qu'elles sont très peu accidentogènes) et que pour les autres, le taux d'accidents par année en 2017 était de 0,67 ($\mu = 0,669$, μ signifiant λ pour le package **gamlss**). À nouveau, nous pouvons effectuer un test formel avec la fonction **ks.test**.

```
params <- as.list(model_poissonzi$estimate)
ks.test(data_accidents$nb_accident,
       y = pZIP, mu = params$mu, sigma = params$sigma)
```

```
##
## One-sample Kolmogorov-Smirnov test
##
## data: data_accidents$nb_accident
## D = 0.85476, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Encore une fois, on doit rejeter l'hypothèse selon laquelle le test suit une distribution de poisson avec excès de zéros. Ces deux exemples montrent à quel point ce test est restrictif.

2.5.5 La transformation des variables

2.5.5.1 Les transformations visant à atteindre la normalité

Comme énoncé au début de cette section, plusieurs méthodes statistiques nécessitent que la variable quantitative soit normalement distribuée. C'est notamment le cas de l'analyse de variance et des tests t (abordés dans les chapitres suivants) qui fourniront des résultats plus robustes lorsque la variable est normalement distribuée. Plusieurs transformations sont possibles, les plus courantes étant la racine carrée, le logarithme et l'inverse de la variable. Selon plusieurs auteurs (notamment, Tabacknick et *et al.* (2007, p. 89)), en fonction du type (positive ou négative) et du degré d'asymétrie, les transformations suivantes sont possibles afin d'améliorer la normalité de la variable :

- Asymétrie positive modérée : la racine carrée de la variable X avec la fonction `sqrt(df$x)`.
- Asymétrie positive importante : le logarithme de la variable avec `log10(df$x)`
- Asymétrie positive sévère : l'inverse de la variable avec `1/(df$x)`



Attention, pour une valeur égale ou inférieure à 0, on ne peut pas calculer une racine carrée ou un logarithme. Par conséquent, il convient de décaler simplement la distribution vers la droite afin de s'assurer qu'il n'y ait plus de valeurs négative ou égale à 0 :

- `sqrt(df$x - min(df$x+1))` avec pour une asymétrie positive avec des valeurs négatives ou égales à 0
- `log(df$x - min(df$x+1))` pour une asymétrie positive avec des valeurs négatives ou égales à 0

Par exemple, si la valeur minimale de la variable est égale à -10, la valeur minimale de variable décalée sera ainsi de 11.

- Asymétrie négative modérée : `sqrt(max(df$x+1) - df$x)`.
- Asymétrie négative importante : `log(max(df$x+1) - df$x)`
- Asymétrie négative sévère : `1/(max(df$x+1) - df$x)`



Transformation des variables pour atteindre la normalité : ce n'est pas toujours la panacée !

La transformation des données fait et fera encore longtemps débat à la fois parmi les statisticiens, les débutants et utilisateurs avancés des méthodes quantitatives. Field et al. (2012, pp. 193) résument le tout avec humour : « To transform or not transform, that is the question ».

Avantages de la transformation

- L'obtention de *résultats plus robustes*.
- Dans une régression linéaire multiple, la transformation de la variable dépendante peut *remédier au non-respect des hypothèses de base liées à la régression* (linéarité et homoscédasticité des erreurs, absence des valeurs aberrantes, etc.).

Inconvénients de la transformation

- *Une variable transformée est plus difficile à interpréter* puisque cela change l'unité de mesure de la variable. Prenons un exemple concret : vous souhaitez comparer les moyennes de revenu de deux groupes A et B . Vous obtenez une différence de 15000\$, soit une valeur facile à interpréter. Par contre, si la variable a été préalablement transformée en logarithme, il est possible que vous obteniez une différence de 9, ce qui est beaucoup moins parlant. Aussi, en transformant la variable en *log*, vous ne comparez plus les moyennes arithmétiques des deux groupes, mais plutôt leurs moyennes géométriques (Field et al., 2012, pp. 193).
- *Pourquoi perdre la forme initiale de la distribution du phénomène à expliquer ?* Il est possible pour de nombreuses méthodes de choisir la distribution que l'on souhaite utiliser, il n'est donc pas nécessaire de toujours se limiter à la distribution normale. Par exemple, dans les modèles de régression généralisés (GLM), on pourrait indiquer que notre variable indépendante suit une distribution de *Student* plutôt

que de vouloir à tout prix la rendre normale. De même, certains tests non-paramétriques permettent d'analyser des variables ne suivant pas une distribution normale.

Démarche à suivre avant et après la transformation

- *La transformation est-elle nécessaire?* Ne transformez jamais une variable sans avoir analyser rigoureusement sa forme (histogramme avec courbe normale, *skewness* et *kurtosis*, tests de normalité).
- *D'autres options à la transformation d'une variable dépendante (VD) sont-elles envisageables?* Identifiez la forme de la distribution de la VD et utilisez au besoin un modèle GLM adapté à cette distribution. Autrement dit, ne transformez pas automatiquement votre VD pour simplement pouvoir l'introduire dans une régression linéaire multiple.
- *La transformation a-t-elle un apport significatif?* Premièrement, vérifiez si la transformation utilisée (logarithme, racine carrée, inverse, etc.) améliore la normalité de la variable. Ce n'est toujours le cas, pourquoi c'est pire! Prenez soin de comparer les histogrammes, les valeurs de *skewness*, *kurtosis* et des différents tests de normalité avant et après la transformation. Deuxièmement, comparez les résultats de vos analyses statistiques sans et avec transformation, et ce, dans une démarche coût-avantage. Vos résultats sont-ils bien plus robustes? Par exemple, un R^2 qui passe de 0,597 à 0,602 avant et après la transformation des variables avec des associations significatives similaires, mais plus difficiles à interpréter (du fait des transformations), n'est pas forcément un gain significatif. La modélisation en sciences sociales ne vise pas à prédire la trajectoire d'un satellite ou l'atterrissement d'un engin sur Mars! La précision à la quatrième décimale n'est pas une condition! Par conséquent, un modèle un peu moins robuste, mais plus facile à interpréter est parfois préférable.

2.5.5.2 Autres types de transformations

Les trois transformations les plus couramment utilisées sont :

- **La côte z** (*z score* en anglais) qui consiste à soustraire à chaque valeur sa moyenne (soit un centrage), puis à la diviser par son écart-type (soit une réduction) (eq. (2.29)). Par conséquent, on parle aussi de variable centrée-réduite qui a comme propriétés intéressantes une moyenne égale à 0 et un écart-type égale à 1 (la variance est aussi égale à 1 puisque $1^2 = 1$). Nous verrons que cette transformation est largement utilisée dans les méthodes de classification (chapitre 9) et les méthodes factorielles (chapitre 10).

$$z = \frac{x_i - \mu}{\sigma} \quad (2.29)$$

- **La transformation en rangs** qui consiste simplement à trier une variable en ordre croissant, puis à affecter le rang de chaque observation de 1 à n . Cette transformation est très utilisée quand la variable est très anormalement distribuée, notamment pour calculer le coefficient de corrélation de Spearman (section 4.1.3.3) et certains tests non-paramétriques (sections 4.3.2 et 4.4.2).
- **La transformation sur une échelle de 0 à 1** (ou de 0 à 100) qui consiste à soustraite à chaque observation la valeur minimale et à diviser le tout par l'étendue (eq. (2.30)).

$$X_{\in[0-1]} = \frac{x_i - \text{max}}{\text{max} - \text{min}} \text{ ou } X_{\in[0-100]} = \frac{x_i - \text{min}}{\text{max} - \text{min}} \times 100 \quad (2.30)$$

Pour un *dataframe* nommé *df* comprenant une variable *x*, la syntaxe ci-dessous illustre comment obtenir quatre transformations (côte *z*, rangs, 0 à 1 et 0 à 100).

TAB. 2.9 : Illustration des trois transformations

Observation	x_i	Côte z	Rang	0 à 1
1	22,00	-1,45	1	0,00
2	27,00	-0,65	3	0,28
3	25,00	-0,97	2	0,17
4	30,00	-0,16	4	0,44
5	37,00	0,97	7	0,83
6	32,00	0,16	5	0,56
7	35,00	0,65	6	0,72
8	40,00	1,45	8	1,00
Moyenne	31,00	0,00		
Écart-type	6,19	1,00		

```
df2 <- data.frame(x = c(22, 27, 25, 30, 37, 32, 35, 40))

# Transformation centrée-réduite : côte Z
df2$zx <- (df2$x - mean(df2$x)) / sd(df2$x)

# Transformation en rangs avec la fonction rank
df2$rz <- rank(df2$x)

# Transformation en rangs de 0 à 1
df2$x01 <- (df2$x - min(df2$x)) / (max(df2$x) - min(df2$x))

# Transformation en rangs de 0 à 100
df2$x0100 <- (df2$x - min(df2$x)) / (max(df2$x) - min(df2$x)) * 100
```



Ces trois transformations sont parfois utilisées pour générer un indice composite à partir de plusieurs variables ou encore dans une analyse de sensibilité avec les indices de Sobol (1993).

2.5.6 Mise en œuvre dans R

Il existe une multitude de *packages* dédiés au calcul des statistiques descriptives univariées. Par parcimonie, nous en utiliserons uniquement trois : `DescTools`, `nortest` et `stats`. Libre à vous de faire vos recherches sur Internet pour utiliser d'autres *packages* au besoin. Les principales fonctions que nous utilisons ici sont :

- `summary` : pour obtenir un résumé sommaire des statistiques descriptives (minimum, Q1, Q2 Q3, Maximum)
- `mean` : moyenne
- `min` : minimum
- `max` : maximum
- `range` : minimum et maximum
- `quantile` : quartiles
- `quantile((x, probs = seq(.0, 1, by = .2)))` : quintiles
- `quantile((x, probs = seq(.0, 1, by = .1)))` : déciles
- `var` : variance
- `sd` : écart-type
- `Skew` du *package DescTools* : coefficient d'asymétrie
- `Kurt` du *package DescTools* : coefficient d'appatissement

- `ks.test(x, "pnorm", mean=mean(x), sd=sd(x))` du package `nortest` : test de Kolmogorov-Smirnov
- `shapiro.test` du package `DescTools` : test de Shapiro-Wilk
- `lillie.test` du package `DescTools` : du package `nortest` : test de Lilliefors
- `ad.test` du package `DescTools` : test d'Anderson-Darling
- `JarqueBeraTest` du package `DescTools` : test de Jarque-Bera

2.5.6.1 Application à une seule variable

Admettons que vous voulez obtenir des statistiques pour une seule variable présente dans un *dataframe* (`dataMTL$PctFRev`) :

```
library(DescTools)
library(stats)
library(nortest)

# Importation du fichier csv dans un dataframe
dataMTL <- read.csv("data/univariee/DataSR2016.csv")
# Tableau sommaire pour la variable PctFRev
summary(dataMTL$PctFRev)

##      Min. 1st Qu. Median   Mean 3rd Qu.   Max.
## 1.846 11.242 15.471 16.822 20.229 68.927

# PARAMÈTRES DE TENDANCE CENTRALE
mean(dataMTL$PctFRev)    # Moyenne

## [1] 16.82247

median(dataMTL$PctFRev)   # Médiane

## [1] 15.471

# PARAMÈTRES DE POSITION
# Quartiles
quantile(dataMTL$PctFRev)

##        0%       25%       50%       75%      100%
## 1.8460 11.2420 15.4710 20.2285 68.9270

# Quintiles
quantile(dataMTL$PctFRev, probs = seq(.0, 1, by = .2))

##        0%      20%      40%      60%      80%     100%
## 1.846 10.294 13.626 16.918 21.756 68.927

# Déciles
quantile(dataMTL$PctFRev, probs = seq(.0, 1, by = .1))

##        0%      10%      20%      30%      40%      50%      60%      70%      80%      90%     100%
## 1.846  8.402 10.294 12.172 13.626 15.471 16.918 18.868 21.756 26.854 68.927
```

```

# Percentiles personnalisés avec apply
quantile(dataMTL$PctFRev, probs = c(0.01,.05,0.10,.25,.50,.75,.90,.95,.99))

##      1%      5%     10%     25%     50%     75%     90%     95%     99%
## 5.2290 7.1470 8.4020 11.2420 15.4710 20.2285 26.8540 31.7530 45.6010

# PARAMÈTRES DE DISPERSION
range(dataMTL$PctFRev) # Min et Max

## [1] 1.846 68.927

# Étendue
max(dataMTL$PctFRev)-min(dataMTL$PctFRev)

## [1] 67.081

# Écart interquartile
quantile(dataMTL$PctFRev) [4]-quantile(dataMTL$PctFRev) [2]

##      75%
## 8.9865

var(dataMTL$PctFRev) # Variance

## [1] 66.62482

sd(dataMTL$PctFRev) # Écart-type

## [1] 8.162403

sd(dataMTL$PctFRev) / mean(dataMTL$PctFRev) # CV

## [1] 0.4852083

# PARAMÈTRES DE FORME
Skew(dataMTL$PctFRev) # Skewness

## [1] 1.67367

Kurt(dataMTL$PctFRev) # Kurtosis

## [1] 4.858815

# TESTS D'HYPOTHÈSE SUR LA NORMALITÉ
# K-Smirnov
ks.test(dataMTL$PctFRev, "pnorm", mean=mean(dataMTL$PctFRev), sd=sd(dataMTL$PctFRev))

```

```
##  
## One-sample Kolmogorov-Smirnov test  
##  
## data: dataMTL$PctFRev  
## D = 0.10487, p-value = 1.646e-09  
## alternative hypothesis: two-sided
```

```
shapiro.test(dataMTL$PctFRev)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: dataMTL$PctFRev  
## W = 0.88748, p-value < 2.2e-16
```

```
lillie.test(dataMTL$PctFRev)
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data: dataMTL$PctFRev  
## D = 0.10487, p-value < 2.2e-16
```

```
ad.test(dataMTL$PctFRev)
```

```
##  
## Anderson-Darling normality test  
##  
## data: dataMTL$PctFRev  
## A = 21.072, p-value < 2.2e-16
```

```
JarqueBeraTest(dataMTL$PctFRev)
```

```
##  
## Robust Jarque Bera Test  
##  
## data: dataMTL$PctFRev  
## X-squared = 2173.1, df = 2, p-value < 2.2e-16
```

Pour construire un histogramme avec la courbe normale, vous pourrez consulter la section [3.2.1.3](#) ou la syntaxe ci-dessous.

```
moyenne <- mean(dataMTL$PctFRev)  
ecart_type <- sd(dataMTL$PctFRev)  
  
ggplot(data = dataMTL) +  
  geom_histogram(aes(x = PctFRev, y = ..density..),  
    bins = 30, color = "#343a40", fill = "#a8dadc") +  
  labs(y = "densité") +
```

```
stat_function(fun = dnorm, args = list(mean = moyenne, sd = ecart_type),
              color = "#e63946", size = 1.2, linetype = "dashed")
```

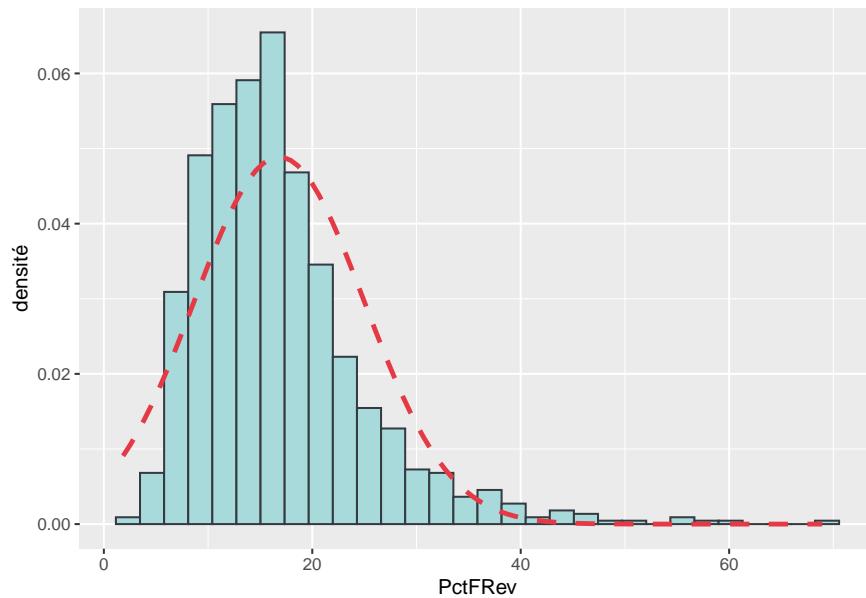


FIG. 2.35 : Histogramme avec courbe normale

2.5.6.2 Application à plusieurs variables

Pour obtenir des sorties de statistiques descriptives pour plusieurs variables, nous vous conseillons :

- de créer un vecteur avec les noms de variables (*VarsSelect* dans la syntaxe ci-dessous)
- d'utiliser ensuite les fonctions *sapply* et *apply*.

```
# Noms des variables du dataframe
names(dataMTL)

## [1] "CTNAME"                 "PopTotal"                "HabKm2"
## [4] "PctFRev"                 "TxChomage"               "PctImmigrant"
## [7] "PctImgRecent"            "PctMenage1pers"          "PctFamilleMono"
## [10] "PctLangueMaternelleFR"  "PctLangueMaternelleAN"  "PctLangueMaternelleAU"

# Vecteur pour trois variables
VarsSelect <- c("HabKm2", "TxChomage", "PctFRev" )

# Tableau sommaire pour les 3 variables
summary(dataMTL[VarsSelect])

##      HabKm2        TxChomage        PctFRev
##  Min.   : 18   Min.   : 1.942   Min.   : 1.846
##  1st Qu.: 1980  1st Qu.: 5.482   1st Qu.:11.242
##  Median : 3773  Median : 7.130   Median :15.471
##  Mean    : 5513  Mean    : 7.743   Mean    :16.822
##  3rd Qu.: 7916  3rd Qu.: 9.391   3rd Qu.:20.229
```

```
##   Max. :50282   Max. :26.882   Max. :68.927
```

```
# PARAMÈTRES DE TENDANCE CENTRALE
sapply(dataMTL[VarsSelect], mean) # Moyenne
```

```
##      HabKm2 TxChomage PctFRev
## 5512.830705    7.743329 16.822470
```

```
sapply(dataMTL[VarsSelect], median) # Médiane
```

```
##      HabKm2 TxChomage PctFRev
## 3773.000    7.130     15.471
```

```
# PARAMÈTRES DE POSITION
# Quartiles
sapply(dataMTL[VarsSelect], quantile)
```

```
##      HabKm2 TxChomage PctFRev
## 0%       18.0     1.9420  1.8460
## 25%     1980.5     5.4825 11.2420
## 50%     3773.0     7.1300 15.4710
## 75%     7915.5     9.3910 20.2285
## 100%    50282.0    26.8820 68.9270
```

```
# Quintiles
apply(dataMTL[VarsSelect], 2, function(x) quantile(x, probs = seq(.0, 1, by = .2)))
```

```
##      HabKm2 TxChomage PctFRev
## 0%       18     1.942    1.846
## 20%     1525    5.116   10.294
## 40%     2953    6.422   13.626
## 60%     4971    7.973   16.918
## 80%     9509   10.000   21.756
## 100%    50282   26.882   68.927
```

```
# Déciles
apply(dataMTL[VarsSelect], 2, function(x) quantile(x, probs = seq(.0, 1, by = .1)))
```

```
##      HabKm2 TxChomage PctFRev
## 0%       18     1.942    1.846
## 10%      455    4.369    8.402
## 20%     1525    5.116   10.294
## 30%     2298    5.780   12.172
## 40%     2953    6.422   13.626
## 50%     3773    7.130   15.471
## 60%     4971    7.973   16.918
## 70%     6918    8.909   18.868
## 80%     9509   10.000   21.756
## 90%    13055   11.749   26.854
```

```

## 100% 50282 26.882 68.927

# Percentiles personnalisés avec apply
apply(dataMTL[VarsSelect], 2,
      function(x) quantile(x, probs = c(0.01,.05,0.10,.25,.50,.75,.90,.95,.99)))

##      HabKm2 TxChomage PctFRev
## 1%     58.5    2.9665  5.2290
## 5%    178.0    3.8980  7.1470
## 10%   455.0    4.3690  8.4020
## 25%  1980.5    5.4825 11.2420
## 50% 3773.0    7.1300 15.4710
## 75% 7915.5    9.3910 20.2285
## 90% 13055.0   11.7490 26.8540
## 95% 15355.0   13.8400 31.7530
## 99% 18578.5   17.1920 45.6010

# PARAMÈTRES DE DISPERSION
sapply(dataMTL[VarsSelect], range) # Min et Max

##      HabKm2 TxChomage PctFRev
## [1,]     18     1.942   1.846
## [2,] 50282    26.882  68.927

# Étendue
sapply(dataMTL[VarsSelect], max) - sapply(dataMTL[VarsSelect], min)

##      HabKm2 TxChomage PctFRev
## 50264.000 24.940    67.081

# Écart interquartile
sapply(dataMTL[VarsSelect], quantile)[4,] - sapply(dataMTL[VarsSelect], quantile)[2,]

##      HabKm2 TxChomage PctFRev
## 5935.0000 3.9085    8.9865

sapply(dataMTL[VarsSelect], var) # Variance

##      HabKm2      TxChomage      PctFRev
## 2.633462e+07 9.880932e+00 6.662482e+01

sapply(dataMTL[VarsSelect], sd) # Écart-type

##      HabKm2      TxChomage      PctFRev
## 5131.726785  3.143395    8.162403

# Coefficient de variation
sapply(dataMTL[VarsSelect], sd) / sapply(dataMTL[VarsSelect], mean)

```

```

##      HabKm2 TxChomage   PctFRev
## 0.9308696 0.4059488 0.4852083

# PARAMÈTRES DE FORME
sapply(dataMTL[VarsSelect], Skew)      # Skewness

##      HabKm2 TxChomage   PctFRev
## 1.967468 1.280216 1.673670

sapply(dataMTL[VarsSelect], Kurt)       # Kurtosis

##      HabKm2 TxChomage   PctFRev
## 8.546403 2.892443 4.858815

# TESTS D'HYPOTHÈSE POUR LA NORMALITÉ
# K-Smirnov
apply(dataMTL[VarsSelect], 2, function(x) ks.test(x, "pnorm", mean=mean(x), sd=sd(x)))

## $HabKm2
##
## One-sample Kolmogorov-Smirnov test
##
## data: x
## D = 0.14899, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
## 
## $TxChomage
##
## One-sample Kolmogorov-Smirnov test
##
## data: x
## D = 0.080183, p-value = 9.778e-06
## alternative hypothesis: two-sided
##
## 
## $PctFRev
##
## One-sample Kolmogorov-Smirnov test
##
## data: x
## D = 0.10487, p-value = 1.646e-09
## alternative hypothesis: two-sided

sapply(dataMTL[VarsSelect], shapiro.test)      # Shapiro-Wilk

##          HabKm2                  TxChomage
## statistic 0.8385086            0.9235146
## p.value   5.648795e-30         1.451222e-21

```

```

## method      "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name  "X[[i]]"                      "X[[i]]"
##          PctFRev
## statistic  0.8874803
## p.value    1.00278e-25
## method      "Shapiro-Wilk normality test"
## data.name  "X[[i]]"

```

```
sapply(dataMTL[VarsSelect], lillie.test)           # Lilliefors
```

```

##          HabKm2
## statistic 0.148988
## p.value   5.689619e-58
## method    "Lilliefors (Kolmogorov-Smirnov) normality test"
## data.name "X[[i]]"
##          TxChomage
## statistic 0.0801829
## p.value   7.758887e-16
## method    "Lilliefors (Kolmogorov-Smirnov) normality test"
## data.name "X[[i]]"
##          PctFRev
## statistic 0.1048704
## p.value   7.43257e-28
## method    "Lilliefors (Kolmogorov-Smirnov) normality test"
## data.name "X[[i]]"

```

```
sapply(dataMTL[VarsSelect], ad.test)                # Anderson-Darling
```

```

##          HabKm2                  TxChomage
## statistic 36.40276              14.9237
## p.value   3.7e-24                3.7e-24
## method    "Anderson-Darling normality test" "Anderson-Darling normality test"
## data.name "X[[i]]"                "X[[i]]"
##          PctFRev
## statistic 21.07194
## p.value   3.7e-24
## method    "Anderson-Darling normality test"
## data.name "X[[i]]"

```

```
sapply(dataMTL[VarsSelect], JarqueBeraTest)         # Jarque-Bera
```

```

##          HabKm2                  TxChomage
## statistic 4270.113              639.2741
## parameter 2                     2
## p.value   0                      0
## method    "Robust Jarque Bera Test" "Robust Jarque Bera Test"
## data.name "X[[i]]"                "X[[i]]"
##          PctFRev
## statistic 2173.082

```

```
## parameter 2
## p.value    0
## method     "Robust Jarque Bera Test"
## data.name  "X[[i]]"
```

2.5.6.3 Transformer une variable dans R

La syntaxe ci-dessous illustre trois exemples de transformation (logarithme, racine carrée et inverse de la variable). Rappelez-vous qu'il faut comparer les valeurs de forme (*skewness* et *kurtosis*) et de forme (tests de Shapiro-Wilk) avant et après les transformations pour identifier celle qui est la plus efficace.

```
library(ggpubr)

# Importation du fichier csv dans un dataframe
dataMTL <- read.csv("data/univariee/DataSR2016.csv")

# Noms des variables du dataframe
names(dataMTL)

## [1] "CTNAME"           "PopTotal"          "HabKm2"
## [4] "PctFRev"          "TxChomage"         "PctImmigrant"
## [7] "PctImgRecent"     "PctMenage1pers"    "PctFamilleMono"
## [10] "PctLangueMaternelleFR" "PctLangueMaternelleAN" "PctLangueMaternelleAU"

# Transformations
dataMTL$HabKm2_log <- log10(dataMTL$HabKm2)
dataMTL$HabKm2_sqrt <- sqrt(dataMTL$HabKm2)
dataMTL$HabKm2_inv <- 1/dataMTL$HabKm2

# Vecteur pour la variable et les trois transformations
VarsSelect <- c("HabKm2", "HabKm2_log", "HabKm2_sqrt", "HabKm2_inv")

# paramètres de forme
sapply(dataMTL[VarsSelect], Skew)      # Skewness

##      HabKm2  HabKm2_log  HabKm2_sqrt  HabKm2_inv
## 1.9674683 -1.2071326   0.4179037   8.2536901

sapply(dataMTL[VarsSelect], Kurt)       # Kurtosis

##      HabKm2  HabKm2_log  HabKm2_sqrt  HabKm2_inv
## 8.54640302 1.55670769  0.04563433 82.85604898

# TESTS D'HYPOTHÈSE SUR LA NORMALITÉ
sapply(dataMTL[VarsSelect], shapiro.test)

##            HabKm2          HabKm2_log
## statistic 0.8385086          0.9113234
## p.value   5.648795e-30          4.11156e-23
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"                  "X[[i]]"
```

```

##             HabKm2_sqrt           HabKm2_inv
## statistic 0.9771699          0.2530266
## p.value   4.638049e-11       8.324983e-52
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"           "X[[i]]"

# Histogrammes avec courbe normale
Graph1 <- ggplot(data = dataMTL) +
  geom_histogram(aes(x = HabKm2, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#a8dadc") +
  labs(x="Habitants au km2", y = "densité")+
  stat_function(fun = dnorm,
                args = list(mean = mean(dataMTL$HabKm2),
                            sd = sd(dataMTL$HabKm2)),
                color = "#e63946", size = 1.2)

Graph2 <- ggplot(data = dataMTL) +
  geom_histogram(aes(x = HabKm2_log, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#a8dadc") +
  labs(x="habitants au km2 (logarithme)", y = "densité")+
  stat_function(fun = dnorm,
                args = list(mean = mean(dataMTL$HabKm2_log),
                            sd = sd(dataMTL$HabKm2_log)),
                color = "#e63946", size = 1.2)

Graph3 <- ggplot(data = dataMTL) +
  geom_histogram(aes(x = HabKm2_sqrt, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#a8dadc") +
  labs(x="habitants au km2 (racine carrée)", y = "densité")+
  stat_function(fun = dnorm,
                args = list(mean = mean(dataMTL$HabKm2_sqrt),
                            sd = sd(dataMTL$HabKm2_sqrt)),
                color = "#e63946", size = 1.2)

Graph4 <- ggplot(data = dataMTL) +
  geom_histogram(aes(x = HabKm2_inv, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#a8dadc") +
  labs(x="habitants au km2 (inverse)", y = "densité")+
  stat_function(fun = dnorm,
                args = list(mean = mean(dataMTL$HabKm2_inv), sd = sd(dataMTL$HabKm2_inv)),
                color = "#e63946", size = 1.2)

ggarrange(plotlist = list(Graph1, Graph2, Graph3, Graph4), ncol = 2, nrow=2)

```

La variable *HabKm2* est asymétrique positive et leptokurtique. Tant les valeurs des statistiques de forme, du test de Shapiro-Wilk que les histogrammes semblent démontrer que la transformation la plus efficace est la racine carrée. Si la variable originale est asymétrique positive, sa transformation logarithme est par contre asymétrique négative. Cela démontre que la transformation logarithmique n'est pas toujours la panacée.

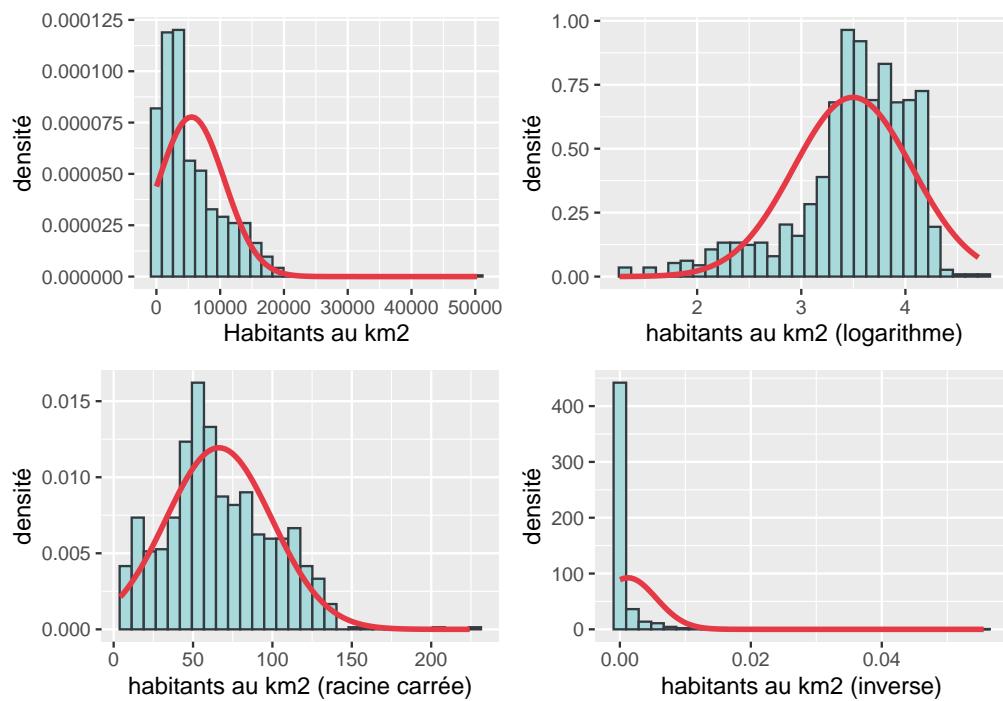


FIG. 2.36 : Histogramme des transformations

2.6 Statistiques descriptives sur des variables qualitatives et semi-qualitatives

2.6.1 Les fréquences

En guise de rappel, les variables nominales, ordinaires et semi-quantitatives comprennent plusieurs modalités pour lesquelles plusieurs types de fréquences sont généralement calculées. Pour illustrer le tout, nous avons extrait du recensement de 2016 de Statistique Canada les effectifs des modalités de la variable sur le principal mode de transport utilisé pour les déplacements domicile-travail, et ce, pour la subdivision de recensement (MRC) de l'île de Montréal (tableau 2.10). Les différents types de fréquences sont les suivantes :

- les fréquences absolues simples (**FAS**) ou fréquences observées représentent le nombre d'observations pour chacune des modalités. Par exemple, sur 857 540 navetteurs domicile-travail (ligne totale), seulement 30 645 optent pour le vélo, alors que 427 530 conduisent un véhicule motorisé (automobile, camion ou fourgonnette) comme principal mode de transport.
- les fréquences relatives simples (**FRS**) sont les proportions de chaque modalité sur le total ($30645/857540 = 0,036$); leur somme est égale à 1. Elles peuvent bien entendu être exprimées en pourcentage ($30645/857540 \times 100 = 3,57$); leur somme est alors égale à 100%. Par exemple, 3,7% des navetteurs utilisent le vélo comme mode de transport principal.
- les fréquences absolues cumulées (**FAC**) représentent la fréquence observée (FAS) de la modalité auxquelles sont additionnées celles qui la précèdent. La valeur de la FAC pour la dernière est donc égale au total.
- À partir des fréquences absolues cumulées (FAC), il est alors possible de calculer les fréquences relatives cumulées (**FRC**) en proportion ($453930/857540 = 0,529$) et en pourcentage ($453930/857540 \times 100 = 52,93$). Par exemple, plus de la moitié des navetteurs utilisent l'automobile comme mode de transport principal (passager ou conducteur).

TAB. 2.10 : Les différents types de fréquences sur une variable qualitative ou semi-qualitative

Mode de transport	FAS	FRS	FRS (%)			
Véhicule motorisé (conducteur)	427 530	0,499	49,86	427 530	0,499	49,86
Véhicule motorisé (passager)	26 400	0,031	3,08	453 930	0,529	52,93
Transport en commun	295 860	0,345	34,50	749 790	0,874	87,43
À pied	69 410	0,081	8,09	819 200	0,955	95,53
Bicyclette	30 645	0,036	3,57	849 845	0,991	99,10
Autre moyen	7 695	0,009	0,90	857 540	1,000	100,00
Total	857 540	1,000	100,00			



Les fréquences cumulées : peu pertinentes pour les variables nominales

Le calcul et l'analyse des fréquences cumulées (absolues et relatives) sont très souvent inutiles pour les variables nominales.

Par exemple, au tableau 2.10, la fréquence cumulée relative (en %) est de 87,43% pour la troisième ligne. Cela signifie que 87,43% des navetteurs se déplacent en véhicule motorisé (conducteur ou passager) ou en transport en commun. Par contre, si la troisième modalité avait été *à pied*, le pourcentage aurait été de 61,02 (52,93 + 8,09). Si vous souhaitez calculer les fréquences cumulées sur une variable nominale, assurez-vous que l'ordre des modalités vous convient et de le modifier au besoin. Sinon, abstenez-vous de les calculer !

Les fréquences cumulées : très utiles pour l'analyse pour des variables ordinaires ou semi-quantitatives

Pour des modalités hiérarchisées (variable ordinaire ou semi-quantitative), l'analyse des fréquences cumulées (absolues et relatives) est par contre très intéressante. Par exemple, au tableau 2.11, elle permet de constater rapidement que sur l'île de Montréal, un peu moins du tiers de la population à moins de 25 ans (35,95%) et 83,33% moins de 65 ans.

Différents graphiques peuvent être construits pour illustrer la répartition des observations : les graphiques en barres (verticales et horizontales) avec les fréquences absolues, les diagrammes circulaires ou en anneau pour les fréquences relatives (figure 2.37). Ces graphiques seront présentés plus en détails dans le chapitre suivant.

2.6.2 Mise en œuvre dans R

La syntaxe ci-dessous permet de calculer les différentes fréquences présentées au tableau 2.11. Notez que pour les fréquences cumulées, nous utilisons la fonction `cumsum`.

```
# Vecteur pour les noms des modalités
Modalite <- c("0 à 14 ans",
           "15 à 24 ans",
           "25 à 44 ans",
           "45 à 64 ans",
           "65 à 84 ans",
           "85 ans et plus")
```

TAB. 2.11 : Les différents types de fréquences sur une variable semi-qualitative

Groupes d'âge	FAS	FRS	FRS (%)			
0 à 14 ans	304 470	0,157	15,68	304 470	0,157	15,68
15 à 24 ans	237 555	0,122	12,23	542 025	0,279	27,91
25 à 44 ans	582 150	0,300	29,98	1 124 175	0,579	57,89
45 à 64 ans	494 205	0,254	25,45	1 618 380	0,833	83,33
65 à 84 ans	271 560	0,140	13,98	1 889 940	0,973	97,32
85 ans et plus	52 100	0,027	2,68	1 942 040	1,000	100,00
Total	1 942 040	1,000	100,00			

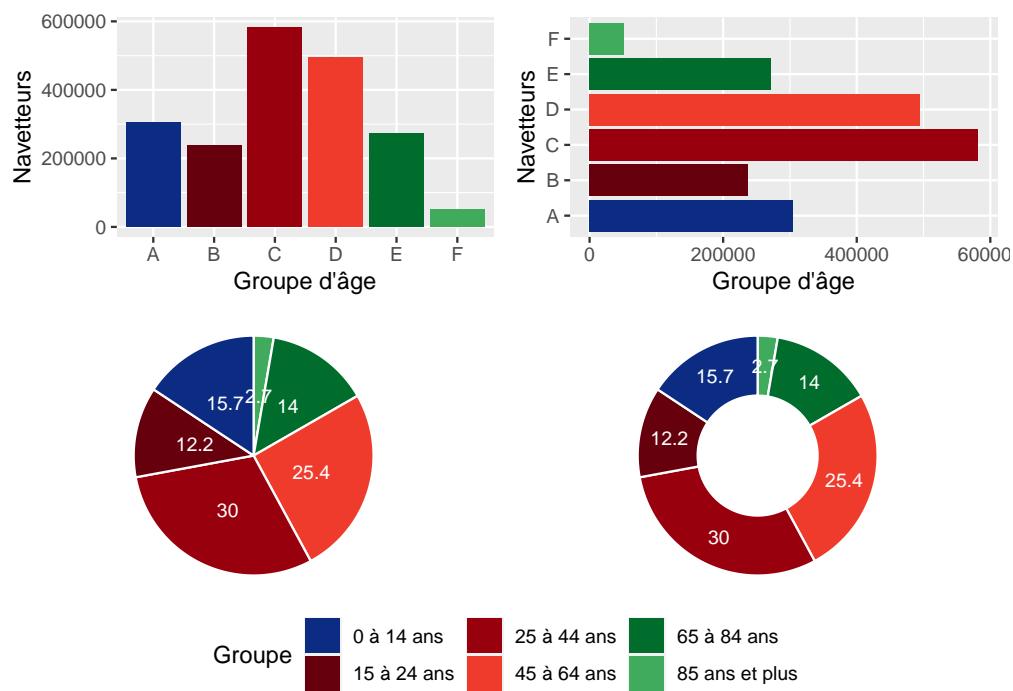


FIG. 2.37 : Différents graphiques pour représenter les fréquences absolues et relatives

```

"25 à 44 ans",
"45 à 64 ans",
"65 à 84 ans",
"85 ans et plus")

# Vecteur pour les fréquences absolues simples (FAS)
Navetteurs <- c(304470, 237555, 582150, 494205, 271560, 52100)
# Somme des FAS
sumFAS <- sum(Navetteurs)
# Construction du dataframe avec les deux vecteurs
df <- data.frame(
  GroupeAge = Modalite,
  FAS = Navetteurs,
  FRS = Navetteurs / sumFAS,
  FRSpct = Navetteurs / sumFAS * 100,
  FAC = cumsum(Navetteurs),
  FRC = cumsum(Navetteurs) / sumFAS,
  FRCpct = cumsum(Navetteurs) / sumFAS * 100
)
df

```

##	GroupeAge	FAS	FRS	FRSpct	FAC	FRC	FRCpct
## 1	0 à 14 ans	304470	0.15677844	15.677844	304470	0.1567784	15.67784
## 2	15 à 24 ans	237555	0.12232240	12.232240	542025	0.2791008	27.91008
## 3	25 à 44 ans	582150	0.29976211	29.976211	1124175	0.5788629	57.88629
## 4	45 à 64 ans	494205	0.25447725	25.447725	1618380	0.8333402	83.33402
## 5	65 à 84 ans	271560	0.13983234	13.983234	1889940	0.9731725	97.31725
## 6	85 ans et plus	52100	0.02682746	2.682746	1942040	1.0000000	100.00000

2.7 Pour aller un peu plus loin : les statistiques descriptives pondérées

Dans la section 2.5, les différentes statistiques descriptives sur des variables quantitatives – paramètres de tendance centrale, de position, de dispersion et de forme – ont été largement abordées. Il est possible de calculer ces différentes statistiques en tenant compte d'une pondération. La statistique descriptive pondérée la plus connue est certainement la moyenne arithmétique pondérée. Son calcul est très simple ; pour chaque observation, deux valeurs sont disponibles :

- x_i , soit la valeur de la variable X pour l'observation i
- w_i , soit la valeur de la pondération pour i .

Prenez soin de comparer les deux équations ci-dessous (à gauche, la moyenne arithmétique ; à droite, la moyenne arithmétique pondérée). Vous constaterez rapidement qu'il suffit simplement de multiplier chaque observation par sa pondération (numérateur) et de diviser ce produit par la somme des pondérations (dénominateur ; et non par n , soit le nombre d'observations comme pour la moyenne arithmétique non pondérée).

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \text{ versus } \bar{m} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} \quad (2.31)$$

Calcul d'autres statistiques descriptives pondérées

Nous n'allons pas reporter ici les formules des versions pondérées de toutes les statistiques descriptives. Retenez toutefois le principe suivant permettant de les calculer à partir de l'exemple du tableau 2.12. Pour la variable X , dupliquons respectivement 20, 80, 50, 200 fois les observations 1 à 4. Si nous calculons la moyenne arithmétique sur ces valeurs dupliquées, alors cette valeur sera identique à la celle de la moyenne arithmétique pondérée. Le même principe reposant sur la duplication des valeurs s'applique à l'ensemble des statistiques descriptives.

Dans un article récent, Alvarenga et al. (2018) évaluent l'accessibilité aux aires de jeux dans les parcs de la Communauté métropolitaine de Montréal (CMM). Pour les 881 secteurs de recensement de la CMM, ils ont calculé la distance à l'aire de jeux la plus proche à travers le réseau de rues. Ce résultat, cartographié à la figure 2.38, permet d'avancer le constat suivant : « la quasi-totalité des secteurs de recensement de l'agglomération de Montréal présente des distances de l'aire de jeux la plus proche inférieures à 500 m, alors que les secteurs situés à plus d'un kilomètre d'une aire de jeux sont très majoritairement localisés dans les couronnes nord et sud de la CMM » (De Alvarenga et al., 2018, p. 238).

Pour chaque secteur de recensement, Alvarenga et al. (2018) disposent des données suivantes :

- x_i , soit la distance à l'aire de jeux la plus proche pour le secteur de recensement i et
- w_i , la pondération, soit le nombre d'enfants de moins de dix ans.

TAB. 2.12 : Calcul de la moyenne pondérée

Observation	x_i	w_i	$x_i \times w_i$
1	200	20	4 000
2	225	80	18 000
3	275	50	13 750
4	300	200	60 000
Somme	1 000	350	95 750
Moyenne	250		
Moyenne pondérée			274

Il est alors possible de calculer les statistiques descriptives de la proximité à l'aire de jeux la plus proche en tenant compte du nombre d'enfants résidant dans chaque secteur de recensement (tableau 2.13). Cet exercice permet de conclure que : « [...] globalement, les enfants ont une bonne accessibilité aux aires de jeux sur le territoire de la CMM. [...] Les enfants sont en moyenne à un peu plus de 500 m de l'aire de jeux la plus proche (moyenne = 559 ; médiane = 512). Toutefois, les valeurs percentiles extrêmes signalent que respectivement 10% et 5% des enfants résident à près de 800 m et à plus de 1000 m de l'aire de jeux la plus proche » (2018, p. 236).

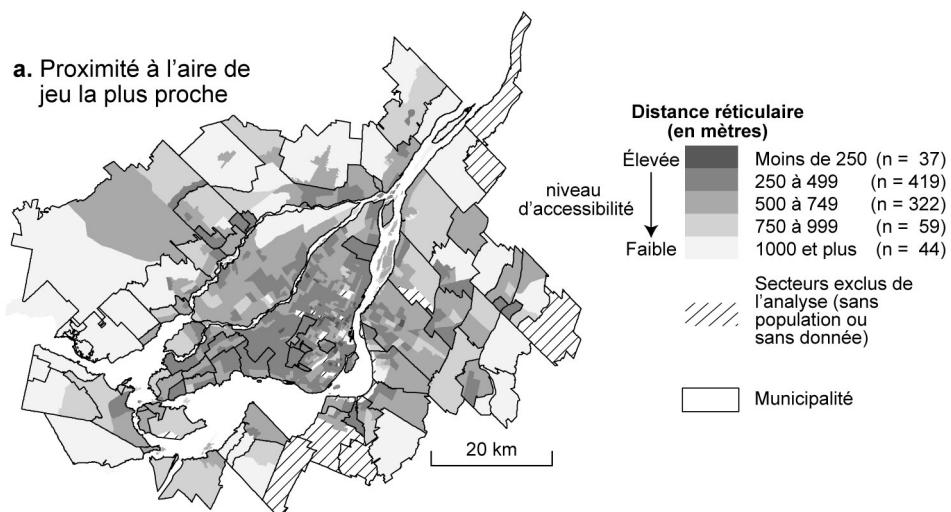


FIG. 2.38 : Accessibilité aux aires de jeux par secteur de recensement, Communauté métropolitaine de Montréal, 2016

De nombreux *packages* sont disponibles pour calculer des statistiques pondérées, dont notamment `Weighted.Desc.Stat` et `Hmisc` utilisés dans la syntaxe ci-dessous.

```
library(foreign)
library(Hmisc)
library(Weighted.Desc.Stat)

df <- read.dbf("data/bivariee/SR_AireJeux_PopMoins10.dbf")

head(df, n = 5)
```

```
##      SRNOM PopMoins10 AireJeux
## 1 0659.06      380 600.1921
## 2 0410.02      390 324.4396
## 3 0863.01      325 524.3323
## 4 0734.05      875 574.6682
## 5 0073.00      100 352.9505
```

TAB. 2.13 : Statistiques de l'aire de jeux la plus proche par secteur de recensement pondérées par la population de moins de 10 ans

N	Moyenne	P5	P10	Q1	Médiane	Q3	P90	P95
881	559	282	327	408	512	640	799	1 006

```

# xi (variable) et wi (pondération)
x <- df$AireJeux
w <- df$PopMoins10

# Calcul des paramètres de position
# Moyenne
Hmisc::wted.mean(x, w)

## [1] 559.8026

Weighted.Desc.Stat::w.mean(x, w)

## [1] 559.8026

# Quartiles et percentile
Hmisc::wted.quantile(x, weights=w, probs=c(.05, .10, .25, .50, .75, .90, .95))

##      5%       10%      25%      50%      75%      90%      95%
## 281.3623 327.3056 406.0759 511.5880 639.4813 798.6559 1011.5493

# Paramètres de dispersion avec le package Weighted.Desc.Stat
# Variance, écart-type et coefficient de variation
w.var(x,w)

## [1] 82818.18

w.sd(x,w)

## [1] 287.7815

w.cv(x,w)

## [1] 0.5140767

# Paramètres de forme avec le package Weighted.Desc.Stat
# Skewness et kurtosis
w.skewness(x, w)

## [1] 4.735351

w.kurtosis(x, w)

## [1] 41.17146

```

Chapitre 3

La magie des graphiques

Dans ce chapitre, nous découvrirons les incroyables capacités graphiques de R. Pour ce faire, nous couvrirons en profondeur les capacités du *package ggplot2* du **tidyverse**. Il s'agit de loin du meilleur *package* pour réaliser des graphiques selon nous.



Dans cette section, nous utiliserons principalement les packages suivants :

- Pour créer des graphiques :
 - * **ggplot2**, le seul, l'unique
 - * **ggbpubr** pour combiner des graphiques
 - * **gthemes** thèmes complémentaires pour les graphiques
- Pour les couleurs :
 - * **rcolorbrewer** pour avoir accès des palettes de couleurs
- Pour les graphiques spéciaux :
 - * **chorddiag** pour construire des graphiques d'accord
 - * **fmsb** pour construire des graphiques ren radar
 - * **treemap** pour construire un graphique *treemap*
 - * **wordcloud2** pour construire un nuage de mots fmsb
- Pour les cartes :
 - * **classInt** pour calculer les intervalles des classes
 - * **ggtern** pour afficher une échelle
 - * **tmap** pour la cartographie
- Autres *packages* :
 - * **dplyr** et **reshape2** pour manipuler des données
 - * **pdftools** pour extraire les textes des fichiers pdf
 - * **udpipe** pour obtenir des dictionnaires linguistiques
 - * **sf** pour manipuler des *simple feature collection*



Qu'est-ce que la visualisation de données ?

La représentation visuelle de données consiste à transposer des informations en une représentation graphique facilitant la lecture de ces dernières. Il s'agit autant d'un ensemble de méthodes, d'un art que d'un moyen de communication. Voici deux exemples marquants avant de détailler ce propos.

Cette première illustration permet de visualiser le volume de plastique que représente la consommation d'eau en bouteille : 480 milliards de bouteilles vendues en 10 ans ! Ce chiffre astronomique est inimaginable. En revanche, une montagne de plastique¹ de 2400 mètres surplombant Manhattan marque davantage les esprits.

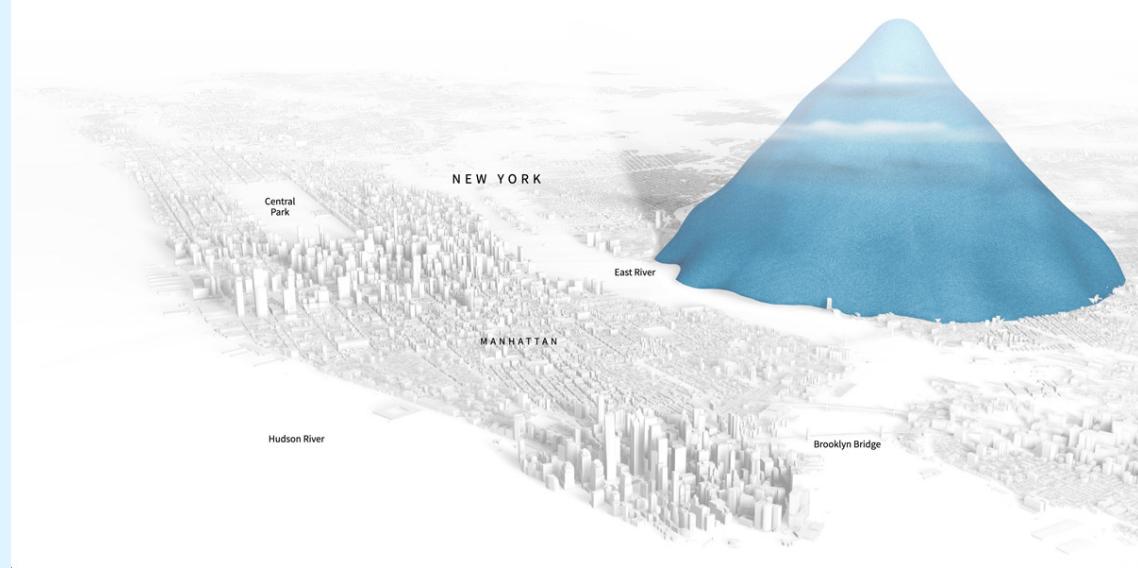


FIG. 3.1 : Noyer dans le plastique selon Reuters Graphics

Ce second graphique² représente quatre informations pour 234 villes à travers le monde :

- la croissance démographique (axe des abscisses)
- la vulnérabilité au changement climatique (axes de ordonnées)
- la taille des villes (taille des cercles)
- le continent sur lequel est localisé chaque ville (couleur des cercles).

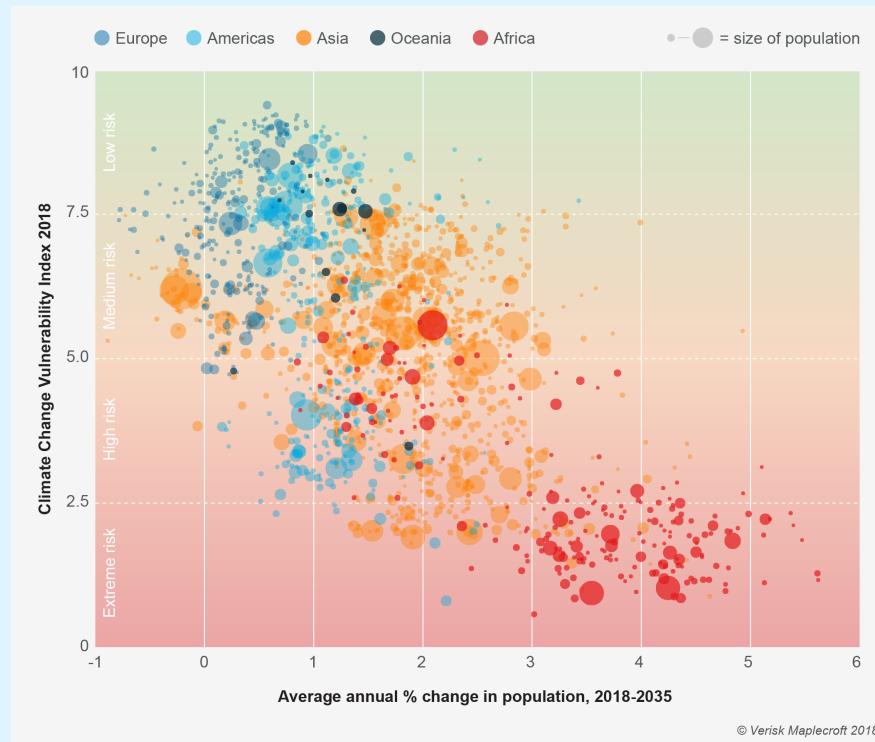


FIG. 3.2 : Changement climatique et vulnérabilité par Verisk Maplecroft

Le graphique est à la fois très accrocheur et esthétique. En un coup d'œil, on constate que les villes avec une forte croissance démographique sont aussi les plus vénérables (lecture des deux axes) et qu'elles sont surtout localisées en Afrique et secondairement en Asie (en rouge et orange), quelle que soit leur taille (taille du cercle). À l'inverse, les villes européennes et américaines (en bleu) sont beaucoup moins vulnérables aux changements climatiques et avec une croissance démographique plus faible, qu'elles soient des petites, moyennes ou grandes villes.

Souvent négligée, la visualisation de données est perçue comme un tâche triviale : il s'agit simplement de représenter une donnée sous forme d'un graphique car c'est l'option la plus pratique ou prenant le moins de place. Pourtant, les avantages de la visualisation des données sont limites. Par exemple, la visualisation de données intègre aujourd'hui des supports dynamiques comme des animations, des figures interactives ou des applications webs. R offre d'ailleurs des possibilités très intéressantes en la matière avec des *packages* comme `shiny`, `plotly`, ou `leaflet`. Toutefois, nous ne couvrirons pas ici ces méthodes plus récentes en visualisation des données qui devraient faire l'objet d'un autre livre en tant quel tel.

Les principaux avantages de la visualisation des données :

- **Analyse exploratoire des données** (*exploratory data analysis - EDA* en anglais). Visualiser des données est crucial pour détecter des problèmes en tout genre (données manquantes, valeurs extrêmes ou aberrantes, non respect de conditions d'application de tests statistiques, etc.), mais aussi pour repérer de nouvelles associations entre les variables.
- **Communication de vos résultats**. La raison d'être d'un graphique est de délivrer un message clair relatif à un résultat obtenu suite à une analyse rigoureuse de vos données. Si votre graphique n'apporte aucune information claire, il vaut mieux ne pas le présenter, le diffuser. Les représentations ne sont pas neutres. Les couleurs et les formes ont des significations particulières en fonction de la culture et du contexte. Posez-vous donc toujours la question : à quel public est destiné le message ? Évitez de surcharger vos visualisation de données, sinon l'essence du message sera perdu.
- **Aide à la décision**. Une illustration (graphique ou carte) peut être un outil facilitant la prise de décisions.

3.1 Philosophie du ggplot2

`ggplot2` fait partie du `tidyverse` et dispose donc d'une logique de fonctionnement particulière. Cette dernière se nomme *The Grammar of Graphics* (les deux G sont d'ailleurs à l'origine du nom `ggplot2`) et a été proposée par Hadley Wickham (le créateur du `tidyverse` !) dans un article intitulé *A layered grammar of graphics* (Wickham, 2010). Nous proposons de synthétiser ici les concepts et principes centraux qui sous-tendent la production de graphiques avec `ggplot2`.

3.1.1 Une grammaire

Hadley Wickham propose une grammaire pour unifier la création de graphiques. L'idée est donc de dépasser les simples dénominations comme un nuage de points, un diagramme en boîte, un graphique en ligne, etc, pour comprendre ce qui relie tous ces graphiques. Ces éléments communs et centraux sont les géométries, les échelles et systèmes de coordonnées, et les annotations (figure 3.3) :

- Les **géométries** sont les formes utilisées pour représenter les données. Il peut s'agir de points, lignes, cercles, rectangles, arcs de cercles, etc.
- Les **échelles et systèmes de coordonnées** permettent de contrôler la localisation des éléments dans un graphique en convertissant les données depuis leur échelle originale (dollars, kilomètres, pourcentages, etc.) vers l'échelle du graphique (pixels).
- Les **annotations** recoupent l'ensemble des informations complémentaires ajoutées au graphique comme son titre et sous-titre, la source des données, la mention sur les droits d'auteurs, etc.

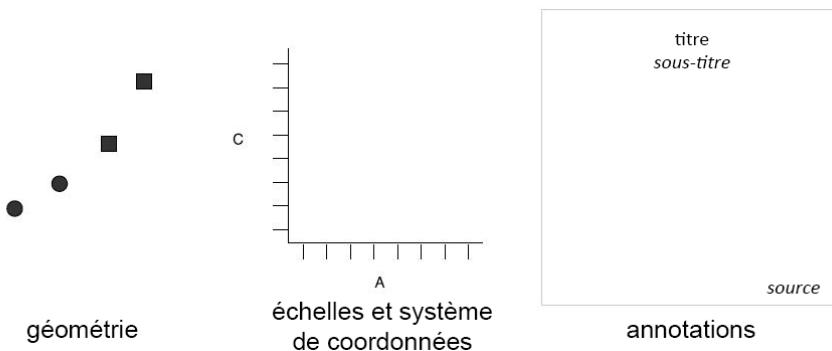


FIG. 3.3 : Trois composantes d'un graphique, adapté de @wickham2010layered

En plus de ces trois éléments, il est bien sûr nécessaire de disposer de **données**. Ces dernières sont assignées à des dimensions du graphique pour être représentées (notamment les axes X et Y et la couleur). Cette étape est appelée **aesthetics mapping** dans **ggplot2**.

Lorsque l'on combine des données, leur assignation a des dimensions, un type de géométries, des échelles et un système de coordonnées, on obtient un **calque** (ou *layer*). Un graphique peut comprendre plusieurs calques comme nous le verrons dans les prochaines sections.

Prenons un premier exemple très simple et construisons un nuage de points à partir du jeu de données *iris* fourni de base dans R. Nous allons représenter la relation qui existe entre la longueur et la largeur des sépals de ces fleurs. Pour commencer, nous devons charger le package **ggplot2** et instancier un graphique avec la fonction **ggplot**.

```
library(ggplot2)
data(iris)
names(iris)

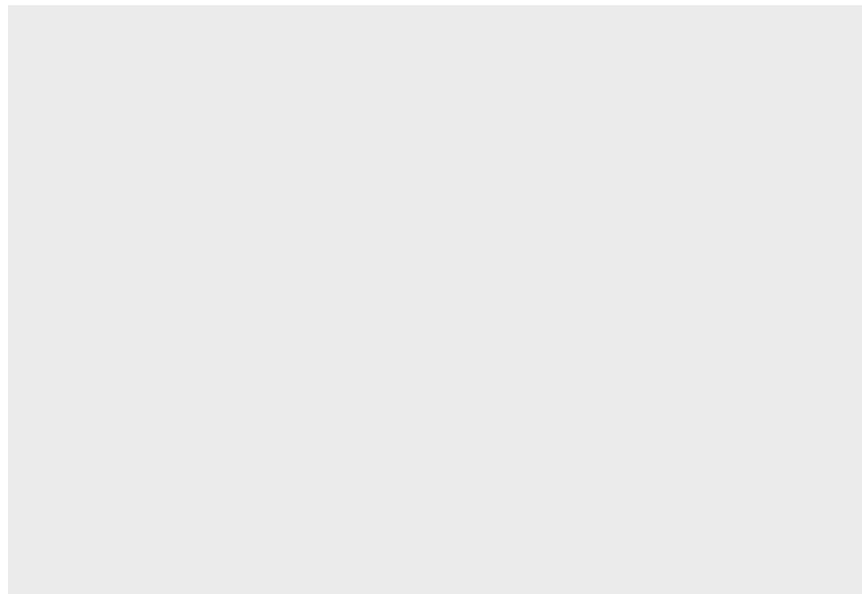
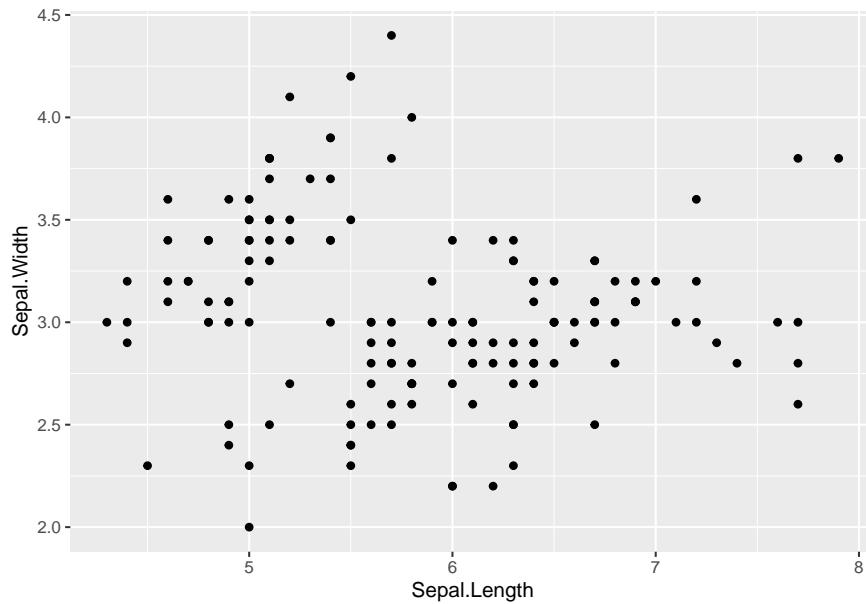
## [1] "Sepal.Length" "Sepal.Width"   "Petal.Length"  "Petal.Width"   "Species"

ggplot()
```

Pour le moment, le graphique est vide. La seconde étape consiste à lui ajouter des données (au travers du paramètre **data**) et à définir les dimensions à associer aux données (avec le paramètre **mapping** et la fonction **aes()**). Dans notre cas, nous voulons utiliser les coordonnées X pour représenter la largeur des sépals, et les coordonnées Y pour représenter la longueur des sépals. Enfin, nous souhaitons représenter les observations par des points, nous utiliserons donc la géométrie **geom_point**.

```
ggplot(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  geom_point()
```

Ce graphique ne comprend qu'un seul calque avec une géométrie de type point. Chaque calque est ajouté avec l'opérateur **+** qui permet de superposer des calques, le dernier apparaissant au dessus des autres. Les arguments **mapping** et **data** sont définis ici dans la fonction **ggplot** et sont donc appliqués à tous les calques qui composeront le graphique. Il est aussi possible de définir **mapping** et **data** au sein des fonctions des géométries :

**FIG. 3.4 :** Base d'un graphique**FIG. 3.5 :** Ajout des dimensions au graphique

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris)
```

La troisième étape consiste à ajouter au graphique ses annotations. Pour notre cas, il faudrait ajouter un titre, un sous-titre, et des intitulés plus clairs pour les axes X et Y, ce qu'il est possible de faire avec la fonction `labs`.

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
```

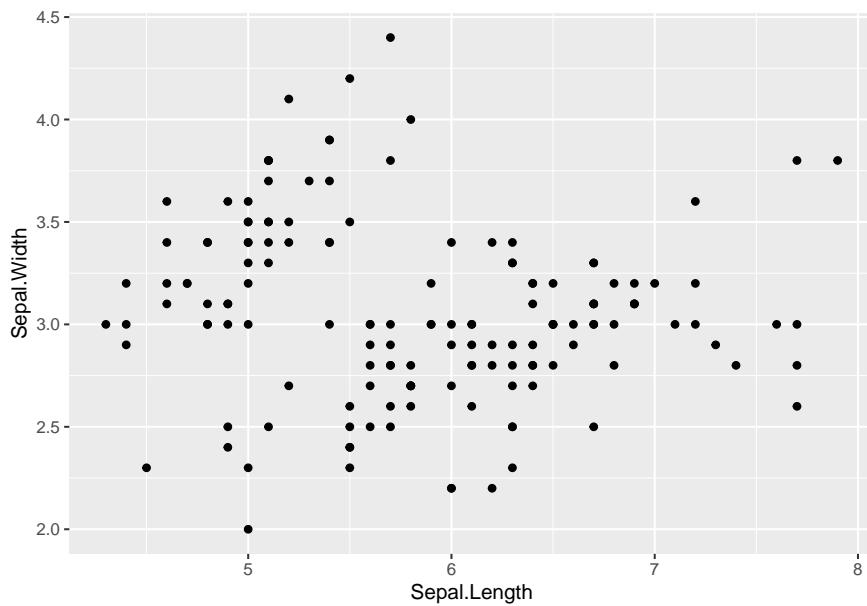


FIG. 3.6 : Autre spécification des arguments mapping et data

```
x = "Longueur des sépals",
y = "Largeur des sépals")
```

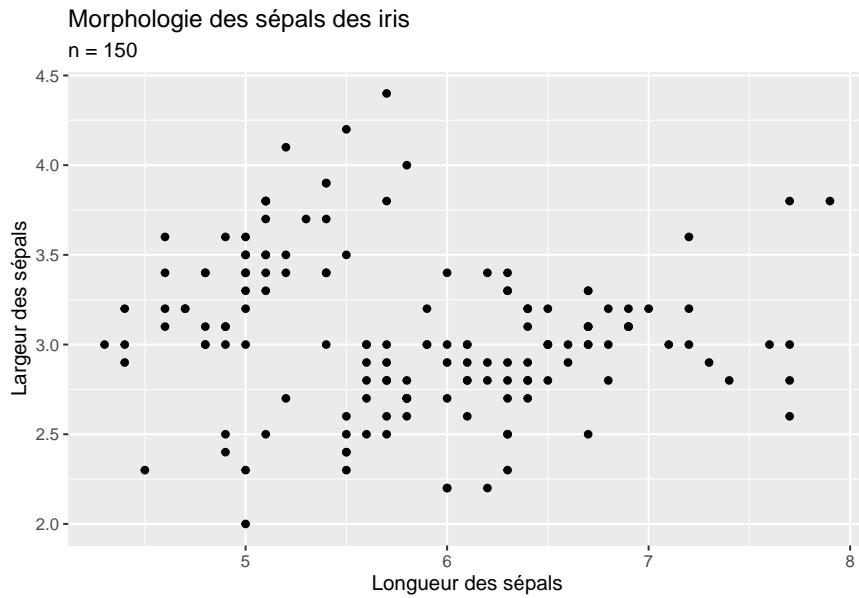


FIG. 3.7 : Ajouter les annotations

3.1.2 Les types de géométries

ggplot2 permet d'utiliser un très grand nombre de géométries différentes. Dans le tableau 3.1, nous avons reporté les principales géométries disponibles afin que vous puissiez vous faire une idée du bestiaire existant. Il ne s'agit que d'un extrait des principales fonctions. Sachez qu'il existe aussi des *packages* proposant des géométries supplémentaires pour compléter **ggplot2**.

TAB. 3.1 : Principales géométries proposée par **ggplot2**

Géométrie	Fonction
point	geom_point
ligne	geom_line
chemin	geom_path
boîte à moustache	geom_boxplot
diagramme violon	geom_violin
histogramme	geom_histogram
barre	geom_bar
densité	geom_density
texte	geom_label
barre d'erreur	geom_errorbar
surface	geom_ribbon

3.1.3 L'habillage

Nous avons montré dans le premier exemple comment rajouter le titre, le sous-titre et le nom des axes sur un graphique. Il est également possible de rajouter un texte sous le graphique (généralement la source des données avec l'argument `caption`) et des annotations textuelles (`annotate`). Pour ces dernières, il convient de spécifier leur localisation (cordonnées `x` et `y`) et le texte à intégrer (`label`); elles sont ensuite ajoutées au graphique avec l'opérateur `+`. Ajoutons deux annotations pour identifier deux fleurs spécifiques.

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  annotate("text", x = 6.7, y = 2.5, # position de la note
           label = "une virginica", # texte de la note
           hjust = "left", vjust = "top", # ajustement
           size = 3, fontface = "italic") +
  annotate("text", x = 5.7, y = 4.4, # position de la note
           label = "une setosa", # texte de la note
           hjust = "left", vjust = "top", # ajustement
           size = 3, fontface = "italic")
```

Comme vous pouvez le constater, de nombreux paramètres permettent de contrôler le style des annotations. Pour avoir la liste des arguments disponibles, n'hésitez pas à afficher l'aide de la fonction : `help(annotate)`.

En plus des annotations de type texte, il est possible d'ajouter des annotations de type géométrique. Nous pourrions ainsi délimiter une boîte encadrant les fleurs de l'espèce setosa.

```
setosas <- subset(iris, iris$Species == "setosa")
sepal.length_extent <- c(min(setosas$Sepal.Length),max(setosas$Sepal.Length))
sepal.width_extent <- c(min(setosas$Sepal.Width),max(setosas$Sepal.Width))

ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
```

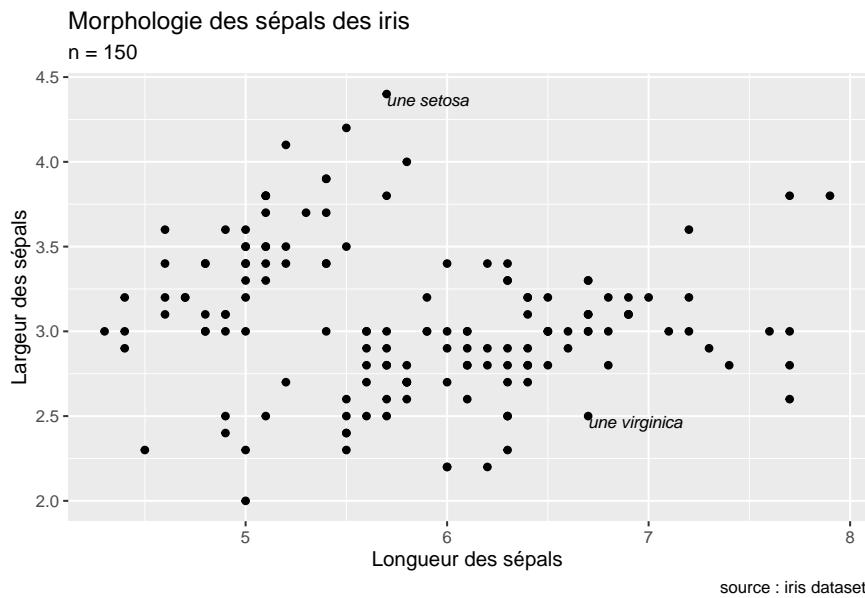


FIG. 3.8 : Ajouter des annotations textuelles

```
y = "Largeur des sépals",
caption = "source : iris dataset" +
annotate("text", x = 6.7, y = 2.5, # position de la note
        label = "une virginica", # texte de la note
        hjust = "left", vjust = "top", # ajustement
        size = 3, fontface = "italic") +
annotate("text", x = 5.7, y = 4.4, # position de la note
        label = "une setosa", # texte de la note
        hjust = "left", vjust = "top", # ajustement
        size = 3, fontface = "italic") +
annotate("rect",
        ymin = sepal.width_extent[[1]],
        ymax = sepal.width_extent[[2]],
        xmin = sepal.length_extent[[1]],
        xmax = sepal.length_extent[[2]],
        fill = rgb(0,0,0), # remplissage transparent
        color = "green") # contour de couleur verte
```

Vous noterez que comme le dernier calque ajouté au graphique est le rectangle, il recouvre tous les calques existant, y compris les précédentes annotations. Pour corriger cela, il suffit de changer l'ordre des calques.

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  annotate("rect",
        ymin = sepal.width_extent[[1]],
        ymax = sepal.width_extent[[2]],
        xmin = sepal.length_extent[[1]],
        xmax = sepal.length_extent[[2]],
```

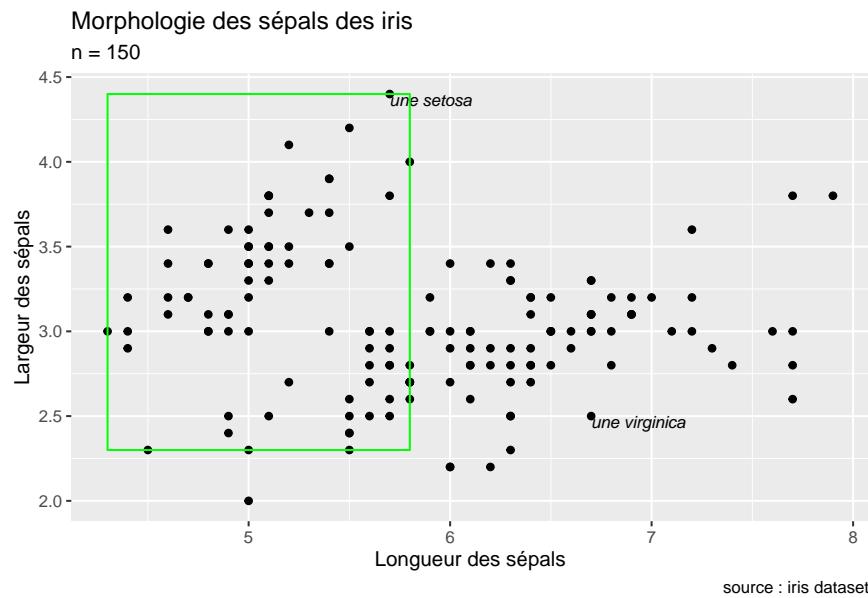


FIG. 3.9 : Ajouter des annotations géométriques

```
fill = rgb(0,0,0,0), # remplissage transparent
color = "green") + # contour de couleur verte
annotate("text", x = 6.7, y = 2.5, # position de la note
           label = "une virginica", # texte de la note
           hjust = "left", vjust = "top", # ajustement
           size = 3, fontface = "italic") +
annotate("text", x = 5.7, y = 4.4, # position de la note
           label = "une setosa", # texte de la note
           hjust = "left", vjust = "top", # ajustement
           size = 3, fontface = "italic")
```

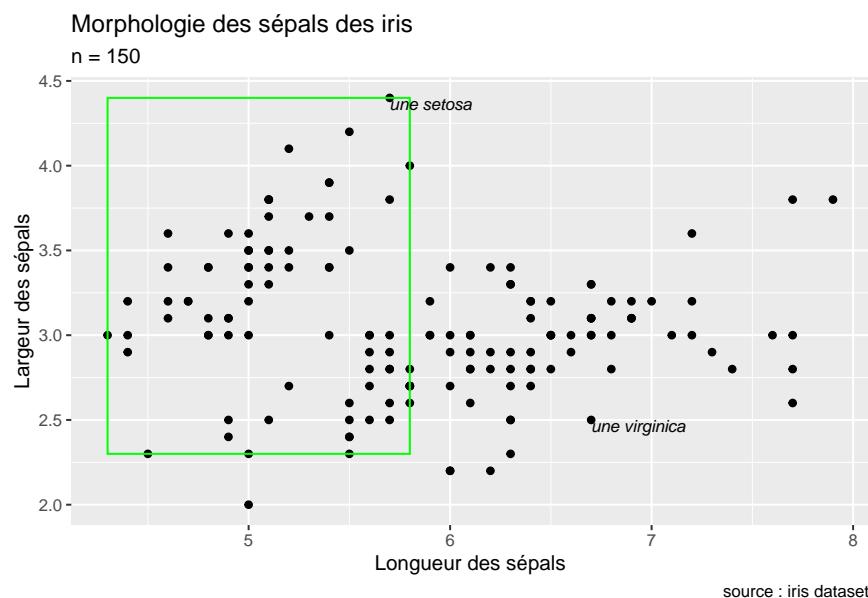


FIG. 3.10 : Gérer l'ordre des annotations

3.1.4 Utiliser des thèmes

De nombreux autres éléments peuvent être modifiés dans un graphique comme les paramètres des polices, l'arrière-plan, la grille de repères, etc. Il peut être fastidieux de paramétrier tous ces éléments. Une alternative intéressante est d'utiliser des thèmes déjà préconstruits. `ggplot2` propose une dizaine de thèmes, constatons leur impact sur le graphique précédent.

- Le thème classique

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  theme_classic()
```

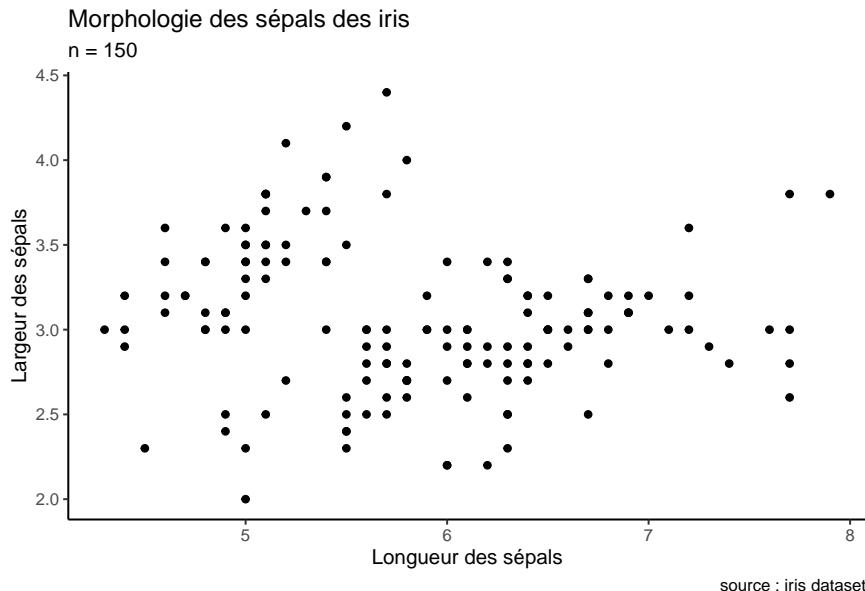


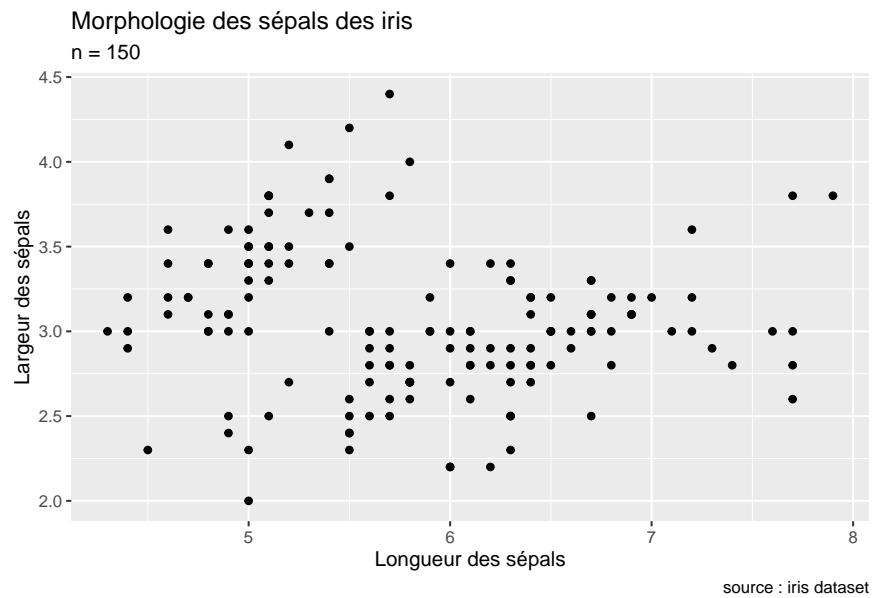
FIG. 3.11 : Thème classique

- Le thème gris

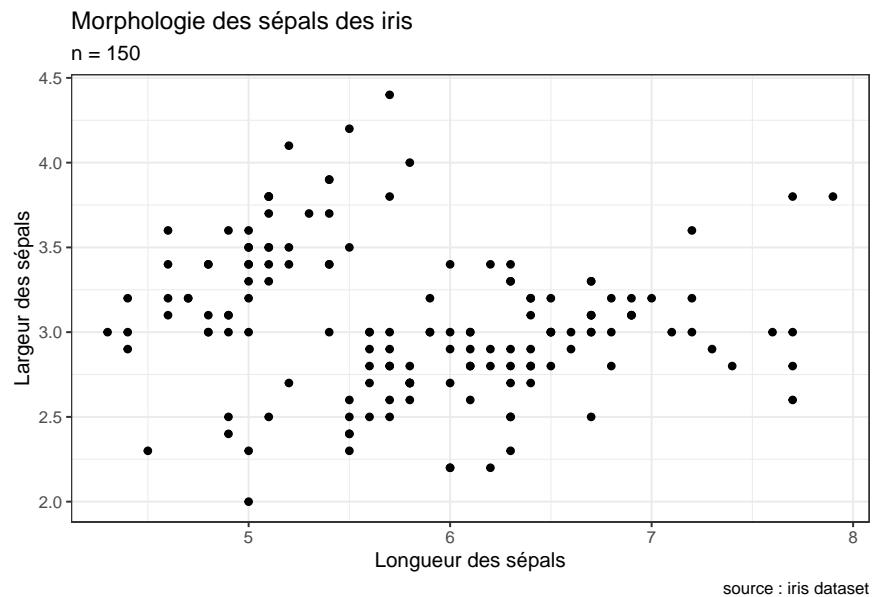
```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  theme_gray()
```

- Le thème noir et blanc

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
```

**FIG. 3.12 :** Thème gris

```
x = "Longueur des sépals",
y = "Largeur des sépals",
caption = "source : iris dataset") +
theme_bw()
```

**FIG. 3.13 :** Thème noir et blanc

- Le thème minimal

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
```

```
x = "Longueur des sépals",
y = "Largeur des sépals",
caption = "source : iris dataset") +
theme_minimal()
```

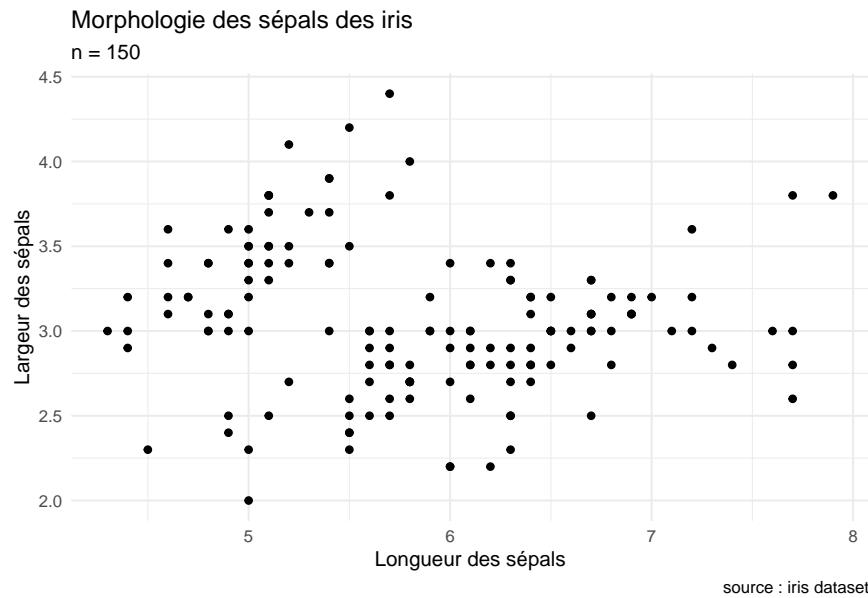


FIG. 3.14 : Thème minimal

Il est aussi possible d'utiliser le *package ggthemes* qui apporte des thèmes complémentaires intéressants dont :

- Le thème tufte (à l'ancienne...)

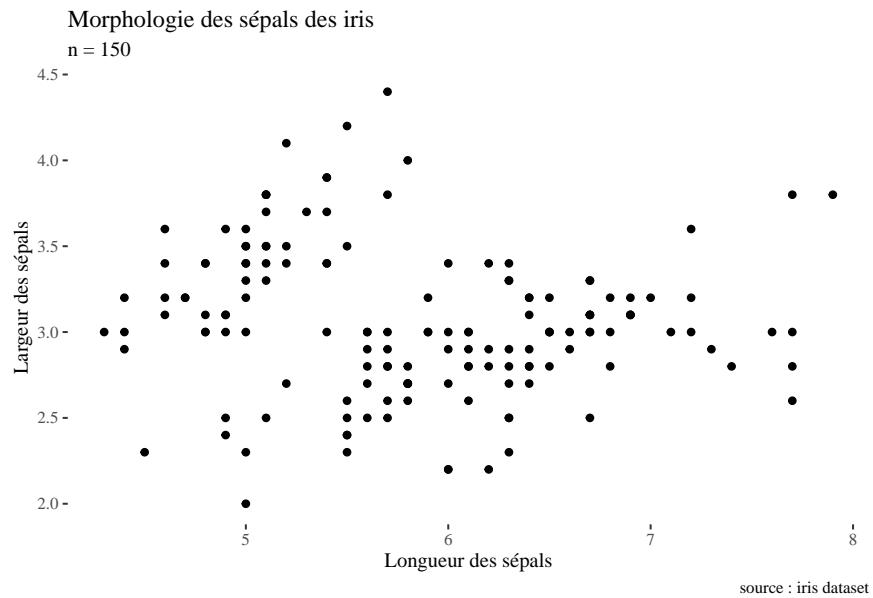
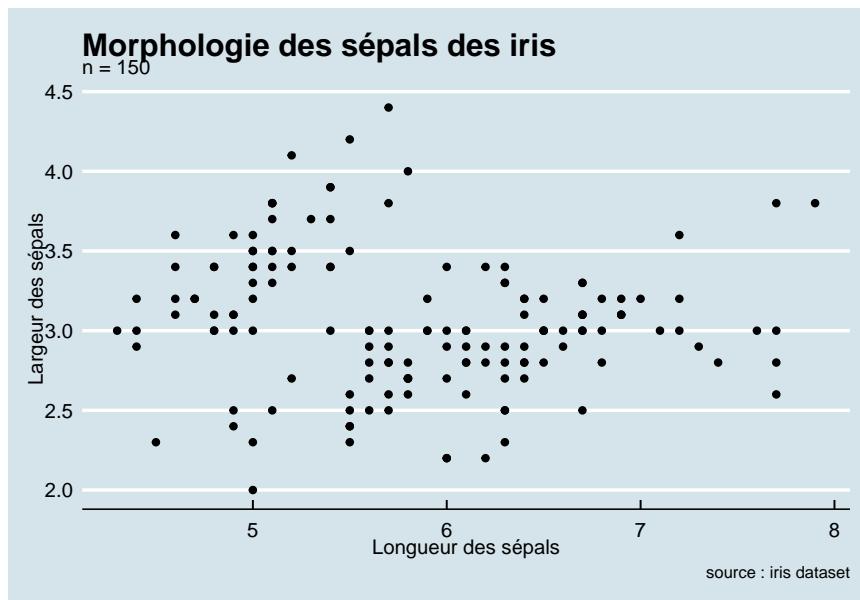
```
library(ggthemes)

ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  theme_tufte()
```

- Le thème economist (inspiré de la revue du même nom)

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  theme_economist()
```

- Le thème solarized (plus original)

**FIG. 3.15 :** Thème tufte**FIG. 3.16 :** Thème economist

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  theme_solarized()
```

Il en existe bien d'autres et vous pouvez composer vos propres thèmes. N'hésitez pas à explorer la documentation de **ggplot2** et **ggthemes** pour en apprendre plus!

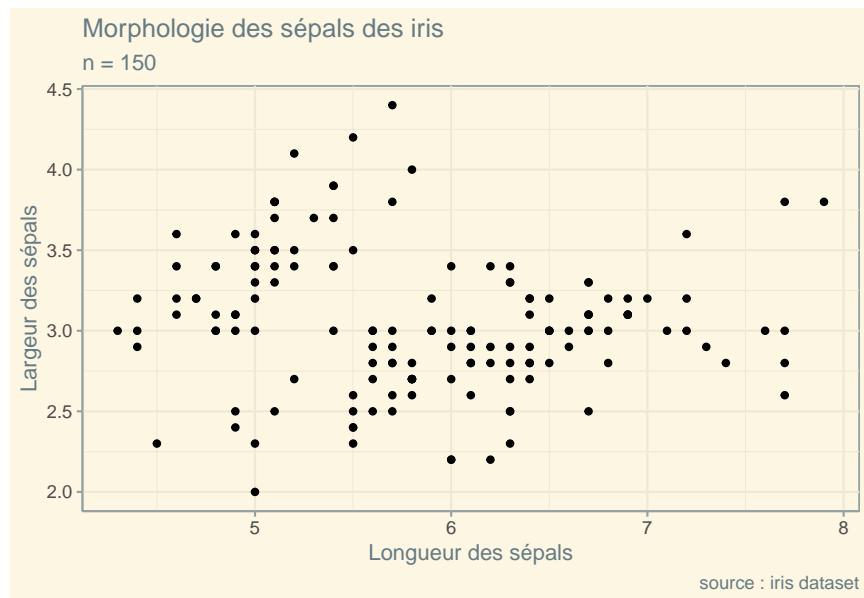


FIG. 3.17 : Thème solarized

3.1.5 Composer une figure avec plusieurs graphiques

Il est très fréquent de vouloir combiner plusieurs graphiques dans une même figure. Deux cas se distinguent :

1. Les données pour les différents graphiques proviennent du même *DataFrame* et peuvent être distinguées selon une variables catégorielle. L'objectif est alors de dupliquer le même graphique, mais pour des sous-groupes de données. Dans ce cas, nous recommandons d'utiliser la fonction `facet_wrap` de **ggplot2**.
2. Les graphiques sont complètement indépendants. Dans ce cas, nous recommandons d'utiliser la fonction `ggarrange` du package **ggpubr**.

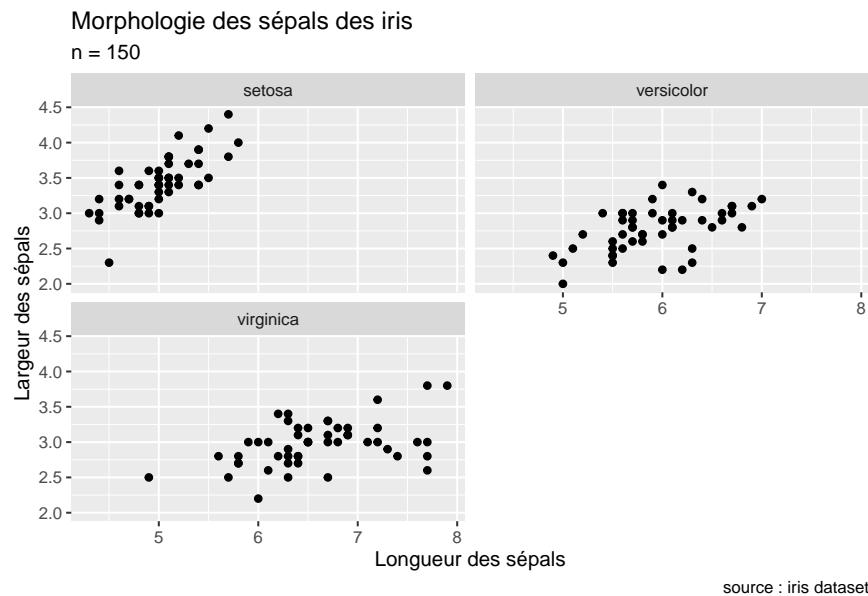
3.1.5.1 ggplot2 et ses facettes

Nous pourrions souhaiter réaliser une figure composite avec le jeu de données iris et séparer notre nuage de points en trois graphiques distincts selon les espèces des iris. Pour cela, il faut au préalable convertir la variable espèce en facteur.

```
iris$Species_fac <- as.factor(iris$Species)

ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  facet_wrap(vars(Species_fac), ncol=2)
```

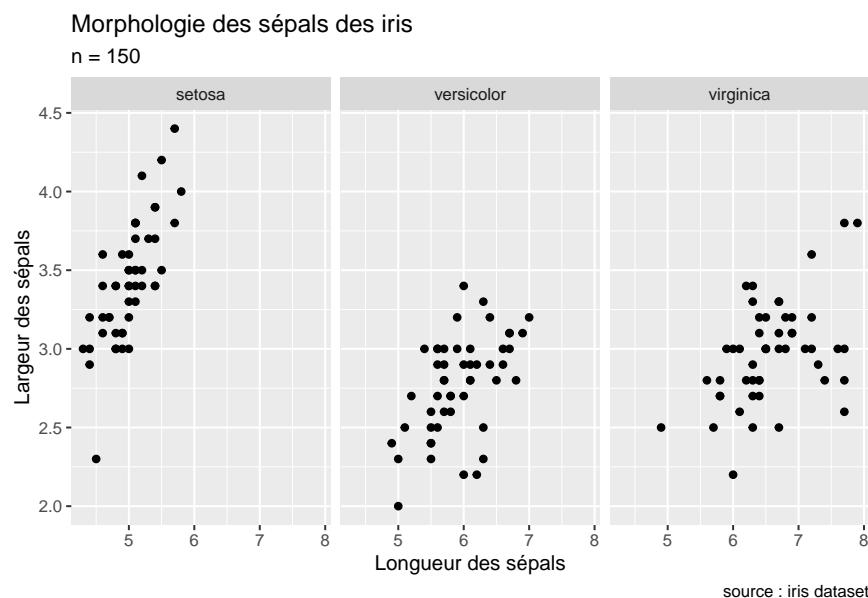
Notez que le nom de la variable (ici `Species_fac`) doit être spécifié au sein d'une sous-fonction `vars : vars(Species_fac)`. Nous aurions aussi pu réaliser le graphique sur une seule ligne en spécifiant `ncol = 3`.



source : iris dataset

FIG. 3.18 : Graphique à facettes

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépals des iris", subtitle = "n = 150",
       x = "Longueur des sépals",
       y = "Largeur des sépals",
       caption = "source : iris dataset") +
  facet_wrap(vars(Species_fac), ncol=3)
```



source : iris dataset

FIG. 3.19 : Graphique à facette en une ligne

3.1.5.2 Arranger des graphiques

La solution avec les facettes est très pratique, mais également très limitée puisqu'elle ne permet pas de créer une figure avec des graphiques combinant plusieurs types de géométries. `ggarrange` du package `ggpubr` permet tout simplement de combiner des graphiques déjà existant. Créons trois nuages de points comparant plusieurs variables en fonction de l'espèce des iris, puis combinons les. Nous allons également attribuer aux points une couleur en fonction de l'espèce des fleurs afin de mieux les distinguer en associant la variable `Species` au paramètre `color`.

```
library(ggpubr)

plot1 <- ggplot(data = iris) +
  geom_point(aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  labs(subtitle = "Caractéristiques des sépals",
       x = "Longueur",
       y = "Largeur")

plot2 <- ggplot(data = iris) +
  geom_point(aes(x = Petal.Length, y = Petal.Width, color = Species)) +
  labs(subtitle = "Caractéristiques des pétales",
       x = "Longueur",
       y = "Largeur")

liste_plots <- list(plot1, plot2)
comp_plot <- ggarrange(plotlist = liste_plots, ncol = 2, nrow = 1,
                       common.legend = TRUE, legend = "bottom") #gérer la légende

annotate_figure(comp_plot,
               top = text_grob("Morphologie des sépals et pétales chez des iris",
                               face = "bold", size = 12, just = "center"),
               bottom = text_grob("source : iris dataset",
                                  face = "italic", size = 8, just = "left"))

)
```

Vous constaterez que quatre étapes sont nécessaires :

1. Créer les graphiques et les enregistrer dans des objets (ici `plot1` et `plot2`).
2. Encapsuler ces objets dans une liste (ici `liste_plots`).
3. Composer la figure finale avec la fonction `ggarrange`.
4. Ajouter les annotations à la figure composite.

L'argument `common.legend` permet d'indiquer à la fonction `ggarrange` de regrouper les légendes des deux graphiques. Dans notre cas, les deux graphiques ont les mêmes légendes, il est donc judicieux de les regrouper. L'argument `legend` contrôle la position de la légende, et peut prendre les valeurs : `top`, `bottom`, `left`, `right` ou `none` (absence de légende). La fonction `annotate_figure` permet d'ajouter des éléments de texte au-dessus, au-dessous et sur les cotés de la figure composite.

3.1.6 La couleur

Dans un graphique, la couleur peut être utilisée à la fois pour représenter une variable quantitative (dégradé de couleur ou discréétisation), ou une variable qualitative (couleur par catégorie). Dans `ggplot2`, il est possible d'attribuer une couleur au contour des géométries avec l'argument `color` et au remplissage avec l'argument `fill`. Il est possible de spécifier une couleur de trois façons dans R :

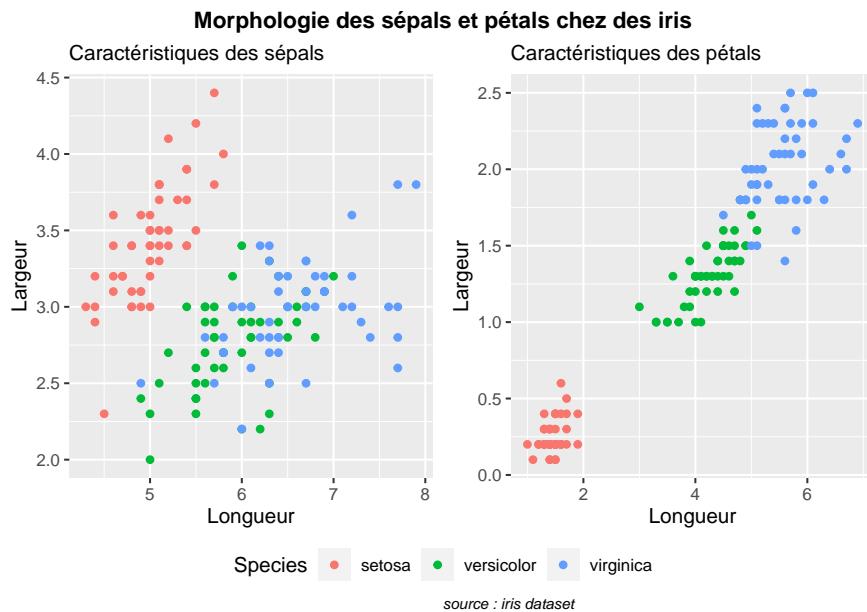


FIG. 3.20 : Figure avec plusieurs graphiques avec ggarrange

- En utilisant le nom de la couleur dans une chaîne de caractère : "chartreuse4". R dispose 657 noms de couleurs prédéfinis. Pour tous les afficher, utilisez la fonction `colors()` qui permet de les visualiser (figure 3.21) .
- En indiquant le code hexadécimal de la couleur. Il s'agit d'une suite de six lettres et de chiffres précédée par un dièse : "#99ff33"
- En utilisant une notation RGB (rouge, vert, bleu). Cette notation doit contenir quatre nombres entre 0 et 1 (0% et 100%), le premier indiquant la quantité de rouge, le second de vert, le troisième de bleu, et le quatrième la transparence. Ces quatres nombres sont donnés comme argument à la fonction `rgb` : `rgb(0.6, 1, 0.2, 0)`.

Le choix des couleurs est un problème plus complexe que la manière de les spécifier. Il existe d'ailleurs tout un pan de la sémiologie graphique dédié à la question du choix et de l'association des couleurs. Une première ressource intéressante est ColorBrewer³. Il s'agit d'une sélection de palettes de couleurs particulièrement efficaces et dont certaines sont même adaptées pour les personnes daltoniennes. Il est possible d'accéder directement aux palettes dans R grâce au package **RColorBrewer** et la fonction `brewer.pal` :

```
library(RColorBrewer)
display.brewer.all()
```

Une autre ressource pertinente est le site web [coolors.co⁴](https://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3) qui propose de nombreuses palettes à portée de clic.

³<https://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3>

⁴<https://coolors.co/palettes/trending>

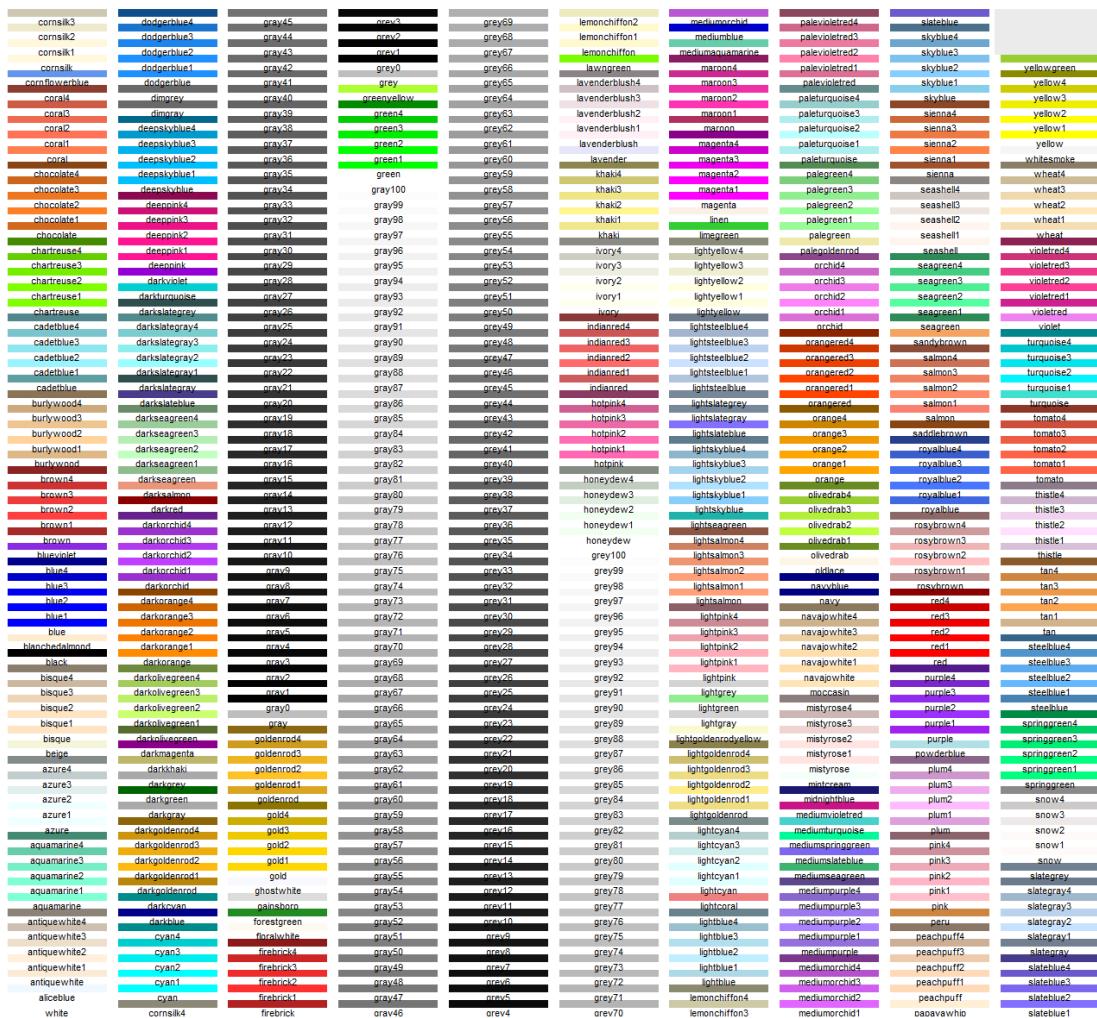
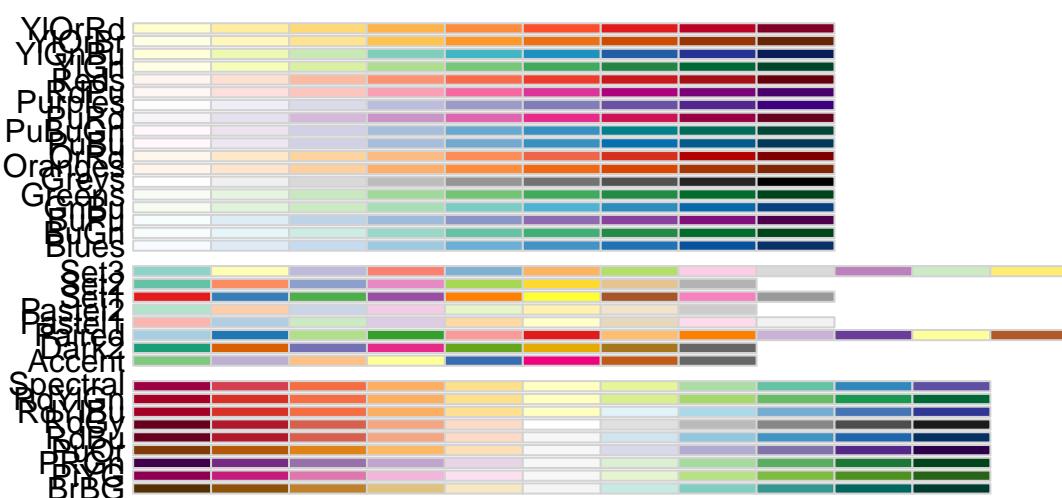


FIG. 3.21 : Couleurs de base



3.2 Principaux graphiques



Puisque vous êtes désormais initié aux bases de la grammaire des graphiques implémentées par **ggplot2**, vous apprendrez dans les sous-sections suivantes à construire les principaux graphiques que vous utiliserez régulièrement et/ou que vous retrouverez présenter dans un article scientifique.

3.2.1 Histogramme

L'histogramme permet de décrire graphiquement la forme de la distribution d'une variable. Pour le réaliser, on utilise la fonction `geom_histogram`. Le paramètre le plus important est le nombre de barres (`bins`) qui composent l'histogramme. Plus ce nombre est grand, plus l'histogramme est précis et à l'inverse, plus il est petit, plus l'histogramme est simplifié. En revanche, il faut éviter d'utiliser un nombre de barres trop élevé comparativement au nombre d'observations disponibles dans le jeu de données, sinon votre histogramme sera plein de trous.

3.2.1.1 histogrammes simples

Générons quatre variables ayant respectivement une distribution Gaussienne, Student, Gamma et Beta, puis réalisons un histogramme pour chacune de ces variables et combinons les avec la fonction `ggarrange`.

```
distrib <- data.frame(
  gaussien = rnorm(1000, mean = 5, sd = 1.5),
  gamma = rgamma(1000, shape = 2, rate = 12),
  beta = rbeta(1000, shape1 = 5, shape2 = 1, ncp = 2),
  student = rt(1000, ncp = 20, df = 5)
)

plot1 <- ggplot(data = distrib) +
  geom_histogram(aes(x = gaussien), bins = 50, color = "#343a40", fill = "#e63946") +
  ylim(c(0,130))

plot2 <- ggplot(data = distrib) +
  geom_histogram(aes(x = gamma), bins = 50, color = "#343a40", fill = "#f1faee") +
  ylim(c(0,130))

plot3 <- ggplot(data = distrib) +
  geom_histogram(aes(x = beta), bins = 50, color = "#343a40", fill = "#a8dadcc") +
  ylim(c(0,130))

plot4 <- ggplot(data = distrib) +
  geom_histogram(aes(x = student), bins = 50, color = "#343a40", fill = "#1d3557") +
  ylim(c(0,130))

histogrammes <- list(plot1, plot2, plot3, plot4)

ggarrange(plotlist = histogrammes, ncol = 2, nrow = 2)

## Warning: Removed 3 rows containing missing values (geom_bar).
```

Notez que cette syntaxe est très lourde. Dans le cas présent, il serait plus judicieux d'utiliser la fonction `facet_wrap`. Pour cela, nous devons au préalable empiler nos données, ce qui signifie changer la forme du `dataframe` actuel qui comprend quatre colonnes (gaussien, gamma, beta et student) et 1000 observations,

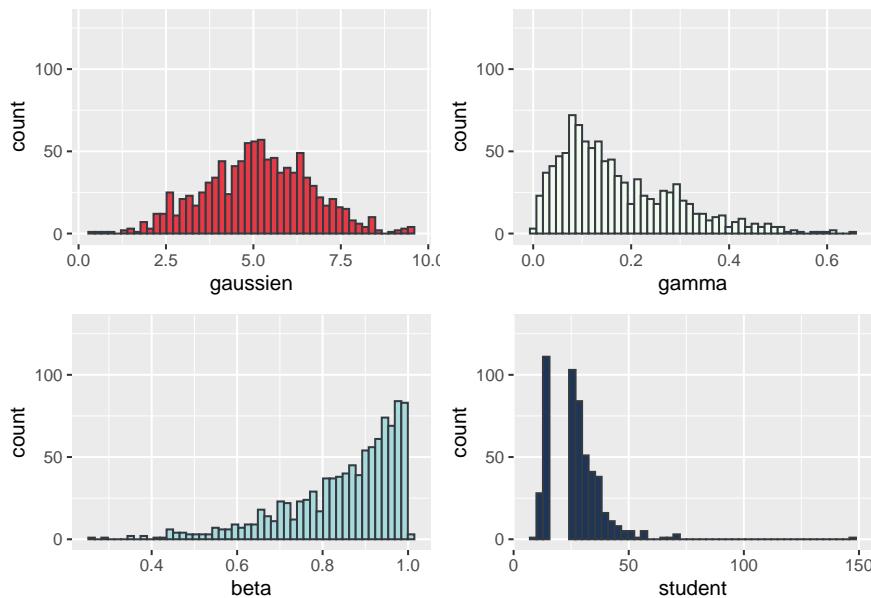


FIG. 3.23 : Histogrammes

pour qu'il n'ait plus que deux colonnes (la valeur originale et le nom de l'ancienne colonne) et 4000 observations. La figure 3.24 décrit graphiquement ce processus qui peut être effectué avec la fonction `melt` du package `reshape2`.

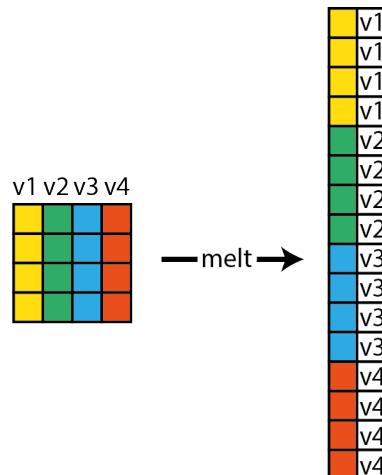


FIG. 3.24 : Empiler les données d'un dataframe

```
library(reshape2)

#faire fondre le jeu de données
melted_distribrs <- melt(distribrs, measure.vars = c("gaussien", "gamma",
                                                       "beta", "student"))

#renommer les colonnes du nouveau dataframe
names(melted_distribrs) <- c("distribution", "valeur")
#convertir la variable catégorielle en facteur
melted_distribrs$distribution <- as.factor(melted_distribrs$distribution)

ggplot(data = melted_distribrs)+
```

```
geom_histogram(aes(x = valeur, fill = distribution), bins = 50, color = "#343a40") +
  ylim(c(0,130)) +
  labs(x = "valeur",
       y = "fréquence")+
  scale_fill_manual(values = c("#e63946","#f1faee","#a8dadc","#1d3557"))+
  facet_wrap(vars(distribution), ncol=2, scales = "free")
```

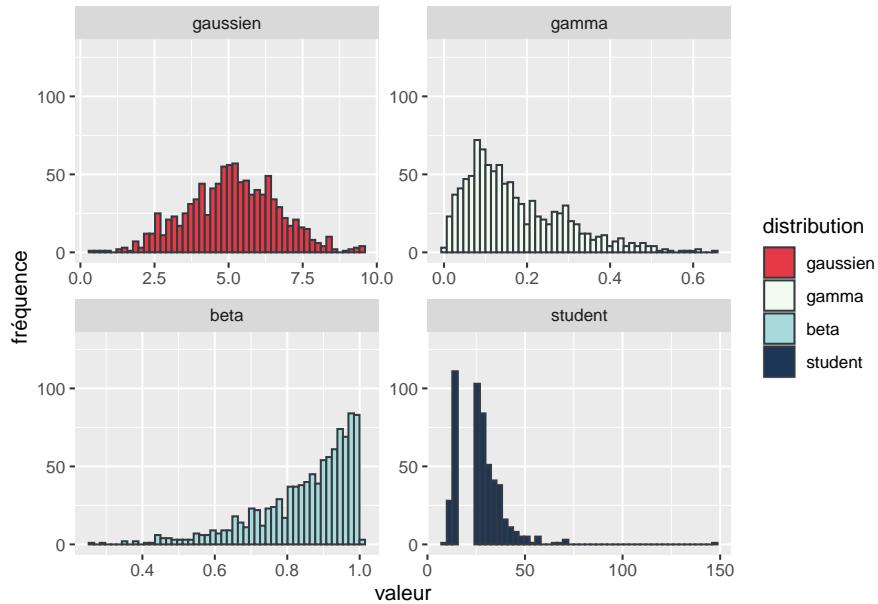


FIG. 3.25 : Histogrammes en facettes

3.2.1.2 histogrammes de densité

Les histogrammes que nous venons de construire utilisent les fréquences des observations pour délimiter la hauteur des barres. Il est possible de changer ce comportement pour plutôt utiliser la densité. L'intérêt est notamment de se rapprocher encore de la définition d'une distribution puisqu'avec cette configuration la somme totale de la surface de l'histogramme est égale à 1. La hauteur de chaque barre représente alors la probabilité d'obtenir l'étendue de valeurs représentées par cette barre. Prenons pour exemple la variable avec la distribution normale que nous venons de voir.

```
plot1 <- ggplot(data = distribs) +
  geom_histogram(aes(x = gaussien, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#1d3557")

plot2 <- ggplot(data = distribs) +
  geom_histogram(aes(x = gaussien, y = ..count..),
                 bins = 30, color = "#343a40", fill = "#a8dadc")

ggarrange(plotlist = list(plot1, plot2), ncol = 2)
```

Le graphique de droite (fréquence) nous indique donc que plus de 100 observations ont une valeur d'environ 5 (entre 4,76 et 5,34, compte tenu de la largeur de la barre), ce qui se traduit par une probabilité de presque 30% d'obtenir cette valeur en tirant une observation au hasard dans le jeu de données.

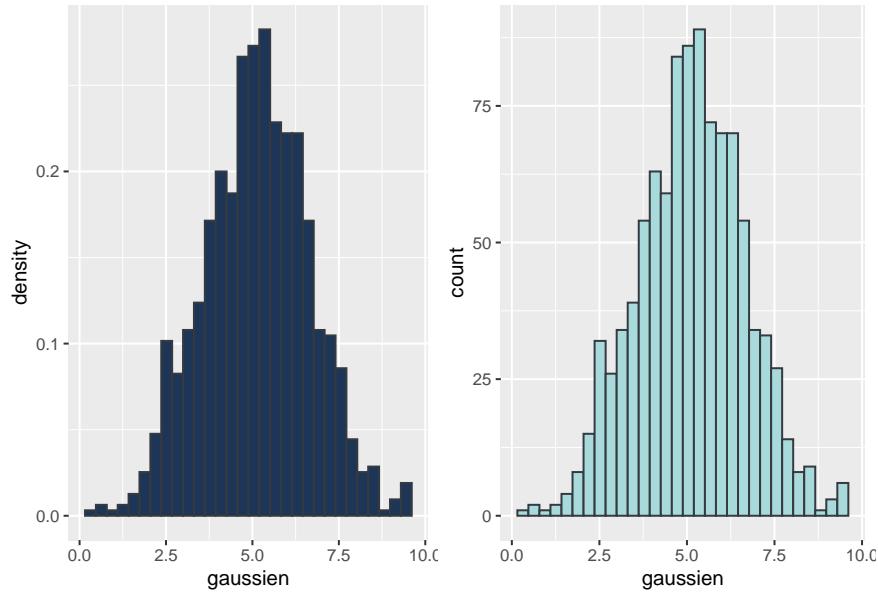


FIG. 3.26 : Histogrammes de densité

3.2.1.3 Histogramme avec courbe de distribution

Les histogrammes sont souvent utilisés pour vérifier graphiquement si une distribution empirique s'approche d'une courbe normale. Pour cela, on rajoute sur l'histogramme de la variable empirique la forme qu'aurait une distribution normale parfaite en utilisant la moyenne et l'écart type de la distribution empirique. Pour créer cette figure dans `ggplot2`, il suffit d'utiliser la fonction `stat_function` pour créer un nouveau calque. Pour faire bonne mesure, il est aussi possible d'ajouter une ligne verticale (`geom_vline`) pour indiquer la moyenne de la distribution.

```
moyenne <- mean(distrib$gaussien)
ecart_type <- sd(distrib$gaussien)

ggplot(data = distrib) +
  geom_histogram(aes(x = gaussien, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#a8adac") +
  labs(x = "gaussien",
       y = "densité") +
  stat_function(fun = dnorm, args = list(mean = moyenne, sd = ecart_type),
                color = "#e63946", size = 1.2, linetype = "dashed") +
  geom_vline(xintercept = moyenne, color = 'red', size = 1) +
  annotate("text", x = round(moyenne,2)+0.5, y = 0.31, hjust = 'left',
          label = paste('moyenne : ',round(moyenne,2),sep=''))
```

Dans notre cas, nous savons que notre variable est normalement distribuée (car produite avec la fonction `rnorm`), et l'on peut constater la grande proximité entre l'histogramme et la courbe normale.

3.2.1.4 Histogramme avec coloration des valeurs extrêmes

Il peut être nécessaire d'attirer le regard sur certaines parties de l'histogramme, comme par exemple des valeurs extrêmes. Si nous reprenons notre distribution de Student, nous pouvons clairement distinguer un ensemble de valeurs fortes à droite de la distribution. On pourrait dans notre cas considérer que des valeurs au delà de 50 constituent des cas extrêmes que nous souhaitons représenter dans une autre

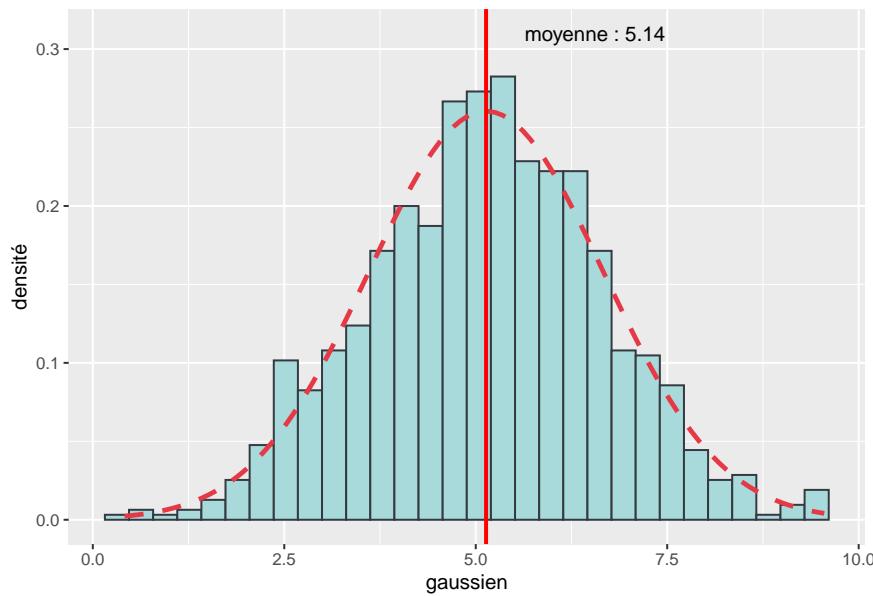


FIG. 3.27 : Histogrammes et courbe normale

couleur. Pour cela, nous devons créer une variable catégorielle nous permettant de distinguer ces cas particuliers.

```
distrib$cas_extreme <- ifelse(distrib$student >=50, "extrême", "normal")

ggplot(data = distrib) +
  geom_histogram(aes(x = student, y = ..count.., fill = cas_extreme),
                 bins = 30, color = "#343a40")+
  scale_fill_manual('cas extrême', values = c("#a8adc","#e63946"))+
  labs(title = 'Distribution de Student',x = "", y = "fréquence")
```

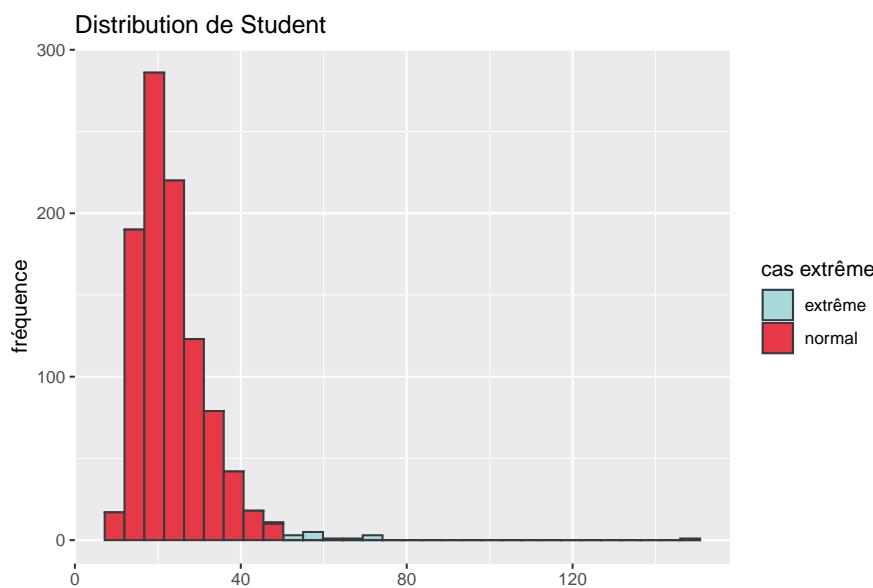


FIG. 3.28 : Histogramme coloré

3.2.2 Graphique de densité

L'histogramme est utilisé pour approximer graphiquement la distribution d'une variable. Sa principale limite est de représenter la variable de façon discontinue. Une alternative intéressante est d'utiliser à la place de l'histogramme une version lissée de celui-ci : le graphique de densité. Cette opération de lissage est réalisée le plus souvent à partir de fonctions *kernel*. Reconstruisons notre figure avec les quatres distributions, mais en utilisant cette fois-ci des graphiques de densité.

```
ggplot(data = melted_distrib)+  
  geom_density(aes(x = valeur, fill = distribution), color = "#343a40") +  
  scale_fill_manual(values = c("#e63946", "#f1faee", "#a8dadc", "#1d3557")) +  
  facet_wrap(vars(distribution), ncol=2, scales = "free")
```

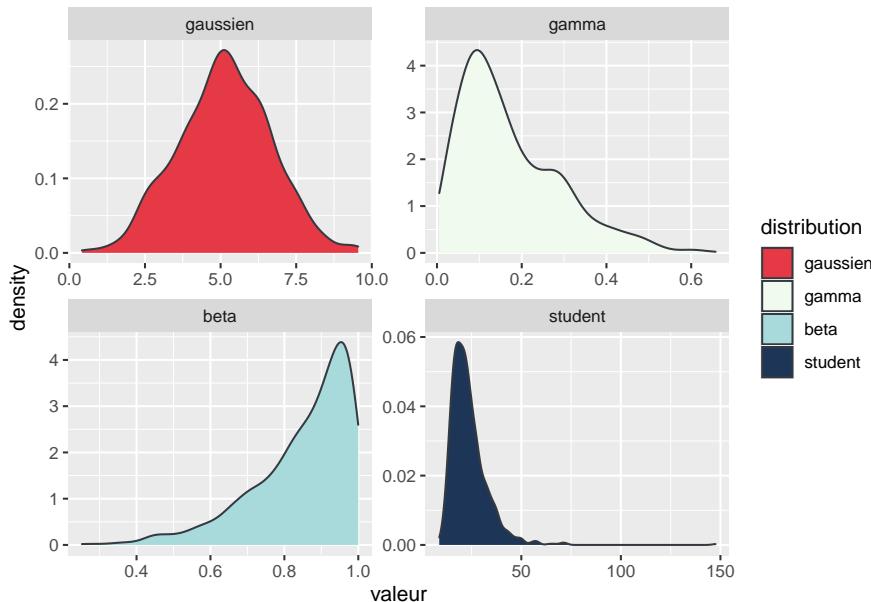


FIG. 3.29 : Graphiques de densité en facette

Les graphiques de densité sont souvent utilisés pour comparer la distribution d'une variable pour plusieurs sous groupe d'une population. Si nous reprenons le jeu de données iris, nous pouvons comparer les longueurs de sépals en fonction des espèces. On constate ainsi que les setosa ont une nette tendance à avoir des sépals plus courts et qu'à l'inverse, les virginica ont les sépals généralement les plus longs.

```
ggplot(data = iris)+  
  geom_density(aes(x = Sepal.Length, fill = Species),  
               color = "#343a40", alpha = 0.4)+  
  labs(x = 'Longueur de sépals',  
       y = '',  
       fill = 'Espèce')
```

3.2.3 Nuage de points

Un nuage de points est un outil très intéressant pour visualiser la relation existante entre deux variables. Prenons un exemple concret et analysons le volume de CO₂ produit annuellement par habitant dans l'ensemble des pays à travers le monde en comparaison avec le niveau d'urbanisation de ces pays. Nous

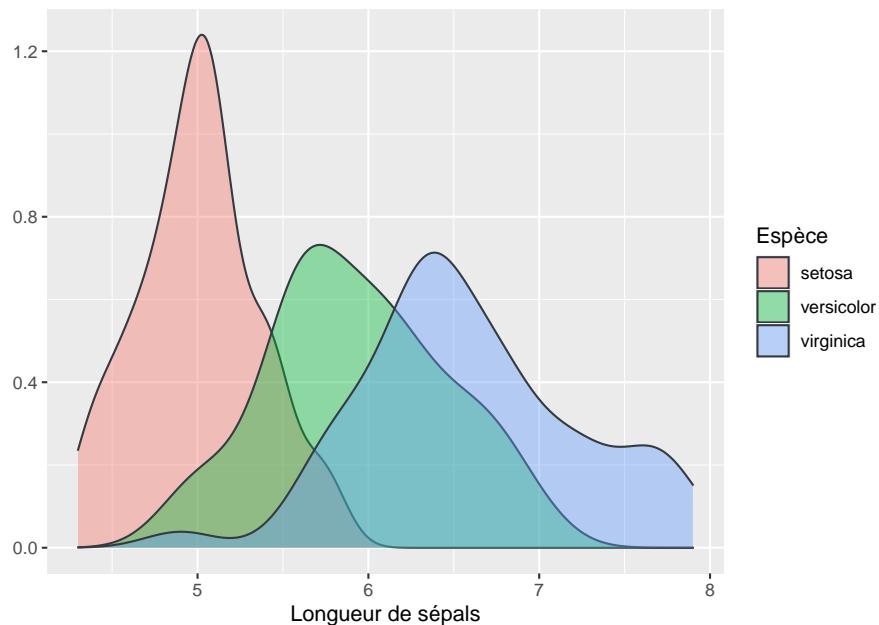


FIG. 3.30 : Graphiques de densité

avons extrait ces données sur le site web de la Banque Mondiale⁵, puis nous les avons structuré dans un fichier *csv*.

```
data_co2 <- read.csv("data/graphique/world_urb_co2.csv")
names(data_co2)

## [1] "country_code" "year"          "Population"    "Urbanisation" "CO2_kt"
## [6] "Country.Name" "CO2t_hab"      "region7"       "region23"
```

3.2.3.1 Nuage de points simple

Commençons par un nuage de points simple avec l'ensemble des données.

```
ggplot(data = data_co2)+
  geom_point(aes(x = Urbanisation, y = CO2t_hab))+
  labs(x = "Niveau d'urbanisation (%)",
        y = 'tonnes de CO2 annuelle / habitant',
        title = 'Relation entre urbanisation et production de CO2 par habitant')
```

À la première lecture de ce graphique, on observe immédiatement un ensemble de points étranges dont le volume de CO2 par habitant annuel est compris entre 150 et plus de 350 tonnes et dont le niveau d'urbanisation est proche de 50%. Isolons ces données pour observer de quoi il s'agit.

```
cas_étrange <- subset(data_co2, data_co2$CO2t_hab >= 150)
print(cas_étrange$Country.Name)

## [1] "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba"
## [10] "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba"
```

⁵<https://donnees.banquemonde.org/indicateur>

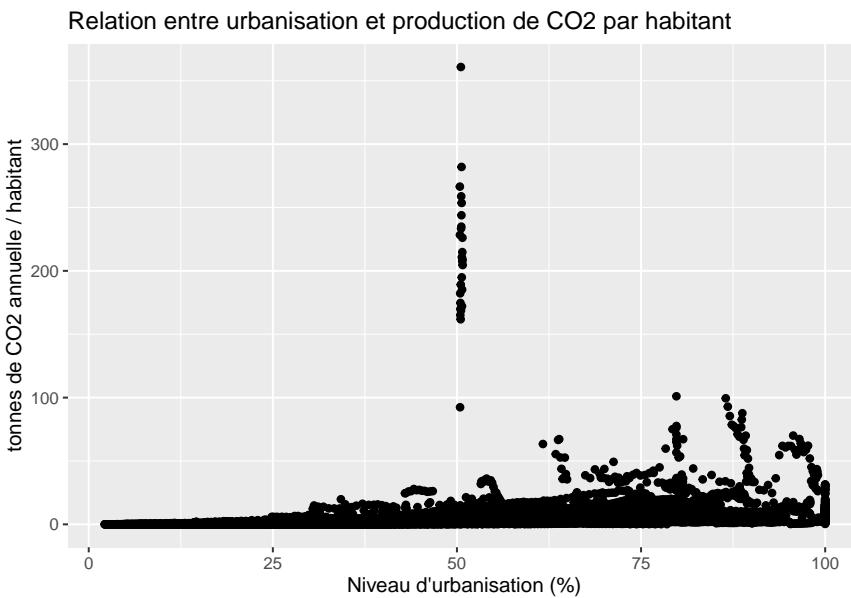


FIG. 3.31 : Nuage de points simple

```
## [19] "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba" "Aruba"
```

Il s'agit d'une petite île néerlandaise des Caraïbes nommée Aruba disposant d'une faible population mais avec des activités très polluantes (raffinerie et extraction d'or). Nous faisons ici le choix de retirer ces observations puisqu'elles sont assez peu représentatives de la tendance mondiale. Cette démarche si simple relève ainsi de l'analyse exploratoire des données! Sans ce graphique, nous n'aurions probablement jamais identifié ces cas problématiques.

```
data_co2 <- subset(data_co2, data_co2$CO2t_hab <= 150)
```

Reconstruisons le nuage de points maintenant que ces données aberrantes ont été retirées.

```
ggplot(data = data_co2)+  
  geom_point(aes(x = Urbanisation, y = CO2t_hab))+  
  labs(x = "Niveau d'urbanisation (%)",  
       y = 'tonnes de CO2 annuelle / habitant',  
       title = 'Relation entre urbanisation et production de CO2 par habitant')
```

Voilà qui est mieux! Cependant, le grand nombre de points restant rend la lecture du graphique assez difficile puisqu'ils se superposent. Une première option à envisager dans ce cas est à la fois d'ajouter de la transparence aux points et de réduire leur taille :

```
ggplot(data = data_co2)+  
  geom_point(aes(x = Urbanisation, y = CO2t_hab), alpha = 0.2, size = 0.5)+  
  labs(x = "Niveau d'urbanisation (%)",  
       y = 'tonnes de CO2 annuelle / habitant',  
       title = 'Relation entre urbanisation et production de CO2 par habitant')
```

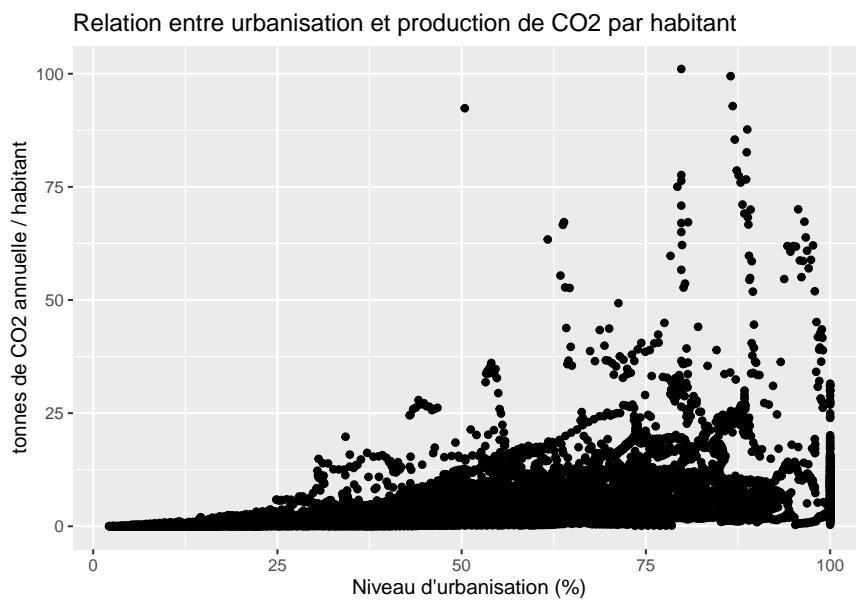


FIG. 3.32 : Nuage de points simple sans données aberrantes

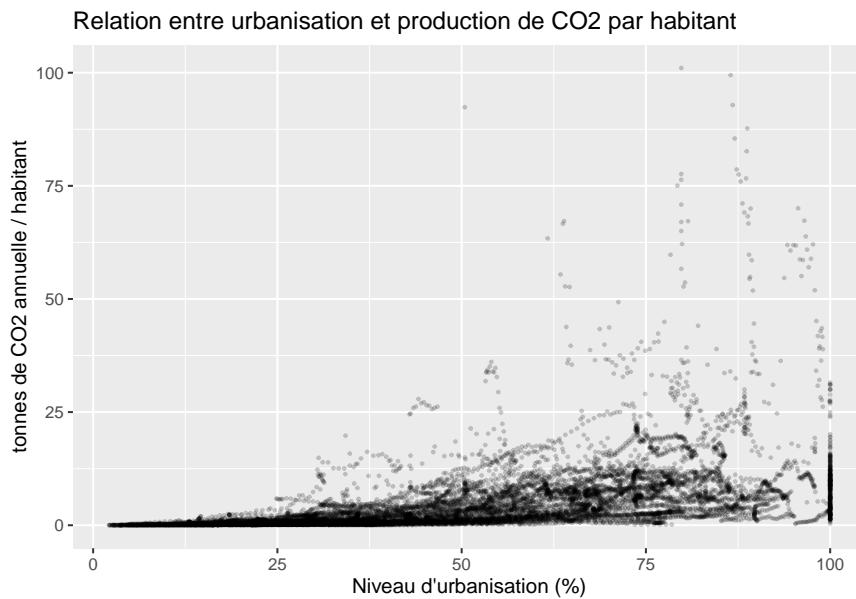


FIG. 3.33 : Nuage de points simple avec transparence

3.2.3.2 Nuage de points avec densité

Bien que la transparence nous aide un peu à distinguer les secteurs du graphique avec le plus de points, il serait plus efficace d'abandonner la géométrie des points pour la remplacer par une géométrie de densité en deux dimensions. Une première approche consiste à diviser l'espace du graphique en petits carrés et à compter le nombre de points tombant dans chaque carré (en somme, un histogramme en deux dimensions).

```
ggplot(data = data_co2)+  
  geom_bin2d(aes(x = Urbanisation, y = C02t_hab), bins = 50) +  
  scale_fill_continuous(type = "viridis") +
```

```
labs(x = "Niveau d'urbanisation (%)",
     y = 'tonnes de CO2 annuelle / habitant',
     title = 'Relation entre urbanisation et production de CO2 par habitant')
```

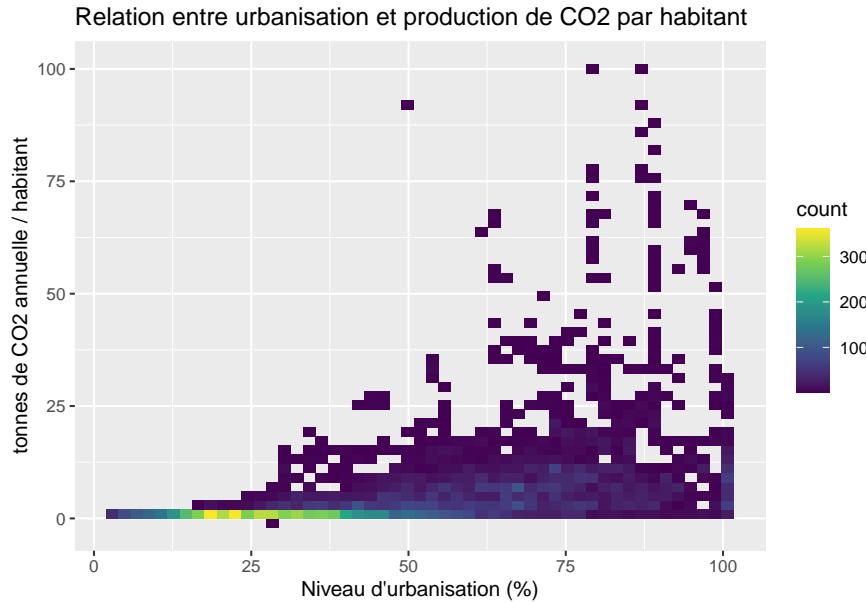


FIG. 3.34 : Densité en deux dimensions par carrés

On observe ainsi une forte concentration dans le bas du graphique, les pays avec des rejets annuels de CO₂ supérieurs à 15 tonnes par habitant sont relativement rares. Pour les personnes préférant les représentations plus élaborées, il est aussi possible de diviser l'espace du graphique avec des hexagones en utilisant le package **hexbin**.

```
ggplot(data = data_co2)+
  geom_hex(aes(x = Urbanisation, y = CO2t_hab), bins = 50) +
  scale_fill_continuous(type = "viridis") +
  labs(x = "Niveau d'urbanisation (%)",
       y = 'tonnes de CO2 annuelle / habitant',
       title = 'Relation entre urbanisation et production de CO2 par habitant')
```

Enfin, il est aussi possible de réaliser une version lissée de ces graphiques avec une fonction *kernel* en deux dimensions (`stat_density_2d`) :

```
ggplot(data = data_co2)+
  stat_density_2d(aes(x = Urbanisation, y = CO2t_hab, fill = ..density..),
                 geom = "raster", n = 50, contour = FALSE) +
  scale_fill_continuous(type = "viridis") +
  labs(x = "Niveau d'urbanisation (%)",
       y = 'tonnes de CO2 annuelle / habitant',
       title = 'Relation entre urbanisation et production de CO2 par habitant')+
  ylim(0,25)
```

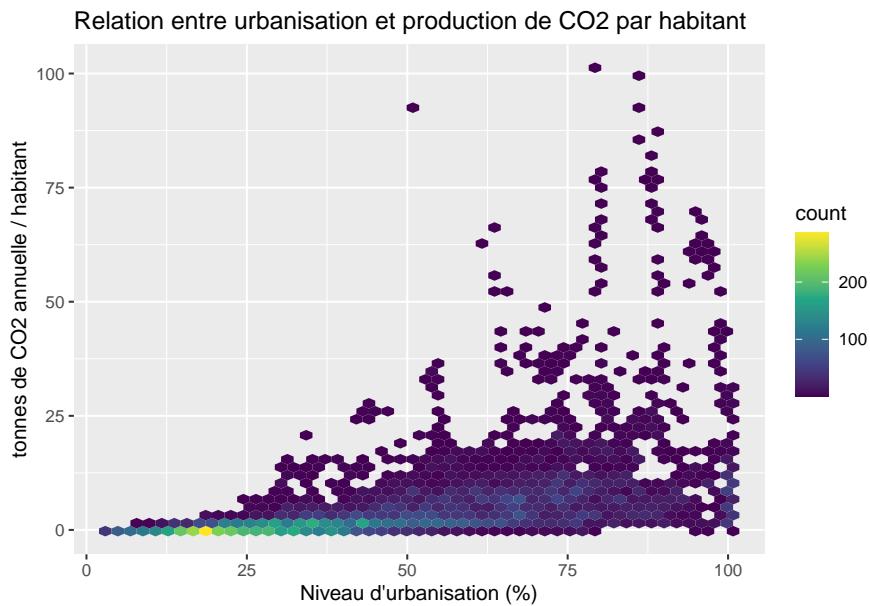


FIG. 3.35 : Densité en deux dimensions par hexagones

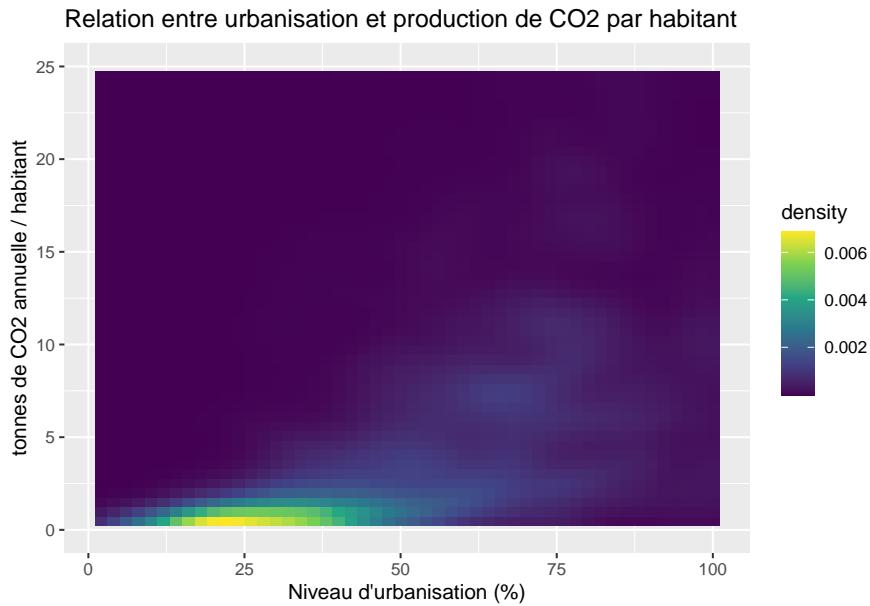


FIG. 3.36 : Densité en deux dimensions lissée

3.2.3.3 Nuage de points et droite de régression

Afin de faire ressortir une éventuelle relation entre les variables représentées sur les deux axes, il est possible d'afficher la droite de régression sur le graphique entre X et Y. Cette opération s'effectue avec la fonction `geom_smooth`.

```
ggplot(data = data_co2)+  
  geom_point(aes(x = Urbanisation, y = C02t_hab), alpha = 0.2, size = 0.5)+  
  geom_smooth(aes(x = Urbanisation, y = C02t_hab), method = lm, color = "red")+  
  labs(x = "Niveau d'urbanisation (%)",  
       y = 'tonnes de CO2 annuelle / habitant',
```

```
title = 'Relation entre urbanisation et production de CO2 par habitant')
```

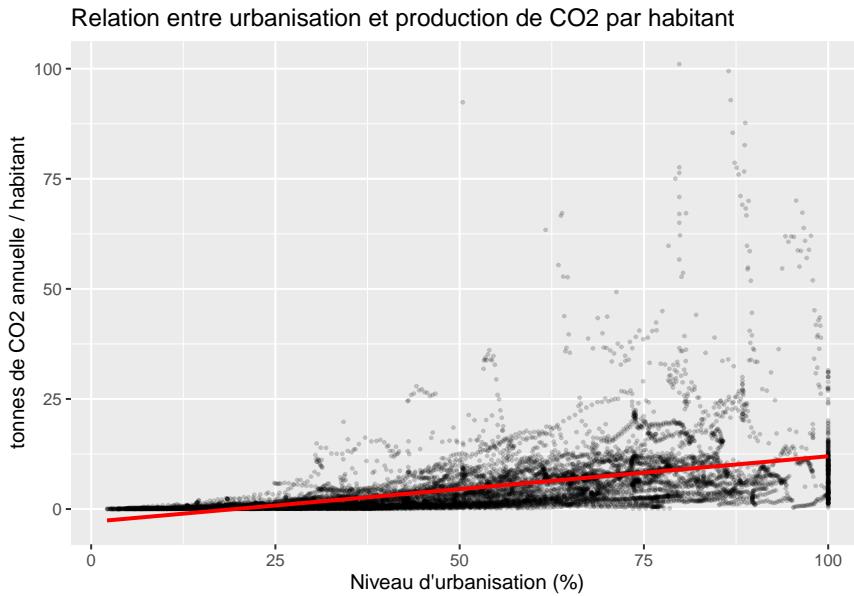


FIG. 3.37 : Nuage de points avec droite de régression

Notez que l'argument `method = lm` permet d'indiquer que nous souhaitons utiliser une régression linéaire (*linear model*) pour tracer la géométrie (une droite de régression). La droite semble bien indiquer une relation positive entre les deux variables : une augmentation de l'urbanisation serait associée avec une augmentation de la production annuelle de CO2 par habitant. On pourrait également vérifier si une relation non linéaire serait plus adaptée au jeu de données. Dans notre cas, une relation quadratique pourrait produire un meilleur ajustement.

```
ggplot(data = data_co2)+  
  geom_point(aes(x = Urbanisation, y = C02t_hab), alpha = 0.2, size = 0.7)+  
  geom_smooth(aes(x = Urbanisation, y = C02t_hab), method = lm,  
              color = "red", formula = y ~ I(x**2))+  
  labs(x = "Niveau d'urbanisation (%)",  
       y = 'tonnes de CO2 annuelle / habitant',  
       title = 'Relation entre urbanisation et production de CO2 par habitant')
```

La régression quadratique (avec x au carré) nous indique ainsi que l'impact du niveau d'urbanisation est plus important à mesure que le niveau d'urbanisation augmente. Vous pouvez également constater que la courbe ne prédit pas de valeurs négatives comparativement à la droite précédente. Il est également possible d'ajuster une courbe sans choisir au préalable sa forme (dans le cas précédent x^2) en utilisant une méthode d'ajustement local appelée *loess*.

```
ggplot(data = data_co2)+  
  geom_point(aes(x = Urbanisation, y = C02t_hab), alpha = 0.2, size = 0.5)+  
  geom_smooth(aes(x = Urbanisation, y = C02t_hab), method = loess,  
              color = "red")+  
  labs(x = "Niveau d'urbanisation (%)",  
       y = 'tonnes de CO2 annuelle / habitant',  
       title = 'Relation entre urbanisation et production de CO2 par habitant')
```

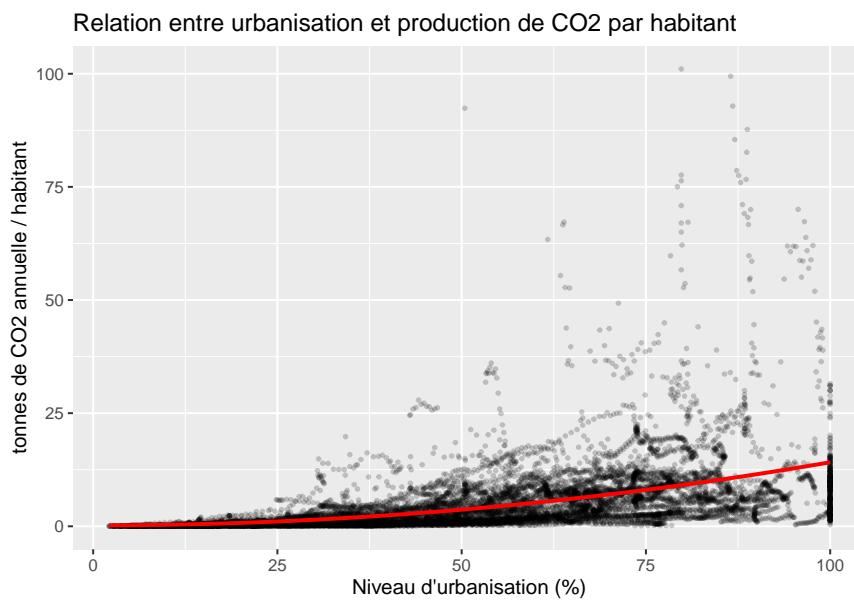


FIG. 3.38 : Nuage de points avec droite de régression exponentielle

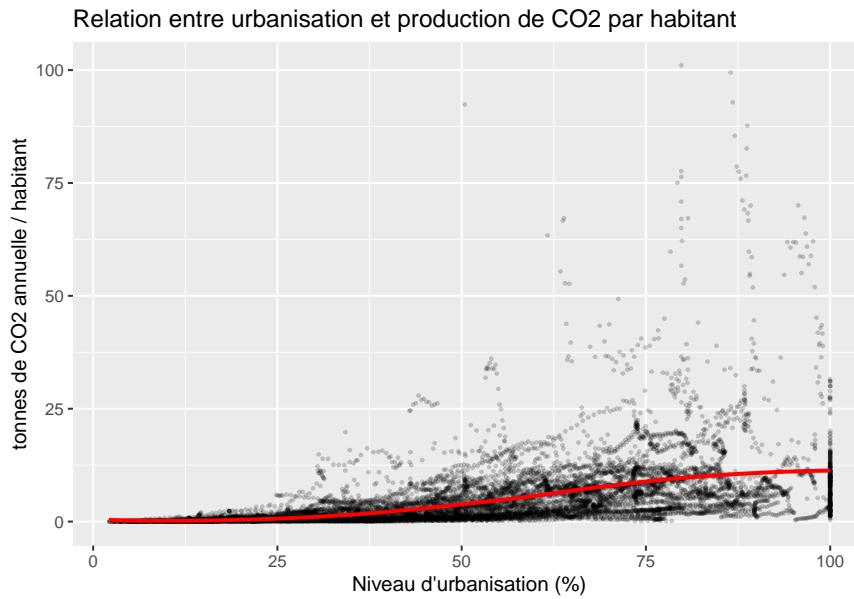


FIG. 3.39 : Nuage de points avec droite de régression non linéaire

La relation non linéaire révèle davantage d'informations : l'augmentation de l'urbanisation est associée à une augmentation de l'émission de CO₂ par habitant uniquement jusqu'à 75% d'urbanisation ; au-delà de ce seuil, la relation ne tient plus. Ces résultats semblent cohérents avec l'évolution classique de l'économie d'un pays passant progressivement d'une économie agricole, à une économie industrialisée et finalement une économie de services.

3.2.4 Graphique en lignes

Un graphique en ligne permet de représenter l'évolution d'une variable, généralement dans le temps. Dans le jeu de données précédent, nous disposons des émissions de CO₂ par habitant de nombreux pays sur plusieurs années. Nous pouvons ainsi représenter l'évolution des émissions pour chaque pays avec

un graphique en ligne. Pour éviter de le surcharger, cet exercice est réalisé uniquement sur les pays de l'Europe de l'Ouest.

```
# conversion de la variable year textuelle en variable numérique
data_co2$an <- as.numeric(data_co2$year)
# extraction des données d'Europe de l'Ouest
data_europe <- subset(data_co2, data_co2$region23 == "Western Europe")
# choix des valeurs pour l'axe des x
x_ticks <- seq(1960,2020,10)

ggplot(data = data_europe)+ 
  geom_path(aes(x = an, y = CO2t_hab, color = Country.Name))+ 
  labs(x = "Années",
       y = 'Tonnes de CO2 annuelle / habitant',
       color = "Pays",
       title = 'Évolution de la production de CO2 par habitant') + 
  scale_x_continuous(breaks = x_ticks, labels = x_ticks)+ 
  theme_tufte()
```

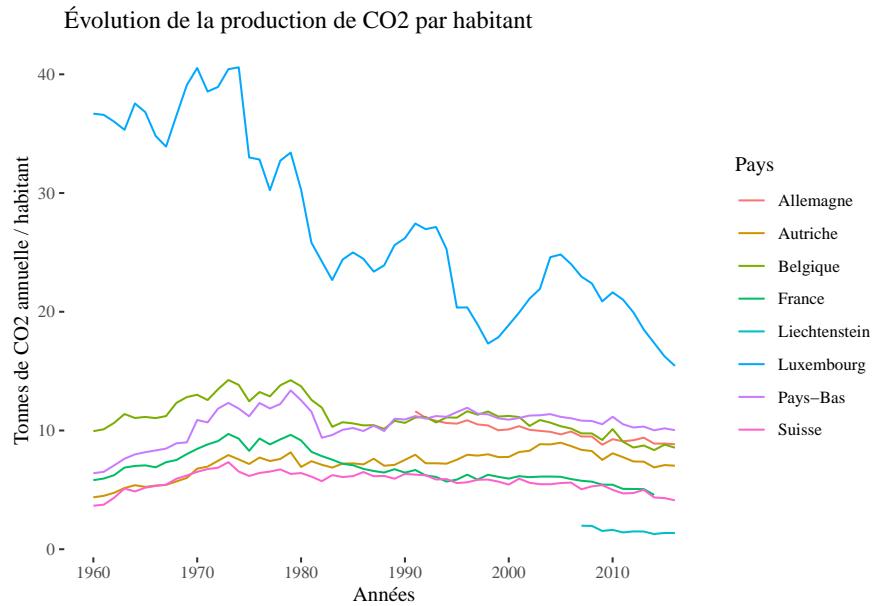


FIG. 3.40 : Graphique en ligne

On remarque notamment qu'aucune donnée avant 2005 n'est disponible pour le Liechtenstein.

3.2.4.1 Barre d'erreur et en bande

Sur un graphique, il est souvent pertinent de représenter l'incertitude que nous avons sur nos données. Cela peut être fait à l'aide de barres d'erreurs ou à l'aide de polygones délimitant les marges d'incertitude. En guise d'exemple, admettons que les données précédentes sont fiables à plus ou moins 10%. En d'autres termes, la valeur d'émission de CO2 annuelle serait relativement incertaine et pourrait se situer dans un intervalle de 10% autour de la valeur fourni par la Banque Mondiale. Nous obtenons ainsi une borne inférieure (valeur donnée - 10%) et une borne supérieure (valeur donnée + 10%). Nous pouvons facilement calculer ces bornes et les faire apparaître dans notre graphique précédent.

```

data_europe$borne_basse <- data_europe$CO2t_hab - 0.1 * data_europe$CO2t_hab
data_europe$borne_haute <- data_europe$CO2t_hab + 0.1 * data_europe$CO2t_hab

ggplot(data = data_europe)+
  geom_point(aes(x = an, y = CO2t_hab, color = Country.Name), size = 0.7)+
  geom_errorbar(aes(x = an, ymin = borne_basse, ymax = borne_haute, color = Country.Name))+ 
  labs(x = "Années",
       y = 'Tonnes de CO2 annuelle / habitant',
       color = "Pays",
       title = 'Évolution de la production de CO2 par habitant') +
  scale_x_continuous(breaks = x_ticks, labels = x_ticks)+ 
  theme_tufte()

```

Évolution de la production de CO2 par habitant

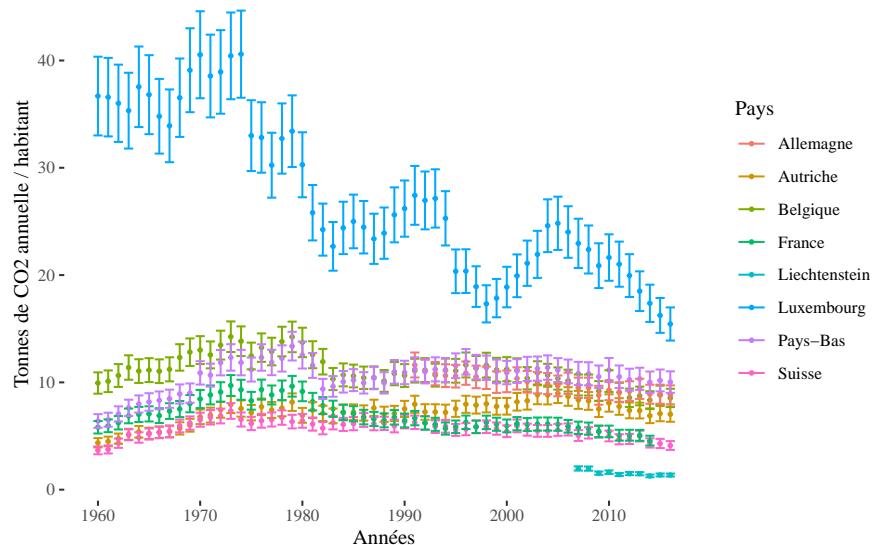


FIG. 3.41 : Graphique en ligne avec barres d'erreur

Ces barres d'erreurs indiquent notamment qu'il n'y a finalement aucun écart significatif entre la Belgique, les Pays-Bas et l'Allemagne à partir des années 1990. Une autre option de représentation est d'utiliser des polygones avec la fonction `geom_ribbon`.

```

ggplot(data = data_europe)+
  geom_path(aes(x = an, y = CO2t_hab, color = Country.Name), size = 0.7)+ 
  geom_ribbon(aes(x = an, ymin = borne_basse, ymax = borne_haute, fill = Country.Name), alpha = 0.4)+ 
  labs(x = "Années",
       y = 'Tonnes de CO2 annuelle / habitant',
       color = "Pays",
       title = 'Évolution de la production de CO2 par habitant') +
  scale_x_continuous(breaks = x_ticks, labels = x_ticks)+ 
  theme_tufte()+
  guides( fill = FALSE)

```

Le message du graphique est le même. Notez que nous avons utilisé ici la fonction `guides` pour retirer de la légende les couleurs associées au remplissage des marges d'erreurs. Ces couleurs sont les mêmes que celles des lignes et il n'est pas utile de dédoubler la légende. De nombreuses méthodes statistiques produisent des résultats accompagnés d'une mesure de l'incertitude associée à ces résultats. Représenter

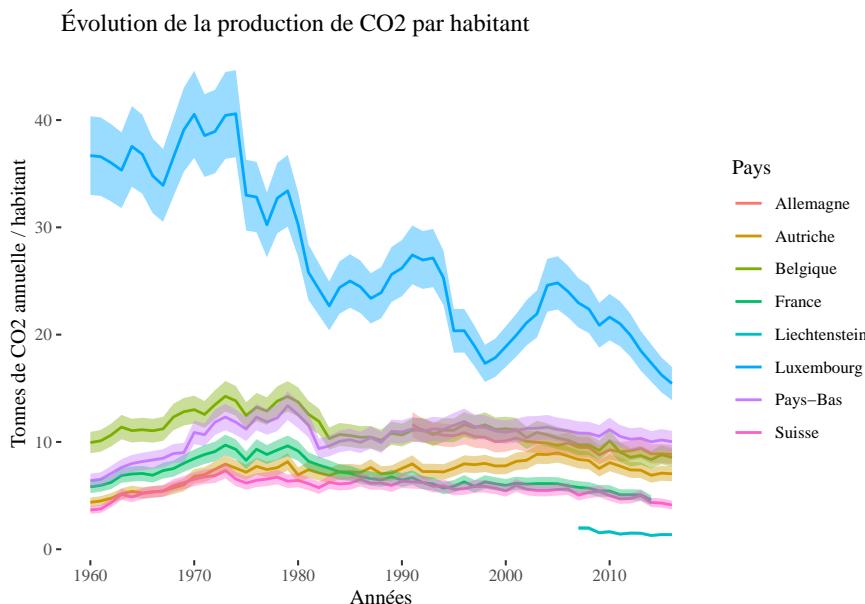


FIG. 3.42 : Graphique en ligne avec marge d'erreur

cette incertitude est crucial pour que le lecteur puisse délimiter la portée des conclusions de vos analyses.

3.2.5 Boites à moustache

Les boites à moustache (*box plot* en anglais) sont des graphiques permettant de comparer les moyennes et les intervalles interquartiles d'une variable continue selon plusieurs groupes d'une population. Si l'on reprend notre exemple précédent, nous pourrions comparer en fonction de la région du monde la moyenne de production annuelle de CO2 par habitant. Pour cela, il suffit d'utiliser la fonction `geom_boxplot`.

```
#retirer les observations n'étant pas associées à une région
data_co2_comp <- subset(data_co2, is.na(data_co2$region7) == F)

ggplot(data = data_co2_comp)+
  geom_boxplot(aes(y = region7, x = C02t_hab))+ 
  labs(x="Tonnes de CO2 par an et habitant", y="Région")
```

La barre centrale d'une boite représente la moyenne. Les extrémités de la boite représentent le premier et le troisième quartile. Plus une boite est allongée, plus les situations sont diversifiées pour les observations appartenant au groupe représenté par la boite. Au contraire, une boite étroite indique un groupe homogène. Notez qu'en inversant les variables dans les axes X et Y, on obtiendrait des boites à moustache verticales. Cependant, les noms des régions étant assez longs, cela nécessiterait d'avoir un graphique très large. Améliorons quelque peu le rendu de ce graphique en ajoutant des titres.

```
ggplot(data = data_co2_comp)+ 
  geom_boxplot(aes(y = region7, x = C02t_hab))+ 
  xlim(c(0,50))+ 
  labs(title = "Niveaux d'émission de CO2 par habitant par région géographique",
       x = "Tonnes de CO2 par an et habitant",
       y = 'Région')
```

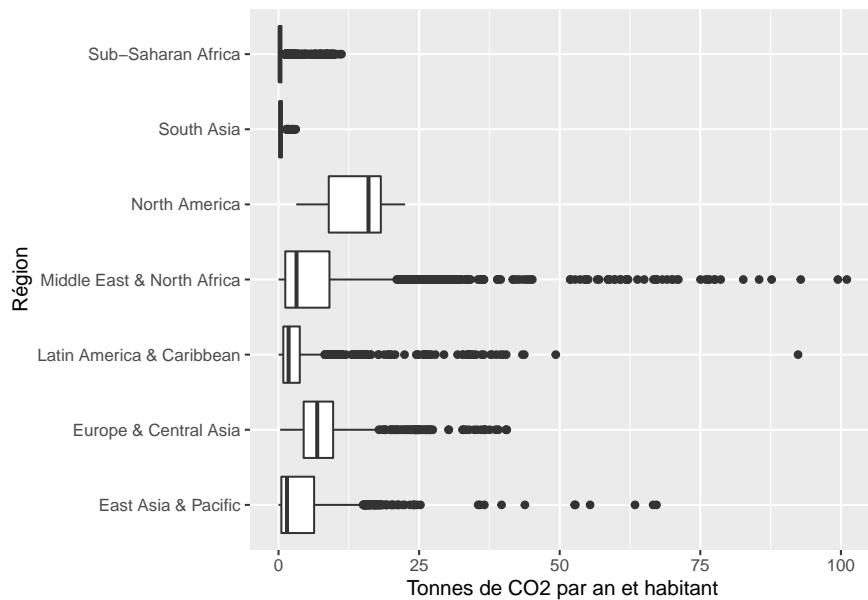


FIG. 3.43 : Boite à moustache

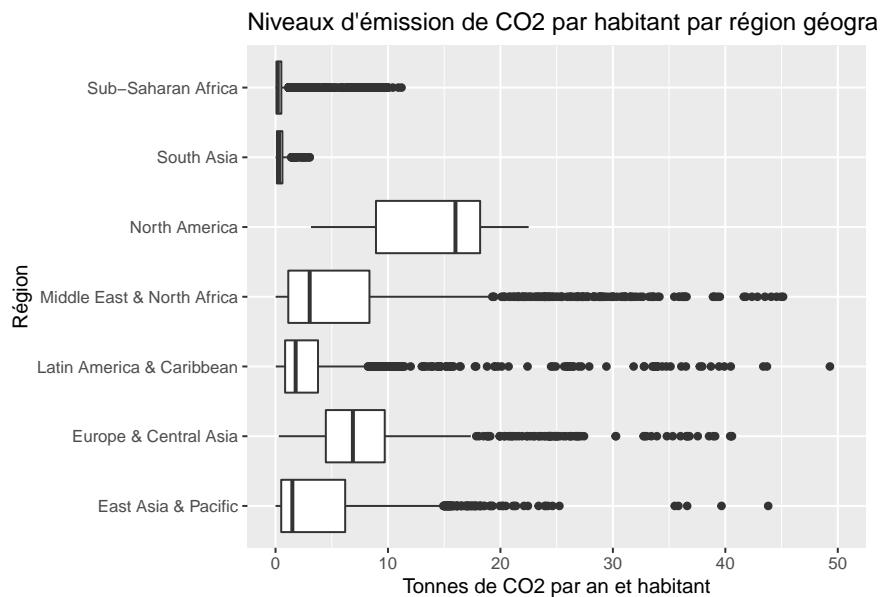


FIG. 3.44 : Boite à moustache améliorée

Les points noirs sur le graphique représentent des valeurs extrêmes, soit des observations situées à plus de 1,5 intervalle interquartile d'une extrémité de la boîte. Pour mieux rendre compte de la densité d'observations le long de chaque boîte à moustache, il est possible de les représenter directement avec la fonction `geom_jitter`.

```
ggplot(data = data_co2_comp)+  
  geom_boxplot(aes(y = region7, x = C02t_hab), outlier.shape = NA)+  
  geom_jitter(aes(y = region7, x = C02t_hab), size = 0.2, alpha = 0.2)+  
  xlim(c(0,50))+  
  labs(title = "Niveaux d'émission de CO2 par habitant par région géographique",  
       x = "Tonnes de CO2 annuelle par habitant",
```

```
y = 'Région')
```

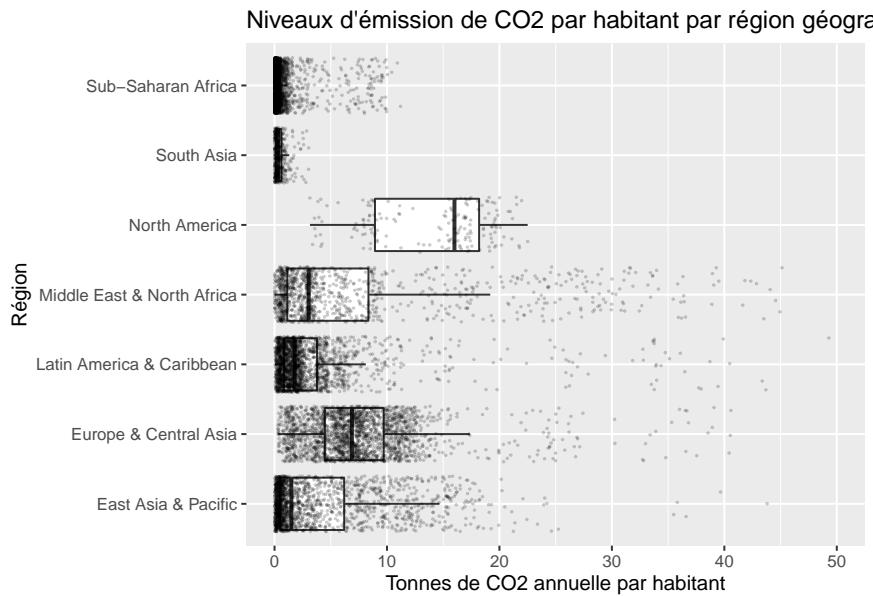


FIG. 3.45 : Boîte à moustache avec observations

Notez que pour éviter que les valeurs extrêmes identifiées par la fonction `geom_boxplot` se superposent avec les points représentant les observations, nous les avons supprimées avec l'argument `outlier.shape = NA`.

3.2.6 Graphique en violons

Les boites à moustaches donnent des informations pertinentes sur le centre et la dispersion d'une variable en fonction de sous groupes de la population. Cependant, une grande partie de l'information reste masquée par la représentation sous forme de boite. Une alternative est de remplacer la simple boite par la distribution de la variable étudiée. On obtient ainsi des graphiques en violon (`geom_violin`). Considérant les très grands écarts entre les régions que nous avons observés avec les boites à moustache, il est préférable de tracer les graphiques en violon en excluant les régions Afrique Sub-Saharienne et Asie du Sud.

```
# retirons les observations de régions que nous ne souhaitons pas garder
data_co2_comp <- subset(data_co2, (! data_co2$region7 %in% c("Sub-Saharan Africa", "South Asia"))
  & !is.na(data_co2$region7)==TRUE)

ggplot(data = data_co2_comp)+ 
  geom_violin(aes(y = region7,x = CO2t_hab))+ 
  xlim(c(0,50))+ 
  labs(title = "Comparaison des niveaux d'émission de CO2 par habitant par région géographique",
       x = "Tonnes de CO2 annuelle / habitant",
       y = '')+ 
  geom_vline(xintercept = 12, linetype = 'dashed', color = 'blue')
```

Ces distributions permettent notamment de souligner que deux groupes distincts se retrouvent en Amérique du Nord. L'un dont les émissions annuelles de CO2 par habitant sont inférieure à 12 tonnes (ligne bleue) et un autre groupe pour lequel elles sont supérieures. En explorant les données, on constate

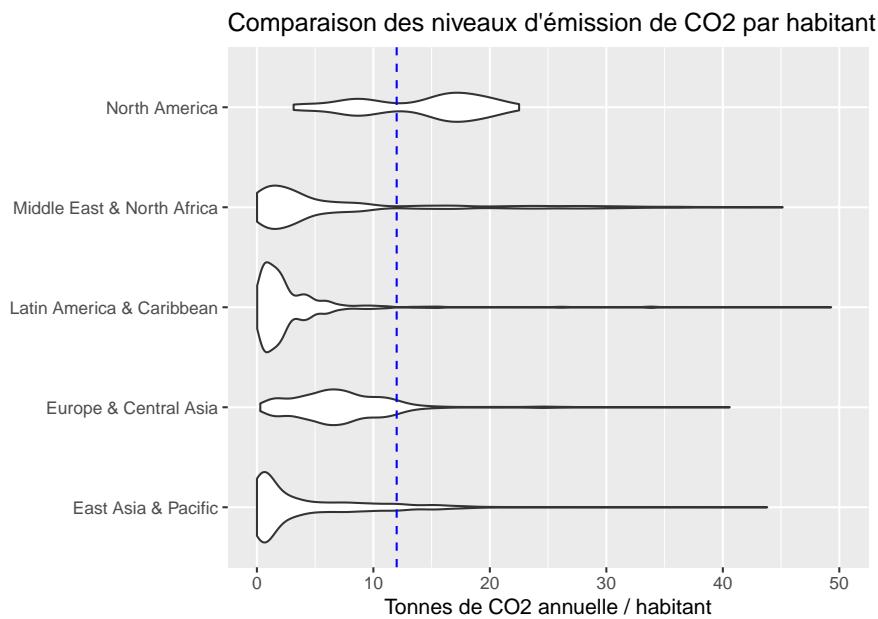


FIG. 3.46 : Graphique en violon

que les Bermudes sont inclus dans le groupe Amérique du Nord, mais ont des niveaux d'émission inférieurs à ceux du Canada et des États Unis, ce qui explique cette distribution bimodale. Cette information était masquée avec les boîtes à moustaches. Finalement, il est aussi possible de superposer graphique en violon et boîte à moustaches pour bénéficier des avantages des deux.

```
ggplot(data = data_co2_comp) +
  geom_violin(aes(y = region7, x = CO2t_hab)) +
  geom_boxplot(aes(y = region7, x = CO2t_hab), width = 0.15) +
  xlim(c(0,50)) +
  labs(title = "Comparaison des niveaux d'émission de CO2 par habitant par région géographique",
       x = "Tonnes de CO2 annuelle / habitant",
       y = '')
```

3.2.7 Graphique en barres

Les graphiques en barres permettent de représenter des quantités (hauteur des barres) réparties dans des catégories (une barre par catégorie). Nous proposons ici un exemple avec des données de déplacements issues de l'*enquête origine destination de Québec de 2017* au niveau des grands secteurs (figure 3.47).

Nous représentons pour chaque secteur le nombre de déplacements moyen entrant et sortant un jour de semaine en heures de pointe. Les données sont présentées sous forme d'une matrice carrée (avec autant de lignes que de colonnes). L'intersection de la ligne A et de la colonne C indique le nombre de personnes partant du secteur A pour se rendre dans le secteur C. À l'inverse, l'intersection de la ligne C et de la colonne A indique le nombre de personnes partant du secteur C pour se rendre dans le secteur A. En sommant les valeurs de chaque ligne, on obtient le nombre total de départs par secteur tandis que le nombre d'arrivées est la somme de chaque colonne. Ces opérations peuvent simplement être effectuées avec les fonctions `rowSums` et `colSums`.

```
# chargement des données
matriceOD <- read.csv('data/graphique/Quebec_2017_OD_MJ.csv',
```

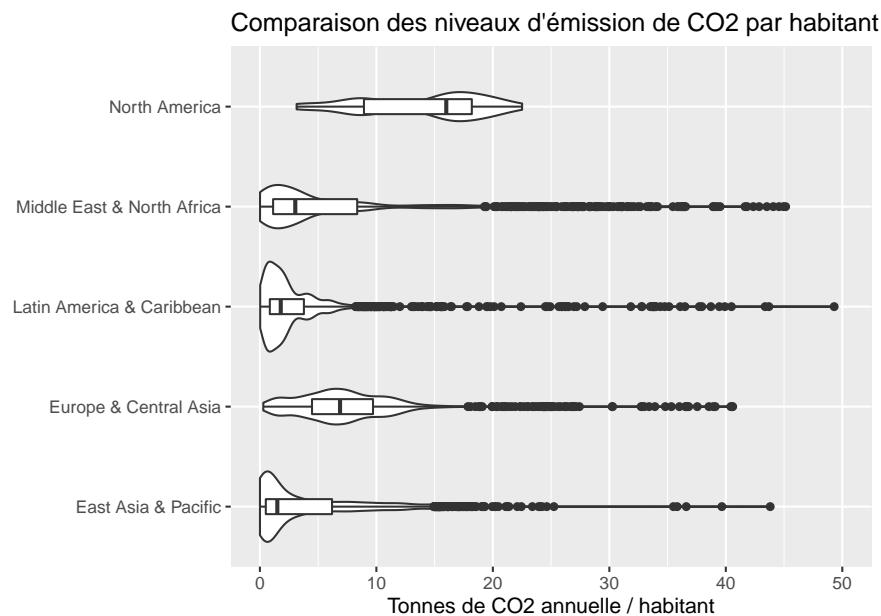


FIG. 3.47 : Graphique en violon et boîte à moustaches

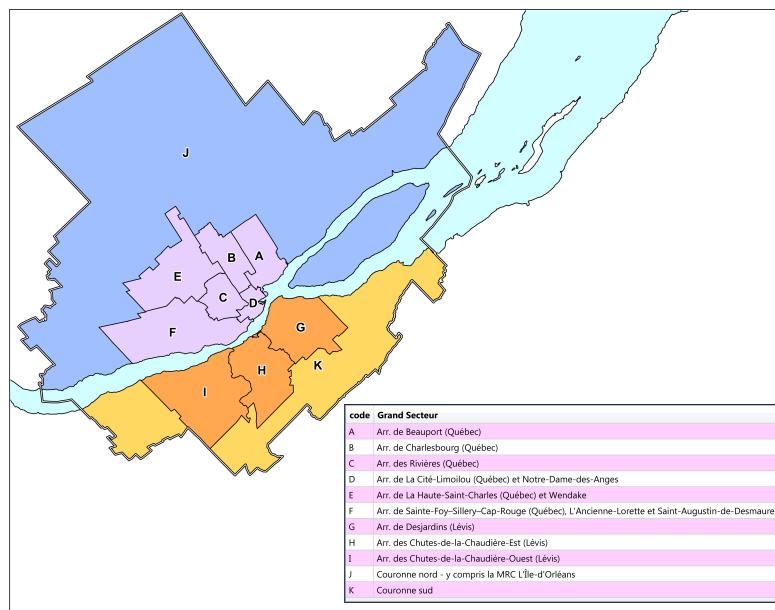


FIG. 3.48 : Grands secteurs de Québec

```

    header = FALSE, sep = ';') # fichier csv sans entête

# calcul des sommes en lignes et en colonnes
tot_depart <- rowSums(matriceOD)
tot_arrivee <- colSums(matriceOD)

# création d'un DataFrame avec les valeurs et les noms des secteurs
df <- data.frame(depart = tot_depart,
                  arrivee = tot_arrivee,
                  secteur = c('Arr. de Beauport (Québec)',
                             'Arr. de Charlesbourg (Québec)',
                             'Arr. des Rivières (Québec)',
                             'Arr. de la Cité-Limoilou (Québec)',
                             'Arr. de la Haute-St-Charles (Québec)',
                             'Arr. de Sainte-Foy-Sillery- Cap-Rouge (Québec)',
                             'Arr.de Desjardins (Lévis)',
                             'Arr. des Chutes-de-la-Chaudière-Est (Lévis)',
                             'Arr. Les Chutes de la Chaudière-Ouest (Lévis)',
                             'Ceinture Nord',
                             'Ceinture Sud',
                             'Hors Territoire'),
                  code = c('A','B','C','D','E','F','G','H','I','J','K','X'))

# création des deux graphiques en barres
plot1 <- ggplot(data = df)+  

  geom_bar(aes(x = code, weight = depart))+  

  labs(subtitle = 'Départs',  

       x = 'total',  

       y = '')

plot2 <- ggplot(data = df)+  

  geom_bar(aes(x = code, weight = arrivee))+  

  labs(subtitle = 'Arrivées',  

       x = 'total',  

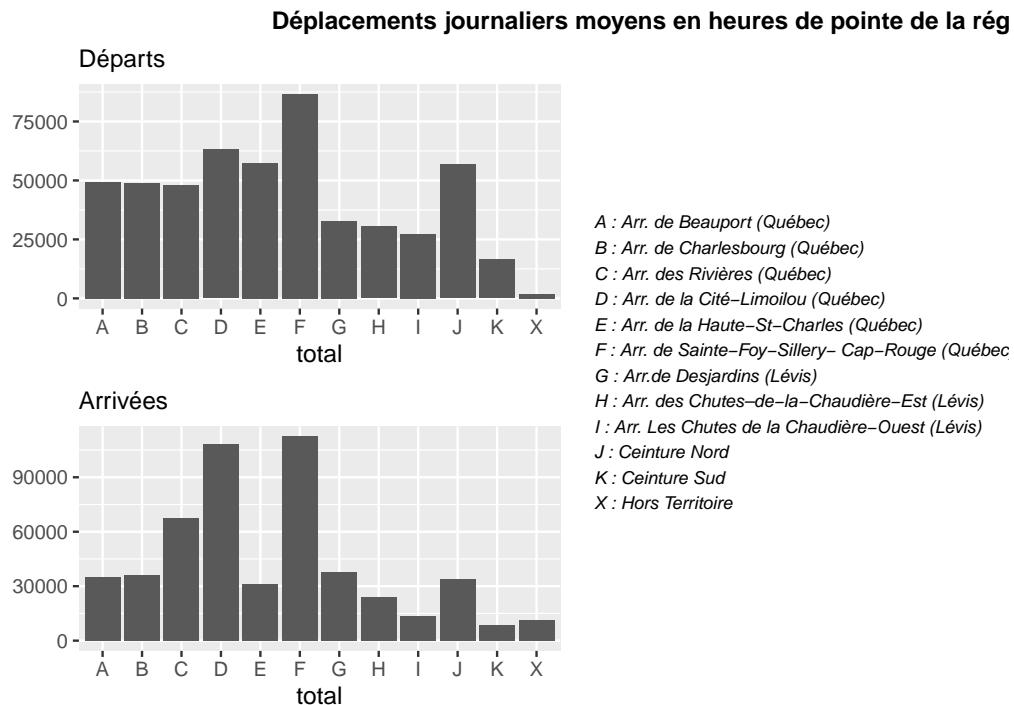
       y = '')

# stocker les graphiques dans une liste et composer une figure
list_plot <- list(plot1, plot2)
tot_plot <- ggarrange(plotlist = list_plot, ncol = 1)

# création d'une légende pour associer le code de chaque secteur
# à son nom. Pour cela on concatène en premier les lettres et les noms,
# on fusionne ensuite le tout en les séparant par le symbole \n représentant
# un saut de ligne.
nom_secteurs <- paste(df$code, df$secteur, sep= ' : ')
string_names <- paste(nom_secteurs, collapse = '\n')

titre <- "Déplacements journaliers moyens en heures de pointe de la région de Québec"
# production finale de la figure
annotate_figure(tot_plot,
                top = text_grob(titre, face = "bold", size = 11, just = "left"),
                right = text_grob(string_names, face = "italic", size = 8,
                                  just = "left", x = 0.05) # position du texte
)

```

**FIG. 3.49 :** Graphiques en barres simples

Plutôt que de représenter les arrivées et les départs dans deux graphiques séparés, il est possible de les empiler dans un même graphique en barres. Nous devons au préalable « faire fondre nos données » avec la fonction `melt`.

```
# faire fondre le jeu de données (empiler les colonnes depart et arrivee)
melted_df <- melt(df, id.vars = c('code'), measure.vars = c('depart','arrivee'))
names(melted_df) <- c('code','deplacement','effectif')
# ajouter les accents dans la colonne déplacement
melted_df$deplacement <- ifelse(melted_df$deplacement == 'depart', 'départ', 'arrivée')

# comparaison du format original et du format "fondu"
head(df)
```

##	depart	arrivee	secteur	code
## V1	49241	34777	Arr. de Beauport (Québec)	A
## V2	48909	36344	Arr. de Charlesbourg (Québec)	B
## V3	48044	67198	Arr. des Rivières (Québec)	C
## V4	63132	108138	Arr. de la Cité-Limoilou (Québec)	D
## V5	57367	30859	Arr. de la Haute-St-Charles (Québec)	E
## V6	86504	112379	Arr. de Sainte-Foy-Sillery- Cap-Rouge (Québec)	F

```
head(melted_df)
```

##	code	deplacement	effectif
## 1	A	départ	49241
## 2	B	départ	48909
## 3	C	départ	48044
## 4	D	départ	63132

```
## 5     E      départ    57367
## 6     F      départ    86504
```

```
# réalisation du graphique
plot1 <- ggplot(data = melted_df) +
  geom_bar(aes(x = code, weight = effectif, fill = deplacement), color = '#e3e3e3') +
  scale_fill_manual(values = c("#e63946", "#1d3557")) +
  labs(title = titre,
       y = 'Effectifs',
       x = '',
       fill = 'Déplacements')

annotate_figure(plot1, right = text_grob(string_names, face = "italic", size = 7,
                                         just = "left", x = 0.05)) # position du texte
```

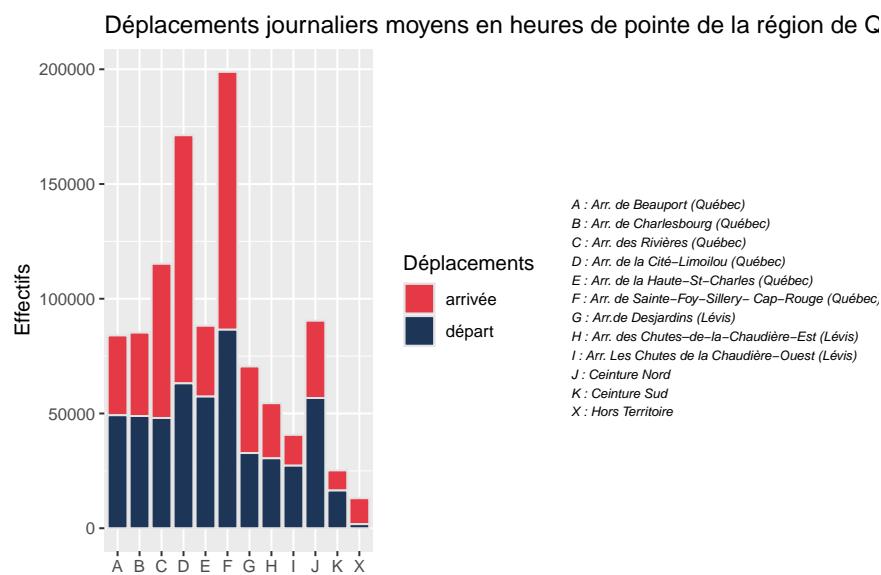


FIG. 3.50 : Graphique en barres empilées

3.2.8 Graphique circulaire

Une alternative directe au graphique en barre est le graphique ou diagramme circulaire, appelé aussi graphique en pointes de tarte (pour ceux à la dent sucrée) ou en camembert (pour les amateurs de fromage). Il est suffisamment connu et utilisé pour qu'aucune présentation ne s'impose. Pour être exact, un graphique en pointes de tarte n'est rien d'autre qu'un graphique en barres dont le système de coordonnées a été modifié. Cela impose cependant de calculer à l'avance la position des étiquettes que l'on souhaite ajouter sur le graphique. Reprenons les données de production mondiale de CO₂ et calculons les productions totales par région géographique en 2015.

```
library(dplyr)

# extraire les données de 2018 pour lesquelles on connaît la région
data_co2_2015 <- subset(data_co2, data_co2$year == "2015" & ! is.na(data_co2$region7))

# effectuer la somme du CO2 par région
co2_2015 <- data_co2_2015 %>%
```

```

group_by(region7) %>%
summarise(total_co2 = sum(CO2_kt,na.rm = TRUE))

# attribuer un code à chaque région pour faciliter la lecture
co2_2015$code <- c("A","B","C","D","E","F","G")

# modifier l'ordre des données, calculer les proportions et la position des labels
df <- co2_2015 %>%
arrange(desc(code)) %>%
mutate(prop = total_co2 / sum(co2_2015$total_co2) *100) %>%
mutate(ypos = cumsum(prop)- 0.5*prop )

# préparer la légende (pourcentages et vrais noms)
nom_region <- rev(paste(df$code, " : ", df$region7, "(", round(df$prop,1), "%)"))
string_region <- paste(nom_region, collapse = '\n')

# construire le graphique
plot1 <- ggplot(df, aes(x="", y=prop, fill=code)) +
  geom_bar(stat="identity", width=1, color="white") +
  coord_polar("y", start=0) +
  theme_void() +
  theme(legend.position="none") +
  geom_text(aes(y = ypos, label = code), color = "white", size=3) +
  scale_fill_grey()+
  labs(title = "Proportion du CO2 émis en 2015")

# ajouter la légende
annotate_figure(plot1,right = text_grob(string_region, face = "italic", size = 9,
just = "left", x = 0.05)) # position du texte)

```

Proportion du CO2 émis en 2015

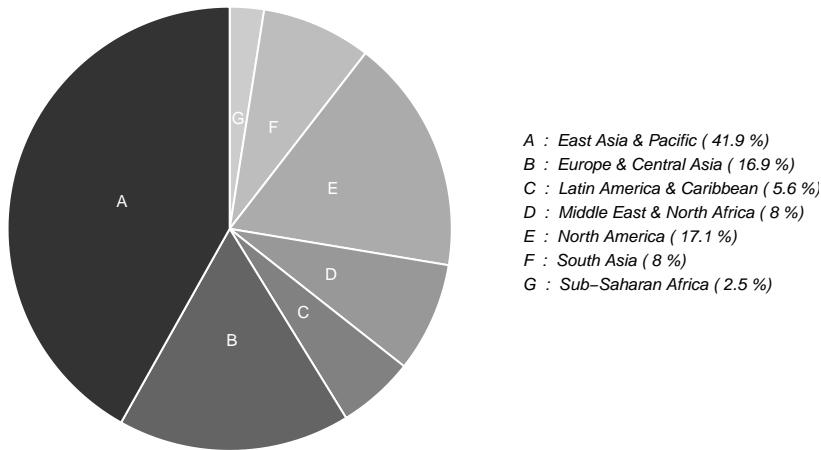


FIG. 3.51 : Graphique en pointes de tarte

Si à la place de la géométrie `geom_bar`, vous utilisez `geom_rect`, vous pouvez convertir votre graphique en pointes de tarte en graphique en anneau (ou en beigne, pour ceux à la dent sucrée) :

```

# calculer la limite inférieure et supérieure du beigne
df$ymax <- cumsom(df$prop)
df$ymin <- c(0, head(df$ymax, n=-1))

# construire le graphique
plot1 <- ggplot(df, aes(ymax=ymax, ymin=ymin,
                        xmax=4, xmin=3,
                        y=prop, fill=code)) +
  geom_rect(stat="identity", color="white") +
  coord_polar("y", start=0) +
  theme_void() +
  theme(legend.position="none") +
  geom_text(aes(x = 3.5,y = ypos, label = code), color = "white", size=3) +
  scale_fill_grey()+
  xlim(c(2,4))+ 
  labs(title = "Proportion du CO2 émis en 2015")

# ajouter la légende
annotate_figure(plot1,right = text_grob(string_region, face = "italic", size = 8,
                                         just = "left", x = 0.05)) # position du texte)

```

Proportion du CO2 émis en 2015

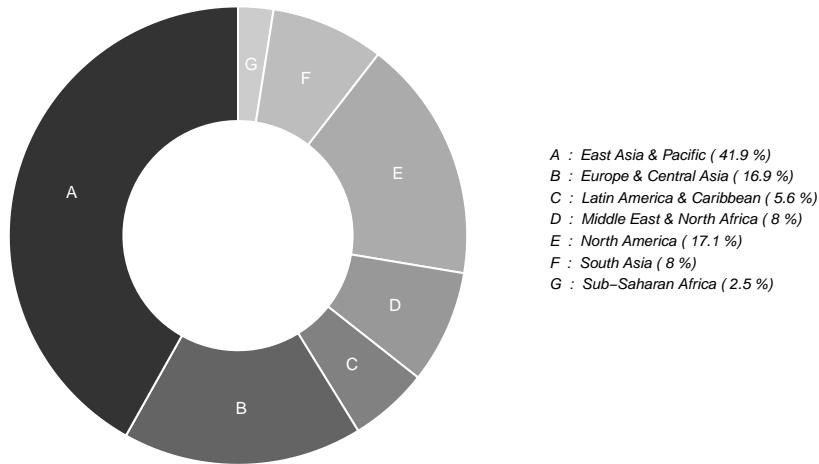


FIG. 3.52 : Graphique en anneau

3.3 Graphiques spéciaux

Dans cette dernière section, nous allons aborder des graphiques plus rarement utilisés. Ils sont toutefois très utiles dans certains contextes du fait de leur capacité à synthétiser des informations complexes.

3.3.1 Graphique en radar

Les graphiques en radar (ou en toile d'araignée) sont utilisés pour comparer une série de variables continues pour plusieurs observations ou groupe d'observations. Chaque variable est associée à un axe et chaque observation est représentée avec un polygone. Prenons comme exemple les données de logement par secteur de recensement dans la région métropolitaine de Montréal en 2016. On pourrait

souhaiter comparer la moyenne des pourcentages des différents types de logements pour les régions des Laurentides, de la Montérégie, de Laval, de Longueuil et de Montréal. Malheureusement, **ggplot2** ne permet pas de dessiner des graphiques en radar satisfaisants, nous devrons donc utiliser le package **fmsb**.

```
library(fmsb)

## 
## Attaching package: 'fmsb'

## The following objects are masked from 'package:DescTools':
## 
##     CronbachAlpha, VIF

data <- read.csv('data/bivariee/sr_rmr_mtl_2016.csv', header = T, encoding = 'UTF-8')

# aggrégeons les données au niveau des régions en calculant la moyenne des pourcentages
variables <- c("MaisonIndi","App5Plus","MaisRangee","AppDuplex","Proprio","Locataire")

data_region <- data[c("Region",variables)] %>%
  group_by(Region) %>%
  summarise_all(.funs = list(mean))

# gérer le nom des colonnes pour ajuster les données aux besoins de
# la fonction radachart
new_names <- c("Region",paste(variables,"_mean",sep=""))
names(data_region) <- new_names
data_region <- data.frame(data_region)
rownames(data_region) <- data_region$Region
data_region$Region <- NULL

# ajouter deux lignes aux données avec les valeurs maximales et minimales
# de chaque colonne. Ces informations aideront la fonction radachart à
# dessiner chacun des axes du radar
data_chart <- rbind(apply(data_region,MARGIN = 2, FUN = max),
                    apply(data_region,MARGIN = 2, FUN = min),
                    data_region
                  )

# choix des couleurs pour l'intérieur des polygones (avec transparence)
couleurs <- c(
  rgb(0.94, 0.28, 0.44, 0.25),
  rgb(1.00, 0.82, 0.40, 0.25),
  rgb(0.02, 0.84, 0.63, 0.25),
  rgb(0.07, 0.54, 0.70, 0.25),
  rgb(0.03, 0.23, 0.30, 0.25)
)

# choix des couleurs pour l'intérieur des polygones (sans transparence)
couleurs_contour <- c(
  rgb(0.94, 0.28, 0.44),
  rgb(1.00, 0.82, 0.40),
  rgb(0.02, 0.84, 0.63),
  rgb(0.07, 0.54, 0.70),
```

```

rgb(0.03, 0.23, 0.30)
)

# dessin du graphique
radarchart(data_chart,
    title = "Comparaison des types de logements dans la RMR",
    pcol = couleurs_contour, pfcol = couleurs,
    plwd = 2, pltly=1,
    cglcol="grey", cglty=1, axislabcol="grey", cglwd=0.8,
    vlcex=0.8,
    vlabels = c("maison individuelle", "immeuble d'appartements",
               "maison \nen rangée", "duplex",
               "propriétaire", "locataire")
)
# ajout d'une légende
legend(x=1.3, y=1, legend = rownames(data_chart[-c(1,2),]), bty = "n",
       pch=20 , col=couleurs , text.col = "black", cex=0.9, pt.cex=1.5)

```

Comparaison des types de logements dans la RMR

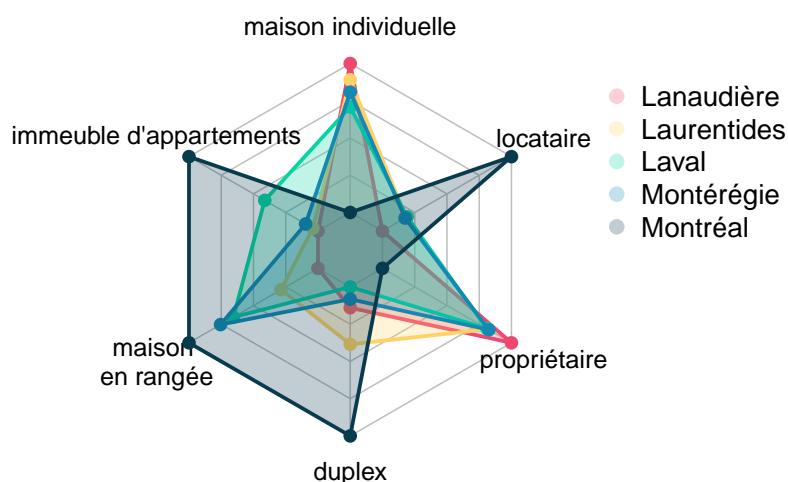


FIG. 3.53 : Graphique en anneau

À la lecture du graphique, on constate rapidement que l'île de Montréal a une situation très différentes des trois autres régions. Laval se distingue également avec une part importante de logements dans des immeubles d'appartements. Ce type de graphique a pour objectif d'orienter le regard sur de potentielles différences dans un contexte multidimensionnelle, mais il comporte quelques inconvénients :

- Les échelles de chaque axe sont différentes. Il est donc essentiel de se rapporter aux valeurs exactes pour estimer si les écarts sont importants en termes absolus.
- La superposition de plusieurs polygones peut rendre la lecture difficile. Une alternative envisageable est de réaliser un graphique par polygone, mais cela prend beaucoup de place dans un document.
- L'utilisation de polygones donne parfois de fausses impressions d'écarts. Dans le précédent graphique, l'œil est attiré en bas à gauche par le polygone de Montréal très différent des autres. Cependant, les écarts sur l'axe *maison en rangée* sont relativement petits comparativement à l'axe *locataire* situé à l'opposé.

3.3.2 Diagramme d'accord

Les diagrammes d'accord (*chord diagram* en anglais) sont utilisés pour représenter des échanges ou des connexions entre des entités. Il peut s'agir par exemple de marchandises importées / exportées entre pays, des messages envoyés entre utilisateurs de réseaux sociaux, de flux de population, etc. Reprenons nos données de l'enquête *origine destination de la région de Québec en 2017* pour illustrer le tout. Nous utiliserons le package **chorddiag**, très facile d'utilisation et produisant des graphiques interactifs, facilitant grandement la lecture de ce type de graphique. Cependant ce package ne fait pas partie du répertoire CRAN, nous devrons l'installer directement depuis *github* avec la fonction `devtools::install_github`.

```
devtools::install_github('mattflor/chorddiag')
```

```
library(chorddiag)

# chargement des données
matriceOD <- read.csv('data/graphique/Quebec_2017_OD_MJ.csv',
                      header = FALSE, sep = ';') # fichier csv sans entête

# transformation du dataframe en matrice
matriceOD <- as.matrix(matriceOD)
codes <- c('A','B','C','D','E','F','G','H','I','J','K','X')
secteurs <- c('Arr. de Beauport',
             'Arr. de Charlesbourg',
             'Arr. des Rivières',
             'Arr. de la Cité-Limoilou',
             'Arr. de la Haute-St-Charles',
             'Arr. de Sainte-Foy-Sillery-Cap-Rouge',
             'Arr.de Desjardins',
             'Arr. des Chutes-de-la-Chaudière-Est',
             'Arr. Les Chutes de la-Chaudière-Ouest',
             'Ceinture Nord',
             'Ceinture Sud',
             'Hors Territoire')

# ajout de noms aux colonnes et aux lignes de la matrice
rownames(matriceOD) <- secteurs
colnames(matriceOD) <- secteurs

#on supprime les trois secteurs Ceinture Nord, Sud et Hors territoire
#qui comprennent de toute façon peu de déplacements
mat <- matriceOD[1:8,1:8]

# choix aléatoire de couleurs pour les lignes
# col <- sample(colors(),nrow(mat),replace = F)

# choix de couleurs
col <- c("#a491d3", "#818aa3", "#C5DCA0", "#F5F2B8",
        "#F9DAD0", "#F45B69", "#22181C", "#5A0001")

# réalisation du graphique : sortie html
if(knitr:::is_html_output()){
  chorddiag(mat, groupColors = col, showTicks = F,
             type = 'bipartite', chordedgeColor = 'white',
             groupnameFontSize = 12, groupnamePadding = 5)
```

```

}

# pour la sortie pdf
if(knitr:::is_latex_output()){
  knitr:::include_graphics('images/magie_graphiques/chord_diagramme.png', dpi = NA)
}

```

Le graphique permet de remarquer que la plupart des flux s'effectuent au sein d'un même secteur. La majorité des déplacements se font au sein du secteur Sainte-Foy (segment rouge central). On peut cependant constater que les secteurs des Rivières, la cité Limoilou et Haute-Saint-Charle attirent une plus grande quantité et diversité de flux. Si vous lisez ce livre dans un navigateur web (et pas au format *pdf*), le graphique est interactif! En plaçant votre souris sur un lien, vous verrez s'afficher le nombre de déplacements qu'il représente.

3.3.3 Nuage de mots

Un nuage de mots est un graphique utilisé en analyse de texte pour représenter les mots les plus importants d'un document. Mesurer l'importance des termes dans un document est une discipline à part entière (*Natural Language Processing*), nous proposons un simple exemple ici avec la méthode *TextRank* (basée sur la théorie des graphs) proposée par [Mihalcea and Tarau \(2004\)](#) et implémentée dans le package **textrank**. Nous aurons également besoin des packages **udpipe** (fournissant des dictionnaires linguistiques), **RColorBrewer** (pour sélectionner une palette de couleurs) et **wordcloud2** (pour générer le graphique). En guise d'exemple, nous avons choisi d'extraire les textes de deux Schémas d'Aménagement et de Développement (SAD), ceux des agglomérations de Québec et Montréal en vigueur en 2020. Il s'agit de deux documents de planification définissant les lignes directrices de l'organisation physique du territoire des municipalités régionales de comté (MRC) ou des agglomérations. Pour ces deux documents, nous nous concentrerons sur le chapitre portant sur les grandes orientations d'aménagement et de développement, soit les pages 30 à 135 pour Québec et 30 à 97 pour Montréal. Pour extraire les textes des fichiers *pdf*, nous utilisons le package **pdftools**.

Nous devons donc réaliser les étapes suivantes pour produire le nuage de mots :

1. Extraire les sections qui nous intéressent des fichiers *pdf*
2. Extraire le texte de ces sections
3. Retirer les caractères représentant les sauts de lignes et les sauts de paragraphes (\n et \r)
4. Concaténer tout le texte en une seule longue chaîne de caractère
5. Utiliser un dictionnaire pour déterminer la nature des mots du texte (noms, adjectifs, verbes, etc.)
6. Utiliser l'algorithme *TextRank* pour identifier les mots clefs
7. Nettoyer les erreurs potentielles parmi les mots clefs
8. Construire le nuage de mots.

Notez que toutes ces étapes de nettoyage ne seraient pas nécessaires si nous utilisions un simple fichier texte comme point de départ. Cependant, il est plus courant de rencontrer des fichiers *pdf*, cet exercice est donc davantage révélateur de la difficulté réelle de la réalisation d'un nuage de mots.

```

library(wordcloud2)
library(udpipe)
library(RColorBrewer)
library(pdftools)
library(textrank)

```

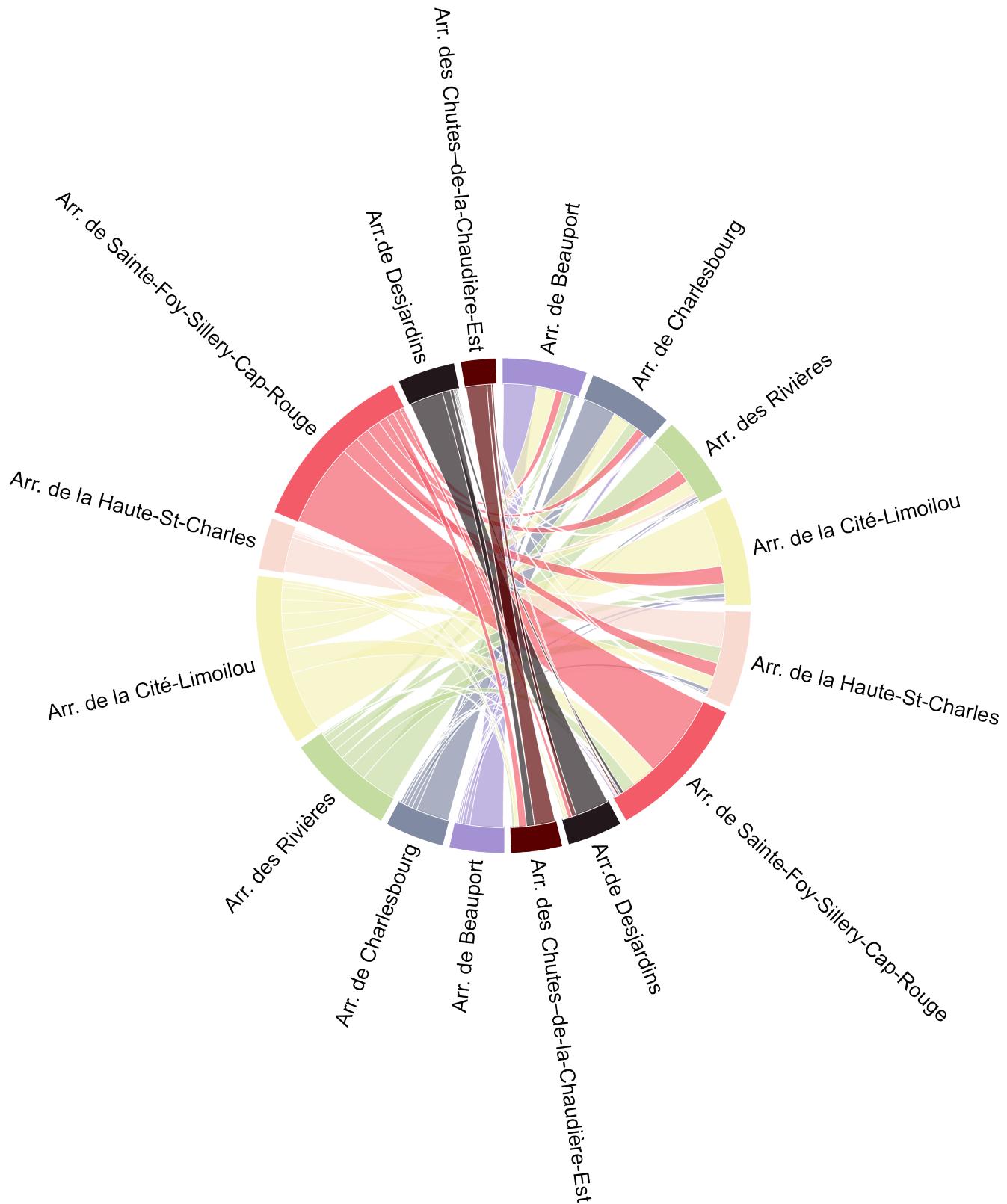


FIG. 3.54 : Diagramme d'accord

```

# Étape 1 : extraire les sections pertinentes des fichiers pdf
extrait_qc <- pdf_subset("data/graphique/SAD_quebec.pdf", pages = c(30:135),
                           output = "data/graphique/SAD_quebec_ext.pdf")
extrait_mtl <- pdf_subset("data/graphique/SAD_montreal.pdf", pages = c(30:97),
                           output = "data/graphique/SAD_montreal_ext.pdf")

# Étape 2 : extraire le texte des fichiers pdf sous forme de vecteur de texte
file_qc <- pdf_text(extrait_qc)
file_mtl <- pdf_text(extrait_mtl)

# Étape 3 : retirer les saut de lignes et les paragraphes
file_qc <- gsub("\r","",x = file_qc)
file_qc <- gsub("\n","",x = file_qc)

file_mtl <- gsub("\r","",x = file_mtl)
file_mtl <- gsub("\n","",x = file_mtl)

# Étape 4 : créer une seule longue chaîne de caractères
# à partir des vecteurs de texte
text_qc <- paste(file_qc, collapse = " ")
text_mtl <- paste(file_mtl, collapse = " ")

# charger le modèle linguistique français
model <- udpipe_load_model('data/graphique/french-sequoia-ud-2.4-190531.udpipe')

# pour télécharger le modèle si ce n'est pas encore fait :
# model <- udpipe_download_model("french-sequoia")
# model <- udpipe_load_model(model)

# Etape 5 : Analyse de la nature des mots du texte avec le dictionnaire fr
# On obtient des dataframes décrivant les mots des textes
annotate_qc <- udpipe_annotation(model, text_qc)
df_qc <- data.frame(annotation_qc)

annotate_mtl <- udpipe_annotation(model, text_mtl)
df_mtl <- data.frame(annotation_mtl)

# Etape 6 : Utilisation de la méthode TextRank
stats_qc <- textrank_keywords(df_qc$lemma,
                               relevant = df_qc$upos %in% c("NOUN", "ADJ"), ngram_max=2)

stats_mtl <- textrank_keywords(df_mtl$lemma,
                               relevant = df_mtl$upos %in% c("NOUN", "ADJ"), ngram_max=2)

# Etape 7 : Nettoyer les coquilles dans les mots clefs
# NB : nous faisons ici le choix de garder des mots clefs uniques (ngram == 1)
# il serait aussi possible de garder des associations de plusieurs mots
dfstats_qc <- subset(stats_qc$keywords, stats_qc$keywords$ngram == 1 &
                        nchar(stats_qc$keywords$keyword)>2)
dfstats_qc$keyword <- gsub("d'", "", dfstats_qc$keyword, fixed = T)
dfstats_qc$keyword <- gsub("l'", "", dfstats_qc$keyword, fixed = T)

dfstats_mtl <- subset(stats_mtl$keywords, stats_mtl$keywords$ngram == 1 &
                        nchar(stats_mtl$keywords$keyword)>2)
dfstats_mtl$keyword <- gsub("d'", "", dfstats_mtl$keyword, fixed = T)

```

```

dfstats_mtl$keyword <- gsub("l","",dfstats_mtl$keyword,fixed = T)

# Etape 8 : Réaliser les nuages de mots
couleurs <- sample(brewer.pal(12, "Paired")) # mise en désordre des couleurs

wordcloud2(data = dfstats_mtl[c("keyword", "freq")],
           color = couleurs, size = 0.5, shuffle = F)

wordcloud2(data = dfstats_qc[c("keyword", "freq")],
           color = couleurs, size = 0.6, shuffle = F)

```



FIG. 3.55 : Nuage de mots pour le SAD de Montréal

Notez qu'à chaque génération du nuage de mots, vous obtiendrez une disposition différente. N'hésitez pas à en essayer plusieurs jusqu'à trouver celle qui vous semble optimale.

3.3.4 Treemaps

Un *treemap* est un graphique permettant de représenter une quantité partagée entre plusieurs observations structurées dans une hiérarchie de groupe. Le jeu de données portant sur les émissions de CO₂ se prête tout à fait à une représentation par *treemaps*. La variable de quantité est bien sûr les émissions de CO₂ par pays; ces pays sont regroupés dans un premier ensemble de régions (déroulage en 23 régions), qui elles-mêmes sont regroupées dans des régions plus larges (déroulage en sept régions). Pour construire un *treemap*, nous allons utiliser le package **treemap**.



FIG. 3.56 : Nuage de mots pour le SAD de Québec

```

library(treemap)
library(RColorBrewer)

# extraire les données de CO2 en 2015
data_co2_2015 <- subset(data_co2,data_co2$year == "2015" & ! is.na(data_co2$region7))

# construire le treemap

treemap(data_co2_2015, index=c("region7","region23"),
        vSize="CO2_kt", type="index",
        title = "CO2 rejetés par pays en 2015",
        fontsize.labels=c(12,8), # taille des étiquettes
        fontcolor.labels=c("white","black"), # couleur des étiquettes
        fontface.labels=c(2,1), # style des polices
        bg.labels=c("transparent"), # arrière plan des étiquettes
        align.labels=list(
            c("center", "center"),
            c("right", "bottom")
        ), # localisation des étiquettes dans les boites
        overlap.labels=0.5, # tolérance de superposition
        inflate.labels=F, # agrandir la taille des étiquettes ou non
        palette = brewer.pal(7,'Paired')
)

```

CO₂ rejetés par pays en 2015

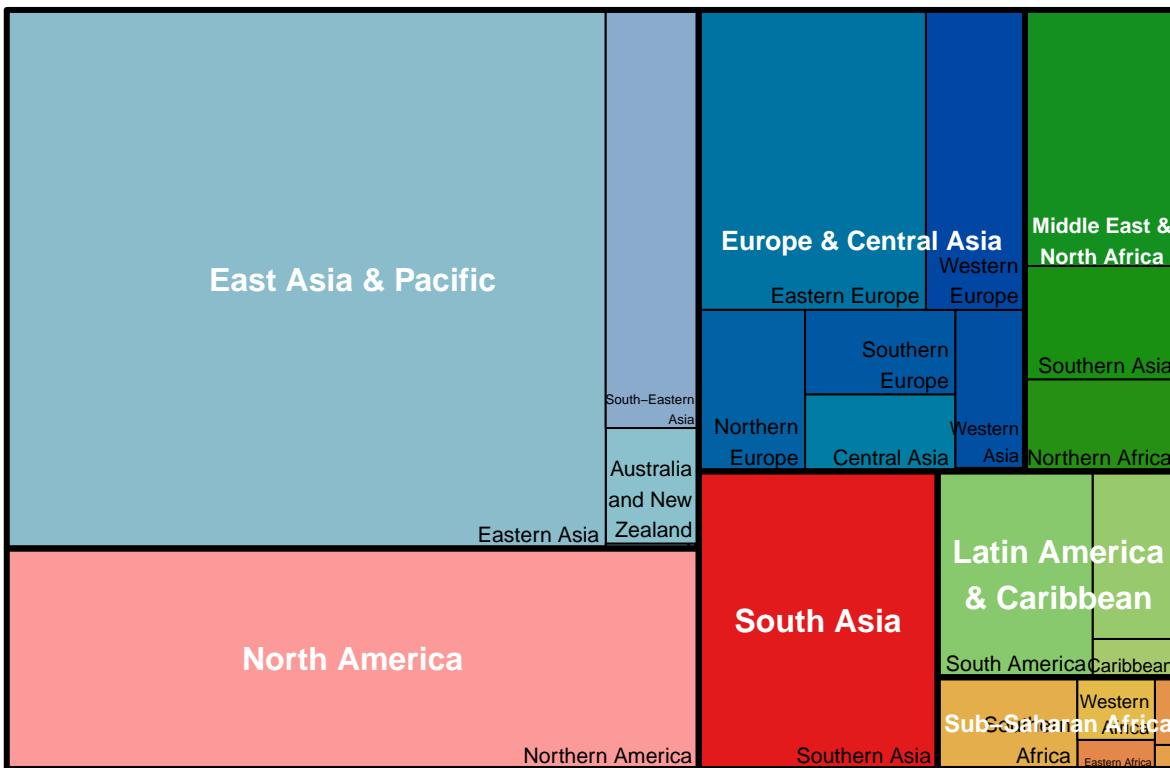


FIG. 3.57 : Treemap

3.4 Les cartes

Toute comme un graphique, une carte est aussi une illustration visuelle, avec la généralisation des données géographiques, il peut être utile de savoir représenter ce type de données. Si R n'est pas un logiciel de cartographie, il est possible de réaliser des cartes assez facilement, directement avec **ggplot2**. Nous avons cependant une préférence pour le package **tmap** qui propose de nombreuses fonctionnalités. Pour tracer des cartes, **tmap** et **ggplot2** ont besoin d'utiliser un format de données comprenant la géométrie (polygones, lignes ou points), la localisation et le système de projection des entités spatiales étudiées. Le format de fichier le plus courant pour ce type de données est le *shapefile* (.shp), mais vous pourrez parfois croiser des fichiers *geojson* (.js), ou encore *geopackages* (.gpkg). Pour lire ces fichiers, il est possible d'utiliser la fonction *readOGR* du package **rgdal**, ou la fonction *st_read* du package **sf**. Notez ici que ces deux fonctions ne produisent pas des *DataFrame*, mais respectivement un *SpatialDataFrame* et un objet *sf* (*simple feature collection*). Sans rentrer dans les détails, sachez que deux packages permettent de manipuler des objets spatiaux dans R : le traditionnel **sp** (avec les *SpatialDataFrame*) et le plus récent **sf** (avec les objets du même nom). Il est assez facile de convertir un objet de **sp** vers **sf** (et inversement) et cette opération est souvent nécessaire car de nombreux packages dédiés à l'analyse spatiale utilisent l'un ou l'autre des formats. Dans le cas de **tmap**, des objets de **sp** et de **sf** peuvent être utilisés sans distinction. En revanche, pour cartographier directement avec **ggplot2**, il est plus facile d'utiliser un objet de type **sf**.

Une carte thématique permet de représenter la répartition spatiale de variables qualitatives ou quantitatives. On les distingue des cartes topographiques dont l'objectif est de représenter la localisation d'objets spécifiques (route, habitation, rivière, lac, etc.). Les premières sont relativement faciles à construire dans R car elles se limitent à quelques symboles relativement peu complexes. Pour les

secondes, on préferera généralement un logiciel comme QGis⁶.

Créons une carte thématique à partir des données de densité de végétation sur l'Île de Montréal avec les packages **ggplot2** puis **tmap**.

Avec **ggplot2**, nous aurons aussi besoin des packages **classInt** pour calculer les intervalles des classes et de **ggsn** pour afficher une échelle.

```
library(sf)
library(classInt)
library(ggsn)

# chargement des données
spatialdf <- st_read("data/bivariee/IlotsVeg2006.shp")

## Reading layer 'IlotsVeg2006' from data source `G:\Projets\Philippe_apparicio\Livre Statistique\VersionGIT\`
## Simple feature collection with 10213 features and 12 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:            xmin: 267518.7 ymin: 5029292 xmax: 306663.7 ymax: 5062652
## projected CRS: NAD83 / MTM zone 8

# création d'une discréétisation en 7 classes égales
values <- c(max(spatialdf$ArbPct)+0.01, spatialdf$ArbPct)

quant <- classIntervals(values, n = 7,
                         style = "quantile",
                         intervalClosure = 'right')

spatialdf$class_col <- cut(spatialdf$ArbPct, breaks = quant$brks, right = F)

# cartographie avec ggplot2
ggplot(data = spatialdf) +
  geom_sf(aes(fill = class_col), color = rgb(0,0,0,0))+
  scale_fill_brewer(palette = "Greens")+
  labs(title = "Végétation dans les îles de recensement",
       'fill' = 'Densité de la canopée (%)')+
  theme(axis.line=element_blank(), axis.text.x=element_blank(),
        axis.text.y=element_blank(), axis.ticks=element_blank(),
        axis.title.x=element_blank(), axis.title.y=element_blank(),
        panel.background=element_blank(),
        panel.border=element_blank(), panel.grid.major=element_blank(),
        panel.grid.minor=element_blank(), plot.background=element_blank(),
        legend.key.size = unit(0.5, "cm"))+
  scalebar(spatialdf, dist = 5, st.size=3, height=0.01, model = 'WGS84',
           dist_unit = "km", transform = F, location = 'bottomright')
```

Il est possible d'arriver à un résultat similaire avec **tmap** avec moins de code!

```
library(tmap)
```

⁶<https://qgis.org/en/site/>

Végétation dans les îles de recensement

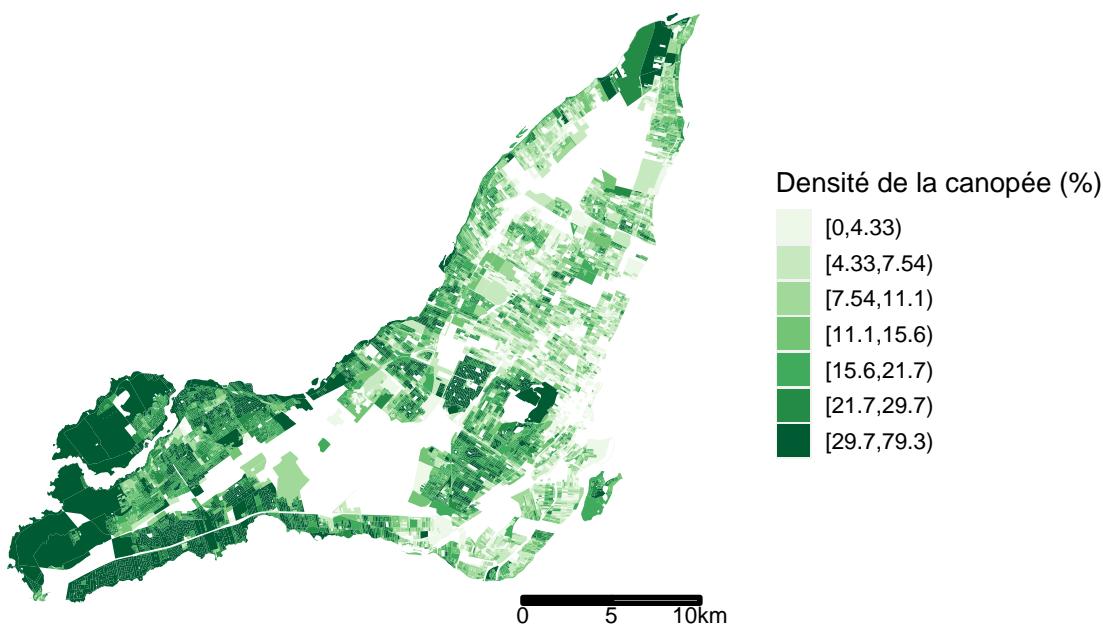


FIG. 3.58 : Carte thématique avec ggplot2

```
colors <- brewer.pal(7,"Greens")

tm_shape(spatialdf) +
  tm_polygons("ArbPct", palette = colors, border.alpha = 0,
              n = 7, style = 'quantile',
              title = 'Densité de la canopée (%)') +
  tm_scale_bar(breaks = c(0,5,10)) +
  tm_layout(title = "Végétation dans les îles de recensement",
            attr.outside = TRUE, frame = FALSE)
```

Les graphiques créés par **tmap** ne peuvent malheureusement pas être combinés avec la fonction **ggarrange**, mais **tmap** dispose de sa propre fonction **tmap_arrange** si vous souhaitez combiner plusieurs cartes.

```
library(tmap)

colors <- brewer.pal(7,"Greens")

colors2 <- brewer.pal(7,"Reds")

carte1 <- tm_shape(spatialdf) +
  tm_polygons("ArbPct", palette = colors, border.alpha = 0,
              n = 7, style = 'quantile',
              title = 'Densité de la canopée (%)') +
  tm_scale_bar(breaks = c(0,5,10)) +
  tm_layout(attr.outside = TRUE, frame = FALSE)

carte2 <- tm_shape(spatialdf) +
```

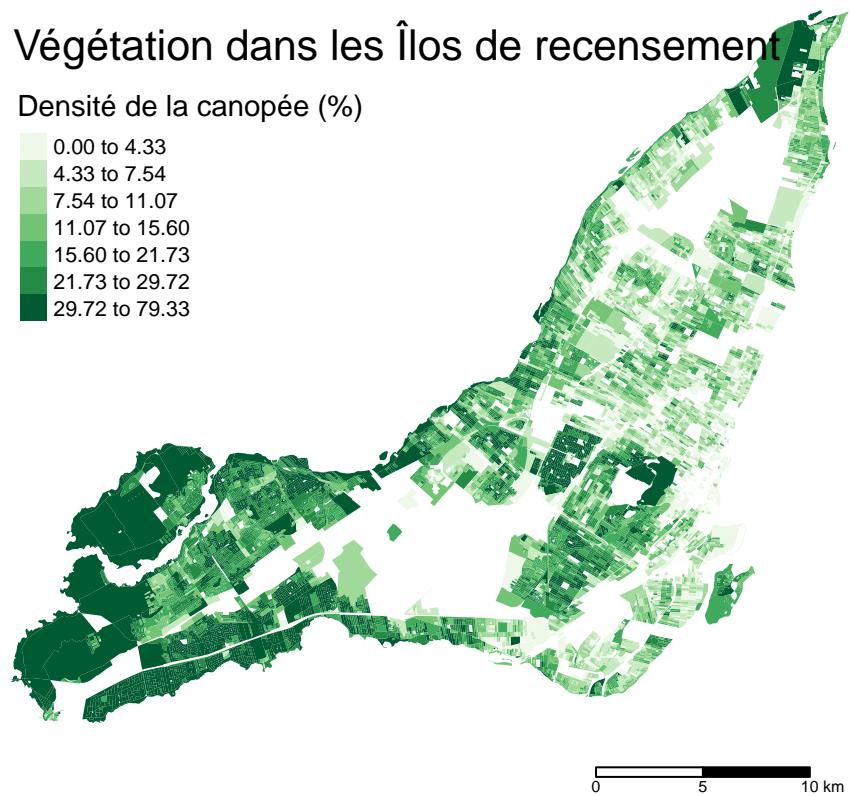


FIG. 3.59 : Carte thématique avec tmap

```
tm_polygons("LogDens", palette = colors2, border.alpha = 0,
            n = 7, style = 'quantile',
            title = 'Densité de logement') +
tm_scale_bar(breaks = c(0,5,10)) +
tm_layout(attr.outside = TRUE, frame = FALSE)

tmap_arrange(carte1, carte2, ncol = 2)
```

3.5 Exporter des graphiques

Tous les graphiques que nous avons construits dans ce chapitre peuvent être exportés assez facilement. Dans RStudio, vous pouvez directement cliquer sur le bouton *export* (figure 3.60)) pour enregistrer votre figure au format image (raster) ou au format PDF (vecteur). Notez qu'avec la seconde option, vous pourrez retoucher votre graphique avec un logiciel externe comme *Inkscape* ou *Illustrator*, ce qui est souvent nécessaire.

Lorsque vous créez un graphique avec **ggplot2**, il est aussi possible de l'exporter avec la fonction **ggsave**. Cette fonctionnalité est très pratique lorsque vous souhaitez automatiser la production de graphiques et ne pas avoir à tous les exporter à la main. Pour en apprendre plus sur l'automatisation de tâches dans R, référez-vous au chapitre XXX.

```
data(iris)

plot1 <- ggplot() +
```

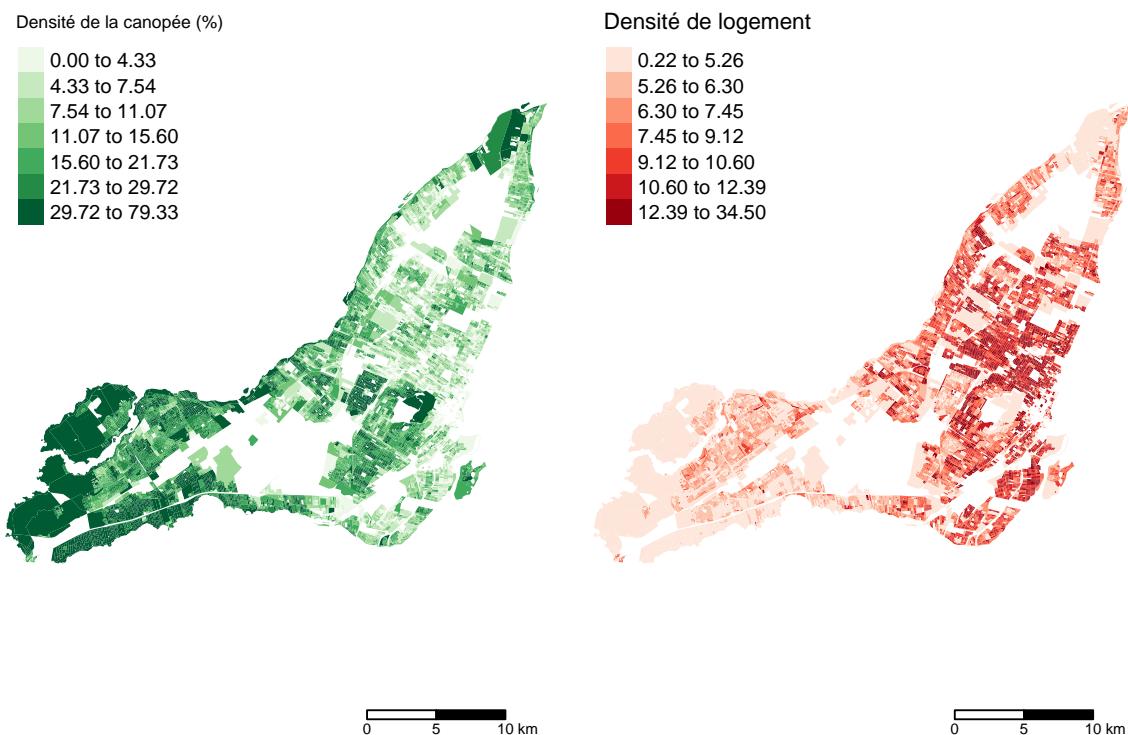


FIG. 3.60 : Combiner des cartes avec tmap



FIG. 3.61 : Exporter un graphique dans RStudio

```
geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris)

ggsave(filename = 'graphique.pdf',
       path = 'mon/dossier',
       plot = plot1,
       width = 10, height = 10, units = "cm")
```

Pour les graphiques n'étant pas réalisés avec **ggplot2**, l'alternative à la fonction **ggsave** est l'ensemble de fonctions **png**, **bmp**, **jpeg**, **tiff** et **pdf**, qui permettent d'exporter n'importe quel graphique dans ces différents formats. Le processus comprend trois étapes :

1. Ouvrir une connexion vers le fichier dans lequel le graphique sera exporté avec une des fonctions **png**, **bmp**, **jpeg**, **tiff** et **pdf**.
2. Réaliser son graphique comme si on souhaitait l'afficher dans RStudio. Il n'apparaîtra cependant pas, car il sera écrit dans le fichier en question à la place.
3. Fermer la connexion au fichier avec la fonction **dev.off** pour définitivement enregistrer le graphique.

```
data(iris)

# 1. Ouvrir la connexion
```

```
png(filename = 'mon/dossier/graphique.png')

# 2. Afficher le graphique
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris)

# 3. fermer la connexion
dev.off()
```

3.6 Conclusion sur les graphiques

Vous avez pu constater que les capacités de représentation graphique de R sont vastes et pourtant nous n'avons fait qu'observer la partie émergée de l'iceberg dans ce chapitre. Il est également possible de réaliser de la visualisation en 3D dans R (`plot3D`, `rgl`), d'animer des graphiques pour en faire des *GIF* ou des vidéos (`ganimate`), de rendre des graphiques interactifs, ou même de construire des plateformes de visualisation de données disponibles en ligne (`shiny`). Vous continuerez à découvrir de nouvelles formes de représentations au fur et à mesure de votre pratique, en apprenant de nouvelles méthodes nécessitant des visualisations spécifiques.

Voici également deux références très utiles qui nous ont notamment aidé à construire ce chapitre :

- The R Graph Gallery⁷, probablement LE site web proposant le plus de matériel sur comment réaliser des graphiques dans R.
- Data to viz⁸, si vous ne savez pas quel graphique pourrait le mieux correspondre à vos données, Data to viz est là pour vous aider. Vous y trouverez un arbre de décision pour vous indiquer quel graphique utiliser dans quelle situation, ainsi que de nombreux conseils sur la visualisation de données.

⁷<https://www.r-graph-gallery.com/>

⁸<https://www.data-to-viz.com/>

Troisième partie

Analyses bivariées

Chapitre 4

Analyses bivariées

Dans ce chapitre, nous présentons les principales méthodes exploratoires et confirmatoires bivariées permettant d'évaluer la relation entre deux variables, et ce, en fonction de leur type : deux variables quantitatives, deux variables qualitatives ou encore une variable quantitative *versus* une variable qualitative (comportant deux modalités ou plus de deux modalités) (figure 4.1).

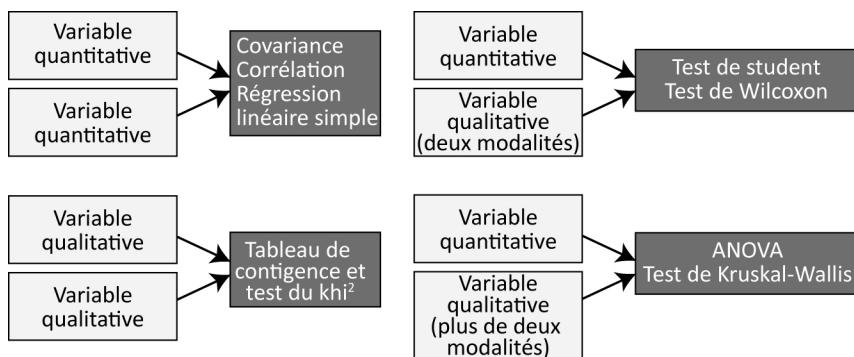


FIG. 4.1 : Les principales méthodes bivariées

Plus spécifiquement, nous présenterons puis mettrons en œuvre dans le logiciel les méthodes suivantes : covariance, corrélation et régression linéaire simple (entre deux variables quantitatives, section 4.1), tableau de contingence et test du χ^2 (entre deux variables qualitatives, section 4.2), t de student (test *t*) et test de Wilcoxon (entre une variable quantitative et une variable qualitative comportant deux modalités, section 4.3), et analyse de variance et test de Kruskal-Wallis (entre une variable quantitative et une variable qualitative comportant plus de deux modalités, section 4.4).



Dans cette section, nous utiliserons principalement les *packages* suivants :

- Pour créer des graphiques :
 - * **ggplot2**, le seul, l'unique
 - * **ggsignif**, pour combiner des graphiques et réaliser des diagrammes quantiles-quantiles
- Pour manipuler des données :
 - * **dplyr**, avec les fonctions *group_by*, *summarize* et les pipes `%>%`
- Pour les corrélations (section 4.1.1) :
 - * **correlation**, de l'ensemble de package **easy_stats**, offrant une large gamme de méthodes de corrélations
 - * **boot** pour réaliser des corrélations avec *bootstrap*
 - * **Hmisc** pour calculer des corrélations de Pearson et Spearman

- * **ppcor**, notamment pour des corrélations partielles
- * **psych** pour obtenir une matrice de corrélation (Pearson, Spearman et Kendall), les intervalles de confiance et les valeurs de p.
- * **stargazer** pour créer des beaux tableaux d'une matrice de corrélation en Html ou en LaTeX ou en ASCII.
- * **corrplot**, pour créer des graphiques de matrices de corrélation
- Pour le tableau de contingence (section 4.1.2) :
 - * **gmodels**, pour construire des tableaux de contingence et calculer les tests t et ses différentes variantes (section 4.2.4)
 - * **vcg**, pour construire un graphique pour un tableau de contingence ((section 4.2.4))
- Pour les test t :
 - * **sjstats** pour réaliser des test t pondérés
 - * **effectsize**, pour calculer les tailles d'effet de tests de t
- Pour la section sur les ANOVA (section 4.4.1) :
 - * **car**, pour les ANOVA classiques
 - * **lmtest** pour le test de Breusch-Pagan d'homogénéité des variances
 - * **rstatix**, intégrant de nombreux tests classiques (comme le test de Shapiro) avec **tidyverse**

4.1 Relation linéaire entre deux variables quantitatives



Deux variables continues varient-elles dans le même sens ou bien en sens contraire? Répondre à cette question est une démarche exploratoire classique en sciences sociales puisque les données socioéconomiques sont souvent associées linéairement. En d'autres termes, lorsque l'une des deux variables tant à augmenter, la seconde augmente également ou diminue systématiquement.

En études urbaines, on pourrait vouloir vérifier si certaines variables socioéconomiques sont associées positivement ou négativement à des variables environnementales jugées positives (comme la couverture végétale ou des mesures d'accessibilité spatiale aux parcs) ou négatives (pollutions atmosphériques et sonores).

Par exemple, au niveau des secteurs de recensement d'une ville canadienne ou américaine, on pourrait vouloir vérifier si le revenu médian des ménages ou encore le coût moyen du loyer varient dans le même sens que la couverture végétale; ou au contraire, en sens inverse des niveaux moyens de dioxyde d'azote ou de bruit routier.

Pour évaluer la linéarité entre deux variables continues, deux statistiques descriptives sont utilisées : la **covariance** (section 4.1.2) et la **corrélation** (section 4.1.3).

4.1.1 Bref retour sur le postulat de la relation linéaire

Vérifier le postulat de la linéarité consiste à évaluer si deux variables quantitatives varient dans le même sens ou bien en sens contraire. Toutefois, la relation entre deux variables quantitatives n'est pas forcément linéaire. En guise d'illustration, la figure 4.2 permet de distinguer quatre types de relations :

- le cas **a** illustre une relation linéaire positive entre les deux variables puisqu'elles vont dans le même sens. Autrement dit, quand les valeurs de X augmentent, celles de Y augmentent aussi. En guise d'exemple, pour les secteurs de recensement d'une métropole donnée, il est fort probable que le coût moyen du loyer soit associé positivement avec le revenu médian des ménages. Graphiquement parlant, il est clair qu'une droite dans ce nuage de points résumerait efficacement la relation entre ces deux variables.
- le cas **b** illustre une relation linéaire négative entre les deux variables puisqu'elles vont en sens

inverse. Autrement dit, quand les valeurs de X augmentent, celles de Y diminuent, et inversement. En guise d'exemple, pour les secteurs de recensement d'une métropole donnée, il est fort probable que le revenu médian des ménages soit associé négativement avec le taux de chômage. De nouveau, une droite résumerait efficacement cette relation.

- pour le cas **c**, il y a une relation entre les deux variables, mais qui n'est pas linéaire. Le nuage de points entre les deux variables prend d'ailleurs une forme parabolique qui traduit une relation curvilinéaire. Concrètement, on observe une relation positive jusqu'à un certain seuil, puis une relation négative.
- pour le cas **d**, la relation entre les deux variables est aussi curvilinéaire; d'abord négative, puis positive.

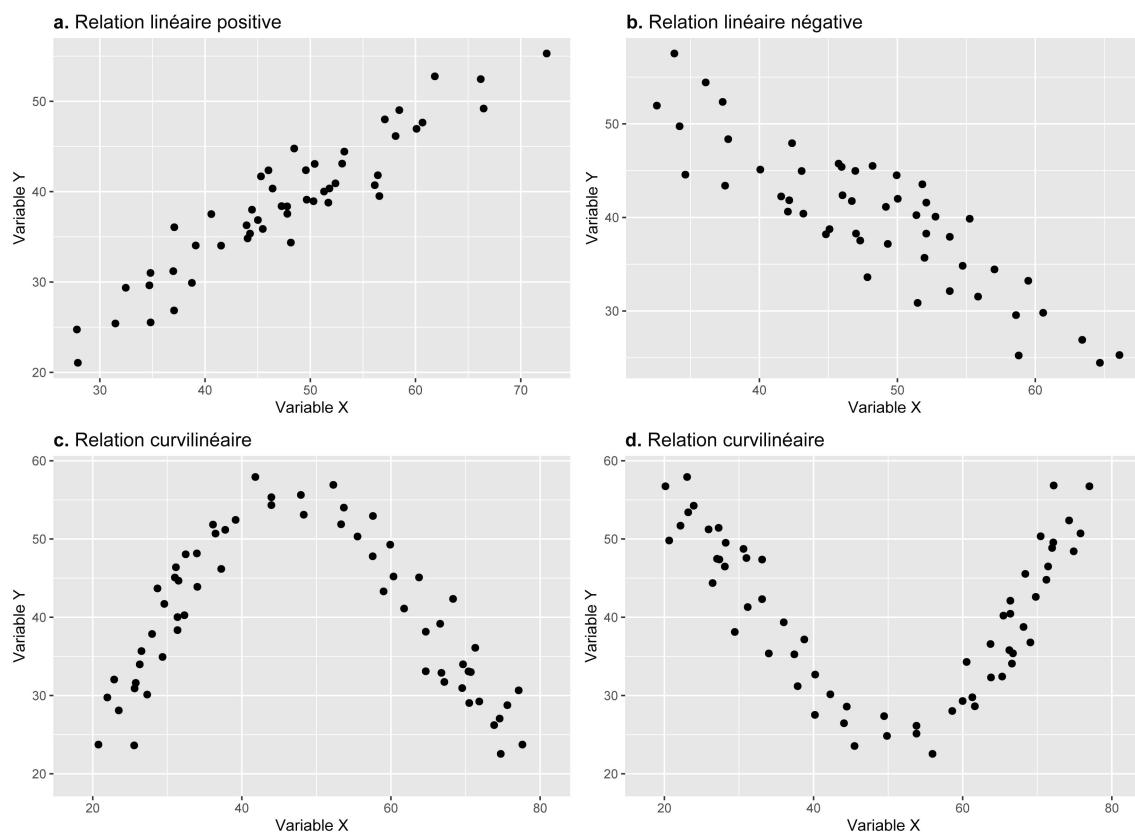


FIG. 4.2 : Relations linéaires et curvilinéaires entre deux variables continues

Prenons un exemple concret. Dans une étude portant sur l'équité environnementale et la végétation à Montréal, Pham *et al.* (2012) ont montré qu'il existe une relation curvilinéaire entre l'âge médian des bâtiments résidentiels (axe des abscisses) et les couvertures végétales (axes des ordonnées) :

- la couverture de la végétation totale et celle des arbres augmentent quand l'âge médian des bâtiments croît jusqu'à atteindre un pic autour de 60 ans (autour de 1950). On peut supposer que les secteurs récemment construits, surtout ceux dans les banlieues, présentent des niveaux de végétation plus faibles. Au fur et au fur que le quartier vieillit, les arbres plantés lors du développement résidentiel deviennent matures — canopée plus importante —, d'où l'augmentation des valeurs de la couverture végétale totale et de celle des arbres.
- Par contre, dans les secteurs développés avant les années 1950, la densité du bâti est plus forte, laissant ainsi moins de place pour la végétation, ce qui explique une diminution des variables relatives à la couverture végétale (figure 4.3).

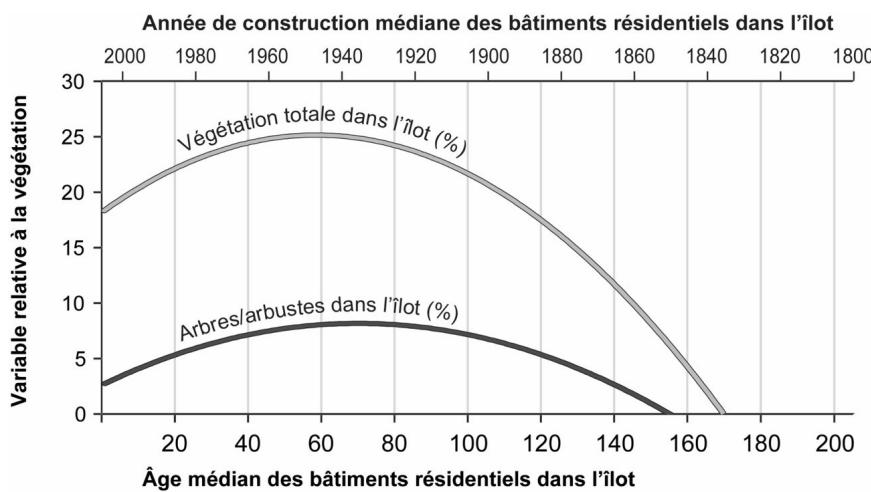


FIG. 4.3 : Exemples de relations curvilinéaires

Dans les sous-sections suivantes, nous décrirons deux statistiques descriptives et exploratoires – la covariance (section 4.1.2) et la corrélation (section 4.1.3) – utilisées pour évaluer la **relation linéaire** entre deux variables continues. Ces deux mesures permettent de mesurer le degré d'association entre deux variables, sans que l'une soit la variable dépendante (variable à expliquer) et l'autre, la variable dépendante (variable explicative). Puis, nous décrirons la régression linéaire simple (section 4.1.4) qui permet justement de prédire une variable dépendante (Y) à partir d'une variable indépendante (X).

4.1.2 Covariance

4.1.2.1 Formulation

La covariance (eq. (4.1)), écrite $cov(x, y)$, est égale à la moyenne du produit des écarts des valeurs des deux variables par rapport à leurs moyennes respectives :

$$cov(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1} = \frac{covariation}{n - 1} \quad (4.1)$$

avec n étant le nombre d'observations; \bar{x} et \bar{y} (prononcez x et y barre) étant les moyennes respectives des variables X et Y .

4.1.2.2 Interprétation

Le numérateur de l'équation (4.1) représente la covariation, soit la somme du produit des déviations des valeurs x_i et y_i par rapport à leurs moyennes respectives (\bar{x} et \bar{y}). La covariance est donc la covariation divisée par le nombre d'observations, soit la moyenne de la covariation. Sa valeur peut être positive ou négative :

- positive quand les deux variables varient dans le même sens, c'est-à-dire que lorsque les valeurs de la variable X s'éloignent de la moyenne, les valeurs de Y s'éloignent aussi dans le même sens; et négative pour une situation inverse.
- Quand la covariance est égale à 0, il n'y a pas de relation entre les variables X et Y . Plus sa valeur absolue est élevée, plus la relation entre les deux variables X et Y est importante.

Ainsi, la covariance correspond à un centrage des variables, c'est-à-dire à soustraire à chaque valeur de la variable sa moyenne correspondante. L'inconvénient majeur de l'utilisation de la covariance est qu'elle

est tributaire des unités de mesure des deux variables. Par exemple, si nous calculons la covariance entre le pourcentage de personnes à faible revenu et la densité de population (habitants au km²) au niveau des secteurs de recensement de la région métropolitaine de Montréal, nous obtenons une valeur de covariance de 34934. En revanche, si la densité de population est exprimée en milliers d'habitants au km², la valeur de la covariance sera de 34,934, alors que la relation linéaire entre les deux variables reste la même tel qu'illustre à la figure 4.4. Pour remédier à ce problème, on privilégie l'utilisation du coefficient de corrélation.

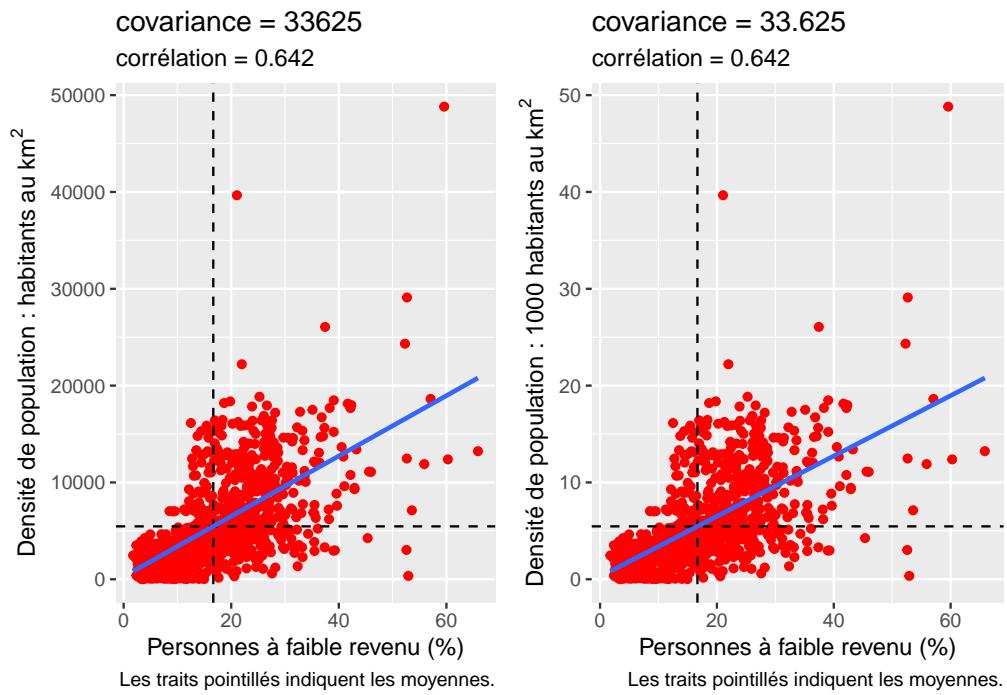


FIG. 4.4 : Covariance et unités de mesure

4.1.3 Corrélation

4.1.3.1 Formulation

Le coefficient de corrélation de Pearson (r) est égal à la covariance (numérateur) divisée par le produit des écart-types des deux variables X et Y (dénominateur). Il représente une standardisation de la covariance. Autrement dit, le coefficient de corrélation repose sur un centrage (moyenne = 0) et une réduction (variance = 1) des deux variables, c'est-à-dire à soustraire à chaque valeur sa moyenne correspondante et à la diviser par son écart-type. Il correspond ainsi à la moyenne du produit des deux variables centrées réduites. Il s'écrit alors :

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2(y_i - \bar{y})^2}} = \sum_{i=1}^n \frac{Z_x Z_y}{n-1} \quad (4.2)$$

La syntaxe ci-dessous démontre que le coefficient de corrélation de Pearson est bien égal à la moyenne du produit de deux variables centrées-réduites.

```
library("MASS")
N <- 1000      # nombre d'observations
```

```

moy_x <- 50 # moyenne de x
moy_y <- 40 # moyenne de y
sd_x <- 10 # écart-type de x
sd_y <- 8 # écart-type de y
rxy <- .80 # corrélation entre X et Y
## création de deux variables fictives normalement distribuées et corrélées entre elles
# Création d'une matrice de covariance
cov <- matrix(c(sd_x^2, rxy*sd_x*sd_y, rxy*sd_x*sd_y, sd_y^2), nrow=2)
# Création du tableau de données avec deux variables
df <- as.data.frame(mvrnorm(N, c(moy_x, moy_y), cov))
# Centrage et réduction des deux variables
df$zV1 <- scale(df$V1, center = TRUE, scale = TRUE)
df$zV2 <- scale(df$V2, center = TRUE, scale = TRUE)
# Corrélation de Pearson
cor1 <- cor(df$V1, df$V2)
# Moyenne du produit des variables centrées-réduites
cor2 <- sum(df$zV1*df$zV2) / (nrow(df)-1)
cat("Corrélation de Pearson = ", round(cor1,5),
    "\nMoyenne du produit des variables centrées-réduites =", round(cor2,5))

## Corrélation de Pearson = 0.8067
## Moyenne du produit des variables centrées-réduites = 0.8067

```

4.1.3.2 Interprétation

Le coefficient de corrélation r varie de -1 à 1 avec :

- 0 quand il n'y a pas de relation linéaire entre les variables X et Y
- -1 quand il y a une relation linéaire négative parfaite
- et 1 quand il y a une relation linéaire positive parfaite.

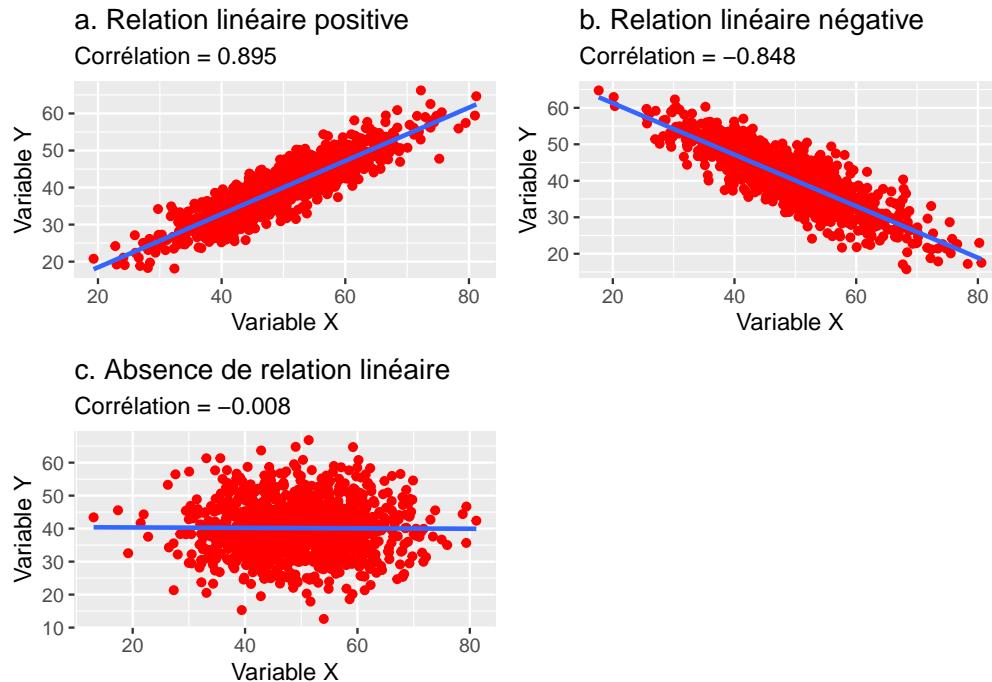


FIG. 4.5 : Relations entre deux variables continues et coefficients de corrélation de Pearson

Concrètement, le signe du coefficient de corrélation indique si la relation est positive ou négative et la valeur absolue du coefficient indique le degré d'association entre les deux variables. Reste à savoir comment déterminer qu'une valeur de corrélation est faible, moyenne ou forte. En sciences sociales, on utilise habituellement les intervalles de valeurs reportées au tableau 4.1. Toutefois, ces seuils sont tout à fait arbitraires. En effet, dépendamment de la discipline de recherche (sciences sociales, sciences de la santé, sciences physiques, etc.), et des variables à l'étude, l'interprétation d'une valeur de corrélation peut varier. Par exemple, en sciences sociales, une valeur de corrélation de 0,2 sera considérée comme très faible alors qu'en sciences de la santé, elle pourrait être considérée comme intéressante. À l'opposé, une valeur de 0,9 en sciences physiques pourrait être considérée comme faible. Il convient alors d'utiliser ces intervalles avec précaution.

Le coefficient de corrélation mis au carré représente le coefficient de détermination et indique la proportion de la variance de la variable Y expliquée par la variable X et inversement. Par exemple, un coefficient de corrélation de -0,70 signale que 49% de la variance de la variable de Y est expliquée par X (figure 4.6).

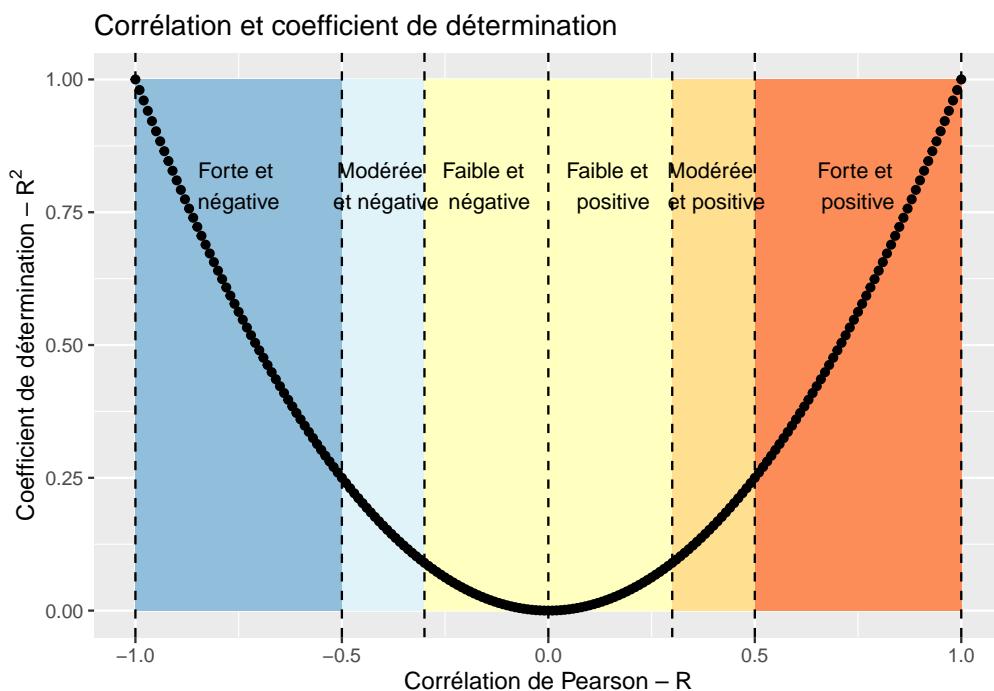


FIG. 4.6 : Coefficient de corrélation et proportion de la variance expliquée

Condition d'application. L'utilisation du coefficient de corrélation de Pearson nécessite que les deux variables continues soient normalement distribuées et qu'elles ne comprennent pas de valeurs aberrantes (extrêmes). D'ailleurs, plus le nombre d'observations sera réduit, plus la présence de valeurs aberrantes aura un impact important sur le résultat du coefficient de corrélation de Pearson. En guise d'exemple, dans le nuage de points à gauche de la figure 4.7, il est possible d'identifier des valeurs extrêmes qui

TAB. 4.1 : Intervalles pour l'interprétation du coefficient de corrélation habituellement utilisés en sciences sociales

Corrélation	Négative	Positive
Faible	de -0,3 à 0,0	de 0,0 à 0,3
Moyenne	de -0,5 à -0,3	de 0,3 à 0,5
Forte	de -1,0 à -0,5	de 0,5 à 1,0

se démarquent nettement dans le jeu de données : six observations avec une densité de population supérieure à 20 000 habitants au km² et deux observations avec un pourcentage de 65 ans et plus supérieur à 55%. Si l'on supprime ces observations (ce qui est défendable dans ce contexte) – soit moins d'un pourcent des observations du jeu de données initial –, la valeur du coefficient de corrélation passe de -0,158 à -0,194, signalant une augmentation du degré d'association entre les deux variables.

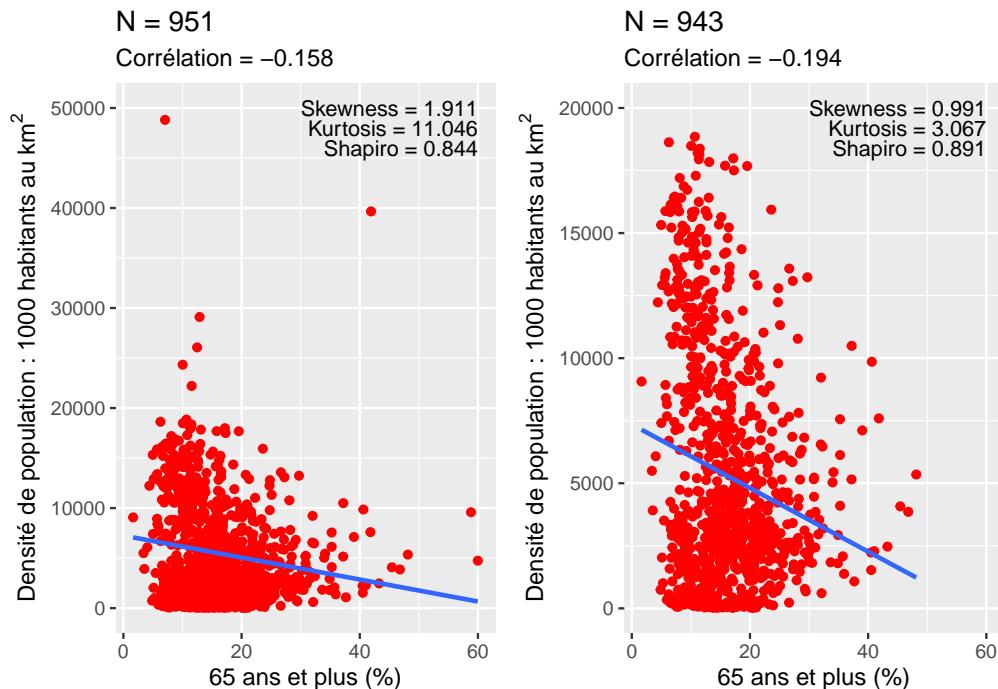


FIG. 4.7 : Illustration de l'effet des valeurs extrêmes sur le coefficient de Pearson

4.1.3.3 Corrélations pour des variables anormalement distribuées (coefficients de Spearman, Tau de kendall)

Lorsque les variables sont fortement anormalement distribuées, le coefficient de corrélation de Pearson est peu adapté pour analyser leurs relations linéaires. Il est alors conseillé d'utiliser deux statistiques non-paramétriques : principalement, le coefficient de corrélation de Spearman (*rho*) et secondairement, le coefficient de Kendall (τ , prononcez Tau), qui varient aussi tous deux de -1 à 1. Calculé sur les rangs des deux variables, le **coefficient de Spearman** est le rapport entre la covariance des deux variables de rangs sur les écart-types des variables de rangs. En d'autres termes, il représente simplement le coefficient de Pearson calculé sur les rangs des deux variables :

$$r_{xy} = \frac{\text{cov}(rg_x, rg_y)}{\sigma_{rg_x} \sigma_{rg_y}} \quad (4.3)$$

La syntaxe ci-dessous démontre clairement que le coefficient de Spearman est bien le coefficient de Pearson calculé sur les rangs (4.1.3.1).

```
df <- read.csv("data/bivariee/sr_rmr_mtl_2016.csv")
# Transformation des deux variables en rangs
df$HabKm2_rang <- rank(df$HabKm2)
df$A65plus_rang <- rank(df$A65plus)
# Coefficient de Spearman avec la fonction cor et la méthode spearman
```

```

cat("Coefficient de Spearman = ",
  round(cor(df$HabKm2, df$A65plus, method = "spearman"),5))

## Coefficient de Spearman = -0.11953

# Coefficient de Pearson sur les variables transformées en rangs
cat("Coefficient de Pearson calculé sur les variables transformées en rangs = ",
  round(cor(df$HabKm2_rang, df$A65plus_rang, method = "pearson"),5))

## Coefficient de Pearson calculé sur les variables transformées en rangs = -0.11953

# Vérification avec l'équation
cat("Covariance divisée par le produit des écart-types sur les rangs :",
  round(cov(df$HabKm2_rang, df$A65plus_rang) / (sd(df$HabKm2_rang)*sd(df$A65plus_rang)),5))

## Covariance divisée par le produit des écart-types sur les rangs : -0.11953

```

Le **coefficient de Kendall** est une autre mesure non-paramétrique calculée comme suit :

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n - 1)} \quad (4.4)$$

avec n_c et n_d qui sont respectivement les nombres de paires d'observations concordantes et discordantes ; et le dénominateur étant le nombre totale de paires d'observations. Des paires sont dites corcordantes quand les valeurs des deux observations vont dans le même sens pour les deux variables ($x_i > x_j$ et $y_i > y_j$ ou $x_i < x_j$ et $y_i < y_j$), et discordantes quand elles vont en sens contraire ($x_i > x_j$ et $y_i < y_j$ ou $x_i < x_j$ et $y_i > y_j$). Contrairement au calcul du coefficient de Spearman, celui de Kendall peut être chronophage : plus le nombre d'observations sera élevé, plus les temps de calcul et la mémoire utilisée sont importants. En effet, avec $n=1000$, le nombre de paires d'observations ($0.5 * n(n - 1)$) sera de 499500, contre près de 50 millions avec $n=10000$ (49 995 000).

À la lecture des deux histogrammes ci-dessus, il est clair que les deux variables *densité de population* et *pourcentage de personnes ayant 65 ou plus* sont très anormalement distribuées. Dans ce contexte, l'utilisation du coefficient de Pearson peut nous amener à mésestimer la relation existante entre les deux variables. Notez que les coefficients de Spearman et de Kendall sont tous les deux plus faibles.

4.1.3.4 Corrélations robustes (*Biweight midcorrelation*, *Percentage bend correlation* et la corrélation *pi* de Shepherd)

Dans l'exemple donné à la figure 4.7, nous avions identifié des valeurs aberrantes que nous avons retirées du jeu de données. Cette pratique peut tout à fait se justifier quand les données sont erronées (un capteur de pollution renvoyant une valeur négative, un questionnaire rempli par un mauvais plaisantin, etc.), mais parfois, les cas extrêmes font partie du phénomène à analyser. Dans ce contexte, les identifier et les retirer peut paraître arbitraire. Une solution plus élégante est d'utiliser des méthodes dites **robustes**, c'est à dire moins sensibles aux valeurs extrêmes. Pour les corrélations, la *Biweight midcorrelation* (Wilcox, 1994) est au coefficient de Pearson ce que la médiane est à la moyenne. Il est donc pertinent de l'utiliser dans des jeux de données présentant potentiellement des valeurs extrêmes. Elle est calculée comme suit :

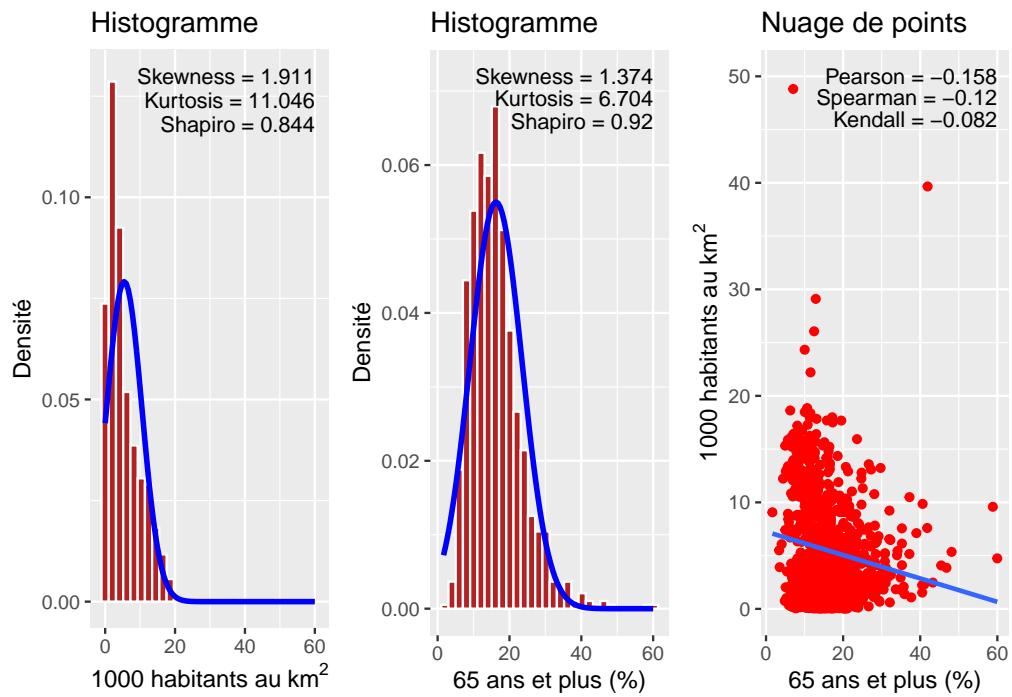


FIG. 4.8 : Comparaison des coefficients de Pearson, Spearman et Kendall sur deux variables anormalement distribuées

$$\begin{aligned}
 u_i &= \frac{x_i - \text{med}(x)}{9 * (\text{med}(|x_i - \text{med}(x)|))} \text{ et } v_i = \frac{y_i - \text{med}(y)}{9 * (\text{med}(|y_i - \text{med}(y)|))} \\
 w_i^{(x)} &= (1 - u_i^2)^2 I(1 - |u_i|) \text{ et } w_i^{(y)} = (1 - v_i^2)^2 I(1 - |v_i|) \\
 I(x) &= \begin{cases} 1, \text{ si } x = 1 \\ 0, \text{ sinon} \end{cases} \\
 \tilde{x}_i &= \frac{(x_i - \text{med}(x))w_i^{(x)}}{\sqrt{(\sum_{j=1}^m [(x_j - \text{med}(x))w_j^{(x)}]^2}}} \text{ et } \tilde{y}_i = \frac{(y_i - \text{med}(y))w_i^{(y)}}{\sqrt{(\sum_{j=1}^m [(y_j - \text{med}(y))w_j^{(y)}]^2)}} \\
 \text{bicor}(x, y) &= \sum_{i=1}^m \tilde{x}_i \tilde{y}_i
 \end{aligned} \tag{4.5}$$

Comme le souligne l'équation (4.5), la *Biweight midcorrelation* est basée sur les écarts à la médiane, plutôt que sur les écarts à la moyenne.

Assez proche de la *Biweight midcorrelation*, la *Percentage bend correlation* se base également sur la médiane des variables X et Y. Le principe général est de donner un poids plus faible dans le calcul de cette corrélation à un certain pourcentage des observations (20% est généralement recommandé) dont la valeur est éloignée de la médiane. Pour une description complète de la méthode, vous pourrez lire l'article de Wilcox (1994).

Enfin, une autre option est l'utilisation de la corrélation *pi* de Sherphred (Schwarzkopf et al., 2012). Il s'agit simplement d'une méthode en deux étapes. Premièrement, les valeurs abberantes sont identifiées à l'aide d'une approche par *bootstrap* utilisant la distance de Mahalanobis (calculant les écarts multivariés entre les observations). Ensuite, le coefficient de *Spearman* est calculé sur les observations restantes.

Appliquons ces corrélations aux données précédentes. Notez que ce simple code d'une dizaine de lignes permet d'explorer rapidement la corrélation entre deux variables selon six mesures de corrélations.

```

library("correlation")
df1 <- read.csv("data/bivariee/sr_rmr_mtl_2016.csv")
methods <- c("pearson","spearman","biweight","percentage","shepherd")
rs <- lapply(methods,function(m){
  test <- correlation::cor_test(data = df1, x="Hab1000Km2",y="A65plus",method = m, ci=0.95)
  return(c(test$r,test$CI_low, test$CI_high))
})
dfCorr <- data.frame(do.call(rbind,rs))
names(dfCorr) <- c("r","IC_05","CI_95")
dfCorr$method <- methods

show_table(dfCorr,
  digits = 2,
  caption = 'Comparaison de différentes corrélations pour les variables densité de population et pourcentage de personnes ayant 65 ans et plus',
  col.names=c("r","IC 5%","IC 95%", "Méthode")
)

```

Il est intéressant de mentionner que ces trois corrélations sont rarement utilisées malgré leur pertinence dans de nombreux cas d'application. Nous faisons face ici à un cercle vicieux dans la recherche : les méthodes les plus connues sont les plus utilisées car elles sont plus facilement acceptées par les autres chercheurs. Des méthodes plus élaborées nécessitent davantage de justification et de discussion, ce qui peut conduire à de multiples sessions de corrections/resoumissions pour qu'un article soit accepté, malgré le fait qu'elles puissent être plus adaptées au jeu de données à l'étude.

4.1.3.5 Significativité des coefficients de corrélation

Quelle que soit la méthode utilisée, il convient de vérifier si le coefficient de corrélation est ou non statistiquement différent de 0. En effet, nous travaillons la plupart du temps avec des données d'échantillonage, et très rarement avec des populations complètes. En collectant un nouvel échantillon, aurions-nous obtenu des résultats différents ? Le calcul de ce degré de significativité nous permet de quantifier notre niveau de certitude quant à l'existence d'une corrélation entre nos deux variables, positive ou négative. Cet objectif est réalisé en calculant la valeur de t et le nombre de degrés de liberté : $t = \sqrt{\frac{n-2}{1-r^2}}$ et $dl = n - 2$ avec r et n étant le coefficient de corrélation et le nombre d'observations. De manière classique, on utilisera la table des valeurs critiques de la distribution de t : si la valeur de t est supérieure à la valeur critique (avec $p = 0,05$ et le nombre de degré de liberté), alors le coefficient est significatif à 5%. En d'autres termes, si la vraie corrélation entre les deux variables (calculable uniquement à partir des populations complètes) était 0, alors la probabilité de collecter notre échantillon aurait été inférieure à 5%. Dans ce contexte, on peut raisonnablement rejeter l'hypothèse nulle (corrélation réelle de 0).

La courte syntaxe illustre comment calculer la valeur de t , le nombre de degrés de liberté et la valeur de

TAB. 4.2 : Comparaison de différentes corrélations pour les variables densité de population et pourcentage de personnes ayant 65 ans et plus

	r	IC 5%	IC 95%	Méthode
	-0,16	-0,22	-0,10	pearson
	-0,12	-0,18	-0,06	spearman
	-0,14	-0,20	-0,07	biweight
	-0,17	-0,23	-0,11	percentage
	-0,12	-0,18	-0,05	shepherd

p pour une corrélation donnée.

```
df <- read.csv("data/bivariee/sr_rmr_mtl_2016.csv")
r <- cor(df$A65plus, df$LogTailInc)      # Corrélation
n <- nrow(df)                           # Nombre d'observations
dl <- nrow(df)-2                      # degrés de liberté
t <- r*sqrt((n-2)/(1-r^2))            # Valeur de T
p <- 2*(1-pt(abs(t),dl))             # Valeur de p
cat("\nCorrélation =", round(r, 4),
    "\nValeur de t =", round(t, 4),
    "\nDegrés de liberté =", dl,
    "\np=", round(p, 4))
```

```
## 
## Corrélation = -0.0693
## Valeur de t = -2.1413
## Degrés de liberté = 949
## p= 0.0325
```

Plus simplement, la fonction `cor.test` permet d'obtenir en une seule ligne de code le coefficient de corrélation, l'intervalle de confiance à 95% et les valeurs de t et de p , tel qu'illustré dans la syntaxe ci-dessous. Si l'intervalle de confiance est à cheval sur 0, c'est-à-dire que la borne inférieure est négative et la borne supérieure positive, alors le coefficient de corrélation n'est pas significatif au seuil choisi (95% habituellement). Dans l'exemple ci-dessous, la relation linéaire entre les deux variables est significativement négative avec une corrélation de Pearson de -0,158 ($P=0,000$) et un intervalle de confiance de [-0,219 -0,095].

```
# Intervalle de confiance à 95%
cor.test(df$HabKm2, df$A65plus, conf.level = .95)
```

```
## 
## Pearson's product-moment correlation
##
## data: df$HabKm2 and df$A65plus
## t = -4.9318, df = 949, p-value = 9.616e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.2194457 -0.0954687
## sample estimates:
## cor
## -0.1580801
```

```
# Vous pouvez accéder à chaque sortie de la fonction cor.test comme suit :
p <- cor.test(df$HabKm2, df$A65plus)
cat("Valeur de corrélation = ", round(p$estimate,3), "\n",
    "Intervalle à 95% = [", round(p$conf.int[1],3), " ", round(p$conf.int[2],3), "]", "\n",
    "Valeur de t = ", round(p$statistic,3), "\n",
    "Valeur de p = ", round(p$p.value,3), sep="")
```

```
## Valeur de corrélation = -0.158
## Intervalle à 95% = [-0.219 -0.095]
```

```

## Valeur de t = -4.932
## Valeur de p = 0

# Corrélation de Spearman
cor.test(df$HabKm2, df$A65plus, method = "spearman")

## 
## Spearman's rank correlation rho
##
## data: df$HabKm2 and df$A65plus
## S = 160482182, p-value = 0.0002202
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.1195333

```

```

# Corrélation de Kendall
cor.test(df$HabKm2, df$A65plus, method="kendall")

```

```

## 
## Kendall's rank correlation tau
##
## data: df$HabKm2 and df$A65plus
## z = -3.7655, p-value = 0.0001662
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##          tau
## -0.08157061

```

On pourra aussi modifier l'intervalle de confiance, par exemple à 90% ou 99%. Le choix de l'intervalle de confiance et du seuil de significativité doivent être définis avant l'étude. Il doit s'appuyer sur les standards de la littérature du domaine étudié, du niveau de preuve attendu et de la quantité de données.

```

# Intervalle à 90%
cor.test(df$HabKm2, df$A65plus, method ="pearson", conf.level = .90)

```

```

## 
## Pearson's product-moment correlation
##
## data: df$HabKm2 and df$A65plus
## t = -4.9318, df = 949, p-value = 9.616e-07
## alternative hypothesis: true correlation is not equal to 0
## 90 percent confidence interval:
## -0.2096826 -0.1055995
## sample estimates:
##          cor
## -0.1580801

```

```
# Intervalle à 99%
cor.test(df$HabKm2, df$A65plus, method ="pearson", conf.level = .99)

##
## Pearson's product-moment correlation
##
## data: df$HabKm2 and df$A65plus
## t = -4.9318, df = 949, p-value = 9.616e-07
## alternative hypothesis: true correlation is not equal to 0
## 99 percent confidence interval:
## -0.23839910 -0.07561336
## sample estimates:
##       cor
## -0.1580801
```

Corrélation et bootstrap. Dans le premier chapitre (LIEN BOOTSTRAP), nous avons abordé la notion de *bootstrap*, soit des méthodes d'inférence statistique basées sur des réplications des données initiales par rééchantillonnage. Il est possible d'appliquer la même méthode aux corrélations afin d'obtenir un intervalle de confiance avec r réplications, tel qu'illustré à partir de la syntaxe ci-dessous.

```
library("boot")
df <- read.csv("data/bivariee/sr_rmr_mtl_2016.csv")
# Fonction pour la corrélation
correlation <- function(df, i, X, Y, cor.type="pearson"){
  # Paramètres de la fonction :
  # data : dataframe
  # X et Y : noms des variables X et Y
  # cor.type : type de corrélation : c("pearson","spearman","kendall")
  # i : indice qui sera utilisé par les réplications (à ne pas modifier)
  cor(df[[X]][i], df[[Y]][i], method=cor.type)
}
# Calcul du Bootstrap avec 5000 réplications
corBootstraped <- boot(data=df, # nom du tableau
                        statistic = correlation, # appel de la fonction à répliquer
                        R=5000, # nombre de réplications
                        X = "A65plus",
                        Y ="HabKm2",
                        cor.type="pearson")
# Histogramme pour les valeurs de corrélation issues du Bootstrap
plot(corBootstraped)
```

```
# Corrélation "bootstrapée"
corBootstraped

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = df, statistic = correlation, R = 5000, X = "A65plus",
##       Y = "HabKm2", cor.type = "pearson")
```

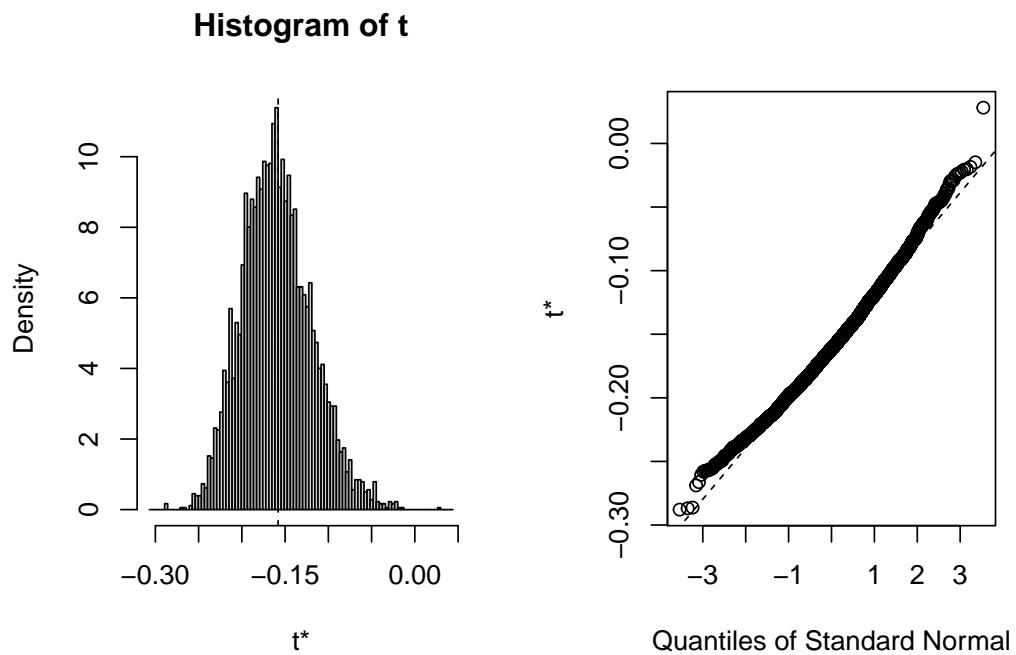


FIG. 4.9 : Histogramme pour les valeurs de corrélation issues du Bootstrap

```

## 
## 
## Bootstrap Statistics :
##      original      bias    std. error
## t1* -0.1580801 -0.00114799  0.04017045

# Intervalle de confiance du bootstrap à 95%
boot.ci(boot.out = corBootstraped, conf = 0.95, type = "all")

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = corBootstraped, conf = 0.95, type = "all")
##
## Intervals :
## Level      Normal          Basic
## 95%   (-0.2357, -0.0782 )  (-0.2416, -0.0856 )
## 
## Level      Percentile        BCa
## 95%   (-0.2306, -0.0745 )  (-0.2185, -0.0460 )
## Calculations and Intervals on Original Scale

# Comparaison de l'intervalle classique basé sur la valeur de T
p <- cor.test(df$HabKm2, df$A65plus)
cat(round(p$estimate,5), " [", round(p$conf.int[1],4), " ", round(p$conf.int[2],4), " ]", sep="")

## -0.15808 [-0.2194 -0.0955]

```

Le *bootstrap* renvoie un coefficient de corrélation de Pearson de -0,158. Les intervalles de confiance obtenues à partir des différentes méthodes d'estimation (normale, basique, pourcentage et bca) ne sont pas à cheval sur 0, indiquant que le coefficient est significatif à 5%.

4.1.3.6 Corrélation partielle



Quelle est la relation entre deux variables continues une fois pris en compte une ou plusieurs variables dites de contrôle? En études urbaines, on pourrait vouloir vérifier si deux variables sont ou non associées une fois contrôlée la densité de population ou encore la distance au centre-ville.

La corrélation partielle permet d'évaluer la relation linéaire entre deux variables quantitatives continues, une fois contrôlé une ou plusieurs autres variables quantitatives (dites variables de contrôle).

Le coefficient de corrélation partielle peut être calculé pour les trois méthodes (Pearson, Spearman et Kendall). Variant aussi de -1 à 1, il est calculé comme suit :

$$r_{ABC} = \frac{r_{AB} - r_{AC}r_{BC}}{\sqrt{(1 - r_{AC}^2)(1 - r_{BC}^2)}} \quad (4.6)$$

avec A et B étant les deux variables pour lesquelles on souhaite évaluer la relation linéaire, une fois contrôlée la variable C ; r étant le coefficient de corrélation (Pearson, Spearman ou Kendall) entre deux variables.

Dans l'exemple ci-dessous, nous voulons estimer la relation linéaire entre le pourcentage de personnes à faible revenu et la couverture végétale au niveau des îlots de l'île de Montréal, une fois contrôlée la densité de population. En effet, plus cette dernière sera forte, plus la couverture végétale sera faible (r de Pearson = -0,603). La valeur du r de Pearson s'élève à -0,546 entre le pourcentage de personnes à faible revenu dans la population totale de l'îlot et la couverture végétale. Une fois la densité de population contrôlée, il chute à -0,316. Pour calculer la corrélation partielle, on pourra utiliser la fonction `pcor.test` du package `ppcor`.

```
library("foreign")
library("ppcor")
dfveg <- read.dbf("data/bivariee/IlotsVeg2006.dbf")
# Corrélation entre les trois variables
round(cor(dfveg[, c("VegPct", "Pct_FR", "LogDens")], method="p"), 3)

##          VegPct Pct_FR LogDens
## VegPct    1.000 -0.513 -0.563
## Pct_FR   -0.513  1.000  0.513
## LogDens  -0.563  0.513  1.000

# Corrélation partielle entre :
# la couverture végétale de l'îlot (%) et
# le pourcentage de personnes à faible revenu
# une fois contrôlée la densité de population
pcor.test(dfveg$Pct_FR, dfveg$VegPct, dfveg$LogDens, method="p")

##      estimate      p.value statistic      n gp Method
## 1 -0.3155194 8.093159e-235 -33.59772 10213  1 pearson
```

```
# Calcul de la corrélation partielle avec la formule :
corAB <- cor(dfveg$VegPct, dfveg$Pct_FR, method = "p")
corAC <- cor(dfveg$VegPct, dfveg$LogDens, method = "p")
corBC <- cor(dfveg$Pct_FR, dfveg$LogDens, method = "p")
CorP <- (corAB - (corAC*corBC)) / sqrt((1-corAC^2)*(1-corBC^2))
cat("Corr. partielle avec ppcor = ",
    round(ppcor.test(dfveg$Pct_FR, dfveg$VegPct, dfveg$LogDens, method="p")$estimate,5),
    "\nCorr. partielle (formule) = ", round(CorP, 5))

## Corr. partielle avec ppcor = -0.31552
## Corr. partielle (formule) = -0.31552
```

4.1.3.7 Mise en œuvre dans R

Comme vous l'aurez compris, il est possible d'arriver au même résultat dans par différents moyens. Pour calculer les corrélations, nous avons utilisé jusqu'à présent les fonctions de base `cor` et `cor.test`. Il est aussi possible de recourir à des fonctions d'autres *packages*, dont notamment :

- **Hmisc** dont la fonction `rcorr` permet de calculer des corrélations de Pearson et Spearman (mais non celle de Kendall) avec la valeur de p .
- **psych** dont la fonction `corr.test` permet d'obtenir la matrice de corrélation (Pearson, Spearman et Kendall), les intervalles de confiance et les valeurs de p .
- **stargazer** pour créer des beaux tableaux d'une matrice de corrélation en *HTML* ou en *LaTeX* ou en ASCII.
- **apaTables** pour créer un tableau avec une matrice de corrélation dans un fichier Word.
- **correlation** pour aller plus loin et explorer les corrélations bayesiennes, robustes, non-linéaires ou multiniveaux.

```
df <- read.csv("data/bivariee/sr_rmr_mtl_2016.csv")
library("Hmisc")
library("stargazer")
library("apaTables")
library("dplyr")
# Corrélations de Pearson et Spearman et valeurs de p
# avec la fonction rcorr de Hmisc pour deux variables
Hmisc:::rcorr(df$RevMedMen, df$Locataire, type="pearson")
```

```
##          x      y
## x  1.00 -0.78
## y -0.78  1.00
##
## n= 951
##
##
## P
##   x   y
## x     0
## y     0
```

```
Hmisc::rcorr(df$RevMedMen, df$Locataire, type="spearman")
```

```
##      x     y
## x  1.00 -0.91
## y -0.91  1.00
##
## 
## n= 951
##
## 
## P
##      x     y
## x  1.00 -0.91
## y -0.91  1.00
## P
```

Matrice de corrélation avec la fonction rcorr de Hmisc pour plus de variables
On crée un vecteur avec les noms des variables à sélectionner
Vars <- c("RevMedMen", "Locataire", "LogTailInc", "A65plus", "ImgRec", "HabKm2", "FaibleRev")
Hmisc::rcorr(df[, Vars] %>% as.matrix())

```
##          RevMedMen Locataire LogTailInc A65plus ImgRec HabKm2 FaibleRev
## RevMedMen      1.00    -0.78     -0.46   -0.07   -0.46   -0.49   -0.74
## Locataire     -0.78     1.00      0.56    0.00    0.64    0.71    0.88
## LogTailInc    -0.46      0.56     1.00   -0.07    0.82    0.48    0.62
## A65plus       -0.07      0.00     -0.07    1.00   -0.06   -0.16   -0.01
## ImgRec        -0.46      0.64      0.82   -0.06    1.00    0.56    0.68
## HabKm2        -0.49      0.71      0.48   -0.16    0.56    1.00    0.64
## FaibleRev     -0.74      0.88      0.62   -0.01    0.68    0.64    1.00
##
## n= 951
##
## 
## P
##          RevMedMen Locataire LogTailInc A65plus ImgRec HabKm2 FaibleRev
## RevMedMen      0.0000    0.0000     0.0441  0.0000  0.0000  0.0000
## Locataire     0.0000            0.0000     0.9594  0.0000  0.0000  0.0000
## LogTailInc  0.0000    0.0000            0.0325  0.0000  0.0000  0.0000
## A65plus       0.0441    0.9594     0.0325            0.0682  0.0000  0.6796
## ImgRec        0.0000    0.0000     0.0000     0.0682            0.0000  0.0000
## HabKm2        0.0000    0.0000     0.0000     0.0000    0.0000            0.0000
## FaibleRev    0.0000    0.0000     0.0000     0.6796  0.0000  0.0000
```

Avec la fonction corr.test de psych pour avoir la matrice de corrélation
(Pearson, Spearman et Kendall), les intervalles de confiance et les valeurs de p
print(psych::corr.test(df[, Vars],
method = "kendall",
ci=TRUE, alpha = 0.05), short=FALSE)
Création d'un tableau pour une matrice de corrélation
changer le paramètre type pour 'html' or 'LaTex'
p <- cor(df[, Vars], method="pearson")

```

stargazer(p, title="Correlation Matrix", type = "text")

## 
## Correlation Matrix
## =====
## RevMedMen Locataire LogTailInc A65plus ImgRec HabKm2 FaibleRev
## -----
## RevMedMen    1   -0.785   -0.461   -0.065   -0.458   -0.489   -0.743
## Locataire   -0.785    1    0.562   -0.002    0.645    0.708    0.879
## LogTailInc  -0.461    0.562    1   -0.069    0.816    0.475    0.622
## A65plus     -0.065   -0.002   -0.069    1   -0.059   -0.158   -0.013
## ImgRec       -0.458    0.645    0.816   -0.059    1    0.561    0.678
## HabKm2      -0.489    0.708    0.475   -0.158    0.561    1    0.642
## FaibleRev   -0.743    0.879    0.622   -0.013    0.678    0.642    1
## -----

```

```

# stargazer(p, title="Correlation Matrix", type = "html")
# stargazer(p, title="Correlation Matrix", type = "latex")
# Créer un tableau avec la matrice de corrélation
# dans un fichier Word (.doc)
apaTables::apa.cor.table(df[, c("RevMedMen", "Locataire", "LogTailInc")],
                           filename = "data/bivariee/TitiLaMatrice.doc",
                           show.conf.interval = TRUE,
                           landscape = TRUE)

```

```

## 
## Means, standard deviations, and correlations with confidence intervals
## 
## 
## Variable      M        SD      1          2
## 1. RevMedMen 66065.50 26635.27
## 
## 2. Locataire  45.05   26.33   -.78**    [-.81, -.76]
## 
## 3. LogTailInc 5.54    4.82    -.46**    .56**    [-.51, -.41] [.52, .60]
## 
## 
## Note. M and SD are used to represent mean and standard deviation, respectively.
## Values in square brackets indicate the 95% confidence interval.
## The confidence interval is a plausible range of population correlations
## that could have caused the sample correlation (Cumming, 2014).
## * indicates p < .05. ** indicates p < .01.
## 
```



Une image vaut mille mots, surtout pour une matrice de corrélation! Le package **corrplot** vous permet justement de construire de belles figures avec une matrice de corrélation (figures 4.10 et 4.11). L'intérêt de ce

type de figure est de repérer rapidement des associations intéressantes lorsque l'on calcule les corrélations entre un grand nombre de variables.

```
library("corrplot")
library("ggpubr")
df <- read.csv("data/bivariee/sr_rmr_mtl_2016.csv")
Vars <- c("RevMedMen", "Locataire", "LogTailInc", "A65plus", "ImgRec", "HabKm2", "FaibleRev")
p <- cor(df[, Vars], method="pearson")
couleurs <- colorRampPalette(c("#053061", "#2166AC", "#4393C3", "#92C5DE",
                                "#D1E5F0", "#FFFFFF", "#FDDBC7", "#F4A582",
                                "#D6604D", "#B2182B", "#67001F"))
corrplot(p, addrect = 3, method="number", diag=FALSE, col=couleurs(100))
```

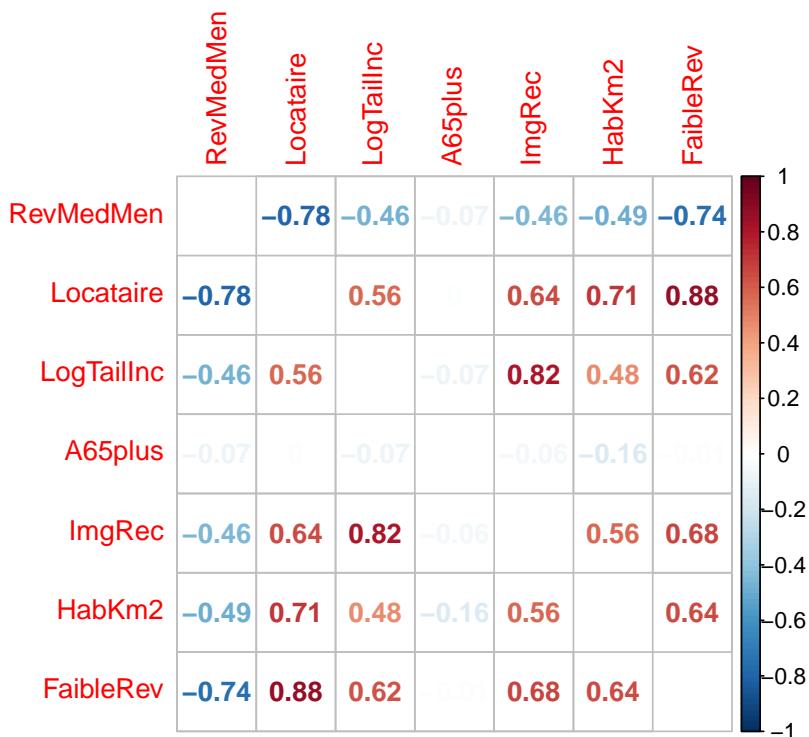


FIG. 4.10 : Matrice de corrélation avec corrplot (chiffres)

```
fig2 <- corrplot.mixed(p, lower="number", lower.col = "black",
                        upper = "ellipse", upper.col=couleurs(100))
```

4.1.3.8 Comment rapporter des valeurs de corrélations ?

Bien qu'il n'y ait pas qu'une seule manière de reporter des corrélations, voici quelques lignes directrices pour vous guider :

- Signaler si la corrélation est faible, modérée ou forte.
- Indiquer si la corrélation est positive ou négative. Toutefois, ce n'est pas une obligation car l'on peut rapidement le constater avec le signe du coefficient.
- Mettre le r et le p en italique et en minuscules.

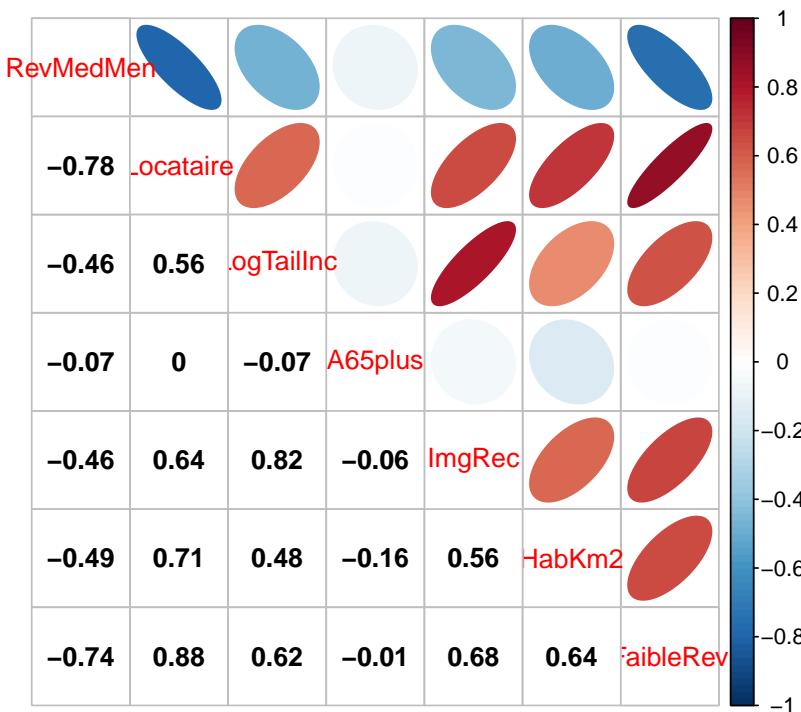


FIG. 4.11 : Matrice de corrélation avec corrplot (chiffres et ellipses)

- Deux décimales uniquement pour le r (sauf si une plus grande précision se justifie dans le domaine d'étude).
- Trois décimales pour la valeur de p . Si elle est inférieure à 0,001, écrire plutôt $p < 0,001$.
- Indiquer éventuellement le nombre de degrés de liberté, soit $r(dl) = \dots$

Voici des exemples :

- La corrélation entre les variables revenu médian des ménages et pourcentage de locataire est fortement négative ($r = -0,78, p < 0,001$).
- La corrélation entre les variables revenu médian des ménages et pourcentage de locataire est forte ($r(949) = -0,78, p < 0,001$).
- La corrélation entre les variables densité de population et pourcentage de logements de taille est modérée ($r = 0,48, p < 0,001$).
- La corrélation entre les variables densité de population et pourcentage de 65 ans et plus n'est pas significative ($r = -0,08, p = 0,07$).

Pour un texte en anglais, référez-vous à : <https://www.socscistatistics.com/tutorials/correlation/default.aspx>.

4.1.4 Régression linéaire simple



Comment expliquer et prédire une variable continue en fonction d'une autre variable ? Répondre à cette question relève de la statistique inférentielle. Il s'agit en effet d'établir une équation simple du type $Y = a + bX$, pour expliquer et prédire les valeurs d'une variable dépendante (Y) à partir d'une variable indépendante (X). L'équation de la régression est construite grâce à un jeu de données (un échantillon). À partir de cette équation, il est possible de prédire la valeur attendue de Y pour n'importe quelle valeur de X . On appelle cette équation un modèle, car elle cherche à représenter la réalité de façon simplifiée.

La régression linéaire simple se distingue ainsi de la **covariance** (section 4.1.2) et de la **corrélation** (section 4.1.3), relevant de la statistique bivariée descriptive et exploratoire.

Par exemple, la régression linéaire simple pourrait être utilisée pour expliquer les notes d'un groupe d'étudiants à un examen (variable dépendante Y) en fonction du nombre d'heures qu'ils ont consacrés à la révision des notes de cours (variable indépendante X). Une fois l'équation de régression déterminée et si le modèle est efficace, nous pourrons prédire les notes des étudiants inscrits au cours la session suivante en fonction du temps qu'ils prévoient de passer à étudier, et ce, avant même qu'ils aient passé l'examen.

Formulons un exemple d'application de la régression linéaire simple en études urbaines. Dans le cadre d'une étude sur les îlots de chaleur urbains, la température de surface (variable dépendante) pourrait être expliquée par la proportion de la superficie de l'îlot couverte par de la végétation (variable indépendante). On supposerait alors que plus cette proportion est importante, plus la température est faible et inversement, soit une relation linéaire négative. Si le modèle est efficace, nous pourrions prédire la température moyenne des îlots d'une autre municipalité pour laquelle nous ne disposons pas d'une carte de température, et repérer ainsi les îlots de chaleur potentiels. Bien entendu, il est peu probable que nous arrivions à prédire efficacement la température moyenne des îlots avec uniquement la couverture végétale comme variable explicative. En effet, bien d'autres caractéristiques de la forme urbaine peuvent influencer ce phénomène comme la densité du bâti, la couleur des toits, les occupations du sol présentes, l'effet des canyons urbains, etc. Il faudrait alors inclure non pas une, mais plusieurs variables explicatives (indépendantes).

Ainsi, on distinguera la **régression linéaire simple** (une variable indépendante, explicative) de la **régression linéaire multiple** (plusieurs variables indépendantes); cette dernière sera largement abordée au chapitre 5.

Dans cette section, nous décrirons succinctement la régression linéaire simple. Concrètement, nous verrons comment déterminer la droite de régression, interpréter ses différents paramètres du modèle et comment évaluer la qualité d'ajustement du modèle. Nous n'aborderons pas les hypothèses liées au modèle de régression linéaire des moindres carrés ordinaires (MCO), ni les conditions d'application. Ces éléments seront expliqués au chapitre 5 consacré à la régression linéaire multiple.



Corrélation, régression simple et causalité : attention aux raccourcis !

Si une variable X explique et prédit efficacement une variable Y , cela ne veut pas dire pour autant que X cause Y . Autrement dit, la corrélation, l'association entre deux variables ne signifie qu'il existe un lien de causalité entre elles.

Premièrement, la variable explicative (X , indépendante) doit absolument précéder la variable à expliquer (Y , dépendante). Par exemple, l'âge (X) peut influencer le sentiment de sécurité (Y). Mais, le sentiment de sécurité ne peut en aucun cas influencer l'âge. Par conséquent, l'âge ne peut conceptuellement pas être la variable dépendante dans cette relation.

Deuxièmement, bien qu'une variable puisse expliquer efficacement une autre variable, elle peut être un **facteur confondant**. Prenons deux exemples bien connus :

- Avoir les doigts jaunes est associé au cancer du poumon. Bien entendu, les doigts jaunes ne causent pas le cancer : c'est un facteur confondant puisque fumer augmente les risques du cancer du poumon et jaunit aussi les doigts.
- Dans un article intitulé *Chocolate Consumption, Cognitive Function, and Nobel Laureates*, Messerli (2012) a trouvé une corrélation positive entre la consommation de chocolat par habitant et le nombre de prix Nobel pour dix millions d'habitants pour 23 pays. Ce résultat a d'ailleurs été rapporté par de nombreux médias (Radio Canada¹, La Presse², Le Point³, etc.), sans pour autant que Messerli (2012) et les journalistes concluent à un lien de causalité entre les deux variables. Tout chercheur sait que la consommation de chocolat ne permet pas d'obtenir des résultats intéressants et de publier dans des revues prestigieuses ; c'est plutôt le café ! Plus sérieusement, il est probable que les pays les plus riches investissent davantage dans la recherche et obtiennent ainsi plus de prix Nobel. Dans les pays les plus

riches, il est aussi probable que l'on consomme plus de chocolat, considéré comme un produit de luxe dans les pays les plus pauvres.

Pour approfondir le sujet sur la confusion entre *corrélation*, *régression simple* et *causalité*, vous pourrez visionner cette courte vidéo ludique⁴ de vulgarisation.

L'association entre deux variables peut aussi être simplement le fruit du hasard. Si l'on explore de très grandes quantités de données (avec un nombre impressionnant d'observations et de variables), soit une démarche relevant du *data mining*, le hasard fera que l'on risque d'obtenir des corrélations surprenantes entre certaines variables. Prenons un exemple concret, admettons que l'on ait collecté 100 variables et que l'on calcule les corrélations entre chaque paire de variables. On obtient une matrice de corrélation de 100 x 100, à laquelle on peut enlever la diagonale et une moitié de la matrice, ce qui nous laisse un total de 4950 corrélations différentes. Admettons que l'on choisisse un seuil de significativité de 5%, on doit alors s'attendre à ce que le hasard produise des résultats significatifs dans 5% des cas. Sur 4950 corrélations, cela signifie qu'environ 247 corrélations seront significatives, et ce, indépendamment de la nature des données. Nous pouvons aisément l'illustrer avec la syntaxe suivante :

```
library("Hmisc")
nbVars <- 100 # nous utilisons 100 variables générées aléatoirement pour l'expérience
nbExperiment <- 1000 # nous reproduirons 1000 fois l'expérience avec les 100 variables
# Le nombre de variables significatives par expérience est enregistrée dans Results
Results <- c()
# iterons pour chaque expérimentation (1000 fois)
for(i in 1:nbExperiment){
  Dataas <- list()
  # générerons 100 variables aléatoires normalement distribuées
  for (j in 1:nbVars){
    Dataas[[j]] <- rnorm(150)
  }
  DF <- do.call("cbind",Dataas)
  # calculons la matrice de corrélation pour les 100 variables
  cor_mat <- rcorr(DF)
  # comptons combien de fois les corrélations étaient significatives
  Sign <- table(cor_mat$P<0.05)
  NbPairs <- Sign[["TRUE"]]/2
  # ajoutons les résultats dans Results
  Results <- c(Results,NbPairs)
}
# transformons Results en un dataframe
df <- data.frame(Values = Results)
# affichons le résultat
ggplot(df, aes(x = Values)) +
  geom_histogram(aes(y =..density..),
                 colour = "black",
                 fill = "white") +
  stat_function(fun = dnorm, args = list(mean = mean(df$Values),
                                         sd = sd(df$Values)),color="blue")+
  geom_vline(xintercept = mean(df$Values),color="red", size=1.2) +
  annotate("text", x=250, y = 0.028,
           label = paste("Nombre moyen de corrélations significatives\nsur 1000 réplications :",
                         round(mean(df$Values),0),sep=""), hjust="left")+
  xlab("Nombre de corrélations significatives")+
  ylab("densité")
```

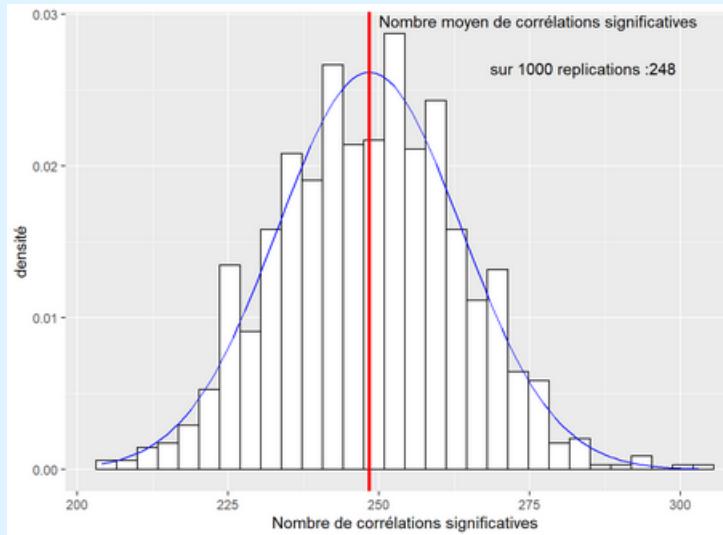


FIG. 4.12 : Corrélations significatives obtenues aléatoirement

4.1.4.1 Principe de base de la régression linéaire simple

La régression linéaire simple vise à déterminer une droite (une fonction linéaire) qui résume le mieux la relation linéaire entre une variable dépendante (Y) et une variable indépendante (X) :

$$\widehat{y}_i = \beta_0 + \beta_1 x_i \quad (4.7)$$

avec \widehat{y}_i et x_i qui sont respectivement la valeur prédictive de la variable indépendante et la valeur de la variable dépendante pour l'observation i . β_0 est la constante (*intercept* en anglais), soit la valeur prédictive de la variable Y quand X est égale à 0. β_1 est le coefficient de régression pour la variable X , soit la pente de la droite. Ce coefficient nous informe sur la relation entre les deux variables : s'il est positif, la relation est positive ; s'il est négatif, la relation est négative, et proche de 0, la relation est nulle (la droite sera alors horizontale). Plus la valeur absolue de β_1 est élevée, plus la pente est forte, et plus la variable Y varie à chaque changement d'une unité de la variable X .

Considérons un exemple fictif de dix municipalités d'une région métropolitaine pour lesquelles nous disposons de deux variables : le pourcentage d'actifs occupés se rendant au travail principalement à vélo et la distance de la municipalité au centre-ville (tableau 4.3).

D'emblée, à la lecture du nuage de points (figure 4.13), on décèle une forte relation linéaire négative entre les deux variables : plus la distance au centre-ville augmente, plus le pourcentage de cyclistes est faible, ce qui est confirmé par le coefficient de corrélation ($r = -0,86$). La droite de régression (en rouge à la figure 4.13) qui résume le mieux la relation entre Vélo (variable dépendante) et KmCV (variable indépendante) s'écrit alors : $Vélo = 30,603 - 1,448 \times KmCV$.

TAB. 4.3 : Données fictives sur l'utilisation du vélo par municipalité

Municipalité	Vélo	KmCV	Municipalité	Vélo	KmCV
A	12,5	14,135	F	18,5	7,195
B	13,5	10,065	G	21,2	7,953
C	15,8	7,762	H	23,0	4,293
D	15,9	11,239	I	25,3	5,225
E	17,6	7,706	J	30,2	2,152

La valeur du coefficient de régression (β_1) est de -1,448. Le signe de ce coefficient décrit une relation négative entre les deux variables. Ainsi, à chaque ajout d'une unité de la distance au centre-ville (exprimée en kilomètres), le pourcentage de cyclistes diminue de 1,448. Retenez que l'unité de mesure de la variable dépendante est très importante pour bien interpréter le coefficient de régression. En effet, si la distance au centre-ville n'était pas exprimée en kilomètres, mais plutôt en mètres, β_1 sera égal à -0,001448. Dans la même optique, l'ajout de 10 km de distance entre une municipalité et le centre-ville fait diminuer le pourcentage de cyclistes de -14,48 points de pourcentage.

Avec, cette équation de régression, il est possible de prédire le pourcentage de cyclistes pour n'importe quelle municipalité de la région métropolitaine. Par exemple, pour des distances de 5, 10 ou 20 kilomètres, les pourcentages de cyclistes seraient de :

- $\hat{y}_i = 30,603 + (-1,448 \times 5\text{km}) = 23,363$
- $\hat{y}_i = 30,603 + (-1,448 \times 10\text{km}) = 8,883$
- $\hat{y}_i = 30,603 + (-1,448 \times 20\text{km}) = 1,643$

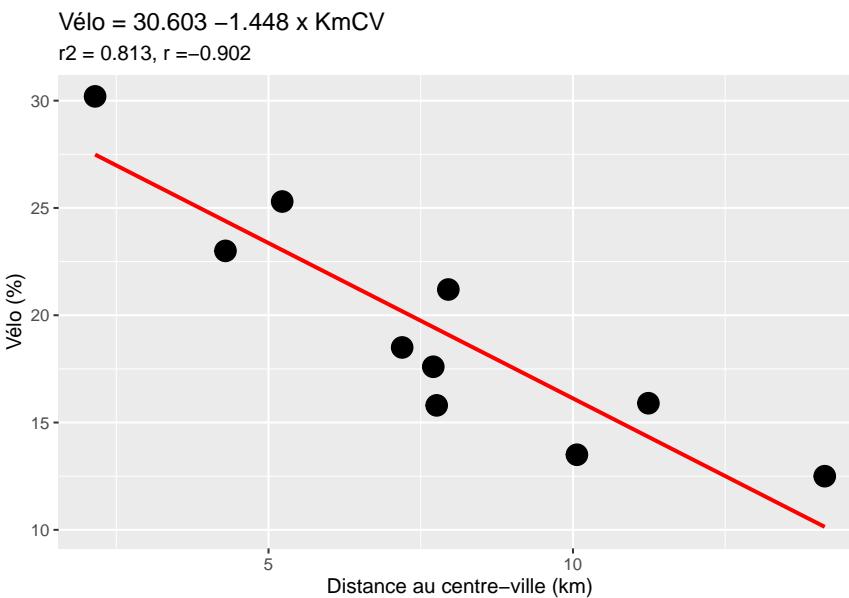


FIG. 4.13 : Relation linéaire entre l'utilisation du vélo et la distance au centre-ville

4.1.4.2 Formulation de la droite de régression des moindres carrés ordinaires

Reste à savoir comment sont estimés les différents paramètres de l'équation, soit β_0 et β_1 . À la figure 4.14, les points noirs représentent les valeurs observées (y_i) et les points bleus les valeurs prédictes (\hat{y}_i) par l'équation du modèle. Les traits noirs verticaux représentent pour chaque observation i , l'écart entre la valeur observée et la valeur prédictée, dénommé résidu (ϵ_i , prononcez epsilon de i ou plus simplement le résidu pour i , ou encore le terme d'erreur de i). Si un point est au-dessus de la droite de régression, la valeur observée sera alors supérieure à la valeur prédictée ($y_i > \hat{y}_i$) et inversement, si le point est au-dessous de la droite ($y_i < \hat{y}_i$). Plus cet écart (ϵ_i) est important, plus l'observation s'éloigne de la prédition du modèle, et par extension moins bon est le modèle. Au tableau 4.4, vous constaterez que la somme des résidus est égale à zéro. La méthode des moindres carrés ordinaires (MCO) vise à minimiser les écarts au carré entre les valeurs observées (y_i) et prédictes ($\beta_0 + \beta_1 x_i$, soit \hat{y}_i) :

$$\min \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 x_i))^2 \quad (4.8)$$

Pour minimiser ces écarts, le coefficient de régression β_1 représente le rapport entre la covariance entre X et Y et la variance de Y (équation (4.9)), tandis que la constante β_0 est la moyenne de la variable Y moins le produit de la moyenne de X et de son coefficient de régression (équation (4.10)).

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{cov(X, Y)}{var(X)} \quad (4.9)$$

$$\beta_0 = \bar{Y} - \beta_1 \bar{X} \quad (4.10)$$

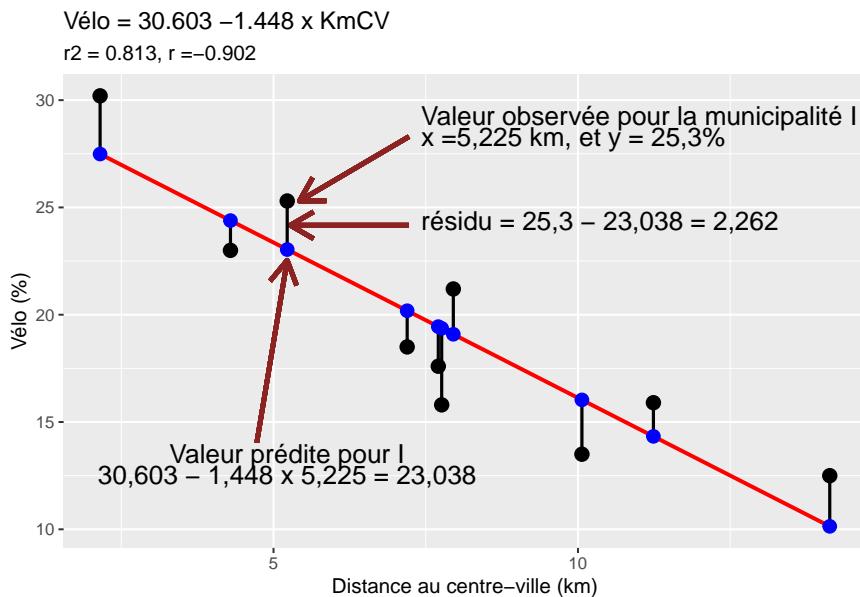


FIG. 4.14 : Droite de régression, valeurs observées, prédites et résidus

4.1.4.3 Mesurer la qualité d'ajustement du modèle

Les trois mesures les plus courantes pour évaluer la qualité d'ajustement d'un modèle de régression linéaire simple sont l'erreur quadratique moyenne (en anglais, *root-mean-square error, RMSE*), le coefficient de détermination (R^2) et la statistique F de Fisher. Pour mieux appréhender le calcul de ces trois mesures, rappelons que l'équation de régression s'écrit :

TAB. 4.4 : Valeurs observées, prédites et résidus

Municipalité	Vélo	KMCV	Valeur prédictive	Résidu	Résidu au carré
A	12,5	14,135	10,138	2,362	5,579
B	13,5	10,065	16,031	-2,531	6,406
C	15,8	7,762	19,365	-3,565	12,709
D	15,9	11,239	14,331	1,569	2,462
E	17,6	7,706	19,446	-1,846	3,408
F	18,5	7,195	20,186	-1,686	2,843
G	21,2	7,953	19,089	2,111	4,456
H	23,0	4,293	24,388	-1,388	1,927
I	25,3	5,225	23,038	2,262	5,117
J	30,2	2,152	27,488	2,712	7,355
Somme	-	-	-	0,000	52,262

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \Rightarrow Y = \beta_0 + \beta_1 X + \epsilon \quad (4.11)$$

Elle comprend ainsi une partie de Y qui est expliquée par le modèle et une autre partie non expliquée : ϵ appelé habituellement le terme d'erreur. Ce terme d'erreur pourrait représenter d'autres variables explicatives qui n'ont pas été prises en compte pour prédire la variable indépendante ou une forme de variation aléatoire inexplicable présente lors de la mesure.

$$Y = \underbrace{\beta_0 + \beta_1 X}_{\text{partie expliquée par le modèle}} + \underbrace{\epsilon}_{\text{partie non expliquée}} \quad (4.12)$$

Par exemple, pour la municipalité A au tableau 4.4, nous avons : $y_A = \hat{y}_A - \epsilon_A \Rightarrow 12.5 = 10.138 + 2.362$. Souvenez-vous que la variance d'une variable est la somme des écarts à la moyenne, divisée par le nombre d'observations. Par extension, il est alors possible de décomposer la variance de Y comme suit :

$$\underbrace{\sum_{i=1}^n (y_i - \bar{y})^2}_{\text{variance de } Y} = \underbrace{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}_{\text{var. expliquée}} + \underbrace{\sum_{i=1}^n (y_i - \hat{y}_i)^2}_{\text{var. non expliquée}} \Rightarrow SCT = SCE + SCR \quad (4.13)$$

avec :

- SCT est la somme des écarts au carré des valeurs observées à la moyenne (en anglais, *total sum of squares*)
- SCE est la somme des écarts au carré des valeurs prédictes à la moyenne (en anglais, *regression sum of squares*)
- SCR est la somme des carrés des résidus (en anglais, *sum of squared errors*).

Autrement dit, la variance totale est égale à la variance expliquée plus la variance non expliquée. Au tableau 4.5, vous pouvez repérer les valeurs de SCT , SCE et SCR et constater que $279,30 = 227,04 + 52,26$ et $27,93 = 22,70 + 5,23$.

Calcul de l'erreur quadratique moyenne

La somme des résidus au carré (SCR) divisée par le nombre d'observations représente donc le carré moyen des erreurs (en anglais, *mean square error - MSE*), soit la variance résiduelle du modèle ($52,26/10 = 5,23$). Plus sa valeur sera faible, plus le modèle sera efficace pour prédire la variable

TAB. 4.5 : Calcul du coefficient de détermination

Municipalité	y_i	\hat{y}_i	ϵ_i	$(y_i - \bar{y})^2$	$(\hat{y}_i - y_i)^2$	ϵ_i^2
A	12,50	10,14	2,36	46,92	84,86	5,58
B	13,50	16,03	-2,53	34,22	11,02	6,41
C	15,80	19,37	-3,57	12,60	0,00	12,71
D	15,90	14,33	1,57	11,90	25,19	2,46
E	17,60	19,45	-1,85	3,06	0,01	3,41
F	18,50	20,19	-1,69	0,72	0,70	2,84
G	21,20	19,09	2,11	3,42	0,07	4,46
H	23,00	24,39	-1,39	13,32	25,38	1,93
I	25,30	23,04	2,26	35,40	13,60	5,12
J	30,20	27,49	2,71	117,72	66,22	7,36
N	10,00	-	-	-	-	-
Somme	193,50	-	0,00	279,30	227,04	52,26
Moyenne	19,35	-	0,00	27,93	22,70	5,23

indépendante. L'erreur quadratique moyenne (en anglais, *root-mean-square error - RMSE*) est simplement la racine carrée de la somme des résidus au carré divisée par le nombre d'observations (n) :

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y})^2}{n}} \quad (4.14)$$

Elle représente ainsi une **mesure absolue des erreurs** qui est exprimée dans l'unité de mesure de la variable dépendante. Dans le cas présent, on a : $\sqrt{5,23} = 2,29$. Cela signifie qu'en moyenne, l'écart absolu (ou erreur absolue) entre les valeurs observées et prédictes est de 2,29 points de pourcentage. De nouveau, une plus faible valeur de **RMSE** indique un meilleur ajustement du modèle. Mais surtout, le RMSE permet d'évaluer avec quelle précision le modèle prédit la variable dépendante. Il est donc particulièrement important si l'objectif principal du modèle est de prédire des valeurs sur un échantillon d'observations pour lequel la variable dépendante est inconnue.

Calcul du coefficient de détermination

Nous avons largement démontré que la variance totale est égale à la variance expliquée plus la variance non expliquée. La qualité du modèle peut donc être évaluée avec le coefficient de détermination (R^2), soit le rapport entre les variances expliquée et totale :

$$R^2 = \frac{SCE}{SCT} \text{ avec } R^2 \in [0, 1] \quad (4.15)$$

Comparativement au RMSE qui est une mesure absolue, le coefficient de détermination est une **mesure relative** qui varie de 0 à 1. Il exprime la proportion de la variance de Y qui est expliquée par la variable X ; autrement dit, plus sa valeur est élevée, plus X influence / est capable de prédire Y . Dans le cas présent, on a : $R^2 = 227.04/279.3 = 0.8129$, ce qui signale que 81,3% de la variance du pourcentage de cyclistes est expliquée par la distance au centre-ville. Tel que signalé dans la section 4.1.3.2, la racine carrée du coefficient de détermination (R^2) est égale au coefficient de corrélation (r) entre les deux variables.

Calcul de la statistique F de Fisher

La statistique F de Fisher permet de vérifier la significativité globale du modèle.

$$F = (n - 2) \frac{R^2}{1 - R^2} = (n - 2) \frac{SCE}{SCR} \quad (4.16)$$

L'hypothèse nulle (H_0 avec $\beta_1 = 0$) est rejetée si la valeur calculée de F est supérieure à la valeur critique de la table F avec $(1, n-2)$ degrés de liberté et un seuil α ($p=0,05$ habituellement) (voir le tableau des valeurs critiques de F , section 12.2). Notez qu'on utilise rarement la table F puisqu'avec la fonction `pf(fobtenu, 1, n-2, lower.tail = FALSE)` l'on obtient directement la valeur de p associée à la valeur de F . Concrètement, si le test F est significatif (avec $p < 0,05$), plus la valeur de F sera élevée, plus le modèle sera efficace (et plus le R^2 sera également élevé).

Notez que la fonction `summary` renvoie les résultats du modèle, dont notamment le test F de Fisher.

```
# utiliser la fonction summary
summary(modele)

##
## Call:
## lm(formula = Velo ~ KmCV, data = data)
```

```

## 
## Residuals:
##      Min       1Q   Median      3Q      Max
## -3.5652 -1.8062  0.0906  2.2241  2.7125
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 30.6032    2.0729 14.763 4.36e-07 ***
## KmCV        -1.4478    0.2456 -5.895 0.000364 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.556 on 8 degrees of freedom
## Multiple R-squared:  0.8129, Adjusted R-squared:  0.7895 
## F-statistic: 34.75 on 1 and 8 DF,  p-value: 0.0003637

```

Dans le cas présent, $F = (10-2)\frac{0.8129}{1-0.8129} = (10-2)\frac{227.04}{52.26} = 34.75$ avec une valeur de $p < 0.001$. Par conséquent, le modèle est significatif.

4.1.4.4 Mise en œuvre dans R

Comment calculer une régression linéaire simple dans ? Rien de plus simple avec la fonction `lm(formula = y ~ x, data= dataframme)`.

```

df <- read.csv("data/bivariee/Reg.csv", stringsAsFactors = F)
## Création d'un objet pour le modèle
monmodele <- lm(Velo ~ KmCV, df)
## Sorties du modèle avec la fonction summary
summary(monmodele)

```

```

## 
## Call:
## lm(formula = Velo ~ KmCV, data = df)
## 
## Residuals:
##      Min       1Q   Median      3Q      Max
## -3.5652 -1.8062  0.0906  2.2241  2.7125
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 30.6032    2.0729 14.763 4.36e-07 ***
## KmCV        -1.4478    0.2456 -5.895 0.000364 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.556 on 8 degrees of freedom
## Multiple R-squared:  0.8129, Adjusted R-squared:  0.7895 
## F-statistic: 34.75 on 1 and 8 DF,  p-value: 0.0003637

```

```
## Calcul du MSE et du RMSE
MSE <- mean(momodele$residuals^2)
RMSE <- sqrt(MSE)
cat("MSE=", round(MSE, 2), "; RMSE=", round(RMSE, 2), sep="")

## MSE=5.23; RMSE=2.29
```

4.1.4.5 Comment rapporter une régression linéaire simple

Nous avons calculé une régression linéaire simple pour prédire le pourcentage d'actifs occupés utilisant le vélo pour se rendre au travail en fonction de la distance au centre-ville (en kilomètres). Le modèle obtient un F de Fisher significatif ($F(1,8)=34,75, p < 0,001$) et un R^2 de 0,813. Le pourcentage de cyclistes peut être prédit par l'équation suivante : $30,603 - 1,448 \times$ (distance au centre-ville en km).

4.2 Relation entre deux variables qualitatives



Deux variables qualitatives sont-elles associées entre elles ? Plus spécifiquement, certaines modalités d'une variable qualitative sont-elles associées significativement à certaines modalités d'une autre variable qualitative.

Prenons l'exemple de deux variables qualitatives : l'une intitulée *groupe d'âge* comprenant trois modalités (15 à 29 ans, 30 à 44 ans, 45 à 64 ans) ; l'autre intitulée *mode de transport habituel pour se rendre au travail* comprenant quatre modalités (véhicule motorisé, transport en commun, vélo, marche).

Comparativement aux deux autres groupes, on pourrait supposer que les jeunes se déplacent proportionnellement plus en modes de transport actifs (vélo et marche) et en transports en commun. À l'inverse, il est possible que les 45 à 64 ans se déplacent majoritairement en véhicule motorisé.

Pour vérifier l'existence d'associations significatives entre les modalités de deux variables qualitatives, il est possible de construire un **tableau de contingence** (section 4.2.1), puis de réaliser le **test du χ^2** (section 4.2.2).

4.2.1 Construction de tableau de contingence

Les données du tableau de contingence ci-dessous décrivent 279 projets d'habitation à loyer modique (HLM) dans l'ancienne ville de Montréal, croisant les modalités de la période de construction (en colonnes) et de la taille (en ligne) des projets HLM (Apparicio, 2002). Les différents éléments du tableau sont décrits ci-dessous.

- **Les fréquences observées**, nommées communément f_{ij} , correspondent aux observations appartenant à la fois à la $i^{ème}$ modalité de la variable en ligne et à la $j^{ème}$ modalité de la variable en colonne. À titre d'exemple, on compte 14 HLM construits entre 1985 et 1989 comprenant moins de 25 logements.
- **Les marges** du tableau sont les totaux pour chaque modalité en ligne ($n_{i.}$) et en colonne ($n_{.j}$). En guise d'exemple, sur les 279 projets HLM, 53 comprennent de 25 à 49 logements et 56 ont été construites entre 1968 et 1974. Bien entenu, la somme des marges en ligne ($n_{i.}$) est égale au nombre total d'observations (n_{ij}), tout comme la somme de marges en colonne ($n_{.j}$).
- **Trois pourcentages** sont disponibles (total, en ligne, en colonne). Ils sont respectivement la fréquence observée divisée par le nombre d'observations ($f_{ij}/n_{ij} \times 100$), par la marge en ligne ($f_{ij}/n_{i.} \times 100$) et en colonne ($f_{ij}/n_{.j} \times 100$). En guise d'exemple, 5% des 279 projets HLM ont été construits entre 1985 et 1989 et comprennent moins de 25 logements (pourcentage total, soit

$14/279 \times 100$). Aussi, plus de la moitié des habitations de moins de 25 logements ont été construits entre 1990 et 1994 (pourcentage en ligne, $41/80 \times 100$). Finalement, près de 36% des logements construits avant 1975 ont 100 logements et plus ($20/56 \times 100$).

- **Les fréquences théoriques**, représentent les valeurs que l'on devrait observer théoriquement s'il y avait indépendance entre les modalités des deux variables : si la répartition des deux modalités des deux variables étaient dûes au hasard. Pour le croisement de deux modalités, la fréquence théorique est égale au produit des marges divisé par le nombre total d'observations ($ft_{ij} = (n_{i\cdot} n_{\cdot j}) / n_{\cdot\cdot}$). Par exemple, la fréquence théorique pour le croisement des modalités *moins de 25 logements* et *avant 1975* est égale à : $(80 \times 56) / 279 = 16,06$. Nous observons ici que la valeur théorique (16,06) est bien supérieure à la valeur réelle (6). On observe donc moins de HLM de moins de 25 logements avant 1975 que ce que l'on pourrait attendre du hasard.
- **La déviation** est la différence entre la fréquence observée et la fréquence théorique ($f_{ij} - ft_{ij}$). Plus la déviation est grande, plus on s'écarte d'une situation d'indépendance entre les deux modalités i et j . La somme des déviations sur une ligne ou sur une colonne est nulle. Si la déviation ij est nulle, la fréquence théorique est égale à la fréquence observée, ce qui signifie qu'il y a indépendance entre les modalités i et j . Une déviation positive traduit, quant à elle, une attraction entre les modalités i et j , ou autrement dit, une surreprésentation du phénomène ij ; tandis qu'une déviation négative renvoie à une répulsion entre les modalités i et j , soit une sous-représentation du phénomène ij . Dans le cas précédent, on observait 6 habitations de moins de 25 logements construits avant 1975 et une fréquence théorique de 16,0. La déviation est donc -10,06, soit une sous-représentation du phénomène.
- **La contribution au khi^2** est égale à la déviation au carré divisée par la fréquence théorique : $\chi^2_{ij} = (f_{ij} - ft_{ij})^2 / ft_{ij}$. Plus sa valeur est forte, plus il y a association entre les deux modalités. La somme des contributions au khi^2 représente le khi^2 total pour l'ensemble du tableau de contingence (ici à 63,54) que nous abordons dans la section suivante.

```
## 
##   Cell Contents
##   |-----|
##   |           Count |
##   |           Expected Values |
##   |           Chi-square contribution |
##   |           Row Percent |
##   |           Column Percent |
##   |           Total Percent |
##   |           Residual |
##   |-----|
## 
## Total Observations in Table:  279
## 
##           | TabKhi2$Periode
## TabKhi2$Taille | Av. 1975 | 1975-79 | 1980-84 | 1985-89 | 1990-94 | Row Total |
## -----|-----|-----|-----|-----|-----|-----|
## < 25 log. |       6 |      11 |       8 |      14 |      41 |       80 |
##             | 16.06 | 13.76 | 13.76 | 13.48 | 22.94 |          |
##             | 6.30 | 0.55 | 2.41 | 0.02 | 14.22 |          |
##             | 7.50% | 13.75% | 10.00% | 17.50% | 51.25% | 28.67% |
##             | 10.71% | 22.92% | 16.67% | 29.79% | 51.25% |          |
```

```

##          | 2.15% | 3.94% | 2.87% | 5.02% | 14.70% |          |
##          | -10.06 | -2.76 | -5.76 | 0.52 | 18.06 |          |
## -----
## 25-49 | 10 | 5 | 8 | 8 | 22 | 53 |
##          | 10.64 | 9.12 | 9.12 | 8.93 | 15.20 |          |
##          | 0.04 | 1.86 | 0.14 | 0.10 | 3.05 |          |
##          | 18.87% | 9.43% | 15.09% | 15.09% | 41.51% | 19.00% |
##          | 17.86% | 10.42% | 16.67% | 17.02% | 27.50% |          |
##          | 3.58% | 1.79% | 2.87% | 2.87% | 7.89% |          |
##          | -0.64 | -4.12 | -1.12 | -0.93 | 6.80 |          |
## -----
## 50-99 | 20 | 21 | 22 | 21 | 15 | 99 |
##          | 19.87 | 17.03 | 17.03 | 16.68 | 28.39 |          |
##          | 0.00 | 0.92 | 1.45 | 1.12 | 6.31 |          |
##          | 20.20% | 21.21% | 22.22% | 21.21% | 15.15% | 35.48% |
##          | 35.71% | 43.75% | 45.83% | 44.68% | 18.75% |          |
##          | 7.17% | 7.53% | 7.89% | 7.53% | 5.38% |          |
##          | 0.13 | 3.97 | 4.97 | 4.32 | -13.39 |          |
## -----
## 100 et + | 20 | 11 | 10 | 4 | 2 | 47 |
##          | 9.43 | 8.09 | 8.09 | 7.92 | 13.48 |          |
##          | 11.83 | 1.05 | 0.45 | 1.94 | 9.77 |          |
##          | 42.55% | 23.40% | 21.28% | 8.51% | 4.26% | 16.85% |
##          | 35.71% | 22.92% | 20.83% | 8.51% | 2.50% |          |
##          | 7.17% | 3.94% | 3.58% | 1.43% | 0.72% |          |
##          | 10.57 | 2.91 | 1.91 | -3.92 | -11.48 |          |
## -----
## Column Total | 56 | 48 | 48 | 47 | 80 | 279 |
##          | 20.07% | 17.20% | 17.20% | 16.85% | 28.67% |          |
## -----
## 
## 
## Statistics for All Table Factors
## 
## 
## Pearson's Chi-squared test
## -----
## Chi^2 = 63.54291      d.f. = 12      p = 5.063109e-09
## 
## 
## 
## Minimum expected frequency: 7.917563

```

4.2.2 Test du χ^2

Avec le test du χ^2 , on postule qu'il y a indépendance entre les modalités des deux variables qualitatives, soit l'hypothèse nulle (H_0). Puis, on calcule le nombre de degrés de liberté : $DL = (n - 1)(l - 1)$ avec l et n étant respectivement les nombres de modalités en ligne et en colonne. Pour notre tableau de contingence, nous avons 12 degrés de liberté : $(4 - 1)(5 - 1) = 12$. À partir du nombre de degré de liberté et d'un seuil critique de significativité (prenons 5% ici), nous pouvons trouver la valeur critique

de khi^2 dans le tableau des valeurs critiques du khi^2 : 21,03 section 12.1). Puisque la valeur du khi^2 calculé dans le tableau de contingence est bien supérieure à celle obtenue dans le tableau des valeurs critiques (63,54), on peut rejeter l'hypothèse d'indépendance au seuil de 5%. Autrement dit, si les deux variables n'étaient pas associées, nous aurions eu moins de 5% de chances de collecter des données avec ce niveau d'association, ce qui nous permet de rejeter l'hypothèse nulle (absence d'association). Notez que le test reste significatif avec des seuils de 1% ($p=0,01$) et 0,1% ($p=0,001$) puisque les valeurs critiques sont de 26,22 et 32,91.

Bien entendu, une fois que l'on connaît le nombre de degrés de liberté, on peut directement calculer les valeurs critiques pour différents seuils de signification et éviter ainsi de recourir à la table ci-dessus. Dans la même veine, on peut aussi calculer la valeur de p d'un tableau de contingence en spécifiant le nombre de degrés de liberté et la valeur du khi^2 obtenue.

```
cat("Valeurs critiques du khi2 avec le nombre de degrés de liberté", "\n",
  round(qchisq(p=0.95, df=12, lower.tail = FALSE),3), "avec p=0,05", "\n",
  round(qchisq(p=0.99, df=12, lower.tail = FALSE),3), "avec p=0,01", "\n",
  round(qchisq(p=0.999, df=12, lower.tail = FALSE),3), "avec p=0,0001")
```

```
## Valeurs critiques du khi2 avec le nombre de degrés de liberté
## 5.226 avec p=0,05
## 3.571 avec p=0,01
## 2.214 avec p=0,0001
```

```
cat("Valeurs de p du Khi2 obtenu (63.54291) avec 12 degrés de liberté :", "\n",
  pchisq(q=63.54291, df=12, lower.tail = FALSE))
```

```
## Valeurs de p du Khi2 obtenu (63.54291) avec 12 degrés de liberté :
## 5.063101e-09
```



Outre le khi^2 , d'autres mesures d'association permettent de mesurer le degré d'association entre deux variables qualitatives. Les plus courantes sont reportées au tableau ci-dessous. À des fins de comparaison, le khi^2 décrit précédemment est aussi reporté sur la première ligne du tableau.

Statistique	Formule	Propriété et interprétation
khi^2	$\chi^2 = \sum \frac{(f_{ij} - ft_{ij})^2}{ft_{ij}}$	Mesure classique du Khi^2 calculé à partir des différences entre les fréquences observées et attendues. Valeur de p disponible.
Ratio de vraisemblance du khi^2	$G^2 = 2 \sum f_{ij} \ln \left(\frac{f_{ij}}{ft_{ij}} \right)$	Calculé à partir du ratio entre les fréquences observées et attendues. Valeur de p disponible.
khi^2 de Mantel-Haenszel	$Q_{MH} = (N - 1)r^2$	avec r étant le coefficient de corrélation entre les deux variables qualitatives; par exemple, entre les valeurs des modalités de 1 à 5 de la variable <i>période de construction</i> et celles de 1 à 4 de la variable <i>taille du projet HLM</i> . Ce coefficient est très utile quand les deux variables qualitatives ne sont pas nominales, mais ordinaires . Valeur de p disponible.
Corrélation polychorique	obtenue itérativement par maximum de vraisemblance	Dans le même esprit que le khi^2 de Mantel-Haenszel, la corrélation polychorique s'applique à deux variables ordinaires. Plus spécifiquement, elle formule le postulat que deux variables théoriques normalement distribuées ont été mesurées de façon approximative avec deux échelles ordinaires. Par exemple, en psychologie, le sentiment de bien être et le sentiment de sécurité peuvent être conceptualisés comme deux variables continues normalement distribuées. Cependant, les mesurer directement est très difficile, on a donc recours à des échelles de Likert allant de 1 à 10. Pour cet exemple, il serait pertinent d'utiliser la corrélation polychorique. Comme une corrélation de Pearson, la

Coefficient Phi $\phi = \sqrt{\frac{\chi^2}{n}}$

Simplement le Khi² divisé par le nombre d'observations. Si les deux variables qualitatives comprennent deux modalités chacune (tableau 2x2 dimensions) alors ϕ varie de -1 à 1; sinon de 0 à $\min(\sqrt{c-1}, \sqrt{l-1})$ avec c et l étant le nombre de modalités en colonne et en ligne. Par conséquent, ce coefficient est peu utile pour les tableaux de plus de 2x2 dimensions. Pas de valeur de p disponible.

V de Cramer $V = \sqrt{\frac{\chi^2/n}{\min(c-1, l-1)}}$

Il représente un ajustement du coefficient Phi et varie de 0 à 1. Plus sa valeur est forte plus les deux variables sont associées. À la lecture des deux formules, vous constaterez que pour un tableau de 2 x 2, la valeur du V de Carmer sera égale à celle du Coefficient Phi. Pas de valeur de p disponible.

4.2.3 Mise en œuvre dans

Pour calculer le Khi² entre deux variables qualitatives, on utilise la fonction de base : `chisq.test(x = ..., y = ...)` qui renvoie le nombre de degré de liberté, les valeurs du Khi² et de p .

```
# Importation du csv
dataHLM <- read.csv("data/bivariee/hlm.csv")
# Calcul du Khi2 avec la fonction de base chisq.test
chisq.test(x = dataHLM$Taille, y = dataHLM$Periode)

##
## Pearson's Chi-squared test
##
## data: dataHLM$Taille and dataHLM$Periode
## X-squared = 63.543, df = 12, p-value = 5.063e-09
```

Pour la construction du tableau de contingence, deux options sont possibles dépendamment de la structuration de votre tableau de données initial. Premier cas de figure, votre tableau comprend une ligne par observation avec les différentes modalités dans deux colonnes (ici *Periode* et *Taille*). Dans la syntaxe ci-dessous, pour chacune des deux variables qualitatives, on crée un facteur afin de spécifier un intitulé à chaque modalité (`factor(levels = c(...), labels = c(..))`). Puis, on utilise la fonction `CrossTable` du package **gmodels**. Pour obtenir les fréquences théoriques, les contributions locales au Khi² et les déviations, on spécifie les options suivantes : `expected=TRUE, chisq=TRUE, resid=TRUE`.

```
library("gmodels")
#Premiers enregistrements du tableau
head(dataHLM)
```

```
##   Periode Taille
## 1      5     1
## 2      5     1
## 3      5     2
## 4      5     1
## 5      5     1
## 6      5     2
```

```

# La variable Periode comprend 5 modalités (de 1 à 5)





```

Deuxième cas de figure, vous disposez déjà d'un tableau de contingence, soit les fréquences observées (f_{ij}). On n'utilise donc pas la fonction `CrossTable`, mais directement la fonction `chisq.test`.

```

# Importation des données
df <- read.csv("data/bivariee/data_transport.csv", stringsAsFactors = FALSE)
df # Visualisation du tableau

##                               ModeTransport Homme Femme
## 1 Automobile, camion ou fourgonnette - conducteur 689400 561830
## 2 Automobile, camion ou fourgonnette - passager   21315  40010
## 3 Transport en commun                      181435 238330
## 4 A pied                                43715  54360
## 5 Bicyclette                            24295  13765
## 6 Autre moyen                           8395   6970

```

```

Matrice <- as.matrix(df[, c("Homme", "Femme")])
dimnames(Matrice) <- list(unique(df$ModeTransport), Sexe=c("Homme", "Femme"))
# Notez que vous pouvez saisir vos données directement si vous avez peu d'observations
Femme <- c(689400, 21315, 181435, 43715, 24295, 8395) # Vecteur de valeurs pour les femmes
Homme <- c(561830, 40010, 238330, 54360, 13765, 6970) # Vecteur de valeurs pour les hommes
Matrice <- as.table(cbind(Femme, Homme)) # Création du tableau
# Nom des deux variables et de leurs modalités respectives
dimnames(Matrice) <- list(transport=c("Automobile (conducteur)",
                                         "Automobile (passager)",
                                         "Transport en commun",
                                         "A pied",
                                         "Bicyclette",
                                         "Autre moyen"),
                           sexe=c("Homme", "Femme"))

# Test du Khi2
test <- chisq.test(Matrice)
print(test)

```

```

## 
## Pearson's Chi-squared test
##
## data: Matrice
## X-squared = 29134, df = 5, p-value < 2.2e-16

```

```

# Fréquences observées (Fij)
test$observed

```

```

##                      sexe
## transport           Homme Femme
##   Automobile (conducteur) 689400 561830
##   Automobile (passager)    21315  40010
##   Transport en commun     181435 238330
##   A pied                  43715  54360
##   Bicyclette                24295  13765
##   Autre moyen                 8395   6970

```

```

# Fréquences théoriques (FTij)
round(test$expected, 0)

```

```

##                      sexe
## transport           Homme Femme
##   Automobile (conducteur) 643313 607917
##   Automobile (passager)    31530  29795
##   Transport en commun     215820 203945
##   A pied                  50425  47650
##   Bicyclette                19568  18492
##   Autre moyen                 7900   7465

```

```

# Déviations (Fij - FTij)
round(test$observed-test$expected, 0)

```

```
##                   sexe
## transport           Homme   Femme
## Automobile (conducteur) 46087 -46087
## Automobile (passager)   -10215  10215
## Transport en commun    -34385  34385
## A pied                -6710   6710
## Bicyclette              4727   -4727
## Autre moyen             495    -495
```

```
# Contributions au Khi2
round((test$observed-test$expected)^2/test$expected,2)
```

```
##                   sexe
## transport           Homme   Femme
## Automobile (conducteur) 3301.74 3493.98
## Automobile (passager)   3309.37 3502.05
## Transport en commun     5478.22 5797.18
## A pied                892.81  944.80
## Bicyclette              1141.71 1208.19
## Autre moyen             31.04   32.85
```

```
# Marges en lignes et en colonnes
colSums(Matrice)
```

```
## Homme   Femme
## 968555 915265
```

```
rowSums(Matrice)
```

```
## Automobile (conducteur)  Automobile (passager)  Transport en commun
##                      1251230                  61325          419765
##                      A pied                  Bicyclette      Autre moyen
##                      98075                  38060          15365
```

```
# Grand total
sum(Matrice)
```

```
## [1] 1883820
```

```
# Pourcentages
round(Matrice/sum(Matrice)*100,2)
```

```
##                   sexe
## transport           Homme   Femme
## Automobile (conducteur) 36.60 29.82
## Automobile (passager)   1.13  2.12
## Transport en commun     9.63 12.65
## A pied                2.32  2.89
## Bicyclette              1.29  0.73
```

```
## Autre moyen          0.45  0.37
```

```
# Pourcentages en ligne
round(Matrice/rowSums(Matrice)*100,2)
```

```
##                      sexe
## transport           Homme Femme
## Automobile (conducteur) 55.10 44.90
## Automobile (passager)   34.76 65.24
## Transport en commun    43.22 56.78
## A pied               44.57 55.43
## Bicyclette            63.83 36.17
## Autre moyen            54.64 45.36
```

```
# Pourcentages en colonne
round(Matrice/colSums(Matrice)*100,2)
```

```
##                      sexe
## transport           Homme Femme
## Automobile (conducteur) 71.18 58.01
## Automobile (passager)   2.33  4.37
## Transport en commun    18.73 24.61
## A pied               4.78  5.94
## Bicyclette            2.51  1.42
## Autre moyen            0.92  0.76
```

Pour obtenir les autres mesures d'association, on pourra utiliser la syntaxe suivante :

```
df <- read.csv("data/bivariee/hlm.csv")
# Fonction pour calculer les autres mesures d'association
AutresMesuresKhi2 <- function(x, y){
  testChi2 <- chisq.test(x, y) # Calcul du Chi2
  n <- sum(testChi2$observed) # Nombre d'observations
  c <- ncol(testChi2$observed) # Nombre de colonnes
  l <- nrow(testChi2$observed) # Nombre de lignes
  dl <- (c-1)*(l-1) # Nombre de degrés de libertés
  chi2 <- testChi2$statistic # Khi2
  Pchi2 <- testChi2$p.value # P pour le Khi2

  #Ratio de vraisemblance du khi2
  G <- 2*sum(testChi2$observed*log(testChi2$observed/testChi2$expected)) # G2
  PG <- pchisq(G, df=dl, lower.tail = FALSE) # P pour le G22

  # khi2 de Mantel-Haenszel avec la librarie DescTools
  MHTest <- DescTools::MHChisqTest(testChi2$observed)
  MH <- MHTest$statistic
  PMH <- MHTest$p.value

  # Coefficient de correlation Polychorique
  df <- data.frame("x" = as.factor(x),
                    "y" = as.factor(y))
  polychoricCorr <- correlation::cor_test(df,"x","y",method = "polychoric")
```

```

polyR <- polychoricCorr$rho
polyP <- polychoricCorr$p

# Coefficient Phi et V de Cramer
phi <- sqrt(chi2/n)
vc <- sqrt(chi2/(n*min(c-1,l-1)))

# Tableau pour les sorties
dfsoutie <- data.frame(
  Statistique = c("Khi2",
                 "Ratio de vraisemblance du khi",
                 "Khi2 de Mantel-Haenszel",
                 "Corrélation Polychoric",
                 "Coefficient de Phi",
                 "V de Cramer"),
  Valeur = round(c(Pchi2, G, MH, polyR, phi, vc),3),
  P = round(c(Pchi2, PG, PMH, polyP , NA, NA),10))
return(dfsoutie)
}

dfkhi2 <- AutresMesuresKhi2(df$Periode, df$Taille)

show_table(dfkhi2,
           caption="Mesures d'association entre deux variables qualitatives",
           digits = 3
)

```

4.2.4 Interprétation d'un tableau de contingence

Nous vous proposons une démarche très simple pour vérifier l'association entre deux variables qualitatives avec les étapes suivantes :

- On pose l'hypothèse nulle (H_0), soit l'indépendance entre les deux variables. Si le Khi^2 total du tableau de contingence est inférieur à la valeur critique du Khi^2 avec $p=0,05$ et le nombre de degrés de liberté de la table T , alors il y a bien indépendance. La valeur de p sera alors supérieure à 0,05. L'analyse s'arrête donc là! Autrement dit, il n'est pas nécessaire d'analyser le contenu de votre tableau de contingence puisqu'il n'y pas d'associations significatives entre les modalités des deux variables. Vous pouvez simplement signaler que : selon les résultats du test du Khi^2 , il n'y a pas d'association significative entre les deux variables ($\chi = \dots$ avec $p= \dots$).
- S'il y a dépendance ($khi_{observ}^2 > khi_{critique}^2$), trouver les cellules ij où les contributions au Khi^2 sont les plus fortes, c'est-à-dire où les liens entre les modalités i de la variable en ligne et les

TAB. 4.7 : Mesures d'association entre deux variables qualitatives

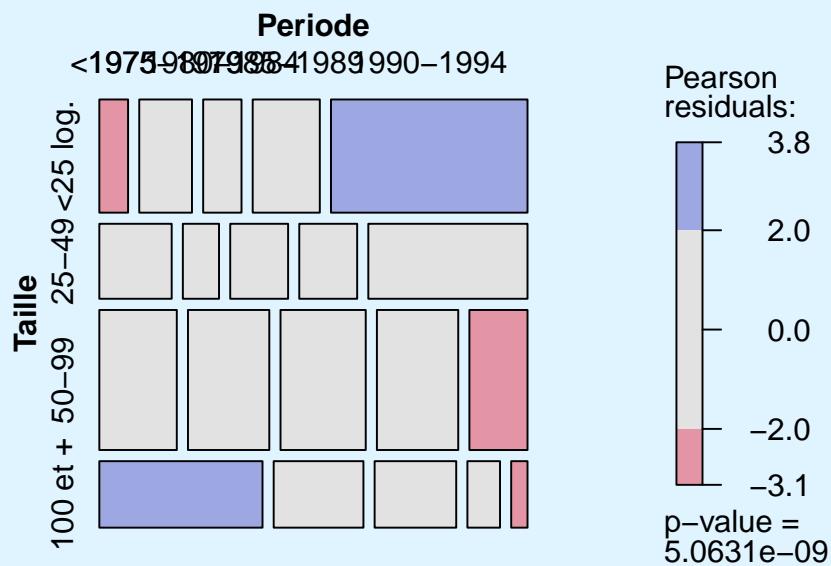
Statistique	Valeur	P
Khi2	0,000	0
Ratio de vraisemblance du khi	67,286	0
Khi2 de Mantel-Haenszel	48,486	0
Corrélation Polychoric	-0,479	0
Coefficient de Phi	0,477	-
V de Cramer	0,276	-

modalités j de la variable en colonne sont les plus marqués. Pour ces cellules, le phénomène ij est surreprésenté si la déviation est positive ou sous-représenté si la déviation est négative. Commentez ces associations et utilisez les pourcentages en lignes ou en colonnes pour appuyer vos propos.



Pour repérer rapidement les cellules où les contributions au χ^2 sont les plus fortes, vous pouvez construire un graphique avec la fonction **mosaic** du package **vcd**. À la figure ??, la taille des rectangles représentent les effectifs entre les deux modalités tandis que les associations sont représentées comme suit : en gris lorsqu'elles ne sont pas significatives, en rouge pour des déviations significatives et négatives et en bleu pour des déviations significatives et positives.

```
library(vcd)
mosaic(~ Taille+Periode, data=dataHLM, shade=TRUE, legend=TRUE)
```



Exemple d'interprétation. «Les résultats du test du χ^2 signalent qu'il existe des associations entre les modalités de la taille et de la période de construction des projets d'habitation ($\chi^2 = 63,5, p < 0,001$). Les fortes contributions au χ^2 et le signe positif ou négatif des déviations correspondantes permettent de repérer cinq associations majeures entre les modalités de taille et de période de construction des projets HLM : **1)** la répulsion entre les projets d'habitation de moins de 25 logements et la période de construction 1964-1974 ; **2)** l'attraction entre les projets d'habitation de 100 logements et plus et la période de construction de 1969-1974 ; **3)** l'attraction entre les projets d'habitation de moins de 25 logements et la période de construction de 1990-1994 ; **4)** la répulsion entre les projets d'habitation de 50 à 99 logements et la période de construction 1990-1994 ; **5)** la répulsion entre les projets d'habitation de 100 logements et plus et la période de construction 1990-1994. On observe donc une tendance bien marquée dans l'évolution du type de construction entre 1970 et 1994 : entre 1969 et 1974, on construit habituellement de grandes habitations dépassant souvent 100 logements ; du milieu des années 1970 à la fin des années 1980, on priviliege la construction d'habitations de taille plus modeste, entre 50 et 100 logements ; tandis qu'au début des années 1990, on opte plutôt pour des habitations de taille réduite (moins de 50 logements). Quelques chiffres à l'appui : sur les 56 habitations réalisées entre 1969 et 1974, 20 ont plus de 100 logements, 20 comprennent entre 50 et 99 logements et seules 10 ont moins de 25 logements. Près de la moitié des habitations construites entre 1975 et 1989 regroupent 50 à 99 logements (43,8% pour la période 1975-1979, 45,8% pour 1980-1984 et 44,7% pour 1985-1989). Par contre, 51% des logements érigés à partir de 1990 disposent de moins de 25 logements» (Apparicio, 2002, 117-118). Notez que cette évolution décroissante est aussi soutenue par le coefficient négatif de la corrélation polychorique.

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

Vous pouvez aussi construire un graphique pour appuyer vos constats, soit avec les pourcentages en ligne ou en colonne (figure 4.15 tirée de Apparicio and Séguin (2006)).

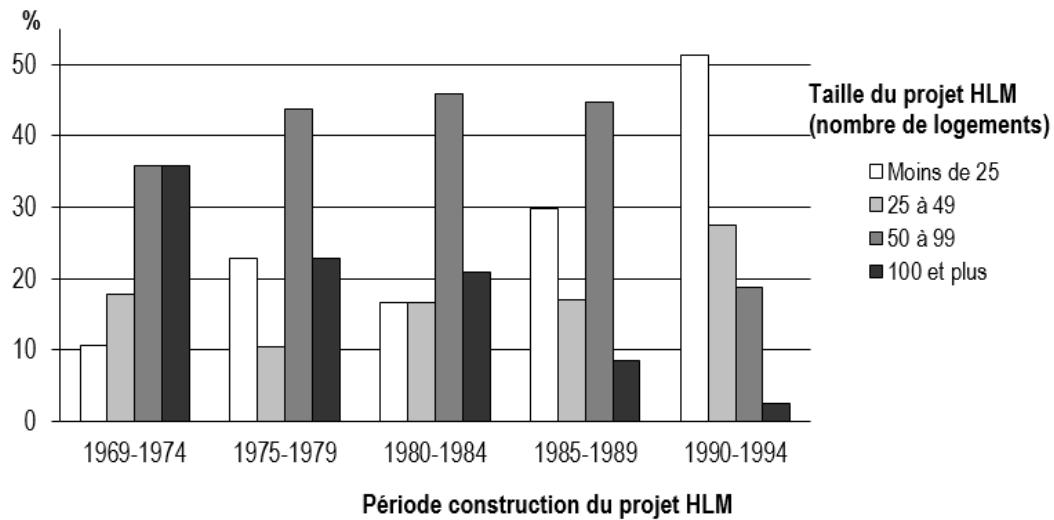


FIG. 4.15 : Taille des ensembles HLM selon la période de construction

Comment rapporter succinctement les résultats d'un Test du Khi² ?

Le test du Khi² a été réalisé pour examiner la relation entre la taille et la période de construction des habitations HLM. Cette relation est significative : $\chi^2(12, N = 279) = 63,5, p < 0,001$. Plus les projets ont été construits récemment, plus ils sont de taille réduite.

Pour un texte en anglais, vous pourrez consulter <https://www.socscistatistics.com/tutorials/chisquare/default.aspx>.

4.3 Relation entre une variable quantitative et une variable qualitative à deux modalités



Les moyennes de deux groupes de population sont-elles significativement différentes ? On souhaite ici comparer deux groupes de population en fonction d'une variable continue. Par exemple, pour deux échantillons respectivement d'hommes et de femmes travaillant dans le même secteur d'activité, on pourrait souhaiter vérifier si les moyennes des salaires des hommes et des femmes sont différentes et ainsi vérifier la présence ou l'absence d'une iniquité systématique. En études urbaines, dans le cadre d'une étude sur un espace public, on pourrait vouloir vérifier si la différence des moyennes du sentiment de sécurité des femmes et des hommes est significative (c'est-à-dire différente de 0).

Pour un même groupe, la moyenne de la différence d'un phénomène donné mesuré à deux moments est-elle ou non égale à zéro ? Autrement dit, on cherche à comparer un même groupe d'individus avant et après une expérimentation, ou dans deux contextes différents. Prenons un exemple d'application en études urbaines. Dans le cadre d'une étude sur la perception des risques associés à la pratique du vélo en ville, 50 individus utilisant habituellement l'automobile pour se rendre au travail sont recrutés. L'expérimentation pourrait consister à leur donner une formation sur la pratique du vélo en ville et à les accompagner quelques jours durant leurs déplacements domicile-travail. On évaluera la différence de leurs perceptions des risques associés à la pratique du vélo sur une échelle de 0 à 100 avant et après l'expérimentation. On pourrait supposer que la moyenne des différences est significativement négative, ce qui indiquerait que la perception du risque a diminué après l'expérimentation ; autrement dit, la perception du risque serait plus faible en fin de période.

4.3.1 Test *t* et ses différentes variantes

Le **t de student**, appelé aussi **test *t*** (*t-test* en anglais), est un test paramétrique permettant de comparer les moyennes de deux groupes (échantillons), qui peuvent être indépendantes ou non :

- **Échantillons indépendants (dits non appariés)**, les observations de deux groupes qui n'ont aucun lien entre eux. Par exemple, on souhaite vérifier si les moyennes du sentiment de sécurité des hommes et des femmes, ou encore si, les moyennes des loyers entre deux villes sont statistiquement différentes. Ainsi, les tailles des deux échantillons peuvent être différentes ($n_a \neq n_b$).
- **Échantillons dépendants (dits appariés)**, les individus des deux groupes sont les mêmes et sont donc associés par paires. Autrement dit, on a deux séries de valeurs de taille identique $n_a = n_b$ et n_{ai} est le même individu que n_{bi} . Ce type d'analyse est souvent utilisée en études cliniques : pour n individus, on dispose d'une mesure quantitative de leur état de santé pour deux séries (l'une avant le traitement, l'autre une fois le traitement terminé). Cela permet de comparer les mêmes individus avant et après un traitement, une expérimentation ; on parle alors d'étude, d'expérience et d'analyse pré-post. Concrètement, on cherche à savoir si la moyenne des différences des observations avant et après est significativement différente de 0. Si c'est le cas, on peut en conclure que l'expérimentation a eu un impact sur le phénomène mesuré (variable continue). Ce type d'analyse pré-post peut aussi être utilisé pour évaluer l'impact du réaménagement d'un espace public (rue commerciale, place publique, parc, etc.). Par exemple, on pourrait questionner le même échantillon de commerçants ou d'usagers avant et après le réaménagement d'une artère commerciale.

Condition d'application. Pour utiliser les tests de Student et de Welch, la variable continue doit être normalement distribuée. Si elle est fortement anormale, on utilisera le test non paramétrique de Wilcoxon (section 4.3.2). Il existe trois principaux tests pour comparer les moyennes de deux groupes :

- Test de Student (test *t*) avec échantillons indépendants et variances similaires (méthode *pooled*). Les variances de deux groupes sont semblables quand leur ratio varie de 0,5 à 2 ($0,5 < (S_{X_A}^2 / S_{X_B}^2) < 2$).
- Test de Welch (appelé aussi Satterthwaite) avec échantillons indépendants quand les variances des deux groupes sont dissemblables.
- Test de Student (test *t*) avec échantillons dépendants.

Il s'agit de vérifier si les moyennes des deux groupes sont statistiquement différentes avec les étapes suivantes :

- On pose l'hypothèse nulle (H_0), soit que les moyennes des deux groupes A et B ne sont pas différentes ($\bar{X}_A = \bar{X}_B$) ou autrement dit, la différence des deux moyennes est nulle ($\bar{X}_A - \bar{X}_B = 0$). L'hypothèse alternative (H_1) est donc $\bar{X}_A \neq \bar{X}_B$.
- On calcule la valeur de t et le nombre de degrés de liberté. La valeur de t sera négative quand la moyenne du groupe A est inférieure au groupe B et inversement.
- On compare la valeur absolue de t ($|T|$) avec celle issue de la table des valeurs critiques T avec le bon nombre de degrés de liberté et en choisissant un degré de signification (habituellement, $p=0,05$). Si ($|t|$) est supérieure à la valeur t critique, alors les moyennes sont statistiquement différentes au degré de signification retenu.
- Si les moyennes sont statistiquement différentes, on peut calculer la taille de l'effet.

Cas 1. Test de student pour des échantillons indépendants avec variances égales (méthode *pooled*). La valeur de t est le ratio entre la différence des moyennes des deux groupes (numérateur) et l'erreur-type groupée des deux échantillons (dénominateur) :

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

$$t = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{\frac{S_p^2}{n_A} + \frac{S_p^2}{n_B}}} \text{ avec } S_p^2 = \frac{(n_A-1)S_{X_A}^2 + (n_B-1)S_{X_B}^2}{n_A+n_B-2}$$

avec n_A, n_B , $S_{X_A}^2$ et $S_{X_B}^2$ étant respectivement les nombres d'observations et les variances pour les groupes A et B , S_p^2 étant la variance groupée des deux échantillons et $n_A + n_B - 2$ étant le nombre de degrés de liberté.

Cas 2. Test de Welch pour des échantillons indépendants (avec variances différentes). Le test de Welch est très similaire au test de student; seul le calcul de la valeur de T est différent, pour tenir compte des variances respectives des groupes :

$$t = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{\frac{S_A^2}{n_A} + \frac{S_B^2}{n_B}}} \text{ et } dl = \frac{\left(\frac{S_{X_A}^2}{n_A} + \frac{S_{X_B}^2}{n_B} \right)^2}{\frac{S_{X_A}^4}{n_A^2(n_A-1)} + \frac{S_{X_B}^4}{n_B^2(n_B-1)}}$$

Dans la syntaxe ci-dessous, nous avons écrit une fonction dénommée `test_independants` permettant de calculer les deux tests pour des échantillons indépendants. Dans cette fonction, vous pourrez repérer comment sont calculés les moyennes, nombres d'observations et variances pour les deux groupes, le nombre de degrés de liberté, les valeurs de t et de p pour les deux tests. Puis, nous avons créé aléatoirement deux jeux de données relativement à la vitesse de déplacement de cyclistes utilisant un vélo personnel ou un vélo en libre service (généralement plus lourd et moins utilisé par des cyclistes expérimentés) :

- Au cas 1, 60 cyclistes utilisant un vélo personnel roulant en moyenne à 18 km/h (écart-type de 1,5) et 50 utilisateurs du système de vélo partage avec une vitesse moyenne de 15 km/h (écart-type de 1,5).
- Au cas 2, 60 cyclistes utilisant un vélo personnel roulant en moyenne à 16 km/h (écart-type de 3) et 50 utilisateurs du système de vélo partage avec une vitesse moyenne de 15 km/h (écart-type de 1,5). Ce faible écart des moyennes, combiné à une plus forte variance va réduire la significativité de la différence entre les deux groupes.

D'emblée, l'analyse visuelle des boîtes à moustaches (figure 4.16) signale qu'au cas 1 contrairement au cas 2, les groupes sont plus homogènes (boîtes plus compactes) et les moyennes semblent différentes (les boîtes sont centrées différemment sur l'axe des ordonnées). Cela est confirmé par les résultats des tests.

```
library("ggplot2")
library("ggpubr")
# fonction -----
tstudent_independants <- function(A, B){
  x_a <- mean(A)          # Moyenne du groupe A
  x_b <- mean(B)          # Moyenne du groupe B
  var_a <- var(A)          # Variance du groupe A
  var_b <- var(B)          # Variance du groupe B
  sd_a <- sqrt(var_a)      # Écart-type du groupe A
  sd_b <- sqrt(var_b)      # Écart-type du groupe B
  ratio_v <- var_a / var_b # ratio des variances
  n_a <- length(A)         # nombre d'observation du groupe A
  n_b <- length(B)         # nombre d'observation du groupe B

  # T-test (variances égales)
  dl_test <- n_a+n_b-2     # degrés de liberté
```

```

PooledVar <- (((n_a-1)*var_a)+((n_b-1)*var_b))/dl_test
t_test <- (x_a-x_b) / sqrt(((PooledVar/n_a)+(PooledVar/n_b)))
p_test <- 2*(1-(pt(abs(t_test), dl_test)))
# Test Welch-Satterwaite (variances inégales)
t_welch <- (x_a-x_b) / sqrt( (var_a/n_a) + (var_b/n_b))
dl_num = ((var_a/n_a) + (var_b/n_b))^2
dl_dem = ((var_a/n_a)^2/(n_a-1)) + ((var_b/n_b)^2/(n_b-1))
dl_welch = dl_num / dl_dem # degrés de liberté
p_welch <- 2*(1-(pt(abs(t_welch), dl_welch)))

cat("\n groupe A (n = ", n_a,), moy = ", round(x_a,1),",
    variance = ", round(var_a,1)," , écart-type = ", round(sd_a,1),
"\n groupe B (n = ", n_b,), moy = ", round(x_b,1)," ,
    variance = ", round(var_b,1)," , écart-type = ", round(sd_b,1),
"\n ratio variance = ",round(ratio_v,2),
"\n t-test (variances égales): t(dl = ", dl_test, ") = ",round(t_test,4),
", p = ", round(p_test,6),
"\n t-Welch (variances inégales): t(dl = ", round(dl_welch,3), ") = ",
round(t_welch,4), ", p = ", round(p_welch,6), sep="")

if (ratio_v > 0.5 && ratio_v < 2) {
  cat("\n Variances similaires. Utilisez le test de Student !")
  p <- p_test
} else {
  cat("\n Variances dissemblables. Utilisez le test de Welch-Satterwaite !")
  p <- p_welch
}

if (p <=.05){
  cat("\n Les moyennes des deux groupes sont significativement différentes.")
} else {
  cat("\n Les moyennes des deux groupes ne sont pas significativement différentes.")
}

# CAS 1 : données fictives -----
# Création du groupe A : 60 observations avec une vitesse moyenne de 18 et un écart-type de 1,5
Velo1A <- rnorm(60,18,1.5)
# Création du groupe B : 50 observations avec une vitesse moyenne de 15 et un écart-type de 1,5
Velo1B <- rnorm(50,15,1.5)
df1 <- data.frame(
  vitesse = c(Velo1A,Velo1B),
  type = c(rep("Vélo personnel",length(Velo1A)), rep("Vélo partage",length(Velo1B)))
)
boxplot1 <- ggplot(data=df1, mapping=aes(x=type,y=vitesse, colour=type)) +
  geom_boxplot(width=0.2) +
  ggtitle("Données fictives (cas 1)") +
  xlab("Type de vélo") +
  ylab("Vitesse de déplacement (km/h)") +
  theme(legend.position = "none")

# CAS 2 : données fictives -----
# Création du groupe A : 60 observations avec une vitesse moyenne de 18 et un écart-type de 3
Velo2A <- rnorm(60,16,3)
# Création du groupe B : 50 observations avec une vitesse moyenne de 15 et un écart-type de 1,5
Velo2B <- rnorm(50,15,1.5)
df2 <- data.frame(

```

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

```

vitesse = c(Velo2A,Velo2B),
type = c(rep("Vélo personnel",length(Velo2A)), rep("Vélo partage",length(Velo2B)))
)
boxplot2 <- ggplot(data=df2, mapping=aes(x=type,y=vitesse, colour=type)) +
  geom_boxplot(width=0.2) +
  ggtitle("Données fictives (cas 2)") +
  xlab("Type de vélo") +
  ylab("Vitesse de déplacement (km/h)") +
  theme(legend.position = "none")
ggarrange(boxplot1, boxplot2, ncol = 2, nrow = 1)

```

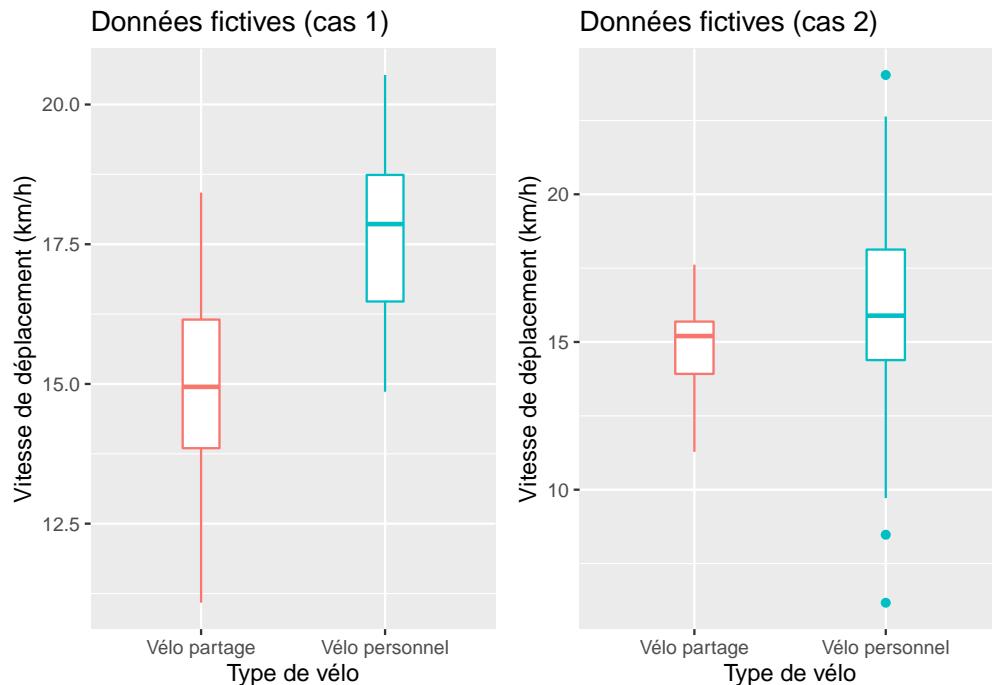


FIG. 4.16 : Boîtes à moustaches sur des échantillons fictifs non appariés

```

# Appel de la fonction pour le cas 1
tstudent_independants(Velo1A, Velo1B)

```

```

## 
##  groupe A (n = 60), moy = 17.7,
##          variance = 2.2, écart-type = 1.5
##  groupe B (n = 50), moy = 15,
##          variance = 2.4, écart-type = 1.5
##  ratio variance = 0.92
##  t-test (variances égales): t(dl = 108) = 9.3956, p = 0
##  t-Welch (variances inégales): t(dl = 102.695) = 9.3583, p = 0
##  Variances similaires. Utilisez le test de Student !
##  Les moyennes des deux groupes sont significativement différentes.

```

```

# Appel de la fonction pour le cas 2
tstudent_independants(Velo2A, Velo2B)

```

```

## groupe A (n = 60), moy = 15.8,
##           variance = 11, écart-type = 3.3
## groupe B (n = 50), moy = 14.9,
##           variance = 1.9, écart-type = 1.4
## ratio variance = 5.86
## t-test (variances égales): t(dl = 108) = 1.8042, p = 0.073991
## t-Welch (variances inégales): t(dl = 81.531) = 1.9254, p = 0.057663
## Variances dissemblables. Utilisez le test de Welch-Satterwaite !
## Les moyennes des deux groupes ne sont pas significativement différentes.

```

4.3.1.1 Principe de base et formulation pour des échantillons dépendants (appariés)

Nous disposons de plusieurs individus pour lesquelles nous avons mesuré un phénomène (variable continue) à deux temps différents : généralement avant et après une expérimentation (analyse pré-post). Il s'agit de vérifier si la moyenne des différences des observations avant et après la période est différente de 0. Pour ce faire, on réalise les étapes suivantes :

- On pose l'hypothèse nulle (H_0), soit que la moyenne des différences entre les deux séries est égale à 0 ($\bar{D} = 0$ avec $d = x_{t_1} - x_{t_2}$). L'hypothèse alternative (H_1) est donc $\bar{D} \neq 0$. Notez que l'on peut tester une autre valeur que 0.
- On calcule la valeur de t et le nombre de degrés de liberté. La valeur de t sera négative quand la moyenne des différences entre X_{t_1} et X_{t_2} est négative et inversement.
- On compare la valeur absolue de t ($|T|$) avec celle issue de la table des valeurs critiques T avec le bon nombre de degrés de liberté et en choisissant un degré de signification (habituellement, $p=0,05$). Si ($|t|$) est supérieure à la valeur t critique, alors les moyennes sont statistiquement différentes au degré de signification retenu.

Pour le test de student avec des échantillons appariées, la valeur de t se calcule comme suit :

$$t = \frac{\bar{D} - \mu_0}{\sigma_D / \sqrt{n}}$$

avec \bar{D} étant la moyenne des différences entre les observations appariées de la série A et de la série B, σ_D l'écart des différences, n le nombre d'observations, et finalement μ_0 la valeur de l'hypothèse nulle que l'on veut tester (habituellement 0). Bien entendu, il est possible fixer une autre valeur pour μ_0 : par exemple, avec $\mu_0 = 10$, on chercherait ainsi à vérifier si la moyenne des différences est significativement différente de 10. Le nombre de degrés de liberté sera égal à $n - 1$.

Dans la syntaxe ci-dessous, nous avons écrit une fonction dénommée `tstudent_dépendants` permettant de réaliser le test de student pour des échantillons appariés. Dans cette fonction, vous pourrez repérer comment sont calculés la différence entre les observations pairees, la moyenne et l'écart-type de cette différence, puis le nombre de degrés de liberté, les valeurs de t et de p pour les deux tests.

Pour illustrer l'utilisation de la fonction, nous avons créé aléatoirement deux jeux de données. Imaginons que ces données décrivent 50 personnes utilisant habituellement l'automobile pour se rendre au travail. Pour ces personnes, nous avons généré des valeurs du risque perçu de l'utilisation du vélo (de 0 à 100), et ce, avant et après une période de 20 jours ouvrables durant lesquels ils devaient impérativement se rendre au travail à vélo.

- Au cas 1, les valeurs de risque ont une moyenne de 70 avant l'expérimentation et de 50 après l'expérimentation, avec des écarts types de 5.

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

- Au cas 2, les valeurs de risque ont une moyenne de 70 avant et 66 après, avec des écarts types de 5.

D'emblee, l'analyse visuelle des boîtes à moustaches (figure 4.17) pairées montrent que la perception du risque semble avoir nettement diminé après l'expérimentation pour le cas 1 mais pas pour le cas 2. Cela est confirmé par les résultats des tests.

```

library("ggplot2")
library("ggpubr")
tstudent_dependants <- function(A, B, mu=0){
  d <- A-B           # différences entre les observations pairees
  moy <- mean(d)     # Moyenne des différences
  e_t <- sd(d)       # Ecart-type des différences
  n   <- length(A)   # nombre d'observations
  dl  <- n-1          # nombre de degrés de liberté (variances égales)

  t <- (moy-mu) / (e_t/sqrt(n)) # valeur de t
  p <- 2*(1-(pt(abs(t), dl)))

  cat("\n groupe A : moy = ", round(mean(A),1)," , var = ",
      round(var(A),1)," , sd = ", round(sqrt(var(A)),1),
      "\n groupe B : moy = ", round(mean(B),1)," , var = ",
      round(var(B),1)," , sd = ", round(sqrt(var(B)),1),
      "\n Moyenne des différences = ", round(mean(moy),1),
      "\n Ecart-type des différences = ", round(mean(e_t),1),
      "\n t(dl = ", dl, ") = ", round(t,2),
      ", p = ", round(p,3), sep="")

  if (p <=.05){
    cat("\n La moyenne des différences entre les échantillons est significative")
  }
  else{
    cat("\n La moyenne des différences entre les échantillons n'est pas significative")
  }
}

# CAS 1 : données fictives -----
Avant1 <- rnorm(50,70,5)
Apres1 <- rnorm(50,50,5)
df1 <- data.frame(Avant=Avant1, Apres=Apres1)
boxplot1 <- ggpaired(df1, cond1 = "Avant", cond2 = "Apres", fill = "condition",
                      palette = "jco", title = "Données fictives (cas 1)")

# CAS 2 : données fictives -----
Avant2 <- rnorm(50,70,5)
Apres2 <- rnorm(50,66,5)
df2 <- data.frame(Avant=Avant2, Apres=Apres2)
boxplot2 <- ggpaired(df2, cond1 = "Avant", cond2 = "Apres", fill = "condition",
                      palette = "jco", title = "Données fictives (cas 2)")

ggarrange(boxplot1, boxplot2, ncol = 2, nrow = 1)

# Test t : appel de la fonction tstudent_dependants
tstudent_dependants(Avant1, Apres1, mu=0)

```

```

## 
##  groupe A : moy = 69.8, var = 13.1, sd = 3.6
##  groupe B : moy = 49.6, var = 22.1, sd = 4.7

```

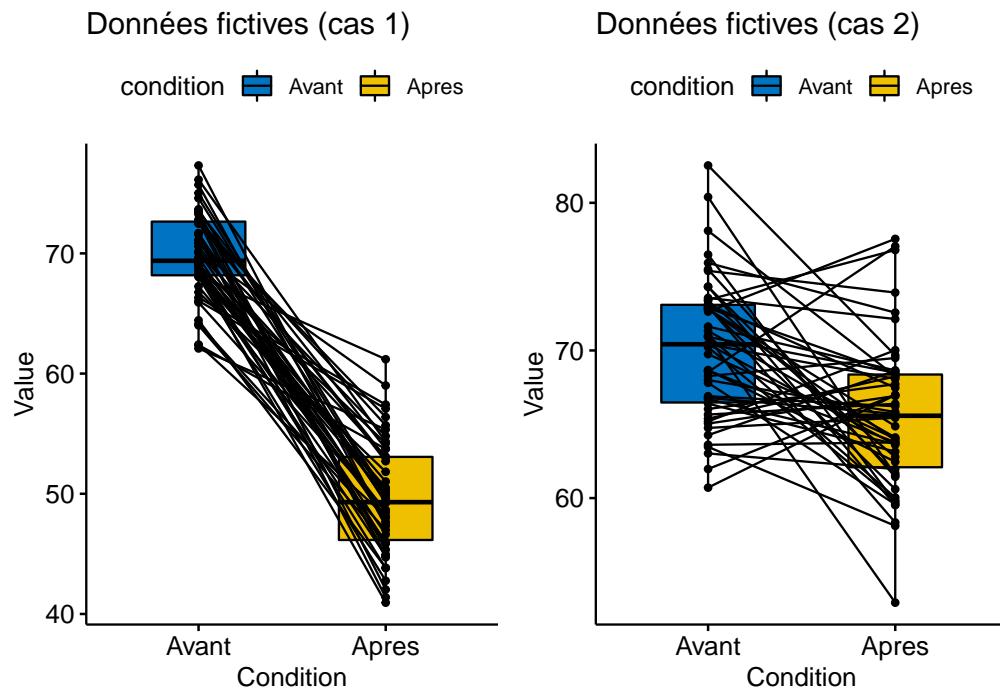


FIG. 4.17 : Boites à moustaches sur des échantillons fictifs appariées

```
## Moyenne des différences = 20.1
## Ecart-type des différences = 6.1
## t(dl = 49) = 23.36, p = 0
## La moyenne des différences entre les échantillons est significative
```

```
tstudent_dependants(Avant2, Apres2, mu=0)
```

```
##
## groupe A : moy = 70.1, var = 22.7, sd = 4.8
## groupe B : moy = 65.6, var = 25.6, sd = 5.1
## Moyenne des différences = 4.5
## Ecart-type des différences = 6.8
## t(dl = 49) = 4.73, p = 0
## La moyenne des différences entre les échantillons est significative
```

4.3.1.2 Mesurer la taille de l'effet

Rappelons que la taille de l'effet permet d'évaluer la magnitude (force) de l'effet d'une variable (ici la variable qualitative à deux modalités) sur une autre (ici la variable continue). Dans le cas de comparaisons de moyennes (avec des échantillons paireés ou non), pour mesurer la taille d'effet, on utilise habituellement le d de Cohen ou encore le g de Hedges; le second étant un ajustement du premier. Notez que nous analyserons la taille de l'effet uniquement si le test student ou de Welch s'est révélé significatif ($p < 0,05$).

Pourquoi utiliser le d de cohens? Deux propriétés en font une mesure particulièrement intéressante. Premièrement, elle est facile à calculer puisque d est le ratio entre la différence de deux moyennes de groupes (A, B) et l'écart-type combiné des deux groupes. Deuxièmement, d représente ainsi une mesure standardisée de la taille de l'effet; elle permet ainsi l'évaluation de la taille d'effet indépendamment

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

de l'unité de mesure de la variable continue. Concrètement, cela signifie que quelle que soit l'unité de mesure de la variable continue X , elle est toujours exprimée en unité d'écart-type de X . Cette propriété facilite ainsi grandement les comparaisons entre des valeurs de d calculées sur différentes combinaisons de variables (au même titre que le coefficient de variation ou le coefficient de corrélation par exemple). Pour des échantillons indépendants de tailles différentes, il s'écrit :

$$d = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{\frac{(n_A-1)S_{X_A}^2 + (n_B-1)S_{X_B}^2}{n_A+n_B-2}}}$$

avec n_A, n_B , $S_{X_A}^2$ et $S_{X_B}^2$ étant respectivement les nombres d'observations et les variances pour les groupes A et B , S_p^2 .

Si les échantillons sont de tailles identiques ($n_A = n_B$), alors d peut s'écrire :

$$d = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{(S_A^2 + S_B^2)/2}} = \frac{\bar{X}_A - \bar{X}_B}{(\sigma_A + \sigma_B)/2}$$

avec σ_A et σ_B étant les écart-types des deux groupes (rappel : l'écart-type est la racine carrée de la variance!).

Le g de Hedge est simplement une correction de d , particulièrement importante quand les échantillons sont de taille réduite.

$$g = d - \left(1 - \frac{3}{4(n_A + n_B) - 9}\right)$$

Moins utilisé en sciences sociales, mais surtout en études cliniques, le delta de Glass est simplement la différence des moyennes des groupes indépendants (numérateur) sur l'écart-type du deuxième groupe (dénominateur). Dans une étude clinique, on a habituellement un groupe qui subit un traitement (groupe de traitement) et un groupe qui a reçu un placebo (groupe de contrôle ou groupe témoin). L'effet de taille est ainsi évalué par rapport au groupe de contrôle :

$$\Delta = \frac{\bar{X}_A - \bar{X}_B}{\sigma_B}$$

Finalement, pour des échantillons dépendants (pairés), il s'écrit simplement $d = \bar{D}/\sigma_D$ avec \bar{D} et σ_D étant la moyenne et l'écart-type des différences entre les observations.

Comment interpréter le d de Cohen? Un effet sera considéré comme faible avec $|d|$ à 0,2, modéré à 0,50 et fort à 0,80 (Cohen, 1992). Notez que ces seuils ne sont que des conventions pour vous guider à interpréter la mesure de Cohen. D'ailleurs, dans son livre intitulé *Statistical power analysis for the behavioral sciences*, il écrit : « all conventions are arbitrary. One can only demand of them that they not be unreasonable » (Cohen, 2013). Plus récemment, (Sawilowsky, 2009) a ajouté d'autres seuils à ceux proposés par Cohen (tableau 4.8).

4.3.1.3 Mise en œuvre dans

Nous avons écrit précédemment les fonctions `tstudent_independants` et `tstudent_dépendants` uniquement pour décomposer les différentes étapes de calcul des tests de Student et de Welch. Il existe des fonctions de base (`t.test` et `var.test`) qui permettent de réaliser l'un ou l'autre de ces deux tests avec une seule ligne de code.

TAB. 4.8 : Conventions pour l'interprétation du d de Cohen

Sawilowsky	Cohen
0,1 : Très faible	0,2 : Faible
0,2 : Faible	0,5 : Moyen
0,5 : Moyen	0,8 : Fort
0,8 : Fort	1,2 : Très fort
1,2 : Très fort	2,0 : Énorme

La fonction `t.test` permet ainsi de calculer les test de Student et de Welch :

- `t.test(x ~ y, data=, mu = 0, paired = FALSE, var.equal = FALSE, conf.level = 0.95)` ou `t.test(x =, y =, mu = 0, paired = FALSE, var.equal = FALSE, conf.level = 0.95)`.
- Le paramètre `paired` sera utilisé pour spécifier si les échantillons sont dépendants (`paired=TRUE`) ou indépendants (`paired=FALSE`).
- Le paramètre `var.equal` sera utilisé pour spécifier si les variances sont égales pour le test de Student (`var.equal=TRUE`) ou dissemblables pour le test de Welch (`var.equal=FALSE`).
- `var.test(x, y)` ou `var.test(x ~ y, data=)` pour vérifier au préalable si les variances sont égales ou non et choisir ainsi un t de Student ou un t de Welch.

Les fonctions `cohens_d` et `hedges_g` du package **effectsize** renvoient respectivement les mesures de d de Cohen et du g de Hedge :

- `cohens_d(x ~ y, data = datafram, paired = FALSE, pooled_sd = TRUE)` ou `cohens_d(x, y, data = datafram, paired = FALSE, pooled_sd = TRUE)`
- `hedges_g(x ~ y, data = datafram, paired = FALSE, pooled_sd = TRUE)` ou `hedges_g(x, y, data = datafram, paired = FALSE, pooled_sd = TRUE)`
- `glass_delta(x ~ y, data = datafram, paired = FALSE, pooled_sd = TRUE)` ou `glass_delta(x, y, data = datafram, paired = FALSE, pooled_sd = TRUE)`

Notez que pour toutes ces fonctions deux écritures sont possibles :

- `x ~ y, data=` avec un `dataframe` dans lequel `x` est une variable continue et `y` et un facteur binaire
- `x, y` qui sont tous deux des vecteurs numériques (variable continue).

Exemple de test pour des échantillons indépendants

La figure 4.18 représente la cartographie du pourcentage de locataires par secteur de recensement (SR) pour la région métropolitaine de recensement de Montréal (RMR) en 2016, soit une variable continue. L'objectif est de vérifier si la moyenne de ce pourcentage des SR de l'agglomération de Montréal est significativement différente de celles de SR hors de l'agglomération.

Les résultats de la syntaxe ci-dessous signalent que le pourcentage de locataires par SR est bien supérieur dans l'agglomération (moyenne = 59,7% ; écart-type = 21,4%) qu'en dehors de l'agglomération de Montréal (moyenne = 27,3% ; écart-type = 20,1%); cette différence de 32,5 points de pourcentage est d'ailleurs significative et très forte ($t = -23,95$; $p \otimes 0,001$, d de Cohen = 1,54).

```
library("foreign")
library("effectsize")
library("ggplot2")
library("dplyr")
# Importation du fichier
```

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

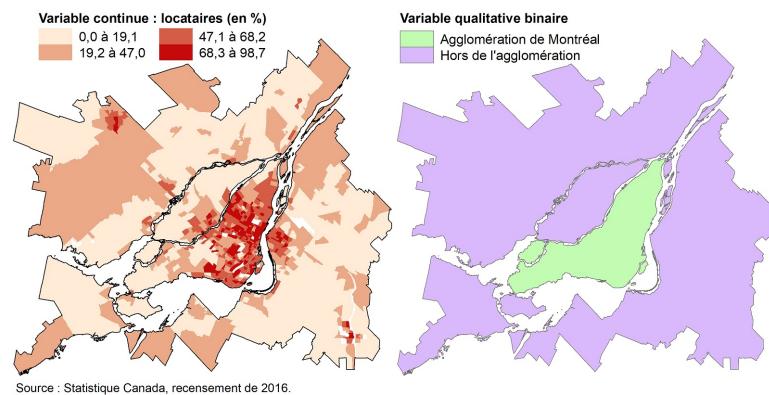
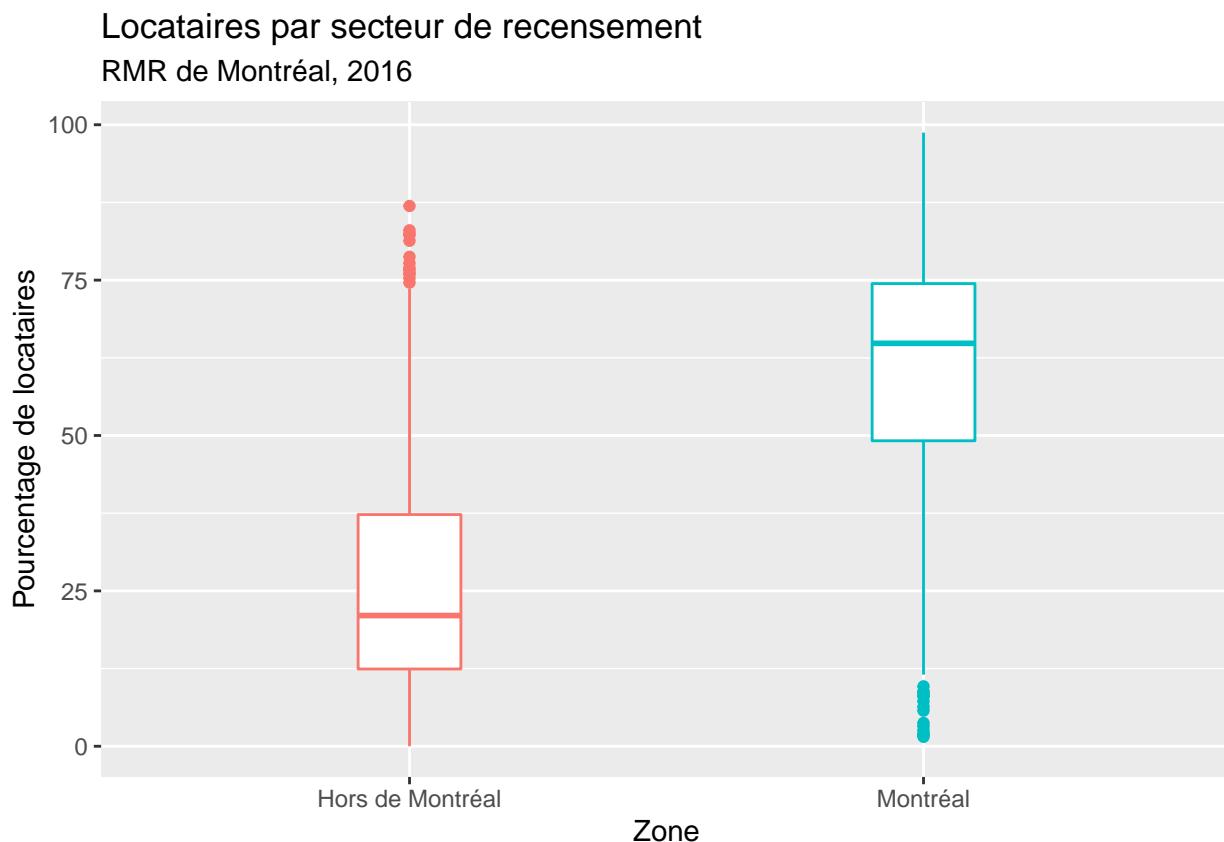


FIG. 4.18 : Pourcentage de locataires par secteur de recensement, RMR de Montréal, 2016

```
dfRMR <- read.dbf("data/bivariee/SRRMRMTL2016.dbf")
# Définition d'un facteur binaire
dfRMR$Montreal <- factor(dfRMR$Montreal,
                            levels= c(0,1),
                            labels = c("Hors de Montréal","Montréal"))

# Comparaison des moyennes -----
#Boite à moustaches (boxplot)
ggplot(data = dfRMR, mapping=aes(x=Montreal,y=Locataire,colour=Montreal)) +
  geom_boxplot(width=0.2)+
  theme(legend.position="none")+
  xlab("Zone")+
  ylab("Pourcentage de locataires")+
  ggtitle("Locataires par secteur de recensement", subtitle="RMR de Montréal, 2016")
```



```
# nombre d'observations, moyennes et écart-types pour les deux échantillons
group_by(dfRMR, Montreal) %>%
  summarise(
    n = n(),
    moy = mean(Locataire, na.rm = TRUE),
    ecarttype = sd(Locataire, na.rm = TRUE)
  )

## # A tibble: 2 x 4
##   Montreal           n     moy   ecarttype
##   <fct>        <int>  <dbl>      <dbl>
## 1 Hors de Montréal  430   27.3      20.1
## 2 Montréal         521   59.7      21.4

# On vérifie si les variances sont égales avec la fonction var.test
# quand la valeur de P est inférieure à 0,05 alors les variances diffèrent
v <- var.test(Locataire ~ Montreal, alternative='two.sided', conf.level=.95, data=dfRMR)
print(v)
```

```
##
## F test to compare two variances
##
## data: Locataire by Montreal
## F = 0.88156, num df = 429, denom df = 520, p-value = 0.1739
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7361821 1.0573195
## sample estimates:
## ratio of variances
## 0.8815563
```

Le test indique que nous n'avons aucune raison de rejeter l'hypothèse nulle selon laquelle les variances sont égales. Pour l'île de Montréal, l'écart-type est de 21,4 et de 20,1 hors de l'île, soit une différence négligeable.

```
# Calcul du T de Student ou du T de Welch
p <- v$p.value
if(p >= 0.05){
  cat("\n Les variances ne diffèrent pas !",
      "\n On utilise le test de student avec l'option var.equal=TRUE", sep="")
  t.test(Locataire ~ Montreal, # variable continue ~ facteur binaire
         data=dfRMR,          # nom du dataframe
         conf.level=.95,       # intervalle de confiance pour la valeur de t
         paired = FALSE,       # échantillons non pairés (indépendants)
         var.equal=TRUE)       # variances égales
} else {
  cat("\n Les variances diffèrent !",
      "\n On utilise le test de Welch avec l'option var.equal=FALSE", sep="")
  t.test(Locataire ~ Montreal, # variable continue ~ facteur binaire
         data=dfRMR,          # nom du dataframe
         conf.level=.95,       # intervalle de confiance pour la valeur de t
```

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

```

        paired = FALSE,           # échantillons non pairés (indépendants)
        var.equal=FALSE          # variances différentes
    }

## Les variances ne diffèrent pas !
## On utilise le test de student avec l'option var.equal=TRUE

##
## Two Sample t-test
##
## data: Locataire by Montreal
## t = -23.95, df = 949, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -35.11182 -29.79341
## sample estimates:
## mean in group Hors de Montréal      mean in group Montréal
##                               27.27340                           59.72601

# Effet de taille à analyser uniquement si le test est significatif
cohens_d(Locataire ~ Montreal, data = dfRMR, paired = FALSE)

```

```

## Cohen's d |      95% CI
## -----
##     -1.56 | [-1.71, -1.41]

```

```

hedges_g(Locataire ~ Montreal, data = dfRMR, paired = FALSE)

```

```

## Hedge's g |      95% CI
## -----
##     -1.56 | [-1.70, -1.41]

```

Notez que le d de Cohen et le g de Hedge sont très proches ici, rappelons que le second est une correction du premier pour des échantillons de taille réduite. Avec 951 observations, nous disposons d'un échantillon suffisamment grand pour que cette correction soit négligeable.

Exemple de syntaxe pour un test de Student pour des échantillons dépendants

```

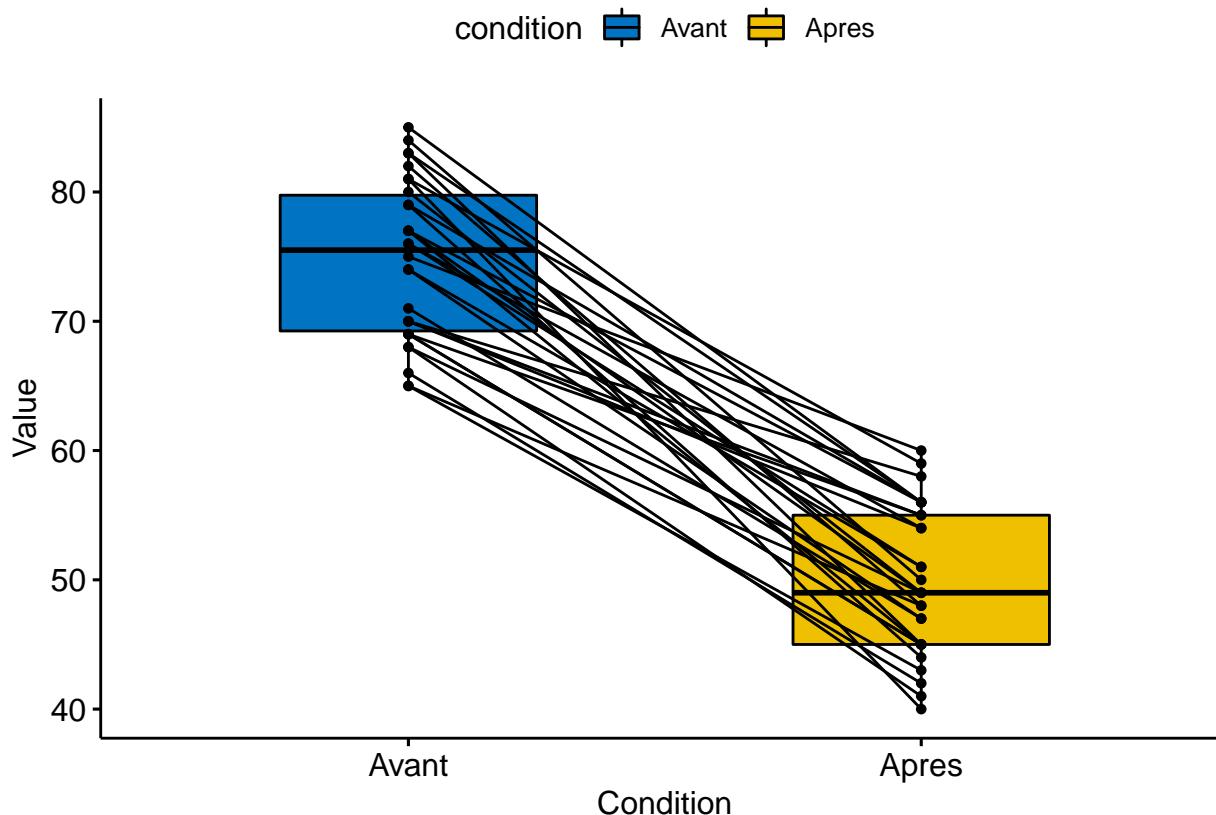
library("ggpubr")
library("dplyr")
Pre <- c(79,71,81,83,77,74,76,74,79,70,66,85,69,69,82,69,81,70,83,68,77,76,77,70,68,80,65,65,75,84)
Post <- c(56,47,40,45,49,51,54,47,44,54,42,56,45,45,48,55,59,58,56,41,56,51,45,55,49,49,48,43,60,50)
# Première façon de faire un tableau : avec deux colonnes Avant et Après
df1 <- data.frame(Avant=Pre, Apres=Post)
head(df1)

##   Avant Apres
## 1     79     56
## 2     71     47
## 3     81     40

```

```
## 4     83    45
## 5     77    49
## 6     74    51
```

```
ggpaired(df1, cond1 = "Avant", cond2 = "Apres", fill = "condition", palette = "jco")
```



```
# Nombre d'observations, moyennes et écart-types
cat(nrow(df1), " observations",
"\nPOST. moy = ", round(mean(df1$Avant),1), ", e.t. = ", round(sd(df1$Avant),1),
"\nPRE. moy = ", round(mean(df1$Apres),1), ", e.t. = ", round(sd(df1$Apres),1), sep="")
```

```
## 30 observations
## POST. moy = 74.8, e.t. = 6.1
## PRE. moy = 49.9, e.t. = 5.7
```

```
t.test(Pre, Post, paired = TRUE)
```

```
##
## Paired t-test
##
## data: Pre and Post
## t = 18.701, df = 29, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 22.11740 27.54926
## sample estimates:
```

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

```
## mean of the differences
##                                24.83333
```

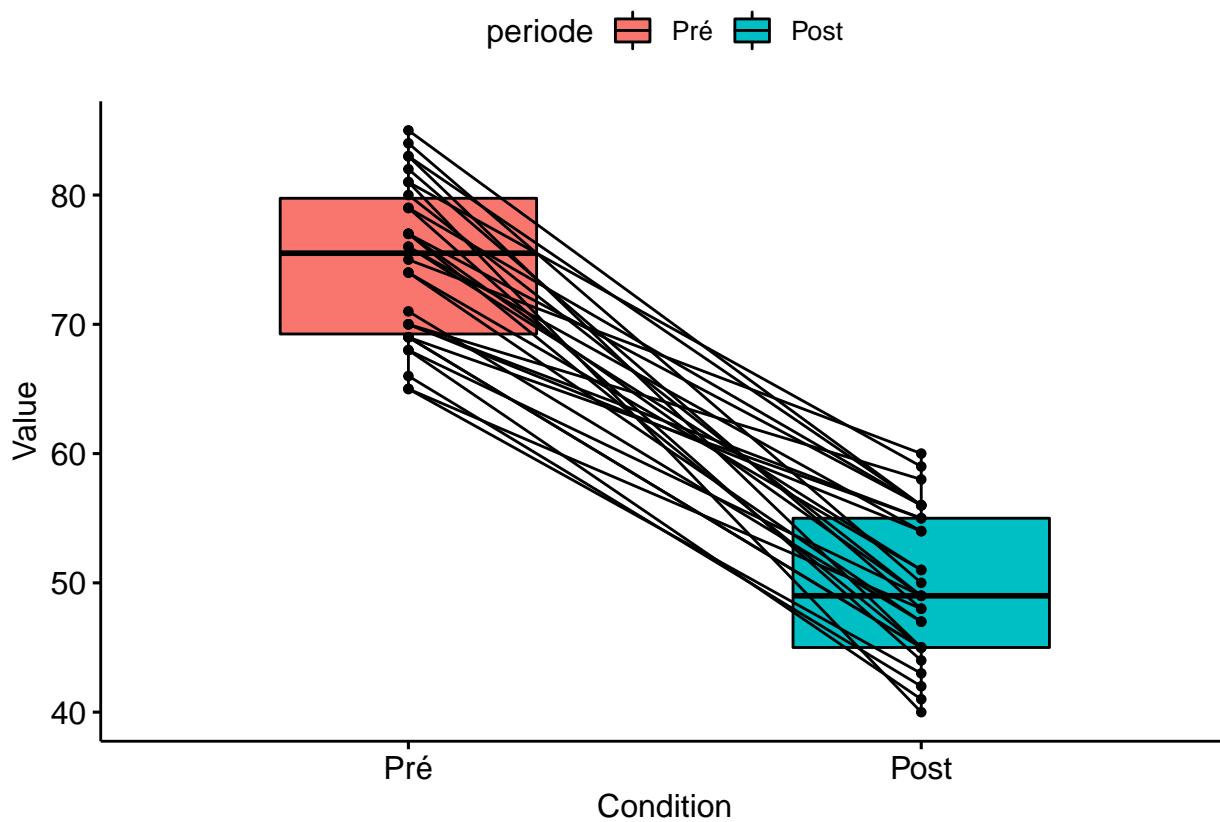
```
# Deuxième façon de faire un tableau : avec une colonne pour la variable continue et une autre pour la variable qual
n <- length(Pre)*2
df2 <- data.frame(
  id=(1:n),
  participant=(1:length(Pre)),
  risque=c(Pre,Post)
)
df2$periode <- ifelse(df2$id <= length(Pre), "Pré", "Post")
head(df2)
```

```
##   id participant risque periode
## 1  1           1    79     Pré
## 2  2           2    71     Pré
## 3  3           3    81     Pré
## 4  4           4    83     Pré
## 5  5           5    77     Pré
## 6  6           6    74     Pré
```

```
# nombre d'observations, moyennes et écart-types pour les deux échantillons
group_by(df2, periode) %>%
  summarise(
    n = n(),
    moy = mean(risque, na.rm = TRUE),
    et = sd(risque, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 4
##   periode     n   moy     et
##   <chr>   <int> <dbl>  <dbl>
## 1 Post       30  49.9  5.67
## 2 Pré        30  74.8  6.10
```

```
ggpaired(data=df2, x= "periode", y="risque", fill = "periode")
```



```
t.test(risque ~ periode, data=df2, paired = TRUE)
```

```
##
##  Paired t-test
##
## data: risque by periode
## t = -18.701, df = 29, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -27.54926 -22.11740
## sample estimates:
## mean of the differences
## -24.83333
```

4.3.1.4 Comparer des moyennes pondérées



En études urbaines et en géographie, le recours aux données agrégées (non individuelles) est fréquent, par exemple au niveau des secteurs de recensement (comprenant généralement entre 2500 à 8000 habitants). Dans ce contexte, un secteur de recensement plus peuplé devrait avoir un poids plus important dans l'analyse. Il est possible d'utiliser les versions pondérées des tests présentés précédemment. Prenons deux exemples pour illustrer le tout :

- Pour chaque secteur de recensement des îles de Montréal et de Laval, nous avons calculé la distance au parc le plus proche à travers le réseau de rues avec un Système d'Information Géographique (SIG). On souhaite vérifier si les enfants âgés de moins de 15 ans résidant sur l'île de Montréal bénéficient en moyenne d'une meilleure accessibilité au parc.

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

- Dans une étude pour sur la concentration de polluants atmosphérique dans l'environnement autour des écoles primaires montréalaises, Carrier *et al.* (2014) souhaitaient vérifier si les élèves fréquentant les écoles les plus défavorisés sont plus exposés au dioxyde d'azote (NO_2) dans leur milieu scolaire. Pour ce faire, ils ont réalisé un test t sur un tableau avec comme observations les écoles primaires et trois variables : la moyenne NO_2 (variable continue), les quintiles extrêmes d'un indice de défavorisation (premier et dernier quintiles, variable qualitative) et le nombres d'élèves inscrits par école (variable pour la pondération).

Pour réaliser un test t pondéré, nous pouvons utiliser la fonction `weighted_ttest` du package **sjstats**.

En guise d'exemple appliqué, dans la syntaxe ci-dessous, nous avons refait le même test t que précédemment (`Locataire ~ Montreal`) en pondérant chaque secteur de recensement par le nombre de logements qu'il comprend.

```
library("sjstats")
library("dplyr")
# Calcul des statistiques pondérées
group_by(dfRMR, Montreal) %>%
  summarise(
    n = sum(Logement),
    MoyPond = weighted_mean(Locataire, Logement),
    ecarttypePond = weighted_sd(Locataire, Logement)
  )

## # A tibble: 2 x 4
##   Montreal           n  MoyPond  ecarttypePond
##   <fct>       <int>    <dbl>        <dbl>
## 1 Hors de Montréal 856928     28.4        19.9
## 2 Montréal         870354     60.0        20.8

# Test t non pondéré
t.test(Locataire ~ Montreal, dfRMR,
       paired = FALSE, var.equal = TRUE, conf.level=.95)

##
## Two Sample t-test
##
## data: Locataire by Montreal
## t = -23.95, df = 949, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -35.11182 -29.79341
## sample estimates:
## mean in group Hors de Montréal      mean in group Montréal
##                      27.27340                      59.72601

# Test t pondérée
weighted_ttest(Locataire ~ Montreal + Logement, dfRMR,
               paired = FALSE, ci.lvl=.95)

##
```

```
## Two-Sample t-test (two.sided)
##
## # comparison of Locataire by Montreal
## # t=-23.91 df=928 p-value=0.000
##
##   mean in group Hors de Montréal: 28.396
##   mean in group Montréal       : 60.003
##   difference of mean          : -31.608 [-34.202 -29.013]
```

4.3.1.5 Comment rapporter un test de Student ou de Welch ?

Pour les différentes versions du test, il est important de rapporter la valeur de t , la valeur de p et les moyennes des groupes. Voici quelques exemples :

Test de Student ou de Welch pour échantillons indépendants

- Dans la région métropolitaine de Montréal en 2005, le revenu total des femmes (moyenne = 29117 dollars; écart-type = 258022) est bien inférieur à celui des hommes (moyenne = 44463; écart-type = 588081). La différence entre les moyennes des deux sexes (-15345) en faveur des hommes est d'ailleurs significative ($t = -27,09$; $p \approx 0,001$).
- Il y a un effet significatif selon le sexe ($t = -27,09$; $p \approx 0,001$), le revenu total des hommes (moyenne = 44463; écart-type = 588081) étant bien supérieur à celui des femmes (moyenne = 29 117; écart-type = 258 022).
- 50 personnes se rendent au travail à vélo (moyenne = 33,7, écart-type = 8,5) contre 60 en automobile (moyenne = 34, écart-type = 8,7); il n'y a pas de différence significative entre les moyennes d'âge des deux groupes ($t(108) = -0,79$, $p = 0,427$).

Test de Student échantillons dépendants (pairés)

- On constate une diminution significative de la perception du risque après l'activité (moyenne = 49,9, écart-type = 5,7) comparativement à avant (moyenne = 74,8, écart-type = 6,1), avec une différence de -24,8 ($t(29) = -18,7$, $p < 0,001$).
- Les résultats du pré-test (moyenne = 49,9, écart-type = 5,7) et du post-test (moyenne = 74,8, écart-type = 6,1) montrent qu'il y a une diminution significative de la perception du risque ($t(29) = -18,7$, $p < 0,001$).

Pour un texte en anglais, vous pourrez consulter <https://www.socscistatistics.com/tutorials/ttest/default.aspx>.

4.3.2 Test non paramétrique de Wilcoxon



Si la variable continue est fortement anormalement distribuée, il est déconseillé d'utiliser les tests de Student et de Welch. On privilégiera le test des rangs signés de Wilcoxon (*Wilcoxon rank-sum test* en anglais). Attention, il est aussi appelé test U de Mann-Whitney. Ce test permet alors de vérifier si les deux groupes présentent des médianes différentes.

Pour ce faire, on utilise la fonction `wilcox.test` dans laquelle le paramètre `paired` permettra de spécifier si les échantillons sont indépendants ou non (`FALSE` ou `TRUE`).

Dans l'exemple suivant, nous analysons le pourcentage de locataires dans les secteurs de recensements de la région métropolitaine de Montréal. Plus spécifiquement, nous comparons ce pourcentage entre les secteurs présents sur l'île et les secteurs hors de l'île. Il s'agit donc d'un test avec des échantillons indépendants.

4.3. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À DEUX MODALITÉS

```

library("foreign")
library("dplyr")
#####
# Échantillons indépendants
#####
dfRMR <- read.dbf("data/bivariee/SRRMRMTL2016.dbf")
# Définition d'un facteur binaire
dfRMR$Montreal <- factor(dfRMR$Montreal,
                           levels= c(0,1),
                           labels = c("Hors de Montréal","Montréal"))
# Calcul du nombre d'observations, moyennes et écart-types des rangs pour les deux échantillons
group_by(dfRMR, Montreal) %>%
  summarise(
    n = n(),
    moy_rang = mean(rank(Locataire), na.rm = TRUE),
    med_rang = median(rank(Locataire), na.rm = TRUE),
    ecarttype_rang = sd(rank(Locataire), na.rm = TRUE)
  )

## # A tibble: 2 x 5
##   Montreal           n moy_rang med_rang ecarttype_rang
##   <fct>     <int>     <dbl>     <dbl>          <dbl>
## 1 Hors de Montréal 430      216.     216.         124.
## 2 Montréal        521      261      261          151.

# Test des rangs signés de Wilcoxon sur des échantillons indépendants
wilcox.test(Locataire ~ Montreal, dfRMR, paired = FALSE)

```

```

##
## Wilcoxon rank sum test with continuity correction
##
## data: Locataire by Montreal
## W = 33716, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0

```

Nous observons bien ici une différence significative entre le pourcentage de locataires des secteurs de recensement sur l'île (rang médian = 216) et hors de l'île (rang médian = 261).

Pour le second exemple, nous générerons deux jeux de données au hasard représentant une mesure d'une variable pré-traitement (*pre*) et post-traitement (*post*) pour un même échantillon.

```

#####
# Échantillons dépendants
#####
pre <- sample(60:80, 50, replace=T)
post <- sample(30:65, 50, replace=T)
df1 <- data.frame(Avant=pre, Apres=post)
# Nombre d'observations, moyennes et écart-types
cat(nrow(df1), " observations",
  "\nPOST. median = ", round(median(df1$Avant),1),
  " moy = ", round(mean(df1$Avant),1),
  "\nPRE. median = ", round(median(df1$Apres),1),
  "
```

```

", moy = ", round(mean(df1$Apres),1), sep="")
```

```

## 50 observations
## POST. median = 70, moy = 69.9
## PRE. median = 47, moy = 48.7
```

```
wilcox.test(df1$Avant, df1$Apres, paired = TRUE)
```

```

##
## Wilcoxon signed rank test with continuity correction
##
## data: df1$Avant and df1$Apres
## V = 1272, p-value = 9.208e-10
## alternative hypothesis: true location shift is not equal to 0
```

À nouveau, nous obtenons une différence significative entre les deux variables.

Comment rapporter un test de Wilcoxon ?

Lorsque l'on rapporte les résultats d'un test de Wilcoxon, il est important de signaler la valeur du test (W), le degré de signification (valeur de p) et éventuellement la médiane des rangs ou de la variable originale pour les deux groupes. Voici quelques exemples :

- Les résultats du test des rangs signés de Wilcoxon signalent que les rangs de l'île de Montréal sont significativement plus élevés que ceux de l'île de Laval ($W = 1223$, $p \approx 0,001$).
- Les résultats du test de Wilcoxon signalent que les rangs post-tests sont significativement plus faibles que ceux pré-test ($W = 1273,5$, $p \approx 0,001$).
- Les résultats du test de Wilcoxon signalent que la médiane des rangs pré-tests (médiane = 69) est significativement plus forte que celle du post-test (médiane = 50,5) ($W = 1273,5$, $p \approx 0,001$).

4.4 Relation entre une variable quantitative et une variable qualitative à plus de deux modalités



Existe-t-il une relation entre une variable continue et une variable qualitative comprenant plus de deux modalités ? Pour répondre à cette question, on pourra recourir à deux méthodes : l'analyse de variance – ANOVA, *ANalysis Of VAriance* en anglais – et le test non paramétrique de Kruskal-Wallis. La première permet de vérifier si les moyennes de plusieurs groupes d'une population donnée sont ou non significativement différentes ; la seconde si leurs médianes sont différentes.

4.4.1 Analyse de variance

L'analyse de variance (ANOVA) est largement utilisée en psychologie, médecine et pharmacologie. Prenons un exemple classique en pharmacologie pour tester l'efficacité d'un médicament. Quatre groupes de population sont constitués :

- un premier groupe d'individus pour lequel on administre un placebo (un médicament sans substance active), soit le groupe de contrôle ou le groupe témoin;
- un second groupe auquel l'on administre le médicament avec un faible dosage;
- un troisième avec un dosage moyen;

4.4. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À PLUS DE DEUX MODALITÉS

- un quatrième avec un dosage élevé.

La variable continue permettra d'évaluer l'évolution de l'état de santé des individus (par exemple, la variation du taux de globules rouges dans le sang avant et après le traitement). Si le traitement est efficace, on s'attendrait alors à ce que les moyennes des deuxième, troisième et quatrième groupes soient plus élevées que celle du groupe de contrôle. Les différences de moyennes entre les second, troisième et quatrième groupes permettront aussi de repérer quel dosage est le plus efficace. Si nous n'observons aucune différence significative entre les groupes, cela signifie que l'effet du médicament ne diffère pas de l'effet d'un placebo.

L'ANOVA est aussi très utilisée en études urbaines, principalement pour vérifier si un phénomène urbain varie selon plusieurs groupes d'une population donnée ou régions géographiques. En guise d'exemple, le recours à l'ANOVA permettrait de répondre aux questions suivantes :

- les moyennes des niveaux d'exposition à un polluant atmosphérique (variable continue) varient-elles significativement selon le mode de transport utilisé (automobile, vélo, transport en commun) pour des trajets similaires en heures de pointe ?
- pour une métropole donnée, les moyennes des loyers (variable continue) sont-elles différentes entre les logements de la ville centre versus ceux localisés dans la première couronne et ceux de la seconde couronne ?

4.4.1.1 Le calcul des trois variances pour l'ANOVA

L'ANOVA repose sur le calcul de trois variances :

- la **variance totale** (VT) de la variable dépendante continue, soit la somme des carrés des écarts à la moyenne de l'ensemble de la population (équation (4.17));
- la **variance intergroupe** (Var_{inter}) ou variance expliquée (VE), soit la somme des carrés des écarts entre la moyenne de chaque groupe et la moyenne de l'ensemble du jeu de données multipliées par le nombre d'individus appartenant à chacun des groupes (équation (4.18));
- la **variance intragroupe** (Var_{intra}) ou variance non expliquée (VNE), soit la somme des variances des groupes de la variable indépendante (équation (4.19)).

$$VT = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (4.17)$$

$$Var_{inter} \text{ ou } VE = \sum_{i \in g_1} (\bar{y}_{g_1} - \bar{y})^2 + \sum_{i \in g_2} (\bar{y}_{g_2} - \bar{y})^2 + \dots + \sum_{i \in g_n} (\bar{y}_{g_n} - \bar{y})^2 \quad (4.18)$$

$$Var_{intra} \text{ ou } VNE = \sum_{i \in g_1} (y_i - \bar{y}_{g_1})^2 + \sum_{i \in g_2} (y_i - \bar{y}_{g_2})^2 + \dots + \sum_{i \in g_n} (y_i - \bar{y}_{g_n})^2 \quad (4.19)$$

avec \bar{y} est la moyenne de l'ensemble de la population; $\bar{y}_{g_1}, \bar{y}_{g_2}, \bar{y}_{g_n}$ sont respectivement les moyennes des groupes 1 à k (k étant le nombre de modalités de la variable qualitative).

La variance totale (VT) est égale à la somme de la variance intergroupe (expliquée) et la variance intragroupe (non expliquée) (équation (4.20)). Le ratio entre la variance intergroupe (expliquée) et la variance totale est dénommé Eta^2 (équation (4.21)). Il varie de 0 à 1 et exprime la proportion de la variance de la variable continue qui est expliquée par les différentes modalités de la variable qualitative.

$$VT = Var_{inter} + Var_{intra} \text{ ou } VT = VNE + VE \quad (4.20)$$

$$\eta^2 = \frac{Var_{inter}}{VT} \text{ ou } \eta^2 = \frac{VE}{VT} \quad (4.21)$$



La décomposition de la variance totale – égale à la somme des variances intragroupe et intergroupe – est fondamentale en statistique. Nous verrons qu'elle est aussi utilisée pour évaluer la qualité d'une partition d'une population dans le chapitre sur les méthodes de classification (chapitre 10). En ANOVA, on retiendra que :

- plus la variance intragroupe est faible, plus les différents groupes sont homogènes;
- plus la variance intergroupe est forte, plus les moyennes des groupes sont différentes et donc plus les groupes sont dissemblables.

Autrement dit, plus la variance intergroupe (**dissimilarité** des groupes) est maximisée et corollairement plus la variance intragroupe (**homogénéité** de chacun des groupes) est minimisée, plus les groupes sont clairement distincts et plus l'ANOVA sera performante.

Examinons un premier jeu de données fictif sur la vitesse de déplacements de cyclistes (variable continue exprimée en km/h) et une variable qualitative comprenant trois groupes de cyclistes utilisant soit un vélo personnel ($n_A = 5$), soit en libre service ($n_B = 7$), soit électrique ($n_C = 6$) (tableau 4.9). D'emblée, on note que les moyennes de vitesse des trois groupes sont différentes : 17,6 km/h pour les cyclistes avec leur vélo personnel, 12,3 km/h les utilisateurs des vélos en libre service et 23,1 km/h pour les cyclistes avec un vélo électrique.

Pour chaque observation, la troisième colonne du tableau représente les écarts à la moyenne globale mis au carré, tandis que les colonnes suivantes représentent la déviation au carré de chaque observation à la moyenne de son groupe d'appartenance. Ainsi, pour la première observation, on a : $(16,900 - 17,339)^2 = 0,193$ et $(16,900 - 17,580)^2 = 0,46$. La variance totale (VT) est donc égale à la somme de la troisième colonne (424,663), tandis que la variance intragroupe (non expliquée, VNE) est égale à $11,228 + 21,537 + 13,993 = 46,758$. Quant à la variance intergroupe (expliquée, VE), elle est égale à $5 \times (17,580 - 17,339)^2 + 7 \times (12,257 - 17,339)^2 + 6 \times (23,067 - 17,339)^2 = 377,904$.

On a donc $VT = Var_{inter} + Var_{intra}$, soit $424,663 = 377,904 + 46,758$ et $\eta_2 = 377,904/424,663 = 0,89$. Cela signale que 89% de la variance de la vitesse des cyclistes est expliquée par le type de vélo utilisé.

Examinons un deuxième jeu de données fictives pour lequel le type de vélo utilisé n'aurait que peu d'effet sur la vitesse des cyclistes (tableau 4.10). D'emblée, les moyennes des trois groupes semblent très similaires (19,3, 17,9 et 18,7). Les valeurs des trois variances sont les suivantes :

- la variance totale est égale à 121,756.
- la variance intragroupe (non expliquée, VNE) est égale à $9,140 + 50,254 + 56,275 = 115,669$
- la variance intragroupe (expliquée, VE) est égale à $5 \times (19,300 - 18,528)^2 + 7 \times (17,871 - 18,528)^2 + 6 \times (18,650 - 18,528)^2 = 6,087$.

On a donc $VT = Var_{inter} + Var_{intra}$, soit $121,756 = 6,087 + 115,669$ et $\eta_2 = 6,087/121,756 = 0,05$. Cela signale que 5% de la variance de la vitesse des cyclistes est uniquement expliquée par le type de vélo utilisé.

4.4.1.2 Le test de Fisher

Pour vérifier si les moyennes sont statistiquement différentes (autrement dit, si leur différence est significativement différente de 0), on a recours au test F de Fisher. Pour ce faire, on pose l'hypothèse nulle (H_0), soit que les moyennes des groupes sont égales; autrement dit que la variable qualitative n'a

4.4. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À PLUS DE DEUX MO

TAB. 4.9 : Données fictives et calcul des trois variances (cas 1)

Type de vélo	km/h	$(y_i - \bar{y})^2$	$(y_i - \bar{y}_A)^2$	$(y_i - \bar{y}_B)^2$	$(y_i - \bar{y}_C)^2$
A. personnel	16,900	0,193	0,462	–	–
A. personnel	20,400	9,370	7,952	–	–
A. personnel	16,100	1,535	2,190	–	–
A. personnel	17,700	0,130	0,014	–	–
A. personnel	16,800	0,290	0,608	–	–
B. libre service	13,400	15,515	–	1,306	–
B. libre service	11,300	36,468	–	0,916	–
B. libre service	14,000	11,148	–	3,038	–
B. libre service	12,400	24,393	–	0,020	–
B. libre service	13,700	13,242	–	2,082	–
B. libre service	8,500	78,126	–	14,116	–
B. libre service	12,500	23,415	–	0,059	–
C. électrique	22,900	30,926	–	–	0,028
C. électrique	26,000	75,015	–	–	8,604
C. électrique	23,600	39,202	–	–	0,284
C. électrique	21,000	13,404	–	–	4,271
C. électrique	22,300	24,613	–	–	0,588
C. électrique	22,600	27,679	–	–	0,218
grande moyenne	17,339	–	–	–	–
moyenne groupe A	17,580	–	–	–	–
moyenne groupe B	12,257	–	–	–	–
moyenne groupe C	23,067	–	–	–	–
Variance totale	–	424,663	–	–	–
Variance intragroupe	–	–	11,228	21,537	13,993

TAB. 4.10 : Données fictives et calcul des trois variances (cas 2)

Type de vélo	km/h	$(y_i - \bar{y})^2$	$(y_i - \bar{y}_A)^2$	$(y_i - \bar{y}_B)^2$	$(y_i - \bar{y}_C)^2$
A. personnel	17,500	1,056	3,24	–	–
A. personnel	19,000	0,223	0,09	–	–
A. personnel	19,700	1,374	0,16	–	–
A. personnel	18,700	0,030	0,36	–	–
A. personnel	21,600	9,439	5,29	–	–
B. libre service	13,700	23,307	–	17,401	–
B. libre service	20,800	5,163	–	8,577	–
B. libre service	15,100	11,750	–	7,681	–
B. libre service	18,800	0,074	–	0,862	–
B. libre service	21,500	8,834	–	13,167	–
B. libre service	16,500	4,112	–	1,881	–
B. libre service	18,700	0,030	–	0,687	–
C. électrique	16,600	3,716	–	–	4,203
C. électrique	16,300	4,963	–	–	5,523
C. électrique	15,600	8,572	–	–	9,303
C. électrique	20,000	2,167	–	–	1,822
C. électrique	24,600	36,872	–	–	35,402
C. électrique	18,800	0,074	–	–	0,022
grande moyenne	18,528	–	–	–	–
moyenne groupe A	19,300	–	–	–	–
moyenne groupe B	17,871	–	–	–	–
moyenne groupe C	18,650	–	–	–	–
Variance totale	–	121,756	–	–	–
Variance intragroupe	–	–	9,14	50,254	56,275

pas d'effet sur la variable continue (indépendance entre les deux variables). L'hypothèse alternative (H_1) est donc que les moyennes sont différentes. Pour nos deux jeux de données fictives ci-dessus comprenant trois groupes, H_0 signifie que $\bar{y}_A = \bar{y}_B = \bar{y}_C$. La statistique F se calcule comme suit :

$$F = \frac{\frac{Var_{inter}}{k-1}}{\frac{Var_{intra}}{n-k}} \text{ ou } F = \frac{\frac{VE}{k-1}}{\frac{VNE}{n-k}} \quad (4.22)$$

avec n et k étant respectivement les nombres d'observations et de modalités de la variable qualitative. L'hypothèse nulle (les moyennes sont égales) sera rejetée si la valeur du F calculé est supérieure à la valeur critique de la table F avec les degrés de libertés ($k-1, n-k$) et un seuil α ($p=0,05$ habituellement) (voir le tableau des valeurs critiques de F , section 12.2). Notez qu'on utilise rarement la table F puisqu'avec la fonction `aov` on calcule directement la valeur F et celle de p qui lui est associée. Concrètement, si le test F est significatif (avec $p<0,05$), plus la valeur de F sera élevée, plus la différence entre les moyennes sera élevée.

Appliquons rapidement la démarche du test F à nos deux jeux de données fictives qui comprennent 3 modalités pour la variable qualitative et 18 observations. Avec $\alpha=0,05$, 2 degrés de liberté (3-1) au numérateur et 15 au dénominateur (18-3), la valeur critique de F est de 3,68. On en conclut alors que :

- pour le cas A, le F calculé est égal à $F = (377,904/2)/(46,758/15) = 60,62$. Il est supérieur à la valeur F critique ; les moyennes sont donc statistiquement différentes au seuil 0,05. Autrement dit, nous aurions eu moins de 5% de chance d'obtenir un échantillon produisant ces résultats si en réalité la différence entre les moyenne était de 0.
- pour le cas B, le F calculé est égal à $F = (6,087/2)/(115,669/15) = 0,39$. Il est inférieur à la valeur F critique ; les moyennes ne sont donc pas statistiquement différentes au seuil 0,05.

4.4.1.3 Conditions d'application de l'ANOVA et solutions alternatives

Trois conditions d'application doivent être vérifiées avant d'effectuer une analyse de variance sur un jeu de données :

- **Normalité des groupes.** Le test de Fisher repose sur le postulat que les échantillons (groupes) sont normalement distribués. Pour le vérifier, on a recours au test de normalité de Shapiro-Wilk (section 2.5.4.1.3). Notez toutefois que ce test est très restrictif, surtout pour des grands échantillons.
- **Homoscédasticité.** La variance dans les échantillons doit être la même (homogénéité des variances). Pour vérifier cette condition, on utilisera les tests de Levene, de Bartlett ou de Breusch-Pagan.
- **Indépendance des observations (pseudo-réPLICATION).** Chaque individu doit appartenir à un et un seul groupe. En d'autres termes, les observations ne sont pas indépendantes si plusieurs mesures (variable continue) sont faites sur un même individu. Si c'est le cas, on utilisera alors une analyse de variance sur des mesures répétées (voir le bloc à la fin du chapitre).

Quelles sont les conséquences si les conditions d'application ne sont pas respectées ? La non vérification des conditions d'application cause deux problèmes distincts : elle affecte la puissance du test (sa capacité à détecter un effet, si celui-ci existe réellement) et le taux d'erreur de type 1 (la probabilité de trouver un résultat significatif alors qu'aucune relation n'existe réellement, soit un faux-positif) (Glass et al., 1972 ; Lix et al., 1996).

- Si la distribution est asymétrique plutôt que centrée (comme pour une distribution normale), la puissance et le taux d'erreur de type 1 sont tous les deux peu affectés car le test est non-orienté (la différence de moyennes peut être négative ou positive).

4.4. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À PLUS DE DEUX MO

- Si la distribution est leptocurtique (pointue, avec des extrémités de la distribution plus importantes, le taux d'erreur de type 1 est peu affecté, en revanche la puissance du test est réduite. L'inverse s'observe si la distribution est platicurtique (aplatie, c'est-à-dire avec des extrémités de la distribution plus réduites).
- Si les groupes ont des variances différentes, le taux d'erreur de type 1 augmente légèrement.
- Si les observations ne sont pas indépendantes, à la fois le taux d'erreurs de type 1 et la puissance du test sont fortement affectés.
- Si les échantillons sont petits, les effets présentés ci-dessus sont démultipliés.
- Si plusieurs conditions ne sont pas respectées, les conséquences présentées ci-dessus s'additionnent, voir se combinent.

Que faire quand les conditions d'application relatives à la normalité ou à l'homoscédasticité ne sont vraiment pas respectées? Signalons d'emblée que le non respect de ces conditions ne change rien à la décomposition de la variance ($VT=V_{\text{intra}}+V_{\text{inter}}$). Cela signifie que vous pouvez toujours calculer Eta². Par contre, le test de Fisher ne peut pas être utilisé car il sera biaisé comme décrit précédemment. Quatre solutions sont envisageables :

- Lorsque les échantillons sont fortement anormalement distribués, certains auteurs vont simplement transformer leur variable en appliquant une fonction logarithme (le plus souvent) ou racine carré, inverse ou exponentielle, et reporter le test de fisher calculé sur cette transformation. Attention toutefois! Transformer une variable ne va pas systématiquement la rapprocher d'une distribution normale et complique l'interprétation finale des résultats. Par conséquent, avant de recalculer votre test F , il convient de réaliser un test de normalité de Shapiro-Wilk et un test d'homoscédasticité (Levene, Bartlett ou/et Breusch-Pagan) sur la variable continue transformée.
- Détecter les observations qui contribuent le plus à l'anormalité et l'hétéroscélasticité, dites valeurs aberrantes (*outliers* en anglais). Supprimez les et refaites votre ANOVA en vous assurant que les conditions sont désormais respectées. Notez que supprimer des observations peut être une pratique éthiquement questionnable en statistique. Si vos échantillons sont bien constitués et que la mesure collectée n'est pas erronée, pourquoi donc la supprimer? Si vous optez pour cette solution, prenez soin de comparer les résultats avant et après la suppression des valeurs aberrantes. Si les conditions sont respectées après suppression et que les résultats de l'ANOVA (Eta² et test F de Fisher) sont très semblables, conservez donc les résultats de l'ANOVA initiale et signalez que vous avez procédez aux deux tests.
- Lorsque les variances des groupes sont dissemblables, vous pouvez utiliser le test de Welch pour l'ANOVA au lieu du test F de Fisher.
- Dernière solution, lorsque les deux conditions ne sont vraiment pas respectées, utilisez le test non paramétrique de Kruskal-Wallis. Par analogie au t de student, il correspond au test des rangs signés de Wilcoxon. Ce test est décrit dans la section suivante.

Vous l'aurez compris, dans de nombreux cas en statistique, les choix méthodologiques dépendent de la subjectivité du chercheur. Il faut s'adapter au jeu de données et à la culture statistique en vigueur dans votre champs d'étude. N'hésitez pas à réaliser plusieurs tests différents pour évaluer la robustesse de vos conclusions et fiez-vous en premier lieu à ceux pour lesquels votre jeu de données est le plus adapté.

4.4.2 Test non paramétrique de Kruskal-Wallis

Le test non paramétrique de Kruskal-Wallis est une alternative à l'ANOVA classique lorsque le jeu de données présente de graves problèmes de normalité et d'hétéroscélasticité. Cette méthode représente une ANOVA appliquée à une variable continue transformée préalablement en rangs. Du fait de la

transformation en rangs, on ne vérifie plus si les moyennes sont différentes, mais bel et bien si les médianes de la variable continue sont différentes. Pour ce faire, on utilisera la fonction `kruskal.test`.

4.4.3 Mise en œuvre dans

Dans une étude récente, Apparicio *et al.* (2018b) ont comparé les expositions au bruit et à la pollution atmosphérique aux heures de pointe à Montréal en fonction du mode de transport utilisé. Pour ce faire, trois équipes de trois personnes ont été constituées : un cycliste, un automobiliste et un utilisateur du transport en commun, équipés de capteurs de pollution, de sonomètres, de vêtements biométriques et d'une montre GPS. Chaque matin, à huit heures précises, les membres de chaque équipe ont réalisé un trajet d'un quartier périphérique de Montréal vers un pôle d'enseignement (université) ou d'emploi localisé au centre-ville. Le trajet inverse était réalisé le soir à 17h. Au total, une centaine de trajets ont ainsi été réalisés. Des analyses de variance ont ainsi permis de comparer entre les trois modes (automobile, vélo et transport en commun) : les temps de déplacement, les niveaux d'exposition au bruit, les niveaux d'exposition au dioxyde d'azote et la dose totale inhalée de dioxyde d'azote. Nous vous proposons ici d'analyser une partie de ces données.

4.4.3.1 Première ANOVA : différences entre les temps de déplacement

Comme première analyse de variance, nous allons vérifier si les moyennes des temps de déplacement sont différentes entre les trois modes de transport.

Dans un premier temps, nous pouvons calculer les moyennes des différents groupes. On peut alors constater que les moyennes sont très semblables : 37,7 minutes pour l'automobile versus 38,4 et 41,6 pour le vélo et le transport en commun. Aussi, les variances des trois groupes sont relativement similaires.

```
library("rstatix")
# chargement des dataframes
load("data/bivariee/dataPollution.RData")
# Statistiques descriptives pour les groupes (moyenne et écart-type)
df_TrajetsDuree %>%
  # Nom du dataframe
  group_by(Mode) %>%
  # Variable qualitative
  get_summary_stats(DureeMinute, type = "mean_sd") # Variable continue

## # A tibble: 3 x 5
##   Mode   variable     n   mean    sd
##   <chr>  <chr>     <dbl> <dbl> <dbl>
## 1 1. Auto DureeMinute     33  37.7  12.8
## 2 2. Velo DureeMinute     33  38.4  15.2
## 3 3. TC   DureeMinute     33  41.6  11.4
```

Pour visualiser la distribution des données pour les trois groupes, vous pouvez créer des graphiques de densité et en violons (figure 4.19). La juxtaposition des trois distributions montre que les distributions des valeurs pour les trois groupes sont globalement similaires. Cela est corroboré par le fait que les boîtes du graphique en violons sont situées à la même hauteur. Autrement dit, à la lecture des deux graphiques, ils ne semblent pas y avoir de différences significatives entre les trois groupes en terme de temps de déplacement.

```
library("ggplot2")
library("ggpubr")
# Graphique de densité
```

4.4. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À PLUS DE DEUX MODES

```
GraphDens <- ggplot(data = df_TrajetsDuree,
  mapping=aes(x=DureeMinute, colour=Mode, fill=Mode)) +
  geom_density(alpha=0.55,mapping=aes(y=..scaled...))+ 
  labs(title="a. graphique de densité",
  x = "Densité",
  y = "Durée du trajets (en minutes)")

# Graphique en violons
GraphViolon <- ggplot(df_TrajetsDuree, aes(x=Mode, y=DureeMinute)) +
  geom_violin(fill="white") +
  geom_boxplot(width=0.1, aes(x=Mode, y=DureeMinute, fill=Mode))+ 
  labs(title="b. Graphique en violons",
  x = "Mode de transport",
  y = "Durée du trajets (en minutes"))+
  theme(legend.position = "none")
ggarrange(GraphDens, GraphViolon)
```

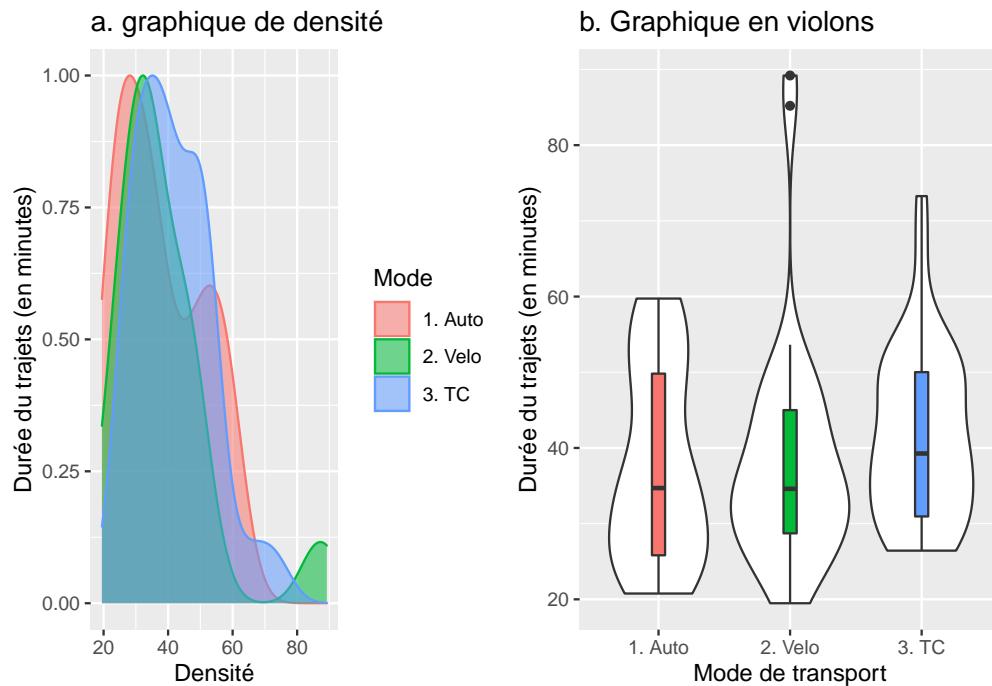


FIG. 4.19 : Graphiques de densité et en violons

Nous pouvons vérifier si les échantillons sont normalement distribués avec la fonction `shapiro.test` du package `rstatix`. À titre de rappel, l'hypothèse nulle (H_0) de ce test est que la distribution est normale. Par conséquent, quand la valeur de P associée à la statistique de Shapiro est supérieure à 0,05 alors on ne peut rejeter l'hypothèse d'une distribution normale (autrement dit, la distribution est anormale). À la lecture des sorties ci-dessous, seul le groupe des utilisateurs en transport en commun présente une distribution proche de la normalité ($p=0,0504$). Ce test étant très restrictif, il est fortement conseillé de visualiser le diagramme quantile-quantile pour chaque groupe (graphique QQ plot) (figure 4.20). Ces graphiques sont utilisés pour déterminer visuellement si une distribution empirique (observées sur des données), s'approche d'une distribution théorique (ici la loi normale). Si effectivement les deux distributions sont proches, les points du diagramme devraient tous tomber sur une ligne droite parfaite. Un intervalle de confiance (représenté ici en gris) peut être construit pour obtenir une interprétation plus nuancée. Dans notre cas, seules deux observations pour le vélo et deux autres pour l'automobile s'éloignent vraiment de la ligne droite. On peut considérer que ces trois distributions s'approchent d'une distribution normale.

```

library("dplyr")
library("ggpubr")
library("rstatix")
# Condition 1 : normalité des échantillons
# Test pour la normalité des échantillons (groupes) : test de Shapiro
df_TrajetsDuree %>%
  group_by(Mode) %>%
  shapiro_test(DureeMinute) # Variable continue

## # A tibble: 3 x 4
##   Mode   variable   statistic      p
##   <chr>  <chr>       <dbl>    <dbl>
## 1 1. Auto DureeMinute  0.905  0.00729
## 2 2. Velo DureeMinute  0.797  0.0000288
## 3 3. TC   DureeMinute  0.936  0.0504

# Graphiques qqplot pour les groupes
ggqqplot(df_TrajetsDuree, "DureeMinute", facet.by = "Mode")

```

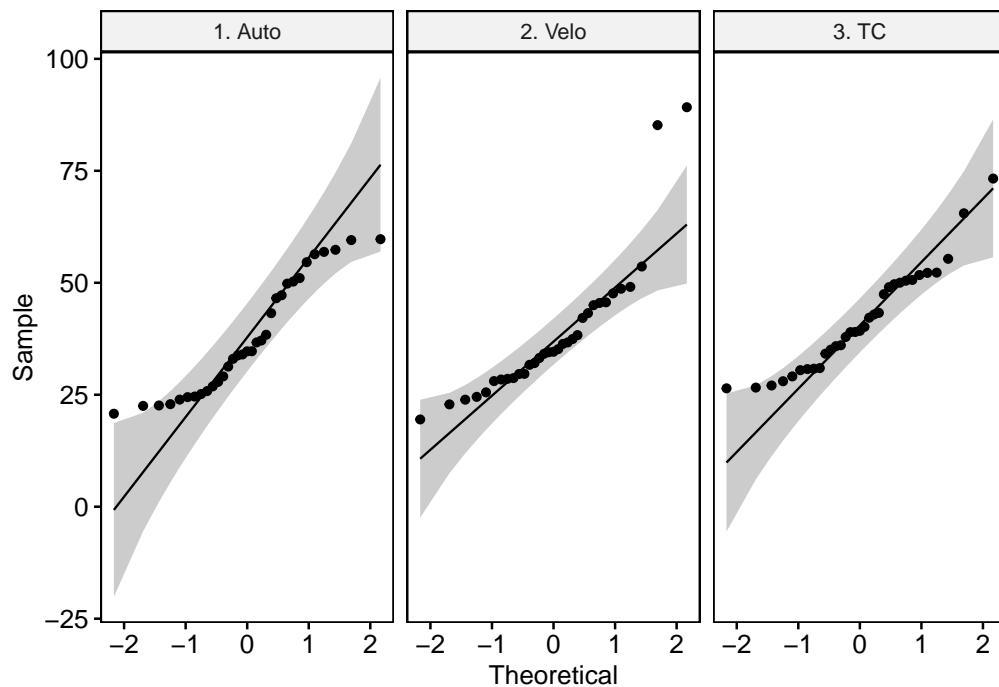


FIG. 4.20 : QQ Plot pour les groupes

Pour vérifier l'hypothèse d'homogénéité des variances, vous pouvez utiliser les tests de Levene, de Bartlett ou de Breusch-Pagan. Les valeurs de P , toutes supérieures à 0,05, signalent que la condition d'homogénéité des variances est respectée.

```

library("rstatix")
library("lmtest")
library("car")
# Condition 2 : homogénéité des variances (homocédasticité)
leveneTest(DureeMinute ~ Mode, data = df_TrajetsDuree)

```

4.4. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À PLUS DE DEUX MODES

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group     2  0.2418 0.7857
##             96
```

```
bartlett.test(DureeMinute ~ Mode, data = df_TrajetsDuree)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data: DureeMinute by Mode
## Bartlett's K-squared = 2.6718, df = 2, p-value = 0.2629
```

```
bptest(DureeMinute ~ Mode, data = df_TrajetsDuree)
```

```
##
## studentized Breusch-Pagan test
##
## data: DureeMinute ~ Mode
## BP = 1.3322, df = 2, p-value = 0.5137
```

Deux fonctions peuvent être utilisées pour calculer l'analyse de variance : la fonction de base `aov`(variable continue ~ variable qualitative, data = votre dataframe) ou bien la fonction `anova_test`(variable continue ~ variable qualitative, data = votre dataframe) du package `rstatix`. Comparativement à `aov`, l'avantage de la fonction `anova_test` est qu'elle calcule aussi le Eta².

```
library("rstatix")
library("lmtest")
library("car")
# ANOVA avec la fonction aov
aov1 <- aov(DureeMinute ~ Mode, data = df_TrajetsDuree)
summary(aov1)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## Mode        2    287   143.2    0.82  0.444
## Residuals  96   16781   174.8
```

```
# calcul de Eta2 avec la fonction eta_sq du package lmtest
eta_sq(aov1)
```

```
## term etasq
## 1 Mode 0.017
```

```
# ANOVA avec la fonction anova_test du package rstatix
anova_test(DureeMinute ~ Mode, data = df_TrajetsDuree)
```

```
## ANOVA Table (type II tests)
##
##   Effect DFn DFD  F      p p<.05   ges
## 1 Mode    2   96 0.82 0.444       0.017
```

La valeur de P associée à la statistique F (0,444) nous permet de conclure qu'il n'y a pas de différences significatives entre les moyennes des temps de déplacements des trois modes de transport.

4.4.3.2 Deuxième ANOVA : différences entre les niveaux d'exposition au bruit

Dans ce second exercice, nous allons analyser les différences d'exposition au bruit. D'emblée, les statistiques descriptives révèlent que les moyennes sont dissemblables : 66,8 dB(A) pour l'automobile versus 68,8 et 74 pour le vélo et le transport en commun. Aussi, la variance du transport en commun est très différente des autres.

```
library("rstatix")
# chargement des dataframes
load("data/bivariee/dataPollution.RData")
# Statistiques descriptives pour les groupes (moyenne et écart-type)
df_Bruit %>%
  group_by(Mode) %>%
  get_summary_stats(laeq, type = "mean_sd") # Variable continue

## # A tibble: 3 x 5
##   Mode   variable     n   mean    sd
##   <chr>  <chr>     <dbl> <dbl> <dbl>
## 1 1. Auto laeq      1094  66.8  4.56
## 2 2. Velo laeq      1124  68.8  4.29
## 3 3. TC   laeq      1207  74.0  6.79
```

À la lecture des graphiques de densité et en violon (figure 4.21), il semble clair que les niveaux d'exposition au bruit sont plus faibles pour les automobilistes et plus élevés pour les cyclistes et surtout les utilisateurs du transport en commun. En outre, la distribution des valeurs d'exposition au bruit dans le transport en commun semble bimodale. Cela s'explique par le fait que les niveaux de bruit sont beaucoup élevés dans le métro que dans les autobus.

```
library("ggplot2")
library("ggpubr")
# Graphique en densité
GraphDens <- ggplot(data = df_Bruit,
  mapping=aes(x=laeq, colour=Mode, fill=Mode)) +
  geom_density(alpha=0.55, mapping=aes(y=..scaled...))+
  labs(title="a. graphique de densité",
       x="Exposition au bruit (dB(A))")

# Graphique en violons
GraphViolon <- ggplot(df_Bruit, aes(x=Mode, y=laeq)) +
  geom_violin(fill="white") +
  geom_boxplot(width=0.1, aes(x=Mode, y=laeq, fill=Mode))+ 
  labs(title="b. Graphique en violons",
       x = "Mode de transport",
       y="Exposition au bruit (dB(A)))"+
  theme(legend.position = "none")
ggarrange(GraphDens, GraphViolon)
```

Le test de Shapiro et les graphiques QQ plot (figure 4.22) révèlent que les distributions des trois groupes sont anormales. Ce résultat n'est pas surprenant si l'on tient compte de la nature logarithmique de l'échelle décibel.

4.4. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À PLUS DE DEUX MODES

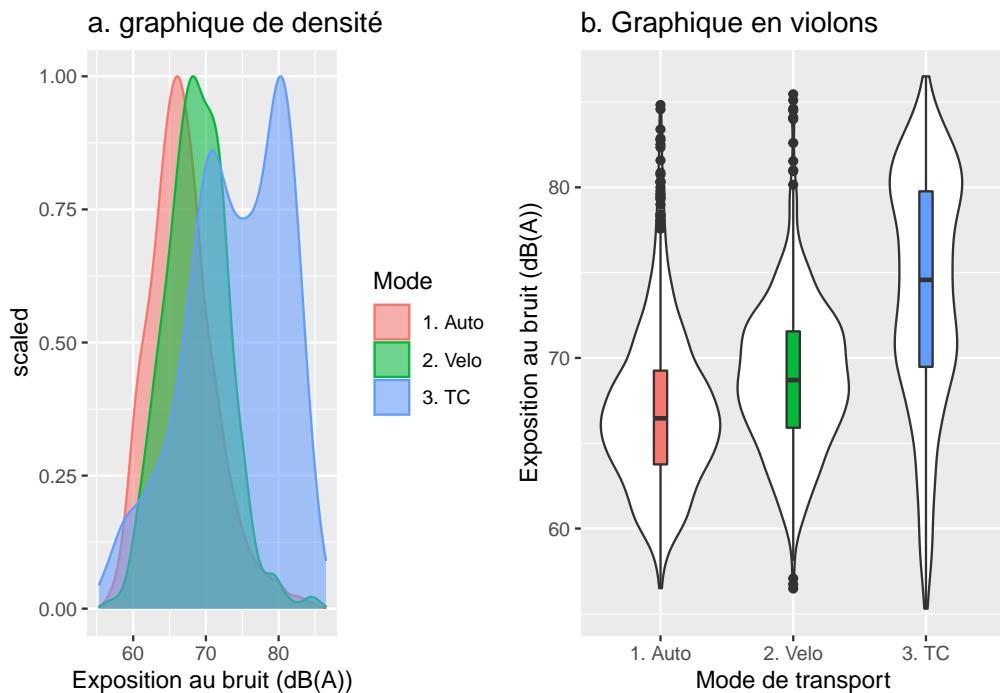


FIG. 4.21 : Graphique de densité et en violons

```

library("dplyr")
library("ggpubr")
library("rstatix")
# Condition 1 : normalité des échantillons
# Test pour la normalité des échantillons (groupes) : test de Shapiro
df_Bruit %>%          # Nom du dataframe
  group_by(Mode) %>%      # Variable qualitative
  shapiro_test(laeq) # Variable continue

## # A tibble: 3 x 4
##   Mode     variable statistic      p
##   <chr>    <chr>      <dbl>    <dbl>
## 1 1. Auto  laeq       0.971 4.92e-14
## 2 2. Velo  laeq       0.992 5.12e- 6
## 3 3. TC    laeq       0.966 3.34e-16

# Graphiques qqplot pour les groupes
ggqqplot(df_Bruit, "laeq", facet.by = "Mode")

```

En outre, selon les valeurs des tests de Levene, de Bartlett ou de Breusch-Pagan, les variances ne sont pas égales.

```

library("rstatix")
library("lmtest")
library("car")
# Condition 2 : homogénéité des variances (homocédasticité)
leveneTest(laeq ~ Mode, data = df_Bruit)

```

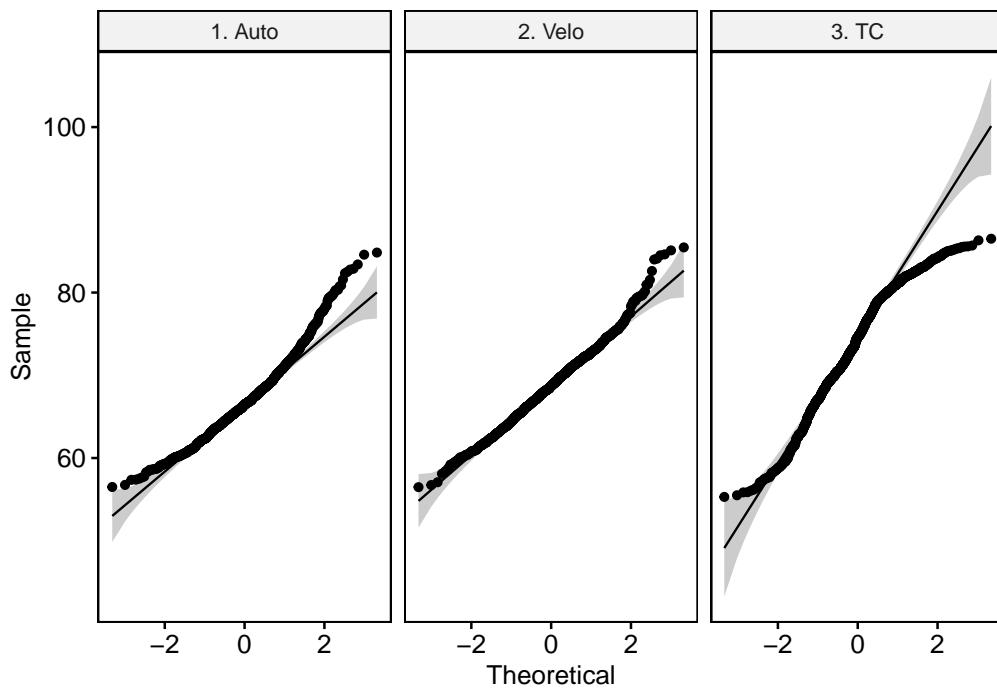


FIG. 4.22 : QQ Plot pour les groupes

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group      2 190.3 < 2.2e-16 ***
##            3422
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
bartlett.test(laeq ~ Mode, data = df_Bruit)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data: laeq by Mode
## Bartlett's K-squared = 306.64, df = 2, p-value < 2.2e-16
```

```
bptest(laeq ~ Mode, data = df_Bruit)
```

```
##
## studentized Breusch-Pagan test
##
## data: laeq ~ Mode
## BP = 279.85, df = 2, p-value < 2.2e-16
```

Étant donné que les deux conditions (normalité et homogénéité des variances) ne sont pas respectées, il serait préférable d'utiliser un test non paramétrique de Kruskal-Wallis. Calculons toutefois préalablement l'ANOVA classique et l'ANOVA de Welch puisque les variances ne sont pas égales). Les valeurs de P des deux tests (Fisher et Welch) signalent que les moyennes d'exposition au bruit sont statistiquement différentes entre les trois modes de transport.

4.4. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À PLUS DE DEUX MODES

```
library("rstatix")
# ANOVA avec la fonction anova_test du package rstatix
anova_test(laeq ~ Mode, data = df_Bruit)

## ANOVA Table (type II tests)
##
##   Effect DFn    DFd      F      p p<.05   ges
## 1   Mode     2 3422 544.214 6.12e-206      * 0.241

# ANOVA avec le test de Welch puisque les variances ne sont pas égales
welch_anova_test(laeq ~ Mode, data = df_Bruit)

## # A tibble: 1 x 7
##   .y.      n statistic   DFn   DFd      p method
## * <chr> <int>      <dbl> <dbl> <dbl>    <dbl> <chr>
## 1 laeq     3425       446.      2 2248. 9.47e-164 Welch ANOVA
```

Une fois démontré que les moyennes sont différentes, le test de Tukey est particulièrement intéressant puisqu'il nous permet de repérer les différences de moyennes significatives deux à deux, tout en ajustant les valeurs de P obtenues en fonction du nombre de comparaisons effectuées. Ci-dessous, on constate que toutes les paires sont statistiquement différentes et que la différence de moyennes entre les automobilistes et les cyclistes est de 1,9 dB(A) et surtout de 7,1 dB(A) entre les automobilistes et les usagers du transport en commun.

```
aov2 <- aov(laeq ~ Mode, data = df_Bruit)
# Test de Tukey pour comparer les moyennes entre elles
TukeyHSD(aov2, conf.level = 0.95)
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = laeq ~ Mode, data = df_Bruit)
##
## $Mode
##          diff      lwr      upr p adj
## 2. Velo-1. Auto 1.941698 1.406343 2.477053 0
## 3. TC-1. Auto  7.113506 6.587309 7.639703 0
## 3. TC-2. Velo  5.171808 4.649307 5.694309 0
```

Le calcul de test non paramétrique de Kruskal-Wallis avec la fonction kruskal.test démontre aussi que les médianes des groupes sont différentes ($p<0,001$). De manière comparable au test de Tukey, la fonction pairwise.wilcox.test permet aussi de repérer les différences significatives entre les paires de groupes. Pour conclure, tant l'ANOVA que le test non paramétrique de Kruskal-Wallis indiquent que les trois modes de transport sont significativement différents quant à l'exposition au bruit, avec des valeurs plus faibles pour les automobilistes comparativement aux cyclistes et aux usagers du transport en commun.

```
# Test de Kruskal-Wallis
kruskal.test(laeq ~ Mode, data = df_Bruit)

##
```

```
## Kruskal-Wallis rank sum test
##
## data: laeq by Mode
## Kruskal-Wallis chi-squared = 784.74, df = 2, p-value < 2.2e-16
```

```
# Calcul de la moyenne des rangs pour les trois groupes
df_Bruit$laeqRank <- rank(df_Bruit$laeq)
df_Bruit %>%
  group_by(Mode) %>%
  get_summary_stats(laeqRank, type = "mean")
```

```
## # A tibble: 3 x 4
##   Mode     variable     n   mean
##   <chr>    <chr>     <dbl> <dbl>
## 1 1. Auto laeqRank 1094 1188.
## 2 2. Velo laeqRank 1124 1572.
## 3 3. TC   laeqRank 1207 2320.
```

```
# Comparaison des groupes avec la fonction pairwise.wilcox.test
pairwise.wilcox.test(df_Bruit$laeq, df_Bruit$Mode, p.adjust.method = "BH")
```

```
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: df_Bruit$laeq and df_Bruit$Mode
##
##          1. Auto 2. Velo
## 2. Velo <2e-16 -
## 3. TC    <2e-16 <2e-16
##
## P value adjustment method: BH
```

4.4.4 Comment rapporter les résultats d'une ANOVA et du test de Kruskal-Wallis

Plusieurs éléments doivent être reportés pour détailler les résultats d'une ANOVA ou d'un test de Kruskal-Wallis : la valeur de F , de W (dans le cas d'une ANOVA de Welch) ou du χ^2 (Kruskal-Wallis), les valeurs de P , les moyennes ou médianes respectives des groupes et éventuellement un tableau détaillant les écarts intergroupes obtenus avec les test de Tukey ou Wilcoxon par paires.

- Les résultats de l'analyse de variance à un facteur démontrent que le mode de transport utilisé n'a pas d'effet significatif sur le temps de déplacement en heures de pointe à Montréal ($F(2,96)=0,82$, $p=0,444$). En effet, pour des trajets de dix kilomètres entre un quartier périphérique et le centre-ville, les cyclistes (Moy=38,4, ET=15,2) arrivent en moyenne moins d'une minute après les automobilistes (Moy=37,7, ET=12,8) et les usagers du transport en commun moins de quatre minutes (Moy=41,6, ET=11,4).
- Les résultats de l'analyse de variance à un facteur démontrent que le mode transport utilisé a un impact significatif sur le niveau d'exposition en heures de pointe à Montréal ($F(2,96)=544$, $p<0,001$ et $Welch(2,96)=446$, $p<0,001$). En effet, les usagers du transport en commun (Moy=74,0, ET=6,79) et les cyclistes (Moy=68,8, ET=4,3) sont significativement plus exposés au bruit que les automobilistes (Moy=66,8, ET=4,56).

4.4. RELATION ENTRE UNE VARIABLE QUANTITATIVE ET UNE VARIABLE QUALITATIVE À PLUS DE DEUX MODES

- Les résultats du test de Kruskal-Wallis démontre qu'il existe des différences significatives d'exposition au bruit entre les trois modes de transport ($\chi^2(2) = 784,74$, $p < 0,001$) avec des moyennes de rangs de 1094 pour l'automobile, 1124 pour le vélo et 1207 pour le transport en commun.



Nous avons vu que l'ANOVA permet de comparer les moyennes d'une variable continue à partir d'une variable qualitative comprenant plusieurs modalités (facteur) pour des observations indépendantes. Il y a donc une seule variable dépendante (continue) et une seule variable indépendante. Sachez qu'il existe de nombreuses extensions de l'ANOVA classique :

- **une ANOVA à deux facteurs**, soit avec une variable dépendante continue et deux variables indépendantes qualitatives (*two-way ANOVA* en anglais). On évalue ainsi les effets des deux variables (a, b) et de leur interaction (ab) sur une variable continue.
- **une ANOVA multifacteur** avec une variable dépendante continue et plus de deux variables indépendantes qualitatives. Par exemple, avec trois variables qualitatives pour expliquer la variable continue, on inclut les effets de chaque variable qualitative (a, b, c) ainsi que de leurs interactions (ab, ac, bc, abc).
- **L'analyse de covariance (ANCOVA, ANalysis of COVAriance en anglais)** comprend une variable dépendante continue, une variable indépendante qualitative (facteur) et plusieurs variables indépendantes continues dites covariables. L'objectif est alors de vérifier si les moyennes d'une variable dépendante sont différentes pour plusieurs groupes d'une population donnée, après avoir contrôlé l'effet d'une ou plusieurs variables continues. Par exemple, pour une métropole donnée, on pourra vouloir comparer les moyennes de loyers entre la ville-centre et ceux de la première et de la seconde couronnes (facteur), une fois contrôlée la taille de ces derniers (variable covariée continue). En effet, une partie de la variance des loyers s'explique certainement par la taille des logements.
- **L'analyse de variance multivariée (MANOVA, Multivariate ANalysis of VAriance en anglais)** comprend deux variables dépendantes continues ou plus et une variable indépendante qualitative (facteur). Par exemple, on souhaiterait comparer les moyennes d'exposition au bruit et à différents polluants (dioxyde d'azote, particules fines, ozone) (variables dépendantes continues) selon le mode de transport utilisé (automobile, vélo, transport en commun, soit le facteur).
- **L'analyse de covariance multivariée (MANCOVA, Multivariate ANalysis of COVAriance en anglais)**, soit une analyse qui comprend deux variables dépendantes continues ou plus (comme la MANOVA) et une variable qualitative comme variable indépendante (facteur) et un covariable continue ou plus.

Pour le test t , nous avons vu qu'il peut s'appliquer soit à deux échantillons indépendants (non appariés), soit à deux échantillons dépendants (appariés). Notez qu'il existe aussi des extensions de l'ANOVA pour des échantillons paires. On parle alors d'**analyse de variance sur des mesures répétées**. Par exemple, on pourrait évaluer la perception du sentiment de sécurité relativement à la pratique vélo d'hiver pour un échantillon de cyclistes ayant décidé de l'adopter récemment, et ce, à plusieurs moments : avant leur première saison, à la fin de leur premier hiver, à la fin de leur second hiver. Autre exemple, on pourrait sélectionner un échantillon d'individus (100 par exemple) pour lesquels on évalueraient leurs perceptions de l'environnement sonore dans différents lieux de la ville. Comme pour l'ANOVA classique (échantillons non appariés), il existe des extensions de l'ANOVA sur des mesures répétées permettant d'inclure plusieurs facteurs (groupes de population) ; on mesure alors une variable continue pour plusieurs groupes d'individus à différents moments ou conditions différentes. Il est aussi possible de réaliser une ANOVA pour des mesures répétées avec une ou plusieurs covariables continues.

Bref, si l'ANOVA était un roman, elle serait certainement «un monde sans fin» de Ken Follett ! Notez toutefois que la SUPERNOVA, la BOSSANOVA et le CASANOVA ne sont pas des variantes de l'ANOVA !

Quatrième partie

Modèles de régression

Chapitre 5

La régression linéaire multiple

À juste titre, Bressoux (2010, pp 15) signale que «la causalité relève d'une interprétation théorique de la relation et celle-ci n'est jamais déterminée par les calculs statistiques».

Chapitre 6

Régressions linéaires généralisées (GLM)

6.1 Qu'est qu'un modèle GLM?

6.1.1 Principes de base des GLM

6.1.2 Reformulation d'un modèle OLS sous forme GLM

6.2 Les principaux modèles GLM utilisées

6.2.1 Les régressions logistiques

6.2.1.1 Régression logit et probit simple

6.2.1.2 Régression logistique multinomiale

6.2.1.3 Régression logistique ordinale

6.2.2 Les régressions de type Poisson

6.3 D'autres modèles GLM

Student, Beta, GAMMA, Exponentiel

Chapitre 7

Régressions à effets mixtes (GLMM) et régression multiniveaux

7.1 Principes de base des GLMM

Notion de pseudo-réplications

Effets aléatoire *versus* effets fixes

7.2 Application à une variable continue

7.3 Régressions multiniveaux

Chapitre 8

Modèles généralisés additifs

8.1 Principes de base des GAM

8.2 Extensions des GAM

8.2.1 GAMM

8.2.2 GAMMAR

Cinquième partie

Analyses exploratoires multivariées

Chapitre 9

Méthodes factorielles

9.1 Un petit historique

9.2 Analyses de composantes principales (ACP)

9.3 Analyses factorielles de correspondances (AFC)

9.4 Analyses factorielles de correspondances multiples (AFM)

9.5 Analyses factorielles de correspondances mixte

Chapitre 10

Méthodes de classification non-supervisées

10.1 Un aperçu sur la multitude des méthodes de classifications

10.2 Combinaisons méthodes factorielles et méthodes de classifications

10.3 Classification ascendantes hiérachiques

10.4 Nuées dynamiques

10.4.1 k-means

10.4.2 k-median

10.4.3 Les extensions en logique floues : c-means, c-median

Chapitre 11

Conclusion

11.1 Synthèse des méthodes abordées

Road map method

11.2 Autres méthodes non abordées

11.2.1 Régressions par quantile

11.2.2 Régressions par panel

11.2.3 Régressions par Tobit

11.2.4 Analyses de survie

11.2.5 Équations structurelles

11.3 Deux écoles de statistique inférielle : fréquentiste et bayésienne

11.4 Éthique en méthodes quantitatives

Chapitre 12

Annexes

12.1 Tableau des valeurs critiques de χ^2

La courte syntaxe  ci-dessous permet de générer un tableau avec les valeurs critiques du χ^2 pour différents degrés de signification (valeurs de p).

```
library(stargazer)

# vecteur pour les degrés de liberté de 1 à 30, puis 40 et 50
dl <- c(1:30, 40, 50, 100, 250, 500)
# la fonction qchisq permet d'obtenir la valeur théorique en fonction
# d'une valeur de p et d'un nombre de degrés de liberté
tableT <- cbind(dl,
                  p0.05 = round(qchisq(p=0.95, df=dl, lower.tail = FALSE),3),
                  p0.01 = round(qchisq(p=0.99, df=dl, lower.tail = FALSE),3),
                  p0.001 = round(qchisq(p=0.999, df=dl, lower.tail = FALSE),3))
# Impression du tableau avec la library stargazer
stargazer(tableT, type="text", summary=FALSE, rownames=FALSE, align = TRUE, digits = 3,
          title="Distribution des valeurs critiques du Khi2")
```

12.2 Tableau des valeurs critiques de F

La courte syntaxe  ci-dessous permet de générer un tableau avec les valeurs critiques de F avec $p=0,05$.

```
dl1 <- c(1:10, 15, 20, 25, 30, 50, 100, 500)
dl2 <- c(1:30, 40, 50, 100)
matrice <- matrix(ncol=length(dl1), nrow=length(dl2), byrow = TRUE)
for(r in 1:length(dl1)){
  for(c in 1:length(dl2)){
    matrice[c,r] <- round(qf(p=0.05, dl1[r], dl2[c], lower.tail = FALSE),2)
  }
}

tableF_p0.05 <- as.data.frame(matrice, row.names = paste0("dl2=",dl2, sep=""))
colnames(tableF_p0.05) <- paste0("dl1=",dl1, sep="")
```

TAB. 12.1 : Distribution des valeurs critiques du khi2

dl	p=0,05	p=0,01	p=0,001
1	0.004	0.000	0.000
2	0.103	0.020	0.002
3	0.352	0.115	0.024
4	0.711	0.297	0.091
5	1.145	0.554	0.210
6	1.635	0.872	0.381
7	2.167	1.239	0.598
8	2.733	1.646	0.857
9	3.325	2.088	1.152
10	3.940	2.558	1.479
11	4.575	3.053	1.834
12	5.226	3.571	2.214
13	5.892	4.107	2.617
14	6.571	4.660	3.041
15	7.261	5.229	3.483
16	7.962	5.812	3.942
17	8.672	6.408	4.416
18	9.390	7.015	4.905
19	10.117	7.633	5.407
20	10.851	8.260	5.921
21	11.591	8.897	6.447
22	12.338	9.542	6.983
23	13.091	10.196	7.529
24	13.848	10.856	8.085
25	14.611	11.524	8.649
26	15.379	12.198	9.222
27	16.151	12.879	9.803
28	16.928	13.565	10.391
29	17.708	14.256	10.986
30	18.493	14.953	11.588
40	26.509	22.164	17.916
50	34.764	29.707	24.674
100	77.929	70.065	61.918
250	214.392	200.939	186.554
500	449.147	429.388	407.947

```
knitr::kable(
  head(tableF_p0.05, n=nrow(tableF_p0.05)), booktabs = TRUE,
  row.names = TRUE,
  caption = 'Distribution des valeurs critiques de F avec p=0,05'
)
```

TAB. 12.2 : Distribution des valeurs critiques de F avec p=0,05

	dl1=1	dl1=2	dl1=3	dl1=4	dl1=5	dl1=6	dl1=7	dl1=8	dl1=9	dl1=10	dl1=15	dl1=20
dl2=1	161.45	199.50	215.71	224.58	230.16	233.99	236.77	238.88	240.54	241.88	245.95	248.01
dl2=2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40	19.43	19.45
dl2=3	10.13	9.55	9.28	9.12	9.01	8.94	8.89	8.85	8.81	8.79	8.70	8.66
dl2=4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96	5.86	5.80
dl2=5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.77	4.74	4.62	4.56
dl2=6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06	3.94	3.87
dl2=7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.64	3.51	3.44
dl2=8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.35	3.22	3.15
dl2=9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.14	3.01	2.94
dl2=10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.98	2.85	2.77
dl2=11	4.84	3.98	3.59	3.36	3.20	3.09	3.01	2.95	2.90	2.85	2.72	2.65
dl2=12	4.75	3.89	3.49	3.26	3.11	3.00	2.91	2.85	2.80	2.75	2.62	2.54
dl2=13	4.67	3.81	3.41	3.18	3.03	2.92	2.83	2.77	2.71	2.67	2.53	2.46
dl2=14	4.60	3.74	3.34	3.11	2.96	2.85	2.76	2.70	2.65	2.60	2.46	2.39
dl2=15	4.54	3.68	3.29	3.06	2.90	2.79	2.71	2.64	2.59	2.54	2.40	2.33
dl2=16	4.49	3.63	3.24	3.01	2.85	2.74	2.66	2.59	2.54	2.49	2.35	2.28
dl2=17	4.45	3.59	3.20	2.96	2.81	2.70	2.61	2.55	2.49	2.45	2.31	2.23
dl2=18	4.41	3.55	3.16	2.93	2.77	2.66	2.58	2.51	2.46	2.41	2.27	2.19
dl2=19	4.38	3.52	3.13	2.90	2.74	2.63	2.54	2.48	2.42	2.38	2.23	2.16
dl2=20	4.35	3.49	3.10	2.87	2.71	2.60	2.51	2.45	2.39	2.35	2.20	2.12
dl2=21	4.32	3.47	3.07	2.84	2.68	2.57	2.49	2.42	2.37	2.32	2.18	2.10
dl2=22	4.30	3.44	3.05	2.82	2.66	2.55	2.46	2.40	2.34	2.30	2.15	2.07
dl2=23	4.28	3.42	3.03	2.80	2.64	2.53	2.44	2.37	2.32	2.27	2.13	2.05
dl2=24	4.26	3.40	3.01	2.78	2.62	2.51	2.42	2.36	2.30	2.25	2.11	2.03
dl2=25	4.24	3.39	2.99	2.76	2.60	2.49	2.40	2.34	2.28	2.24	2.09	2.01
dl2=26	4.23	3.37	2.98	2.74	2.59	2.47	2.39	2.32	2.27	2.22	2.07	1.99
dl2=27	4.21	3.35	2.96	2.73	2.57	2.46	2.37	2.31	2.25	2.20	2.06	1.97
dl2=28	4.20	3.34	2.95	2.71	2.56	2.45	2.36	2.29	2.24	2.19	2.04	1.96
dl2=29	4.18	3.33	2.93	2.70	2.55	2.43	2.35	2.28	2.22	2.18	2.03	1.94
dl2=30	4.17	3.32	2.92	2.69	2.53	2.42	2.33	2.27	2.21	2.16	2.01	1.93
dl2=40	4.08	3.23	2.84	2.61	2.45	2.34	2.25	2.18	2.12	2.08	1.92	1.84
dl2=50	4.03	3.18	2.79	2.56	2.40	2.29	2.20	2.13	2.07	2.03	1.87	1.78
dl2=100	3.94	3.09	2.70	2.46	2.31	2.19	2.10	2.03	1.97	1.93	1.77	1.68

Bibliographie

- Apparicio, P. (2002). *Apport des systèmes d'information géographique à l'étude de l'insertion des HLM dans les quartiers montréalais*. PhD thesis, Université du Maine.
- Apparicio, P., Carrier, M., Gelb, J., Séguin, A.-M., and Kingham, S. (2016). Cyclists' exposure to air pollution and road traffic noise in central city neighbourhoods of montreal. *Journal of Transport Geography*, 57 :63–69.
- Apparicio, P. and Gelb, J. (2020). Cyclists' exposure to road traffic noise : A comparison of three north american and european cities. *Acoustics*, 2(1) :73–86.
- Apparicio, P., Gelb, J., Carrier, M., Mathieu, M.-È., and Kingham, S. (2018a). Exposure to noise and air pollution by mode of transportation during rush hours in montreal. *Journal of Transport Geography*, 70 :182–192.
- Apparicio, P., Gelb, J., Carrier, M., Mathieu, M.-È., and Kingham, S. (2018b). Exposure to noise and air pollution by mode of transportation during rush hours in montreal. *Journal of Transport Geography*, 70 :182–192.
- Apparicio, P., Gelb, J., Dubé, A.-S., Kingham, S., Gauvin, L., and Robitaille, É. (2017). The approaches to measuring the potential spatial access to urban health services revisited : distance types and aggregation-error issues. *International journal of health geographics*, 16(1) :32.
- Apparicio, P., Gelb, J., and Mathieu, M.-È. (2019). Un atlas-web pour comparer l'exposition individuelle aux pollutions atmosphérique et sonore selon le mode de transport. *Cybergeo : European Journal of Geography*, (903).
- Apparicio, P. and Séguin, A.-M. (2006). L'insertion des hlm montréalaises dans le milieu social environnant. *L'Espace géographique*, 35(1) :63–85.
- Bressoux, P. (2010). *Modélisation statistique appliquée aux sciences sociales*. De boeck Bruxelles.
- Buregeya, J. M., Apparicio, P., and Gelb, J. (2020). Short-term impact of traffic-related particulate matter and noise exposure on cardiac function. *International Journal of Environmental Research and Public Health*, 17(4) :1220.
- Carrier, M., Apparicio, P., Séguin, A.-M., and Crouse, D. (2014). Ambient air pollution concentration in montreal and environmental equity : Are children at risk at school? *Case Studies on Transport Policy*, 2(2) :61–69.
- Cohen, J. (1992). A power primer. *Psychological bulletin*, 112(1) :155–159.
- Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Academic press.
- De Alvarenga, B., Apparicio, P., and Séguin, A.-M. (2018). L'accès aux aires de jeux dans les parcs de la communauté métropolitaine de montréal. *Cahiers de géographie du Québec*, 62(176) :229–246.

- Delaunay, D., Apparicio, P., Séguin, A.-M., Gelb, J., and Carrier, M. (2019). L'identification des zones calmes et un diagnostic d'équité environnementale à montréal. *The Canadian Geographer/Le Géographe canadien*, 63(2) :184–197.
- Field, A. P., Miles, J., and Field, Z. (2012). Discovering statistics using r.
- Gelb, J. and Apparicio, P. (2019). Noise exposure of cyclists in ho chi minh city : A spatio-temporal analysis using non-linear models. *Applied Acoustics*, 148 :332–343.
- Gelb, J. and Apparicio, P. (2020). Modelling cyclists' multi-exposure to air and noise pollution with low-cost sensors—the case of paris. *Atmosphere*, 11(4) :422.
- Glass, G. V., Peckham, P. D., and Sanders, J. R. (1972). Consequences of failure to meet assumptions underlying the fixed effects analyses of variance and covariance. *Review of educational research*, 42(3) :237–288.
- Joanes, D. and Gill, C. (1998). Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society : Series D (The Statistician)*, 47(1) :183–189.
- Lebart, L., Morineau, A., and Piron, M. (1995). *Statistique exploratoire multidimensionnelle*, volume 3. Dunod Paris.
- Lix, L. M., Keselman, J. C., and Keselman, H. (1996). Consequences of assumption violations revisited : A quantitative review of alternatives to the one-way analysis of variance f test. *Review of educational research*, 66(4) :579–619.
- Messerli, F. H. (2012). Chocolate consumption, cognitive function, and nobel laureates. *The new England Journal of Medicine*, 367(16) :1563–1564.
- Mihalcea, R. and Tarau, P. (2004). Textrank : Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.
- Pham, T.-T.-H., Apparicio, P., Séguin, A.-M., Landry, S., Gagnon, M., et al. (2012). Spatial distribution of vegetation in montreal : an uneven distribution or environmental inequity? *Landscape and urban planning*, 107(3) :214–224.
- Pumain, D. and Béguin, M. (1994). *La représentation des données géographiques : statistique et cartographie*. Armand Colin.
- Razali, N. M., Wah, Y. B., et al. (2011). Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*, 2(1) :21–33.
- Reed, W. J. (2002). On the rank-size distribution for human settlements. *Journal of Regional Science*, 42(1) :1–17.
- Reed, W. J. and Jorgensen, M. (2004). The double pareto-lognormal distribution : A new parametric model for size distributions. *Communications in Statistics - Theory and Methods*, 33(8) :1733–1753.
- Sawilowsky, S. S. (2009). New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, 8(2) :467–474.
- Schwarzkopf, D., de Haas, B., and Rees, G. (2012). Better ways to improve standards in brain-behavior correlation analysis. *Frontiers in Human Neuroscience*, 6 :200.
- Sean, O. (2018). Common probability distributions.
- Sobol, I. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematics and Computers in Simulation*, 1(4) :407–414.

- Tabachnick, B. G., Fidell, L. S., and Ullman, J. B. (2007). *Using multivariate statistics*, volume 5. Pearson Boston, MA.
- Wickham, H. (2010). A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1) :3–28.
- Wilcox, R. R. (1994). The percentage bend correlation coefficient. *Psychometrika*, 59(4) :601–616.
- Yap, B. W. and Sim, C. H. (2011). Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, 81(12) :2141–2155.

Index

corrélation, 190

corrélation partielle, 204

covariance, 190, 192

khi2, 218

tableau de contingence, 218