



Méthodes quantitatives en sciences sociales : un grand bol d'R

Philippe Apparicio et Jérémy Gelb

Version : 7 mai 2022

Auteurs : Philippe Apparicio et Jérémie Gelb

Remerciements : Ce manuel a été réalisé avec le soutien de la fabriqueREL. Fondée en 2019, la fabrique-REL est portée par divers établissements d'enseignement supérieur du Québec et agit en collaboration avec les services de soutien pédagogique et les bibliothèques. Son but est de faire des ressources éducatives libres (REL) le matériel privilégié en enseignement supérieur au Québec.

Maquette de la page couverture : Graphe Logo (<https://www.graphelogo.com/>)

Mise en page : Philippe Apparicio et Jérémie Gelb

Relecture : Denise Latreille

© Philippe Apparicio et Jérémie Gelb

Pour citer cet ouvrage : Apparicio P. et J. Gelb (2022). *Méthodes quantitatives en sciences sociales : un grand bol d'R*. Institut national de la recherche scientifique. fabriqueREL. Licence CC BY-SA.



Sauf indications contraires, le contenu de ce manuel électronique est disponible en vertu des termes de la [Licence Creative Commons Attribution - Partage dans les mêmes conditions 4.0 International](#).

Vous êtes autorisé·e à :

- Partager – copier, distribuer et communiquer le matériel par tous moyens et sous tous formats.
- Adapter – remixer, transformer et créer à partir du matériel pour toute utilisation, y compris commerciale.

Selon les conditions suivantes :

- Paternité – Vous devez citer le nom des auteurs originaux.
- Mêmes conditions – Si vous remixez, transformez, ou créez à partir du matériel composant l'œuvre originale, vous devez diffuser l'œuvre modifiée avec la même licence.



Table des matières

Préface	25
Un manuel sous la forme d'une ressource éducative libre	25
Un manuel conçu comme un projet collaboratif	27
Comment lire ce livre ?	27
Comment utiliser les données du livre pour reproduire les exemples ?	28
Structure du livre	28
Pourquoi faut-il programmer en sciences sociales ?	30
Remerciements	31
Dédicace toute spéciale à Cargo et Ambrée	31
À propos des auteurs	33
I Découverte de R	35
1 Prise en main de R	37
1.1 Histoire et philosophie de R	37
1.2 Environnement de travail	40
1.2.1 Installation de R	40
1.2.2 Environnement RStudio	40
1.2.3 Installation et chargement un <i>package</i>	44
1.2.4 Aide disponible	45
1.3 Bases du langage R	46
1.3.1 <i>Hello World!</i>	46
1.3.2 Objets et expressions	46
1.3.3 Fonctions et arguments	48
1.3.4 Principaux types de données	49
1.3.5 Opérateurs	51
1.3.6 Structures de données	53
1.4 Manipulation de données	60
1.4.1 Chargement d'un <i>DataFrame</i> depuis un fichier	60
1.4.2 Manipulation d'un <i>DataFrame</i>	63
1.5 Code R bien structuré	81
1.6 Enregistrement des résultats	84
1.7 Session de travail	85
1.8 Conclusion et ressources pertinentes	86
1.9 Quiz de révision du chapitre	87

II Analyses univariées et graphiques dans R	89
2 Statistiques descriptives univariées	91
2.1 Notion et types de variable	91
2.1.1 Notion de variable	91
2.1.2 Types de variables	93
2.2 Types de données	95
2.2.1 Données secondaires <i>versus</i> données primaires	95
2.2.2 Données transversales <i>versus</i> données longitudinales	95
2.2.3 Données spatiales <i>versus</i> données aspatiales	96
2.2.4 Données individuelles <i>versus</i> données agrégées	96
2.3 Statistique descriptive et statistique inférentielle	98
2.3.1 Population, échantillon et inférence	98
2.3.2 Deux grandes familles de méthodes statistiques	98
2.4 Notion de distribution	99
2.4.1 Définition générale	99
2.4.2 Anatomie d'une distribution	101
2.4.3 Principales distributions	104
2.4.4 Conclusion sur les distributions	119
2.5 Statistiques descriptives sur des variables quantitatives	120
2.5.1 Paramètres de tendance centrale	120
2.5.2 Paramètres de position	120
2.5.3 Paramètres de dispersion	122
2.5.4 Paramètres de forme	126
2.5.5 Transformation des variables	141
2.5.6 Mise en œuvre dans R	144
2.6 Statistiques descriptives sur des variables qualitatives et semi-qualitatives	154
2.6.1 Fréquences	154
2.6.2 Mise en œuvre dans R	156
2.7 Statistiques descriptives pondérées : pour aller plus loin	157
2.8 Quiz de révision du chapitre	161
3 Magie des graphiques	165
3.1 Philosophie du ggplot2	166
3.1.1 Grammaire	166
3.1.2 Types de géométries	169
3.1.3 Habillage	170
3.1.4 Utilisation des thèmes	172
3.1.5 Composition d'une figure avec plusieurs graphiques	177
3.1.6 Couleur	179
3.2 Principaux graphiques	181
3.2.1 Histogramme	181
3.2.2 Graphique de densité	186
3.2.3 Nuage de points	188
3.2.4 Graphique en ligne	194
3.2.5 Boîte à moustaches	196
3.2.6 Graphique en violon	198
3.2.7 Graphique en barre	200
3.2.8 Graphique circulaire	203
3.3 Graphiques spéciaux	205

3.3.1	Graphique en radar	205
3.3.2	Diagramme d'accord	207
3.3.3	Nuage de mots	210
3.3.4	Carte proportionnelle	212
3.4	Cartes	213
3.5	Exportation des graphiques	217
3.6	Conclusion sur les graphiques	219
III	Analyses bivariées	221
4	Relation linéaire entre deux variables quantitatives	223
4.1	Bref retour sur le postulat de la relation linéaire	224
4.2	Covariance	226
4.2.1	Formulation	226
4.2.2	Interprétation	226
4.3	Corrélation	226
4.3.1	Formulation	226
4.3.2	Interprétation	228
4.3.3	Corrélations pour des variables anormalement distribuées (coefficient de Spearman, tau de Kendall)	229
4.3.4	Corrélations robustes (<i>Biweight midcorrelation, Percentage bend correlation</i> et la corrélation <i>pi</i> de Shepherd)	231
4.3.5	Significativité des coefficients de corrélation	233
4.3.6	Corrélation partielle	237
4.3.7	Mise en œuvre dans R	238
4.3.8	Comment rapporter des valeurs de corrélations ?	242
4.4	Régression linéaire simple	243
4.4.1	Principe de base de la régression linéaire simple	246
4.4.2	Formulation de la droite de régression des moindres carrés ordinaires	247
4.4.3	Mesure de la qualité d'ajustement du modèle	248
4.4.4	Mise en œuvre dans R	251
4.4.5	Comment rapporter une régression linéaire simple	252
4.5	Quiz de révision du chapitre	253
5	Relation entre deux variables qualitatives	257
5.1	Construction de tableau de contingence	257
5.2	Test du <i>khi-deux</i>	260
5.3	Mise en œuvre dans R	262
5.4	Interprétation d'un tableau de contingence	267
5.5	Quiz de révision du chapitre	270
6	Relation entre une variable qualitative et une variable quantitative	273
6.1	Relation entre une variable quantitative et une variable qualitative à deux modalités	273
6.1.1	Test <i>t</i> et ses différentes variantes	274
6.1.2	Test non paramétrique de Wilcoxon	291
6.2	Relation entre une variable quantitative et une variable qualitative à plus de deux modalités	293
6.2.1	Analyse de variance	293
6.2.2	Test non paramétrique de Kruskal-Wallis	299
6.2.3	Mise en œuvre dans R	299
6.2.4	Comment rapporter les résultats d'une ANOVA et du test de Kruskal-Wallis	308

6.3 Conclusion sur la troisième partie	309
6.4 Quiz de révision du chapitre	310
IV Modèles de régression	315
7 Régression linéaire multiple	317
7.1 Objectifs de la régression linéaire multiple et construction d'un modèle de régression	318
7.2 Principes de base de la régression linéaire multiple	320
7.2.1 Un peu d'équations...	320
7.2.2 Hypothèses de la régression linéaire multiple	321
7.3 Évaluation de la qualité d'ajustement du modèle	322
7.3.1 Mesures de la qualité d'un modèle	322
7.3.2 Comparaison des modèles incrémentiels	324
7.4 Différentes mesures pour les coefficients de régression	327
7.4.1 Coefficients de régression : évaluer l'effet des variables indépendantes	328
7.4.2 Coefficients de régression standardisés : repérer les variables les plus importantes du modèle	329
7.4.3 Significativité des coefficients de régression : valeurs de t et de p	331
7.4.4 Intervalle de confiance des coefficients	334
7.5 Introduction de variables explicatives particulières	337
7.5.1 Exploration des relations non linéaires	337
7.5.2 Variable indépendante qualitative dichotomique	343
7.5.3 Variable indépendante qualitative polytomique	344
7.5.4 Variables d'interaction	349
7.6 Diagnostics de la régression	352
7.6.1 Nombre d'observations	353
7.6.2 Normalité des résidus	353
7.6.3 Linéarité et homoscédasticité des résidus	354
7.6.4 Absence de multicolinéarité excessive	354
7.6.5 Absence d'observations aberrantes	357
7.7 Mise en œuvre dans R	358
7.7.1 Fonctions <code>lm</code> , <code>summary()</code> et <code>confint()</code>	358
7.7.2 Comparaison des modèles	361
7.7.3 Diagnostic sur un modèle	364
7.7.4 Graphiques pour les effets marginaux	375
7.8 Quiz de révision du chapitre	382
8 Régressions linéaires généralisées (GLM)	385
8.1 Qu'est qu'un modèle GLM?	385
8.1.1 Formulation d'un GLM	386
8.1.2 Autres distributions et rôle de la fonction de lien	387
8.1.3 Conditions d'application	389
8.1.4 Résidus et déviance	389
8.1.5 Vérification l'ajustement	391
8.1.6 Comparaison de deux modèles GLM	396
8.2 Modèles GLM pour des variables qualitatives	397
8.2.1 Modèle logistique binomial	397
8.2.2 Modèle probit binomial	415
8.2.3 Modèle logistique des cotes proportionnelles	416

8.2.4	Modèle logistique multinomial	431
8.2.5	Conclusion sur les modèles pour des variables qualitatives	445
8.3	Modèles GLM pour des variables de comptage	445
8.3.1	Modèle de Poisson	445
8.3.2	Modèle binomial négatif	458
8.3.3	Modèle de Poisson avec excès fixe de zéros	465
8.3.4	Modèle de Poisson avec excès ajusté de zéros	467
8.3.5	Conclusion sur les modèles destinés à des variables de comptage	475
8.4	Modèles GLM pour des variables continues	477
8.4.1	Modèle GLM gaussien	477
8.4.2	Modèle GLM avec une distribution de Student	484
8.4.3	Modèle GLM avec distribution Gamma	490
8.4.4	Modèle GLM avec une distribution bêta	505
8.5	Conclusion sur les modèles linéaires généralisés	518
8.6	Quiz de révision du chapitre	520
9	Régressions à effets mixtes (GLMM)	523
9.1	Introduction	523
9.1.1	Indépendance des observations et effets de groupes	523
9.1.2	Terminologie : effets fixes et effets aléatoires	525
9.2	Principes de base des GLMM	526
9.2.1	GLMM avec constantes aléatoires	526
9.2.2	GLMM avec pentes aléatoires	531
9.2.3	GLMM avec constantes et pentes aléatoires	533
9.3	Conditions d'application des GLMM	538
9.3.1	Vérification de la distribution des effets aléatoires	538
9.3.2	Homogénéité des variances au sein des groupes	540
9.4	Inférence dans les modèles GLMM	541
9.4.1	Inférence pour les effets fixes	542
9.4.2	Inférence pour les effets aléatoires, effet global	542
9.4.3	Inférence pour les effets aléatoires, des constantes et des pentes	543
9.5	Conclusion sur les GLMM	543
9.6	Mise en œuvre des GLMM dans R	543
9.6.1	Ajustement du modèle avec uniquement une constante aléatoire	544
9.6.2	Ajustement du modèle avec constantes et pentes aléatoires	552
9.7	Quiz de révision du chapitre	563
10	Régressions multiniveaux	565
10.1	Modèles multiniveaux : deux intérêts majeurs	566
10.1.1	Répartition de la variance entre les différents niveaux	566
10.1.2	Estimation des coefficients aux différents niveaux	566
10.2	Différents types de modèles multiniveaux	567
10.2.1	Description du jeu de données utilisé	567
10.2.2	Démarche classique pour les modèles multiniveaux	568
10.3	Conditions d'application des régressions multiniveaux	573
10.4	Mise en œuvre dans R	574
10.4.1	Le modèle vide	574
10.4.2	Modèle avec les variables indépendantes du niveau 1	576
10.4.3	Modèle avec les variables indépendantes aux niveaux 1 et 2	578
10.4.4	Modèle complet avec une interaction	580

10.4.5 Comparaison des quatre modèles	582
10.5 Quiz de révision du chapitre	584
11 Modèles généralisés additifs	587
11.1 Introduction	587
11.1.1 Non linéarité fonctionnelle	588
11.1.2 Non linéarité avec des polynomiales	590
11.1.3 Non linéarité par segments	591
11.1.4 Non linéarité avec des <i>splines</i>	592
11.2 <i>Spline</i> de régression et <i>spline</i> de lissage	595
11.3 Interprétation d'une <i>spline</i>	596
11.4 Multicolinéarité non linéaire	596
11.5 <i>Splines</i> avancées	596
11.5.1 <i>Splines</i> cycliques	596
11.5.2 Splines par groupe	598
11.5.3 <i>Splines</i> multivariées et <i>splines</i> d'interaction	598
11.6 Mise en œuvre dans R	599
11.7 GAMM	606
11.8 Quiz de révision du chapitre	615
V Analyses exploratoires multivariées	617
12 Méthodes factorielles	619
12.1 Aperçu des méthodes factorielles	620
12.1.1 Méthodes factorielles et types de données	620
12.1.2 Bref historique des méthodes factorielles	620
12.2 Analyses en composantes principales (ACP)	621
12.2.1 Recherche d'une simplification	621
12.2.2 Aides à l'interprétation	623
12.2.3 Mise en œuvre dans R	632
12.3 Analyse factorielle des correspondances (AFC)	649
12.3.1 Recherche d'une simplification basée sur la distance du khi-deux	652
12.3.2 Aides à l'interprétation	654
12.3.3 Mise en œuvre dans R	662
12.4 Analyse de correspondances multiples (ACM)	669
12.4.1 Aides à l'interprétation	671
12.4.2 Mise en œuvre dans R	680
12.5 Quiz de révision du chapitre	692
13 Méthodes de classification non supervisée	695
13.1 Méthodes de classification : un aperçu	696
13.2 Notions essentielles en classification	697
13.2.1 Distance	698
13.2.2 Inertie	704
13.3 Classification ascendante hiérarchique	706
13.3.1 Fonctionnement de l'algorithme	706
13.3.2 Choisir le bon nombre de groupes	708
13.3.3 Limites de la classification ascendante hiérarchique	711
13.3.4 Mise en œuvre dans R	712
13.4 Nuées dynamiques	720

13.4.1 <i>K-means</i>	720
13.4.2 K-médiannes	723
13.4.3 K-médoïds	723
13.4.4 Mise en oeuvre dans R	724
13.4.5 Extensions en logique floue : <i>c-means</i> , <i>c-medoids</i>	735
13.5 Conclusion sur la cinquième partie	747
13.6 Quiz de révision du chapitre	749
14 Annexes	751
14.1 Table des valeurs critiques de khi-deux	751
14.2 Table des valeurs critiques de Fisher	753
14.3 Table des valeurs critiques de <i>t</i>	757
Bibliographie	759

Liste des tableaux

1.1	Opérateurs mathématiques	51
1.2	Opérateurs relationnels	51
1.3	Opérateurs logiques	52
1.4	Premier DataFrame	58
1.5	Temps nécessaire pour lire les données en fonction du type de fichiers	63
1.6	Avantages et inconvénients du tidyverse	64
1.7	Ressources pertinentes pour en apprendre plus sur R	86
2.1	Revenus moyens et médians des ménages en dollars, municipalités de l'île de Montréal, 2015	121
2.2	Stastistiques descriptives de l'exposition au bruit des cyclistes par minute dans trois villes (dB(A), Laeq 1min)	122
2.3	Calcul des mesures de dispersion sur des données fictives	125
2.4	Illustration de la sensibilité des mesures de dispersion à l'unité de mesure et aux valeurs extrêmes	125
2.5	Résumé de la sensibilité de la moyenne et des mesures de dispersion	126
2.6	Différents tests d'hypothèse pour la normalité	132
2.7	Tests de normalité pour différentes distributions	132
2.8	Comparaison des LogLikelihood des trois distributions	137
2.9	Illustration des trois transformations	143
2.10	Différents types de fréquences sur une variable qualitative ou semi-qualitative	156
2.11	Différents types de fréquences sur une variable semi-qualitative	156
2.12	Calcul de la moyenne pondérée	158
2.13	Statistiques de l'aire de jeux la plus proche, par secteur de recensement, pondérées par la population de moins de 10 ans	159
3.1	Principales géométries proposées par ggplot2	170
4.1	Intervalles pour l'interprétation du coefficient de corrélation habituellement utilisés en sciences sociales	228
4.2	Comparaison de différentes corrélations pour les deux variables	233
4.3	Données fictives sur l'utilisation du vélo par municipalité	246
4.4	Valeurs observées, prédites et résidus	248
4.5	Calcul du coefficient de détermination	249
5.1	Autres mesures d'association entre deux variables qualitatives	261
5.2	Mesures d'association entre deux variables qualitatives	267
6.1	Conventions pour l'interprétation du d de Cohen	282
6.2	Données fictives et calcul des trois variances (cas 1)	296

6.3	Données fictives et calcul des trois variances (cas 2)	296
7.1	Statistiques descriptives pour les variables du modèle	322
7.2	Différentes mesures pour les coefficients	328
7.3	Calcul des coefficients standardisés	331
7.4	Modèle avec une variable indépendante sous forme logarithmique	342
7.5	Modèle avec une variable dichotomique	344
7.6	Transformation d'une variable qualitative en variables muettes pour chaque modalité	345
7.7	Modèle avec une variable polytomique (ville de Montréal en catégorie de référence)	346
7.8	Modèle avec une variable polytomique (Senneville en catégorie de référence)	347
7.9	Modèle avec une variable polytomique (Montréal-Est en catégorie de référence)	348
7.10	Modèle avec la distance au centre-ville (km)	349
7.11	Modèle avec une variable d'interaction entre deux VI continues	350
7.12	Modèle avec les variables d'interaction entre une VI continue et une VI dichotomique	352
8.1	Principaux pseudo- R^2	392
8.2	Exemple de matrice de confusion	394
8.3	Exemple de matrice de confusion multinomiale	395
8.4	Interprétation des valeurs du coefficient de Kappa	396
8.5	Indicateurs de qualité de prédiction	396
8.6	Carte d'identité du modèle logistique binomial	398
8.7	Variables indépendantes utilisées pour prédire le mode de transport le plus utilisé	400
8.8	Matrice de confusion pour le modèle binomial	409
8.9	Matrice de confusion pour le modèle binomial	409
8.10	Résultats du modèle binomial	411
8.11	Carte d'identité du modèle probit binomial	415
8.12	Carte d'identité du modèle logistique des cotes proportionnelles	416
8.13	Variables indépendantes utilisées pour prédire la catégorie de prix de logements Airbnb	419
8.14	Coefficients du modèle logistique des cotes proportionnelles	429
8.15	Carte d'identité du modèle logistique multinomial	432
8.16	Variables indépendantes utilisées dans le modèle logistique multinomial	434
8.17	Coefficients du modèle multinomial A versus B	443
8.18	Coefficients du modèle multinomial A versus C	444
8.19	Coefficients du modèle multinomial A versus D	444
8.20	Carte d'identité du modèle de Poisson	445
8.21	Variables indépendantes utilisées dans le modèle de Poisson	447
8.22	Résultats du modèle de quasi-Poisson	457
8.23	Carte d'identité du modèle binomial négatif	458
8.24	Résultats du modèle binomial négatif	465
8.25	Carte d'identité du modèle de Poisson avec excès fixe de zéros	465
8.26	Carte d'identité du modèle de Poisson avec excès ajusté de zéros	467
8.27	Résultats de la partie Poisson du modèle de Poisson avec excès de zéros ajusté	475
8.28	Résultats de la partie logistique du modèle de Poisson avec excès de zéros ajusté	475
8.29	Carte d'identité du modèle gaussien	478
8.30	Résultats du modèle gaussien	484
8.31	Carte d'identité du modèle de Student	485
8.32	Résultats du modèle Student	490
8.33	Carte d'identité du modèle Gamma	491
8.34	Variables indépendantes utilisées dans le modèle Gamma	492
8.35	Résultats pour le modèle GLM Gamma	504

8.36 Carte d'identité du modèle bêta	507
8.37 Variables indépendantes utilisées dans le modèle bêta	508
8.38 Résultats pour le modèle GLM bêta	517
 9.1 Comparaison des trois modèles à effets aléatoires	536
10.1 Statistiques descriptives pour les variables des modèles multiniveaux	567
10.2 Résultats du modèle vide (modèle 1)	569
10.3 Résultats du modèle avec les variables indépendantes au niveau 1 (modèle 2)	570
10.4 Résultats du modèle avec les variables indépendantes centrées au niveau 1 (modèle 2) . .	571
10.5 Résultats du modèle avec les variables indépendantes aux niveaux 1 et 2 (modèle 3) . .	572
10.6 Résultats du modèle avec une variable d'interaction entre les deux niveaux 1 et 2 (modèle 4)	573
 11.1 Exemples de splines avancées	597
12.1 Trois principales méthodes factorielles	620
12.2 Données fictives	622
12.3 Statistiques descriptives pour le jeu de données utilisé pour l'ACP	624
12.4 Résultats de l'ACP pour les valeurs propres	626
12.5 Résultats de l'ACP pour les variables	627
12.6 Matrice de corrélation de Pearson entre les variables utilisées pour l'ACP	628
12.7 Exemple de tableau de contingence pour l'AFC	652
12.8 Exemple d'un tableau de contingence transformé (pourcentage en ligne) pour l'ACP . .	652
12.9 Données brutes du tableau de contingence	653
12.10 Profils des lignes et des colonnes	653
12.11 Données relatives du tableau de contingence (fij)	653
12.13 Jeu de données utilisé pour l'analyse factorielle des correspondances	654
12.12 Distances du khi-deux entre les modalités I et les modalités J	654
12.14 Résultats du test du khi-deux sur le tableau de contingence	655
12.15 Résultats de l'AFC pour les valeurs propres	656
12.16 Vérification des deux propriétés des coordonnées factorielles pour les variables .	658
12.17 Résultats de l'AFC pour les variables	659
12.18 Exemple de variables qualitatives issues d'une enquête	670
12.19 Tableau condensé (données brutes)	670
12.20 Tableau disjonctif complet	670
12.21 Variables qualitatives extraites du sondage sur l'agriculture urbaine de la Ville de Montréal	672
12.22 Résultats de l'ACM pour les valeurs propres	673
12.23 Résultats de l'ACM pour les modalités des variables	675
12.24 Résultats de l'ACM pour les modalités des variables supplémentaires	678
 13.1 Distance du khi-deux entre trois histogrammes	702
13.2 Exemple de données pour la distance de Hamming	703
13.3 Distance de Hamming entre les maisons	703
13.4 Équipements recensés dans les différents parcs de Montréal	712
13.5 Caractéristiques des groupes obtenus lors de la CAH	716
13.6 Caractéristiques des groupes obtenus lors de la CAH (distance euclidienne au carré)	720
13.7 Statistiques descriptives du jeu de données LyonIris	724
13.8 Descriptions des quatre groupes obtenus	732
13.9 Description des groupes avec la méthode c-means	741
 14.1 Distribution des valeurs critiques du khi-deux	752

14.2 Distribution des valeurs critiques de F avec $p = 0,05$	754
14.3 Distribution des valeurs critiques de F avec $p = 0,05$ (suite)	755
14.4 Distribution des valeurs critiques de F avec $p = 0,05$ (suite)	756
14.5 Distribution des valeurs critiques de t	758

Liste des figures

1	Licence Creative Commons du livre	26
2	Téléchargement de l'intégralité du livre	29
3	Cargo et Ambrée, chiots de la Fondation Mira	32
4	Philippe Apparicio et Jérémy Gelb lors d'une collecte de données à vélo à Delhi	34
1.1	Lieu de pèlerinage de R	38
1.2	Nombre d'articles trouvés sur Google Scholar (source : Robert A. Muenchen)	38
1.3	Métaphore sur les langages et programmes d'analyse statistique	40
1.4	Icône des encadrés dédiés aux packages	40
1.5	Console de base de R	41
1.6	Environnement de base de RStudio	41
1.7	RStudio avec le style pastel on dark	42
1.8	Fenêtres de RStudio	43
1.9	Exécuter du code dans RStudio	43
1.10	Description des packages	45
1.11	Arguments de la fonction round	48
1.12	Du vecteur à la matrice	56
1.13	Un array avec trois dimensions	56
1.14	De la donnée au DataFrame	58
1.15	Répartition temporelle des accidents à vélo	73
1.16	Fusion de DataFrames	78
1.17	Jointure de DataFrames	79
1.18	Navigation dans des sections de codes avec RStudio	83
1.19	Structure de dossier recommandée pour un projet avec R	84
1.20	Bouton enregistrer la session	85
1.21	Bouton charger un fichier RDA	85
2.1	Types de variables	93
2.2	Exemple d'agrégation de données individuelles	97
2.3	Distribution théorique d'un lancer de dé	100
2.4	Distribution empirique d'un lancer de dé (n=10)	100
2.5	Distribution empirique d'un lancer de dé	101
2.6	Distributions uniformes continues	102
2.7	Dix-huit distributions essentielles, figure inspirée de Sean (2018)	104
2.8	Distribution binomiale	106
2.9	Distribution géométrique	107
2.10	Distribution binomiale négative	108
2.11	Distribution de Poisson	109
2.12	Distribution de Poisson avec excès de zéros	110

2.13	Distribution gaussienne	110
2.14	Distribution gaussienne asymétrique	111
2.15	Distribution log-gaussienne	112
2.16	Distribution de Student	113
2.17	Distribution de Cauchy	114
2.18	Distribution du khi-deux	114
2.19	Distribution exponentielle	115
2.20	Distribution Gamma	116
2.21	Distribution bêta	117
2.22	Distribution de Weibull	118
2.23	Distribution de Pareto	118
2.24	Exemples de cartographie avec une discréétisation selon les quantiles	121
2.25	Graphique en violon, boîte à moustaches et intervalle interquartile	123
2.26	Formes d'une distribution et coefficients d'asymétrie et d'aplatissement	126
2.27	Asymétrie d'une distribution	128
2.28	Applatissement d'une distribution	129
2.29	Histogrammes et courbe normale	130
2.30	Diagrammes quantile-quantile	131
2.31	Distribution empirique des temps de retard des bus à Toronto en janvier 2019	136
2.32	Comparaison des distributions ajustées aux données de retard des bus	137
2.33	Distribution empirique du nombre d'accidents par intersection impliquant un ou une cycliste à Montréal en 2017 dans les quartiers centraux	139
2.34	Ajustement des distributions de Poisson et Poisson avec excès de zéros	140
2.35	Histogramme avec courbe normale	148
2.36	Histogramme des transformations	155
2.37	Différents graphiques pour représenter les fréquences absolues et relatives	157
2.38	Accessibilité aux aires de jeux par secteur de recensement, Communauté métropolitaine de Montréal, 2016	159
3.1	Trois composantes d'un graphique, adapté de @wickham2010layered	167
3.2	Base d'un graphique	168
3.3	Ajout des dimensions au graphique	168
3.4	Autre spécification des arguments mapping et data	169
3.5	Ajout de titres	169
3.6	Ajout d'annotations textuelles	171
3.7	Ajout d'annotations géométriques	172
3.8	Gestion de l'ordre des annotations	173
3.9	Thème classique	173
3.10	Thème gris	174
3.11	Thème noir et blanc	174
3.12	Thème minimal	175
3.13	Thème tufte	176
3.14	Thème economist	176
3.15	Thème solarized	177
3.16	Graphique à facettes	178
3.17	Graphique à facettes en une ligne	178
3.18	Figure avec plusieurs graphiques avec ggarrange	180
3.19	Couleurs de base	181
3.20	Palette de couleurs de ColorBrewer	182
3.21	Histogrammes	183

3.22	Empiler les données d'un DataFrame	183
3.23	Histogrammes à facettes	184
3.24	Histogrammes de densité	185
3.25	Histogramme et courbe normale	186
3.26	Histogramme coloré	187
3.27	Graphiques de densité à facette	187
3.28	Graphiques de densité superposés	188
3.29	Nuage de points simple	189
3.30	Nuage de points simple avec transparence	190
3.31	Nuage de points simple	191
3.32	Densité en deux dimensions par hexagones	191
3.33	Densité lissée en deux dimensions	192
3.34	Nuage de points avec droite de régression quadratique	193
3.35	Graphique en ligne	194
3.36	Graphique en ligne avec barres d'erreur	195
3.37	Graphique en ligne avec marge d'erreur	196
3.38	Boîtes à moustaches	197
3.39	Boîtes à moustaches améliorées	197
3.40	Boîtes à moustaches avec observations	198
3.41	Graphiques en violon	199
3.42	Graphiques en violon et boîtes à moustaches	199
3.43	Grands secteurs de Québec	200
3.44	Graphiques en barre simples	202
3.45	Graphique en barre empilée	203
3.46	Graphique en tarte	204
3.47	Graphique en anneau	205
3.48	Graphique en anneau	207
3.49	Diagramme d'accord	209
3.50	Nuage de mots pour le SAD de Montréal	212
3.51	Nuage de mots pour le SAD de Québec	213
3.52	Treemap	214
3.53	Carte thématique avec ggplot2	216
3.54	Carte thématique avec tmap	217
3.55	Combiner des cartes avec tmap	218
3.56	Exporter un graphique dans RStudio	218
4.1	Relations linéaires et curvilinéaires entre deux variables continues	225
4.2	Exemples de relations curvilinéaires	225
4.3	Covariance et unités de mesure	227
4.4	Relations entre deux variables continues et coefficients de corrélation de Pearson	228
4.5	Coefficient de corrélation et proportion de la variance expliquée	229
4.6	Illustration de l'effet des valeurs extrêmes sur le coefficient de Pearson	230
4.7	Comparaison des coefficients de Pearson, Spearman et Kendall sur deux variables anormalement distribuées	231
4.8	Histogramme pour les valeurs de corrélation issues du Bootstrap	236
4.9	Matrice de corrélation avec corrplot (chiffres)	241
4.10	Matrice de corrélation avec corrplot (chiffres et ellipses)	242
4.11	Corrélations significatives obtenues aléatoirement	246
4.12	Relation linéaire entre l'utilisation du vélo et la distance au centre-ville	247
4.13	Droite de régression, valeurs observées, prédites et résidus	248

5.1	Figure avec la fonction mosaic du package vcd	268
5.2	Taille des projets d'habitation à loyer modique selon la période de construction	269
6.1	Boîtes à moustaches sur des échantillons fictifs non appariés	277
6.2	Boîtes à moustaches sur des échantillons fictifs appariés	280
6.3	Pourcentage de locataires par secteur de recensement, région métropolitaine de recensement de Montréal, 2016	283
6.4	Graphiques de densité et en violon	300
6.5	QQ Plot pour les groupes	301
6.6	Graphique de densité et en violon	304
6.7	QQ Plot pour les groupes	305
6.8	Les principales méthodes bivariées	309
7.1	Exemple de cadre conceptuel	319
7.2	Relations linéaire et curvilinéaire	341
7.3	Effet marginal de la densité de population	343
7.4	Effet marginal d'une variable dichotomique	345
7.5	Effet marginal d'une variable polytomique	348
7.6	Effet marginal de l'interaction entre deux variables continues	350
7.7	Effets marginaux de deux variables continues en cas d'absence d'interaction	351
7.8	Graphique de l'effet marginal de l'interaction entre une variable quantitative et qualitative	352
7.9	Vérification de la normalité des résidus	354
7.10	Distribution des résidus en fonction des valeurs prédites	355
7.11	Repérage graphique les valeurs influentes du modèle	358
7.12	Différentes parties obtenues avec la fonction summary(Modèle)	360
7.13	Diagnostic : la normalité des résidus	366
7.14	Distribution des résidus en fonction des valeurs prédites	367
7.15	Repérage graphique des valeurs influentes du modèle	370
7.16	Normalité des résidus avant et après la suppression des valeurs influentes	372
7.17	Amélioration de l'homoscédasticité des résidus	373
7.18	Effets marginaux pour des variables continues	376
7.19	Effet marginal d'une variable avec un fonction polynomiale d'ordre 2	377
7.20	Effet du logarithme de la densité	377
7.21	Effet marginal d'une variable dichotomique	378
7.22	Effet marginal d'une variable polytomique	379
7.23	Effet marginal de l'interaction entre deux variables continues	381
7.24	Graphique de l'effet marginal de l'interaction entre une variable quantitative et qualitative	382
8.1	Exemple de données issues d'une distribution de Bernoulli	388
8.2	Ajustement d'une droite de régression aux données issues d'une distribution de Bernoulli	388
8.3	Utilisation de la fonction de lien logistique	389
8.4	Distances de Cook pour le modèle binomial avec toutes les observations	402
8.5	Distances de Cook pour le modèle binomial sans les valeurs aberrantes	403
8.6	Distribution des résidus simulés pour le modèle binomial	404
8.7	Diagnostic des résidus simulés par le package DHARMA	405
8.8	Point d'équilibre entre sensibilité et spécificité	408
8.9	Rapports de cote pour les différents pays de l'UE	411
8.10	Rapports de cote pour les différents lieux de résidence	413
8.11	Effet de l'âge sur la probabilité d'utiliser le vélo comme moyen de déplacement pour son trajet le plus fréquent	415
8.12	Distribution des prix des logements Airbnb	418

8.13 Distances de Cook pour le modèle logistique des cotes proportionnelles	421
8.14 Diagnostic général des résidus simulés du modèle des cotes proportionnelles	422
8.15 Diagnostic des variables indépendantes et des résidus simulés du modèle des cotes proportionnelles	423
8.16 Diagnostic des variables indépendantes et des résidus simulés du modèle des cotes proportionnelles (après correction)	425
8.17 Prédiction de la probabilité d'appartenance aux trois catégories de prix en fonction de la densité de végétation	431
8.18 Distances de Cook pour le modèle logistique multinomial	436
8.19 Diagnostic général des résidus simulés pour le modèle multinomial	438
8.20 Diagnostic des variables indépendantes et des résidus simulés pour le modèle multinomial .	439
8.21 Diagnostic général des résidus simulés pour le modèle multinomial (version 3)	440
8.22 Diagnostic général des résidus simulés pour le modèle multinomial (version 4)	441
8.23 Distribution originale du nombre d'accidents par intersection	448
8.24 Distances de Cook pour le modèle de Poisson	449
8.25 Distances de Cook pour le modèle de Poisson après avoir retiré les valeurs aberrantes .	450
8.26 Représentation de la sur-dispersion des données dans le modèle de Poisson	452
8.27 Comparaison de la distribution originale et des simulations pour le modèle de quasi-Poisson	454
8.28 Comparaison de la distribution originale et des simulations pour le modèle de Poisson .	454
8.29 Analyse globale des résidus simulés pour le modèle de quasi-Poisson	455
8.30 Comparaison des résidus simulés et de chaque variable indépendante	456
8.31 Distances de Cook pour le modèle binomial négatif	459
8.32 Distances de Cook pour le modèle binomial négatif (après avoir retiré quatre observations fortement influentes)	461
8.33 Diagnostic général des résidus simulés pour le modèle binomial négatif	461
8.34 Comparaison de la distribution originale et des simulations pour le modèle binomial négatif	463
8.35 Représentation de la sur-dispersion des données dans le modèle de Poisson	464
8.36 Diagnostic général des résidus simulés du modèle de Poisson avec excès de zéros ajusté .	469
8.37 Diagnostic général des résidus simulés du modèle de Poisson avec excès de zéros ajusté (sans valeurs aberrantes)	470
8.38 Comparaison de la distribution originale et des simulations pour le modèle de Poisson avec excès de zéros ajusté	472
8.39 Processus de sélection d'un modèle pour une variable de comptage	476
8.40 Distances de Cook pour le modèle gaussien	479
8.41 Distances de Cook pour le modèle gaussien après suppression des observations influentes	480
8.42 Diagnostic général des résidus simulés pour le modèle gaussien	481
8.43 Comparaison de la distribution originale de la variable et des simulations issues du modèle	482
8.44 Comparaison de la distribution originale de la variable et des simulations issues du modèle	483
8.45 Effet du paramètre nu sur une distribution de Student	485
8.46 Distances de Cook pour un modèle GLM avec une distribution de Student	486
8.47 Distances de Cook pour un modèle GLM avec une distribution de Student après suppression des valeurs fortement influentes	487
8.48 Diagnostic général des résidus simulés pour le GLM avec distribution de Student	488
8.49 Distribution des résidus simulés du modèle GLM avec distribution de Student	488
8.50 Simulations issues des modèles gaussien et de Student, comparées aux données originales	489
8.51 Distribution des temps de trajet diurne à Montréal	493
8.52 Distances de Cook pour le modèle Gamma	494
8.53 Distances de Cook pour le modèle Gamma (sans les observations fortement influentes)	495
8.54 Comparaison de la distribution originale et de simulations issues du modèle Gamma .	496
8.55 Distribution des résidus simulés du modèle Gamma	497

8.56 Diagnostic général des résidus simulés du modèle Gamma	497
8.57 Diagnostic général des résidus simulés du modèle Gamma (après suppression d'environ 620 valeurs aberrantes)	499
8.58 Comparaison de la variance attendue par le modèle et la variance observée dans les données pour le modèle Gamma	500
8.59 Effet de l'arrondissement de départ sur les temps de trajet à Montréal	502
8.60 Effet de l'heure de départ sur les temps de trajet à Montréal	503
8.61 Effet de la distance à vol d'oiseau sur les temps de trajet à Montréal	505
8.62 Distances de Cook pour le modèle GLM bêta	509
8.63 Distances de Cook pour le modèle GLM bêta (suppression de deux observations influentes)	510
8.64 Distances de Cook pour le modèle GLM bêta (suppression de trois observations influentes)	511
8.65 Diagnostic général des résidus simulés du modèle bêta	512
8.66 Relation entre chaque variable indépendante et les résidus simulés du modèle bêta	513
8.67 Relation entre la variable Arrondissement et les résidus simulés du modèle bêta	514
8.68 Comparaison entre la distribution originale et les simulations issues du modèle	515
8.69 Rapports de cote pour les arrondissements dans le modèle bêta	517
8.70 Effets marginaux des variables pourcentage de personnes à faible revenu et densité de végétation	519
8.71 Résumé graphique des principaux GLM abordés	520
 9.1 Structure hiérarchique entre élèves, classes et écoles	524
9.2 Influence du temps de travail sur la performance scolaire d'élèves	527
9.3 Influence du temps de travail sur la performance scolaire d'élèves en tenant compte de l'effet de leur classe (effet fixe)	528
9.4 Influence du temps de travail sur la performance scolaire d'élèves en tenant compte de l'effet de leur classe (effet aléatoire)	529
9.5 Comparaison des effets des classes pour le modèle à effets fixes versus le modèle à effets aléatoires	530
9.6 Influence du temps de travail sur la performance scolaire d'élèves en interaction avec la classe (effet fixe)	531
9.7 Influence du temps de travail sur la réussite scolaire d'élèves en interaction avec la classe (effet aléatoire)	532
9.8 Influence du temps de travail sur la réussite scolaire d'élèves en interaction avec la classe (effet aléatoire)	533
9.9 Influence du temps de travail sur la performance scolaire d'élèves en tenant compte de l'effet de leur classe et de l'effet de la classe sur l'efficacité du temps de travail (effet fixe) .	534
9.10 Influence du temps de travail sur la performance scolaire d'élèves en tenant compte de l'effet de leur classe et de l'effet de la classe sur l'efficacité du temps de travail (effet aléatoire) .	535
9.11 Comparaison des effets fixes et aléatoires pour le modèle intégrant l'effet des classes et l'interaction entre les classes et le temps de travail	536
9.12 Différentes structures de données hiérarchiques (imbriquée versus croisée)	537
9.13 Distribution normale univariée des constantes aléatoires	539
9.14 Multiples distributions normales univariées des constantes et pentes aléatoires	540
9.15 Distribution normale bivariée des constantes et des pentes aléatoires	540
9.16 Homogénéité de la variance pour les différents groupes d'un modèle GLMM gaussien .	541
9.17 Distributions obtenues par bootstrap de la variance de l'effet aléatoire, de l'ICC et du R carré conditionnel	549
9.18 Constantes aléatoires estimées par Pays (IC par simulations)	550
9.19 Constantes aléatoires estimées par Pays (IC par bootstrap)	553
9.20 Incertitude autour des paramètres de variance obtenue par bootstrap	557

9.21 Constantes aléatoires estimées par pays (intervalles de confiance obtenus par simulations)	558
9.22 Distribution normale univariée des constantes et des pentes aléatoires	560
9.23 Distribution normale bivariée des constantes et des pentes aléatoires	561
 11.1 Patron journalier du dioxyde d'azote et de l'ozone à Paris	588
11.2 Relation non linéaire exponentielle	589
11.3 Autres relations non linéaires	589
11.4 Relation non linéaire plus complexe	590
11.5 Visualisation de plusieurs polynomiales	590
11.6 Sur et sous-ajustement d'une polynomiale	591
11.7 Régression par segment	592
11.8 Bases de la spline triangulaire	593
11.9 Spline triangulaire multipliée par ces coefficients	593
11.10Spline triangulaire	594
11.11Comparaison de différentes bases	594
11.12Pénalisation des splines	595
11.13Spline cyclique pour modéliser la concentration de dioxyde d'azote	597
11.14Spline cyclique variant par groupe	598
11.15Spline d'interaction bivariée	599
11.16Comparaison d'une spline et d'une polynomiale	602
11.17Comparaison de deux splines spatiales	606
11.18Constantes aléatoires pour les arrondissements	608
11.19Constantes aléatoires pour les arrondissements avec intervalle de confiance	609
11.20Prédictions pour les différents arrondissements pour une AD fictive moyenne	610
11.21Pentes et constantes aléatoires pour les arrondissements	612
11.22Relation entre les effets aléatoires des arrondissements et la variable population à faible revenu	614
 12.1 Principe de base des analyses factorielles	620
12.2 Tableau pour une ACP	621
12.3 Corrélation, allongement du nuage de points et axes factoriels	623
12.4 Cartographie des dix variables utilisées pour l'ACP	624
12.5 Graphiques personnalisés pour les valeurs propres pour l'ACP	627
12.6 Coordonnées factorielles des variables	630
12.7 Premier plan factoriel de l'ACP pour les variables	630
12.8 Premier plan factoriel pour les individus	631
12.10Variables et individus supplémentaires pour l'ACP	631
12.9 Cartographie des coordonnées factorielles des individus	632
12.11Graphiques pour les valeurs propres de l'ACP avec factoextra	641
12.12Contributions des variables à la première composante avec factoextra	642
12.13Contributions des variables à la deuxième composante avec factoextra	642
12.14Qualité des variables sur les trois premières composantes avec factoextra	643
12.15Premier plan factoriel de l'ACP pour les variables avec factoextra	644
12.16Premier plan factoriel de l'ACP pour les individus avec factoextra	644
12.17Graphiques personnalisés pour les valeurs propres	646
12.18Histogrammes personnalisés avec les coordonnées factorielles pour les variables	648
12.19Graphiques personnalisés avec les contributions des variables	649
12.20Graphiques personnalisés avec les cosinus carrés des variables	650
12.21Graphique personnalisé avec la qualité des variables sur les axes retenus de l'ACP	651
12.22Cartographie des coordonnées factorielles des individus	651

12.23 Tableau de contingence pour une AFC	652
12.24 Cartographie des modalités de la variable mode de transport utilisée pour l'AFC	655
12.25 Histogramme des valeurs propres de l'AFC	657
12.26 Premier plan factoriel de l'AFC pour les variables	660
12.27 Cartographie de coordonnées factorielles des individus pour l'AFC	661
12.28 Ajout de modalités supplémentaires sur le premier plan factoriel de l'AFC	661
12.29 Graphiques pour les valeurs propres de l'AFC avec factoextra	665
12.30 Contributions des variables avec factoextra	666
12.31 Premier plan factoriel de l'AFC pour les variables et les individus avec factoextra	667
12.32 Ajout de modalités supplémentaires sur le premier plan factoriel l'AFC avec factoextra	668
12.33 Cartographie de coordonnées factorielles des individus pour l'AFC	669
12.34 Tableau pour une ACM	669
12.35 Graphiques pour les valeurs propres pour l'ACM	673
12.36 Graphiques pour les résultats des modalités de l'axe 1 de l'ACM	675
12.37 Graphiques pour les résultats des modalités de l'axe 2 de l'ACM	676
12.38 Premier plan factoriel de l'ACM pour les modalités	677
12.39 Graphiques pour les résultats des modalités de l'axe 3 de l'ACM	678
12.40 Premier plan factoriel de l'ACM avec toutes les modalités incluant celles supplémentaires	679
12.41 Trajectoires des variables ordinaires sur le premier plan factoriel de l'ACM	679
12.42 Premier plan factoriel de l'ACM pour les individus	680
12.43 Premier plan factoriel de l'ACM pour les individus avec coloration d'une variable	681
12.44 Graphique pour les valeurs propres de l'ACM avec factoextra	685
12.45 Graphiques pour les valeurs propres de l'ACM avec factoextra	686
12.46 Exemple de graphiques pour les résultats des modalités	687
12.47 Premier plan factoriel de l'ACM pour les modalités	688
12.48 Premier plan factoriel de l'ACM pour les modalités supplémentaires	689
12.49 Trajectoires des variables ordinaires sur le premier plan factoriel de l'ACM	689
12.50 Premier plan factoriel de l'ACM pour les individus avec factoextra	690
12.51 Premier plan factoriel de l'ACM pour les individus avec coloration d'une variable avec factoextra	691
 13.1 Principe de base des méthodes de classification non supervisée	696
13.2 Classifications stricte et floue	697
13.3 Synthèse des principales méthodes de classification (Gelb et Apparicio 2021)	697
13.4 Situation de base pour le calcul de distance	698
13.5 Représentation de la distance euclidienne	699
13.6 Effets de différentes transformations sur la distribution d'une variable	700
13.7 Représentation de la distance de Manhattan	701
13.8 Trois histogrammes pour illustrer le calcul de la distance du khi-deux	701
13.9 Représentation de l'inertie du jeu de données IRIS	705
13.10 Représentation de l'inertie par groupe pour le jeu de données IRIS	706
13.11 Du tableau de données à la matrice de distance	707
13.12 Principe de fonctionnement de la classification ascendante hiérarchique (auteur : David Sheehan)	708
13.13 Méthode du coude	709
13.14 Méthode de l'indice de silhouette	710
13.15 Méthode GAP	711
13.16 Valeur de l'indice de silhouette pour différents nombres de groupes	714
13.17 Valeur de l'indice de silhouette pour différents nombres de groupes (distance euclidienne au carré)	718

13.18 Classifications stricte et floue	723
13.19 Inertie expliquée pour différents nombres de groupes pour le k-means	726
13.20 Indice de silhouette pour différents nombres de groupes pour le k-means	727
13.21 Méthode GAP pour différents nombres de groupes pour le k-means	728
13.22 Cartographie des groupes obtenus avec la méthode du k-means	729
13.23 Graphiques en radar pour les groupes issus du k-means	730
13.24 Graphiques en violon pour les groupes issus du k-means	731
13.25 Comparaison géographique des résultats obtenus pour le k-means, le k-medians et le k-medoids	734
13.26 Sélection des paramètres k et m pour l'algorithme c-means	737
13.27 Cartographie des probabilités d'appartenir aux quatre groupes identifiés par l'algorithme c-means	739
13.28 Graphique en radar pour les résultats du c-means	740
13.29 Indices de Jacard obtenus sur 1000 réplications du k-means	745
13.30 Distributions des valeurs des centres du groupe 4 sur 1000 itérations	746
13.31 Distributions des valeurs des centres du groupe 2 sur 1000 itérations	747
13.32 Complémentarité entre les méthodes factorielles et les méthodes de classification non supervisée	748

Préface

Ce livre vise à décrire une panoplie de méthodes quantitatives utilisées en sciences sociales avec le logiciel ouvert R. Il a d'ailleurs été écrit intégralement dans R avec rmarkdown¹. Le contenu est pensé pour être accessible à tous et toutes, même à ceux et celles n'ayant presque aucune base en statistique ou en programmation. Les personnes plus expérimentées y découvriront des sections sur des méthodes plus avancées comme les modèles à effets mixtes, les modèles multiniveaux, les modèles généralisés additifs ainsi que les méthodes factorielles et de classification. Ceux et celles souhaitant migrer progressivement d'un autre logiciel statistique vers R trouveront dans cet ouvrage les éléments pour une transition en douceur. La philosophie de ce livre est de donner toutes les clefs de compréhension et de mise en œuvre des méthodes abordées dans R. La présentation des méthodes est basée sur une approche compréhensive et intuitive plutôt que mathématique, sans pour autant que la rigueur statistique ne soit négligée. Servez-vous votre boisson chaude ou froide favorite et installez-vous dans votre meilleur fauteuil. Bonne lecture!

Un manuel sous la forme d'une ressource éducative libre

Pourquoi un manuel de statistique en sciences sociales sous licence libre ? Les logiciels libres sont aujourd'hui très répandus. Comparativement aux logiciels propriétaires, l'accès au code source permet à quiconque de l'utiliser, de le modifier, de le dupliquer et de le partager. Le logiciel R, dans lequel sont mises en œuvre les méthodes quantitatives décrites dans ce livre, est d'ailleurs à la fois un langage de programmation et un logiciel libre (sous la licence publique générale GNU GPL2²). Par analogie aux logiciels libres, il existe aussi des **ressources éducatives libres (REL)** « dont la licence accorde les permissions désignées par les 5R (**Retenir – Réutiliser – Réviser – Remixer – Redistribuer**) et donc permet nécessairement la modification» (*fabriqueREL*³). La licence de ce livre, CC BY-SA (figure 1), permet donc de :

- **Retenir**, c'est-à-dire télécharger et d'imprimer gratuitement le livre. Notez qu'il aurait été plutôt surprenant d'écrire un livre payant sur un logiciel libre et donc gratuit. Aussi, nous aurions été très embarrassés que des personnes étudiantes avec des ressources financières limitées doivent payer pour avoir accès au livre, sans pour autant savoir préalablement si le contenu est réellement adapté à leurs besoins.
- **Réutiliser**, c'est-à-dire utiliser la totalité ou une section du livre sans limitation et sans compensation financière. Cela permet ainsi à d'autres personnes enseignantes de l'utiliser dans le cadre d'activités pédagogiques.
- **Réviser**, c'est-à-dire modifier, adapter et traduire le contenu en fonction d'un besoin pédagogique précis puisqu'aucun manuel n'est parfait, tant s'en faut! Rappelons que le livre a d'ailleurs été écrit

¹<https://rmarkdown.rstudio.com/>

²https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU

³<https://fabriquerel.org/rel/>

intégralement dans R avec rmarkdown⁴. Quiconque peut ainsi télécharger gratuitement le code source du livre sur github⁵ et le modifier à sa guise (voir l'encadré intitulé *Suggestions d'adaptation du manuel*).

- **Remixer**, c'est-à-dire « de combiner la ressource avec d'autres ressources dont la licence le permet aussi pour créer une nouvelle ressource intégrée » (*fabriqueREL*⁶).
- **Redistribuer**, c'est-à-dire distribuer en totalité ou partiellement le manuel ou une version révisée sur d'autres canaux que le site Web du livre (par exemple, sur le site Moodle de votre université ou en faire une version imprimée).



Illustration adaptée de *Les licences Creative Commons*, par la fabriqueREL sous licence CC BY.

FIG. 1 : Licence Creative Commons du livre

La licence de ce livre, CC BY-SA (figure 1), oblige donc à :

- Attribuer la paternité de l'auteur dans vos versions dérivées, ainsi qu'une mention concernant les grandes modifications apportées, en utilisant la formulation suivante : Apparicio, Philippe et Jérémie Gelb. 2022. *Méthodes quantitatives en sciences sociales : un grand bol d'R*. Institut national de la recherche scientifique. CC BY-SA (4.0).
- Utiliser la même licence ou une licence similaire à toutes versions dérivées.



Suggestions d'adaptation du manuel

Notez que pour chaque méthode statistique abordée dans le livre sont disponibles une description détaillée et une mise en œuvre dans R. Par conséquent, plusieurs adaptations du manuel sont possibles :

- Conserver uniquement les chapitres sur les méthodes statistiques ciblées dans votre cours.
- En faire une version imprimée et la distribuer aux personnes étudiantes.
- Modifier la description d'une ou plusieurs méthodes en effectuant les mises à jour directement dans les chapitres.
- Insérer ses propres jeux de données dans les sections intitulées *Mise en œuvre dans R*.
- Modifier les tableaux et figures.
- Ajouter une série d'exercices.
- Rédiger un nouveau chapitre.
- Modifier des syntaxes R. Plusieurs *packages* R peuvent être utilisés pour mettre en œuvre telle ou telle méthode statistique. Ces derniers évoluent aussi très vite et de nouveaux *packages* sont proposés fréquemment! Par conséquent, il peut être judicieux de modifier une syntaxe R du livre en fonction de ses habitudes de programmation dans R (utilisation d'autres *packages* que ceux utilisés dans le manuel

⁴<https://rmarkdown.rstudio.com/>

⁵https://LAEQ.github.io/livre_statistique_Phil_Jere/

⁶<https://fabriquerel.org/rel/>

par exemple) ou de bien mettre à jour une syntaxe à la suite de la parution d'un nouveau *package* plus performant ou intéressant.

- Toute autre adaptation qui permet de répondre au mieux à un besoin pédagogique.

Un manuel conçu comme un projet collaboratif

Il existe actuellement de nombreux livres sous licence ouverte écrits avec rmarkdown⁷ avec le *package* bookdown (Xie 2016), répertoriés sur le site de <https://bookdown.org/>. Sans surprise, R étant un logiciel libre dédié aux méthodes statistiques et à la science des données, plusieurs abordent les méthodes quantitatives, notamment :

- Beyond Multiple Linear Regression : Applied Generalized Linear Models and Multilevel Models in R⁸ (Roback et Legler 2021), CC BY-NC-SA.
- Introduction to Econometrics with R⁹ (Handk et al. 2019), CC BY-NC-SA.
- Statistical Inference via Data Science : A ModernDive into R and the Tidyverse¹⁰ (Ismay et Kim 2019), CC BY-NC-SA.
- R Graphics Cookbook, 2nd edition¹¹ (Chang 2018), CC BY.

Par contre, la grande majorité de ces livres numériques rédigés avec bookdown sont en anglais. À notre connaissance, ce projet constitue le premier manuel numérique en français sur les méthodes quantitatives appliquées aux sciences sociales réalisé avec bookdown. La première version du livre étant lancée, il est grand temps de planifier les suivantes! Pour ce faire, nous considérons ce livre comme un **projet collaboratif visant à mobiliser la communauté universitaire francophone qui enseigne les statistiques en sciences sociales avec R**. Plusieurs raisons motivent cette vision collaborative :

- **Rien n'est parfait!** Cette première version comprend sûrement des coquilles et certaines sections mériteraient d'être améliorées. Les commentaires et suggestions visant à améliorer son contenu sont les bienvenus.
- **La table des matières doit être impérativement extensible!** De nombreuses méthodes statistiques très utilisées en sciences sociales ne sont pas abordées dans ce livre et mériteraient d'être ajoutées dans une version ultérieure : certaines extensions des régressions linéaires (régressions Rigge et Lasso, Tobit, quantile, etc.), les modèles d'équations simultanées, les analyses de données longitudinales (entre autres, modèles de survie, régression par panel), les modèles d'équations structurelles et bien d'autres! Par conséquent, si vous êtes intéressé(e)s, à ajouter un nouveau chapitre ou une partie du livre, nous vous invitons vivement à communiquer avec nous ou à diffuser sous une licence similaire votre version dérivée. L'objectif étant de continuer à faire tourner la roue du libre et, idéalement, que les futures versions soient corédigées par une communauté d'auteurs et d'autrices spécialistes en méthodes quantitatives.

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 vous conseillons de le lire de gauche à droite et page par page! Plus sérieusement, le livre comprend plusieurs types de blocs de texte qui, nous l'espérons, en facilitent la lecture.

⁷<https://rmarkdown.rstudio.com/>

⁸<https://bookdown.org/robact/bookdown-BeyondMLR/>

⁹<https://www.econometrics-with-r.org/>

¹⁰<https://moderndive.com/>

¹¹<https://r-graphics.org/>



Bloc packages. Habituellement localisé au début d'un chapitre, il comprend la liste des *packages R* utilisés pour un chapitre.



Bloc objectif. Il comprend une description des objectifs d'un chapitre ou d'une section.



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



Bloc pour aller plus loin. Il comprend des références ou des extensions d'une méthode statistique abordée dans une section.



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



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

Comment utiliser les données du livre pour reproduire les exemples ?

Ce livre propose des exemples détaillés et appliqués dans R pour chacune des méthodes abordées. Ces exemples se basent sur des jeux de données structurés et mis à disposition avec le livre. Ils sont disponibles sur le *repo github* dans le sous-dossier `data`, à l'adresse https://github.com/LAEQ/livre_statistique_Phil_Jere/tree/master/data.

Pour télécharger l'intégralité des données, vous pouvez utiliser le lien suivant : https://downgit.github.io/#/home?url=https://github.com/LAEQ/livre_statistique_Phil_Jere/tree/master/data. Cela est rendu possible grâce à l'outil DownGit¹².

Une autre option est de télécharger le *repo* complet du livre directement sur *github* (https://github.com/LAEQ/livre_statistique_Phil_Jere) en cliquant sur le bouton `Code`, puis le bouton `Download ZIP` (figure 2). Les données se trouvent alors dans le sous-dossier nommé `data`.

Structure du livre

Le livre est organisé autour de cinq grandes parties.

Partie 1. La découverte de R. Dans cette première partie, nous discutons brièvement de l'histoire et de la philosophie de R. Nous voyons ensuite comment installer R et RStudio. Les bases du langage R (particulièrement les principaux objets que sont le vecteur, la matrice, la liste et le *dataframe*) ainsi que la manipulation des données avec R sont aussi largement abordés dans le chapitre 1.

Partie 2. Analyses univariées et représentations graphiques. Cette seconde partie comprend deux chapitres. Dans le chapitre 2, nous décrivons dans un premier temps les différents types de données (primaires *versus* secondaires, transversales *versus* longitudinales, spatiales *versus* aspatiales, individuelles

¹²<https://downgit.github.io/#/home>

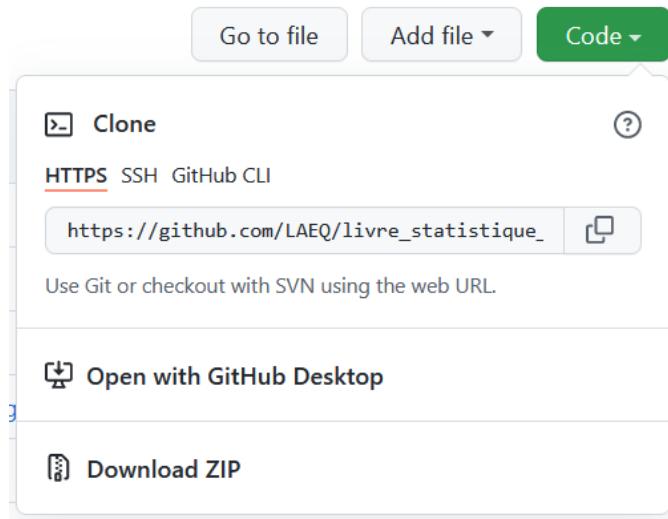


FIG. 2 : Téléchargement de l'intégralité du livre

versus agrégées), les différents types de variables quantitatives (discrètes et continues) et qualitatives (nominales et ordinaires) et les principales distributions de variables utilisées en sciences sociales (uniforme, Bernoulli, binomiale, géométrique, binomiale négative, poisson, poisson avec excès de zéros, gaussienne, gaussienne asymétrique, log-normale, Student, Cauchy, Chi-carré, exponentielle, Gamma, bêta, Weibull et Pareto). Dans un second temps, nous abordons les statistiques descriptives pour des variables quantitatives (paramètres de tendance centrale, paramètres de position, paramètres de dispersion, paramètres de forme), puis qualitatives (fréquences absolues, relatives et cumulées).

Dans le chapitre 3, nous illustrons les incroyables capacités graphiques de R en mettant en œuvre les principaux graphiques (histogramme, graphique de densité, nuage de points, graphique en lignes, boîtes à moustache, graphique en violon, graphique en barre, graphique circulaire), quelques graphiques particuliers (graphique en radar, diagramme d'accord, nuage de mots, carte proportionnelle) et une initiation aux cartes choroplèthes.

Partie 3. Analyses bivariées. Cette troisième partie comprend trois chapitres dans lesquelles sont présentées les principales méthodes exploratoires et confirmatoires bivariées permettant d'évaluer la relation entre deux variables. Plus spécifiquement, nous présentons puis mettons en œuvre dans R les méthodes permettant d'explorer les relations entre deux variables quantitatives (covariance, corrélation et régression linéaire simple) dans le chapitre 4, deux variables qualitatives (tableau de contingence et test du khi-deux) dans le chapitre 5 et une variable quantitative avec une variable qualitative avec deux modalités (tests de Student, de Welch et de Wilcoxon) ou avec plus de deux modalités (ANOVA et test de Kruskal-Wallis) dans le chapitre 6.

Partie 4. Modèles de régression. Dans cette quatrième partie, sont présentées les principales méthodes de statistique inférentielle utilisées en sciences sociales : la régression linéaire multiple (chapitre 7), les régressions linéaires généralisées (chapitre 8), les régressions à effets mixtes (chapitre 9), les régressions multiniveaux (chapitre 10), (chapitre 11) et les modèles généralisés additifs (chapitre 11).

Partie 5. Analyses exploratoires multivariées. Dans cette cinquième partie, sont abordées les méthodes de statistique exploratoire et descriptive permettant de décrire des tableaux de données comprenant plusieurs variables. Nous décrivons d'abord les méthodes de réduction de données : les méthodes factorielles dans le chapitre 12 (analyses de composantes principales, analyses factorielles de correspondances, analyses factorielles de correspondances multiples) et les méthodes de classification non supervisées dans le chapitre 13 (classification ascendante hiérarchique, k-moyennes, k-médianes, k-médoïdes et leurs exten-

sions en logique floue comme les c-moyennes et c-médianes).

Pourquoi faut-il programmer en sciences sociales ?

Vous contrasterez rapidement que R est un véritable langage de programmation. L'apprentissage de ce langage de programmation est-il pour autant pertinent pour les étudiants et étudiantes en sciences sociales ? Il est vrai que la programmation n'est pas une compétence qui vient d'emblée à l'esprit lorsque l'on s'intéresse à la recherche aux sciences sociales. Pourtant, elle est de plus en plus importante, et ce, pour plusieurs raisons :

- Une part toujours plus grande des phénomènes sociaux se produisent ou peuvent s'observer au travers d'environnements numériques. Être capable d'exploiter efficacement ces outils permet d'extraire des données riches sur des phénomènes complexes, tel qu'en témoignent des études récentes sur la propagation de la désinformation sur les réseaux sociaux (Allcott et Gentzkow 2017), la migration des personnes (Spryatos et al. 2019), la propagation et les risques de contamination de la COVID-19 (Boulos et Geraghty 2020). Le plus souvent, les interfaces (API par exemple) permettant d'accéder à ces données nécessitent des habiletés en programmation.
- La quantité de données numériques ouvertes et accessibles en ligne croît chaque année sur des sujets très divers. La plupart des villes et des gouvernements ont maintenant leur portail de données ouvertes auxquelles s'ajoutent les données produites par des projets collaboratifs comme OpenStreetMap¹³ ou NoisePlanet¹⁴. Récupérer ces données et les structurer pour les utiliser à des fins de recherche nécessitent le plus souvent des compétences en programmation.
- Les méthodes quantitatives connaissent également un développement très important. Les logiciels propriétaires peinent à suivre la cadence de ce développement, contrairement aux logiciels à code source ouvert (comme R) qui permettent d'avoir accès aux dernières méthodes. Il est souvent long et coûteux de développer une interface graphique pour un logiciel, ce qui explique que la plupart de ces programmes en sont dépourvus et nécessitent alors de savoir programmer pour les utiliser.
- Savoir programmer donne une liberté considérable en recherche. Cette compétence permet notamment de ne plus être limité(e) aux fonctionnalités proposées par des logiciels spécifiques. Il devient possible d'innover tant en matière de structuration, d'exploration et d'analyse des données que de représentation des résultats en écrivant ses propres fonctions. Cette flexibilité contribue directement à la production d'une recherche de meilleure qualité et plus diversifiée.
- Programmer permet également d'automatiser des tâches qui autrement seraient extrêmement répétitives comme : déplacer et renommer une centaine de fichiers ; retirer les lignes inutiles dans un ensemble de fichiers et les compiler dans une seule base de données ; vérifier parmi des milliers d'adresses lesquelles sont valides ; récupérer chaque jour les messages postés sur un forum. Autant de tâches faciles à automatiser si l'on sait programmer.
- Dans un logiciel avec une interface graphique, il est compliqué de conserver un historique des opérations effectuées. Programmer permet au contraire de garder une trace de l'ensemble des actions effectuées au cours d'un projet de recherche. En effet, le code utilisé reste disponible et permet de reproduire (ou d'adapter) la méthode et les résultats obtenus, ce qui est essentiel dans le monde de la recherche. À cela s'ajoute le fait que chaque ligne de code que vous écrivez vient s'ajouter à un capital de code que vous possédez, car elles pourront être réutilisées dans d'autres projets !

¹³<https://www.openstreetmap.org>

¹⁴https://noise-planet.org/map_noisecapture/index.html

Remerciements

De nombreuses personnes ont contribué à l'élaboration de ce manuel. Ce projet a bénéficié du soutien pédagogique et financier de la *fabriqueREL*¹⁵ (ressources éducatives libres). Les différentes rencontres avec le comité de suivi nous ont permis de comprendre l'univers des ressources éducatives libres (REL) et notamment leurs fameux 5R¹⁶ (Retenir — Réutiliser — Réviser — Remixer — Redistribuer), de mieux définir le besoin pédagogique visé par ce manuel, d'identifier des outils et des ressources pédagogiques pertinents pour son élaboration. Ainsi, nous remercions chaleureusement les membres de suivi de la *fabriqueREL* pour leur support inconditionnel :

- Myriam Beaudet, bibliothécaire à l'Université de Sherbrooke.
- Marianne Dubé, conseillère pédagogique à l'Université de Sherbrooke et coordonnatrice de la *fabriqueREL*.
- Myrian Grondin, bibliothécaire à l'Institut national de la recherche scientifique (INRS).
- Claude Potvin, conseiller en formation à l'Université Laval.
- Serge Allary, vice-recteur adjoint aux études de l'Université de Sherbrooke.

Nous tenons aussi à remercier sincèrement les étudiants et étudiantes du cours **Méthodes quantitatives appliquées aux études urbaines (EUR8219)** du programme de maîtrise en études urbaines de l'INRS. Leurs commentaires et suggestions nous ont permis d'améliorer grandement les versions préliminaires de ce manuel qui ont été utilisées dans le cadre de ce cours.

Nous remercions les membres du comité de révision pour leurs commentaires et suggestions très constructifs. Ce comité est composé de trois étudiantes et deux professeurs de l'INRS¹⁷ :

- Victoria Gay-Gauvin, étudiante à la maîtrise en études urbaines.
- Salomé Vallette, étudiante au doctorat en études urbaines.
- Diana Pena Ruiz, étudiante au doctorat en études des populations.
- Benoît Laplante¹⁸, professeur enseignant aux programmes de maîtrise et de doctorat en études des populations.
- Xavier Leloup¹⁹, professeur enseignant au programme de doctorat en études urbaines.

Finalement, nous remercions Denise Latreille, réviseure linguistique et chargée de cours à l'Université Sherbrooke, pour la révision du manuel.

Dédicace toute spéciale à Cargo et Ambrée

Fait cocasse, l'écriture de ce livre a démarré lorsque Philippe Apparicio était famille d'accueil d'un chiot de la Fondation Mira²⁰, un organisme à but non lucratif qui forme des chiens-guides et d'assistance pour accroître l'autonomie et l'inclusion sociale des personnes vivant avec un handicap visuel ou moteur, ainsi que des jeunes présentant un trouble du spectre de l'autisme (TSA). En fin de rédaction du livre, ce fût au tour de Jérémy Gelb d'être famille d'accueil d'un autre chiot Mira. Nous remercions chaleureusement la Fondation Mira²¹ pour nous avoir donné l'occasion de vivre cette expérience incroyable. Ce livre est donc dédié au beau Cargo et à la belle Ambrée qui nous ont tant supportés dans l'écriture du livre. Il n'y a rien de plus relaxant que d'écrire un livre de statistique avec un chiot qui dort à ses pieds!

¹⁵ <https://fabriquerel.org/>

¹⁶ <https://fabriquerel.org/rel/>

¹⁷ <https://inrs.ca/>

¹⁸ <https://inrs.ca/la-recherche/professeurs/benoit-laplante/>

¹⁹ <https://inrs.ca/la-recherche/professeurs/xavier-leloup/>

²⁰ <https://www.mira.ca/fr/>

²¹ <https://www.mira.ca/fr/>

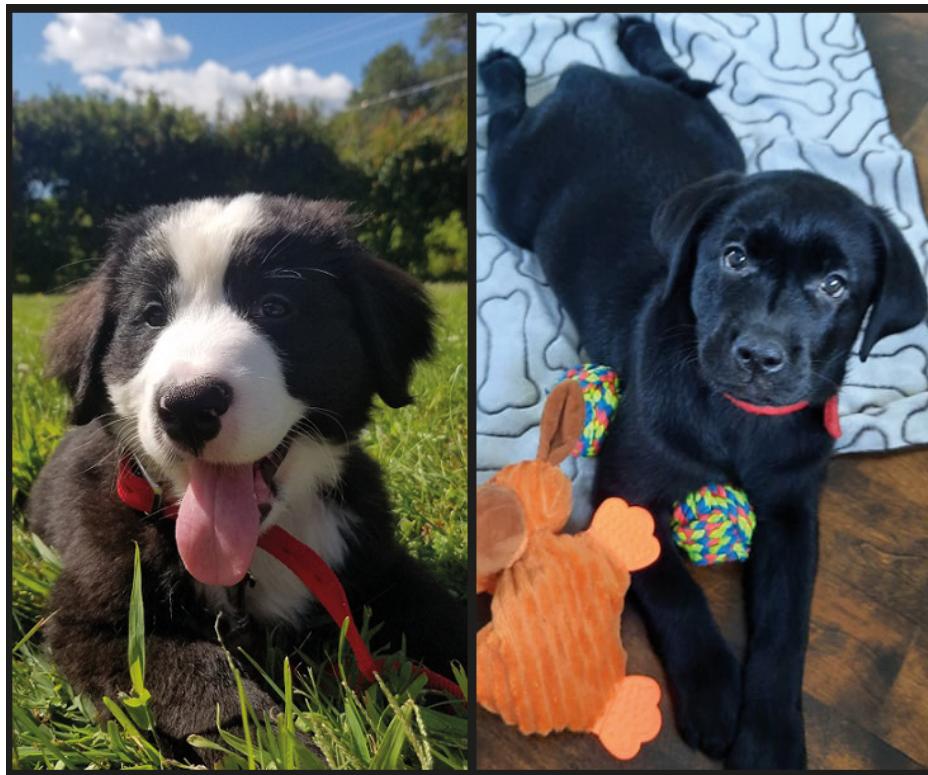


FIG. 3 : Cargo et Ambrée, chiots de la Fondation Mira

À propos des auteurs

Philippe Apparicio (<http://www.inrs.ca/philippe-apparicio>) est professeur titulaire au Centre Urbanisation Culture Société de l’Institut national de la recherche scientifique (INRS, <http://www.inrs.ca/>). Il enseigne au programme de maîtrise en études urbaines (<https://inrs.ca/les-etudes/programmes-d-etudes/etudier-en-sciences-sociales/>) 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/>). 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, le bruit et le vélo en ville. Il a publié une centaine d’articles scientifiques dans différents domaines des études urbaines et de la géographie.

Jérémy Gelb a obtenu un doctorat en études urbaines à l’INRS en 2022, sous la supervision de Philippe Apparicio. Il est 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*²² et *spNetwork*²³, 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 plusieurs années. Avec d’autres collègues, ils ont copublié plusieurs articles (Apparicio, Gelb et al. 2021; Apparicio, Maignan et Gelb 2021; Gelb et Apparicio 2021a; Gelb et Apparicio 2021b; Apparicio et Gelb 2020; Buregeya, Apparicio et Gelb 2020; Gelb et Apparicio 2020; Apparicio, Gelb et Mathieu 2019; Delaunay et al. 2019; Gelb et Apparicio 2019; Apparicio et al. 2018; Apparicio et al. 2017; Apparicio, Carrier et al. 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 des Suds (notamment indiennes, africaines et latino-américaines) et Jérémy dans les villes des Nords (européennes et nord-américaines).

²²<https://cran.r-project.org/web/packages/geocmeans/index.html>

²³<https://cran.r-project.org/web/packages/spNetwork/index.html>



FIG. 4 : Philippe Apparicio et Jérémie Gelb lors d'une collecte de données à vélo à Delhi

Première partie

Découverte de R

Chapitre 1

Prise en main de R

Dans ce chapitre, nous revenons brièvement sur l'histoire de R et la philosophie qui entoure le logiciel. Nous donnons quelques conseils pour son installation et la mise en place d'un environnement de développement. Nous présentons les principaux objets qui sous-tendent le travail effectué avec R (*DataFrame*, vecteur, matrice, etc.) et comment les manipuler avec des exemples appliqués. Si vous maîtrisez déjà R, nullement besoin de lire ce chapitre !



Dans ce chapitre, nous utilisons 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 soutenu par une fondation : *R Foundation for Statistical Computing*. Il est principalement écrit en C et en Fortran, deux langages de programmation de « bas niveau », proches du langage machine. À l'inverse, R est un langage de « haut niveau », car plus proche du langage humain.

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 version expérimentale 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 nous le comparons à ses concurrents SPSS (1968), SAS (1976) et Stata (1984).

R a cependant réussi à s'imposer tant dans le 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](http://r4stats.com/articles/popularity) (figure 1.2).

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



FIG. 1.1 : Lieu de pèlerinage de R

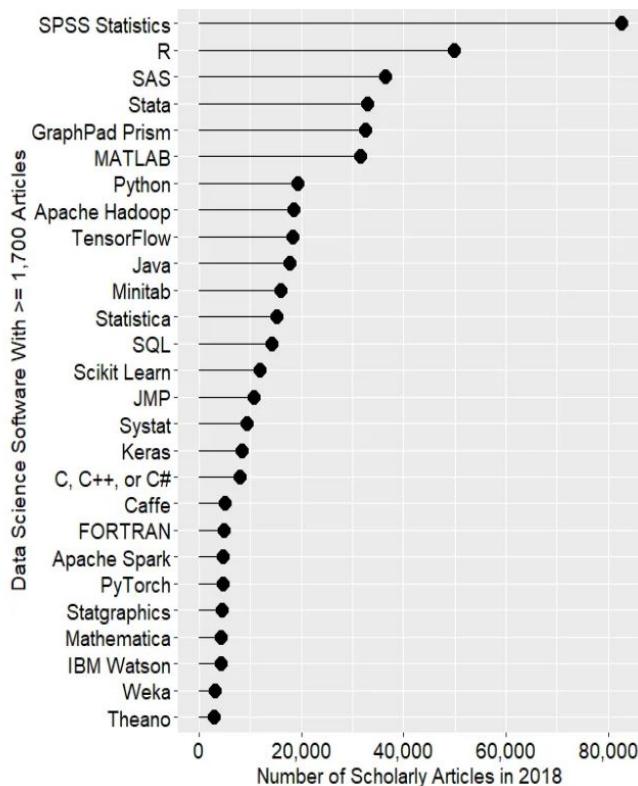


FIG. 1.2 : Nombre d'articles trouvés sur Google Scholar (source : Robert A. Muenchen)

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 multiplateforme, 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.), avec des systèmes de *big data* (Spark, Hadoop), avec des systèmes d'information géographique (QGIS, ArcGIS) et même avec 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 commerciaux (SPSS, SAS, STATA). Avec R, vous pouvez accomplir de nombreuses tâches : monter un site web, créer un robot collectant des données en ligne, combiner des fichiers PDF, composer des diapositives pour une présentation ou même éditer un livre (comme celui-ci), mais aussi, et surtout, réaliser des analyses statistiques.

Un des principaux attraits de R est la quantité astronomique de *packages* actuellement disponibles. **Un package est un ensemble de nouvelles fonctionnalités développées par des personnes utilisatrices 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 *dplyr* permettent de manipuler des tableaux de données; le *package car* offre de nombreux outils pour faciliter l'analyse de modèles de régressions, etc. Ce partage de *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 trouver des solutions de rechange (et ainsi apprendre la syntaxe de 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 d'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 une autre personne. 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, soit CRAN⁴ et 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 personnes utilisant R. Ces derniers ressemblent à de 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 (figure 1.3) : R apparaît clairement comme une communauté hétéroclite, mais diversifiée et adaptable.

Dans ce livre, nous détaillons les *packages* utilisés dans chaque section avec un encadré spécifique, accompagné de l'icône présentée à la figure 1.4.

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...

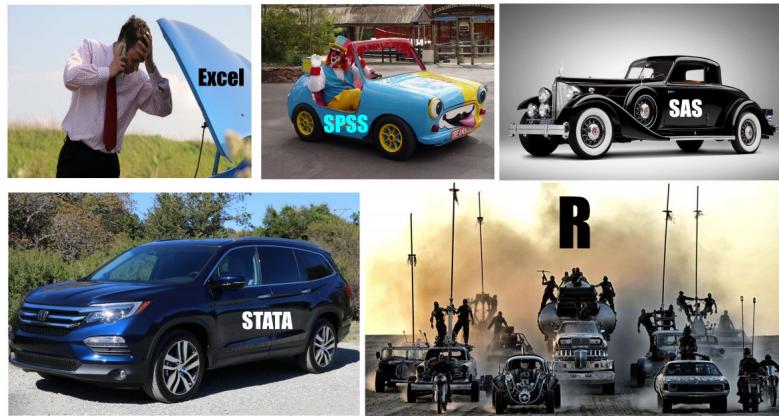


FIG. 1.3 : Métaphore sur les langages et programmes d'analyse statistique



FIG. 1.4 : Icône des encadrés dédiés aux packages

1.2 Environnement de travail

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

1.2.1 Installation de R

La première étape pour travailler avec R est bien sûr de l'installer. Pour cela, 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 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. Cela 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 Environnement RStudio

Rares sont les adeptes de R qui préfèrent travailler directement avec la console classique. Nous vous recommandons vivement d'utiliser RStudio, un environnement de développement (*IDE*) dédié à R offrant une intégration très intéressante d'une console, d'un éditeur de texte, d'une fenêtre de visualisation des données et 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-bagages, des garde-boues et une selle confortable. Vous pouvez télécharger⁸ et installer RStudio sur Windows, Linux

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 : Console de base de R

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 une assistance technique. Il existe d'autres environnements de développement pour travailler avec R (VisualStudio, Jupyter, Tinn-R, Radiant, RIDE, etc.), mais RStudio offre à ce jour la meilleure option en termes de facilité d'installation, de prise en main et de fonctionnalités proposées (voir l'interface de RStudio à la figure 1.6).

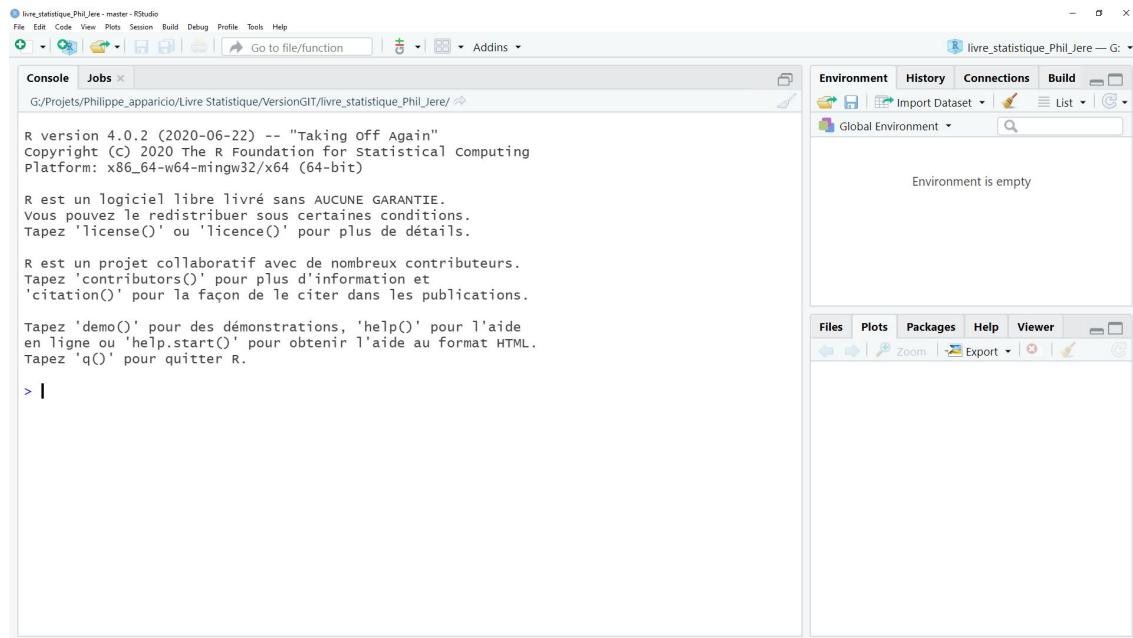


FIG. 1.6 : Environnement de base de RStudio

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*,

puis sélectionnez 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 vous de choisir le style qui vous convient.

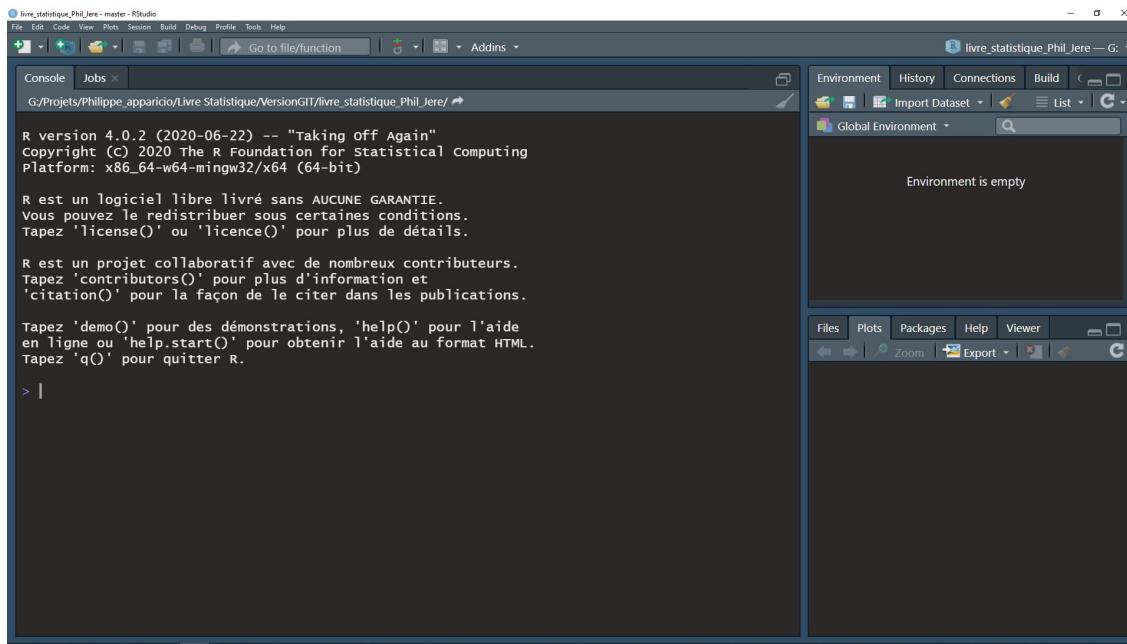


FIG. 1.7 : RStudio avec le style pastel on dark

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 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, les fonctions et les 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 dossiers 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 la souris) et utilisez le raccourci *Ctrl+Entrée* ou cliquez sur le bouton *Run* (avec la flèche verte) pour envoyer cette syntaxe à la console qui l'exécutera immédiatement. Notez que rien ne se passe tant que le code n'est pas envoyé à la console. Il s'agit donc de deux étapes distinctes : écrire son code, puis l'envoyer à la console. Constatez 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).

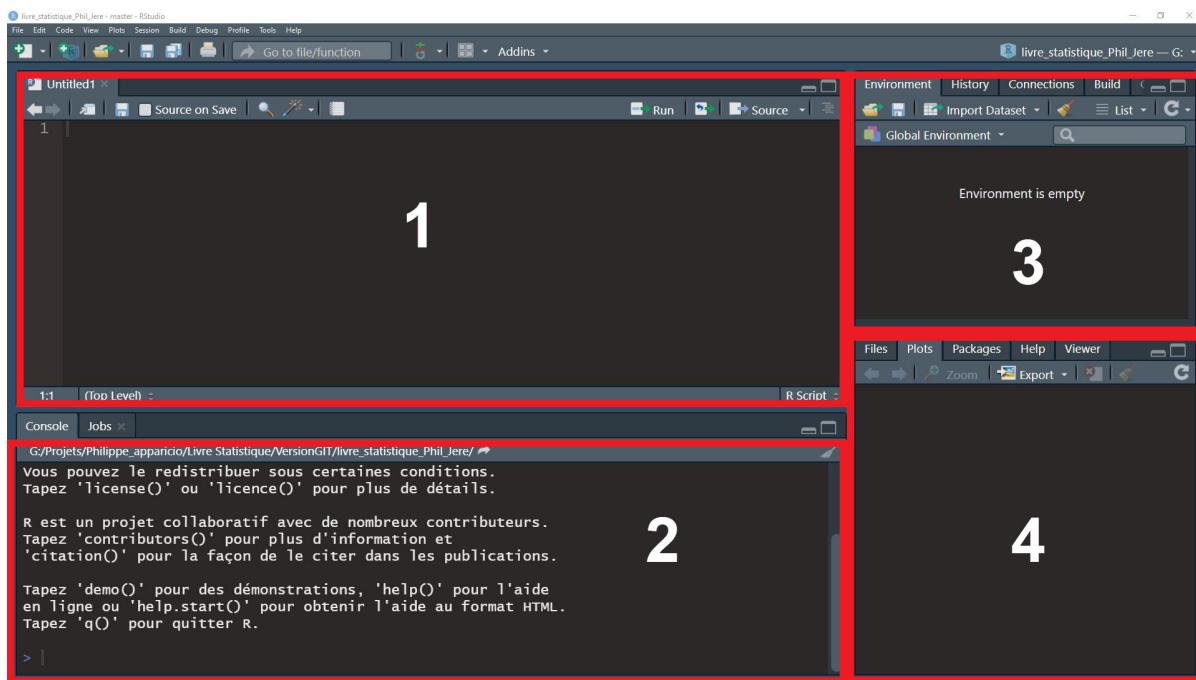


FIG. 1.8 : Fenêtres de RStudio

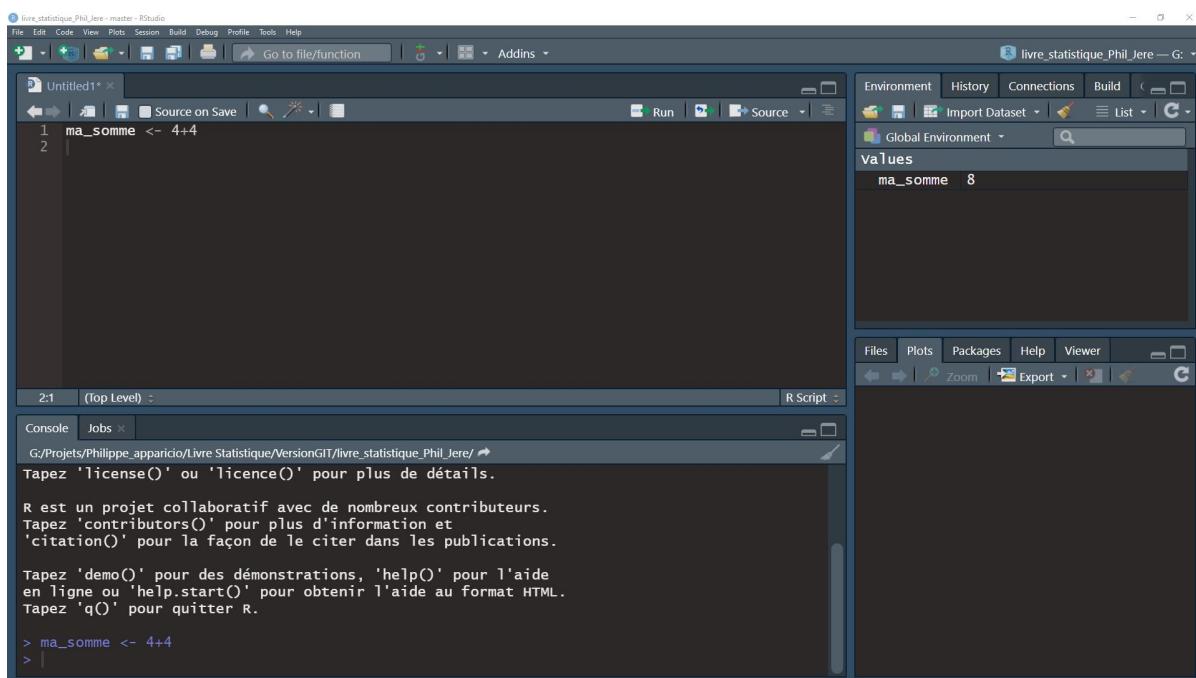


FIG. 1.9 : Exécuter du code dans RStudio

Pour conclure cette section, nous vous invitons à enregistrer votre première syntaxe R (*File/Save As*) dans un fichier `.R` que vous pouvez appeler `mon_premier_script.R` par exemple. 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. Cela 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 Installation et chargement un package

Dans la section sur la Philosophie de R, nous avons souligné la place centrale jouée par les *packages*. Notez que les termes *paquet* et plus rarement *librairie* sont parfois utilisés en français. Voyons ensemble comment installer un *package*, par exemple celui intitulé `lubridate`, qui nous permettra plus tard de manipuler des données temporelles.

1.2.3.1 Installation d'un package depuis CRAN

Pour installer un *package*, il est nécessaire d'être connecté à Internet puisque 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, il demeure disponible localement sur votre ordinateur, à moins de le désinstaller explicitement avec la fonction `remove.packages`.

1.2.3.2 Installation d'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 nous pouvons même y trouver des versions en développement avec des fonctionnalités encore non intégrées dans la version sur CRAN. Reprenons le cas de `lubridate`, mais sur GitHub (il est disponible ici¹⁰). Pour l'installer, nous devons d'abord installer un autre *package* appelé `remotes` (depuis CRAN).

```
install.packages("remotes")
```

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

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

1.2.3.3 Chargement d'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, il suffit d'utiliser la fonction `library`. 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

⁹<https://github.com/>

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

à mesure de votre code, mais ce dernier perd alors en lisibilité. Notez que pour chaque nouvelle session (redémarrage de R), il faut recharger les *packages* dont vous avez besoin avec la fonction `library`.

```
library(lubridate)
```

Si vous obtenez un message d'erreur du type :

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

Cela signifie 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 une faute de frappe dans le nom du *package*. Vous pouvez également redémarrer RStudio et réessayer d'installer ce *package*.

1.2.4 Aide disponible

Lorsque vous installez des *packages* dans R, vous téléchargez aussi leur documentation. Tous les *packages* de CRAN disposent d'une documentation, ce qui n'est pas forcément vrai pour ceux sur 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.

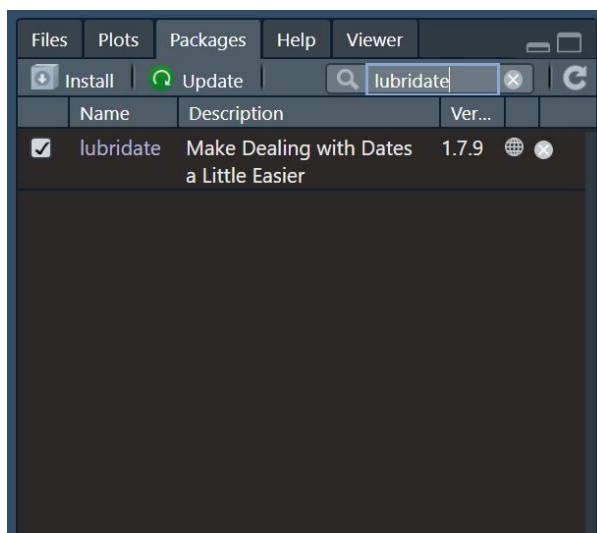


FIG. 1.10 : Description des packages

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 ne vous souvenez plus à quel *package* la fonction appartient, lancez une recherche en utilisant un double point d'interrogation :

```
??now
```

Vous découvrirez ainsi que la fonction `now` n'existe pas que dans `lubridate`, ce qui souligne l'importance de bien connaître les *packages* que nous installons et que nous chargeons dans notre session !

Maintenant que nous avons fait le tour de l'environnement de travail, nous pouvons passer aux choses sérieuses, soit les bases du langage R.

1.3 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 abordons 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 les manipulations que vous pouvez effectuer avec R, avant de poursuivre avec de la lecture des chapitres suivants dédiés aux analyses statistiques. Vous pourrez revenir consulter cette section au besoin. Notez aussi que la maîtrise des différents objets et des différentes 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 référence des objets et des 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 néophytes comment afficher le message `Hello World` à l'écran avec le langage en question.

```
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) avec 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 tirets bas (_) et doit commencer par une lettre. R est sensible à la casse ; en d'autres termes, les variables `Ma_somme`, `ma_sommE`, `ma_SOMME`, et `MA_SOMME` renvoient toutes à un objet différent. Attention donc aux fautes de frappe. 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
```

Portez alors attention 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. Autrement dit, 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 une prochaine section comment sauvegarder des objets et les recharger dans une session ultérieure de R (section 1.6). 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 utilisons la fonction `round`, dont l'objectif est d'arrondir un nombre avec décimales 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)` (figure 1.11), 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 est le nombre de décimales à conserver. Nous pouvons 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.

round(x, digits = 0)
signif(x, digits = 6)

Arguments

- `x` a numeric vector. Or, for `round` and `signif`, a complex vector.
- `digits` integer indicating the number of decimal places (`round`) or significant digits (`signif`) to be used. Negative values are allowed (see 'Details').

FIG. 1.11 : Arguments de la fonction `round`

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'argument. Par exemple, la fonction `now` indique la date précise (avec l'heure) et n'a besoin d'aucun argument pour le faire :

```
now()
```

```
## [1] "2022-05-07 09:29:16 EDT"
```

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

```
round()
```

Erreur : 0 argument passé à 'round' qui en exige 1 ou 2

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")
```

Erreur dans `round("Hello World")` : argument non numérique pour une fonction mathématique

À nouveau le message est très explicite : nous avons passé un argument non numérique à une fonction mathématique. Lisez toujours vos messages d'erreurs : ils permettent de repérer les coquilles et de corriger votre code !

Nous terminons cette section avec une fonction essentielle, `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* ou *doubles*), par exemple 15 et 15.3.
- Les données de type texte qui sont des chaînes de caractères (appelées *strings*) et déclarées entre guillemets "abcdefg".
- Les données booléennes (*booleans*) qui peuvent n'avoir que deux valeurs : vrai (`TRUE`) ou faux (`FALSE`).

Déclarons une variable pour chacun de ces types :

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

Simples ou doubles quotes ?

Pour déclarer des données de type texte, il est possible d'utiliser des quotes simples ' (apostrophe) ou des quotes doubles " (guillemets), cela ne fait aucune différence pour R. Cependant, si la chaîne de caractères que vous créez contient une apostrophe, il est nécessaire d'utiliser des quotes doubles et inversement si votre chaîne de caractère contient des guillemets.

```
phrase1 <- "J'adore le langage R!"
phrase2 <- 'Je cite : "il est le meilleur langage de statistique".'
```

Si la chaîne de caractère contient des guillemets et des apostrophes, il est nécessaire d'utiliser la barre oblique inversée \ pour indiquer à R que ces apostrophes ou ces guillemets ne doivent pas être considérés comme la fin de la chaîne de caractère.

```
phrase3 <- "Je cite : \"j'en rêve la nuit\"."
cat(phrase3)
```

```
## Je cite : "j'en rêve la nuit".
```

Les barres obliques inversées ne font pas partie de la chaîne de caractère, ils sont là pour “échapper” les guillemets qui doivent rester dans la chaîne de caractère. Si une chaîne de caractère doit contenir une barre oblique inversée, alors il faut l'échapper également en utilisant une deuxième barre oblique inversée.

```
phrase4 <- "Une phrase avec une barre oblique inversée : \\"
cat(phrase4)
```

```
## Une phrase avec une barre oblique inversée : \
```

Faites attention à la coloration syntaxique de RStudio! Elle peut vous aider à repérer facilement une chaîne de caractère qui aurait été interrompue par un guillemet ou une apostrophe mal placés.

Si vous avez un doute sur le type de données stockées dans une variable, vous pouvez utiliser la fonction `typeof`. Par exemple, cela permet de repérer si des données qui sont censées être numériques sont en fait stockées sous forme de texte comme dans l'exemple ci-dessous.

```
typeof(age)
```

```
## [1] "double"
```

```
typeof(taille)
```

```
## [1] "double"
```

```
# Ici tailletxt est définie comme une chaîne de caractère car la valeur est
# définie entre des guillemets.
tailletxt <- "175.5"
typeof(tailletxt)
```

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

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

- pour représenter un objet vide, nous utilisons l'objet `NULL`,
- pour représenter une donnée manquante, nous utilisons l'objet `NA`,
- pour représenter un texte vide, nous utilisons une chaîne de caractère de longueur 0, soit "".

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

1.3.5 Opérateurs

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

1.3.5.1 Opérateurs mathématiques

Les opérateurs mathématiques (tableau 1.1) permettent d'effectuer des calculs avec des données de type numérique.

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
%%	Reste de division	15,5 % 2	1,5
/%	Division entière	15,5 %/ 2	7,0

1.3.5.2 Opérateurs relationnels

Les opérateurs relationnels (tableau 1.2) 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.

TAB. 1.2 : Opérateurs relationnels

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

1.3.5.3 Opérateurs logiques

Les opérateurs logiques (tableau 1.3) 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.
- L'opérateur **NOT** (**!**) permet d'inverser une condition. Ainsi, NOT TRUE donne FALSE et NOT FALSE donne TRUE.

TAB. 1.3 : Opérateurs logiques

Opérateur	Description	Syntaxe	Résultat
&	ET	TRUE & FALSE	FALSE
	OU	TRUE FALSE	TRUE
!	NOT	! TRUE	FALSE

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 :

```
# Produit TRUE car nous commençons par tester A < B puis 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 nouveaux comportements pour les opérateurs. Il est par exemple possible d'additionner ou de comparer des objets spéciaux comme des dates, des géométries, des graphes, etc.

1.3.6 Structures de données

Jusqu'à présent, nous avons utilisé des objets ne comprenant qu'une seule valeur. Or, des analyses statistiques nécessitent de travailler avec des volumes de données bien plus grands. Pour stocker des valeurs, nous travaillons avec différentes structures de données : les vecteurs, les matrices, les tableaux de données et les listes.

1.3.6.1 Vecteurs

Les vecteurs sont la brique élémentaire de R. Ils permettent de stocker une série de valeurs **du même type** dans une seule variable. Pour déclarer un vecteur, nous utilisons 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 personnes à un questionnaire.



Il existe dans R une subtilité à l'origine de nombreux malentendus : la distinction 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 contient 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 est donc intéressant de fixer les valeurs possibles du vecteur pour éviter d'en ajouter de nouvelles par erreur. Pour cela, nous devons convertir ce vecteur texte en vecteur de type facteur, ci-après nommé simplement facteur, avec la fonction `as.factor`.

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

Notez qu'à présent, nous pouvons ajouter une nouvelle couleur dans le premier 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, nous préférerons 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 nous convertissons un facteur en valeurs numériques, nous obtenons 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, nous pouvons calculer la moyenne du vecteur `ages` en utilisant la fonction `mean` présente de base dans R.

```
mean(ages)
```

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

Cela démontre bien que le vecteur est la brique élémentaire de R ! Toutes les variables que nous avons déclarées 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) et est généralement utilisée 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 (1er chiffre : lignes; 2e chiffre : colonnes)
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 les lignes d'une matrice. La figure 1.12 présente graphiquement le passage 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 à la matrice 2 nous allons avoir une erreur, car elles n'ont pas les mêmes dimensions.

```
matrice1 == matrice2
```

Erreur dans matrice1 == matrice2 : tableaux de tailles inadéquates

En revanche, nous pouvons transposer la matrice 1 et refaire cette comparaison :

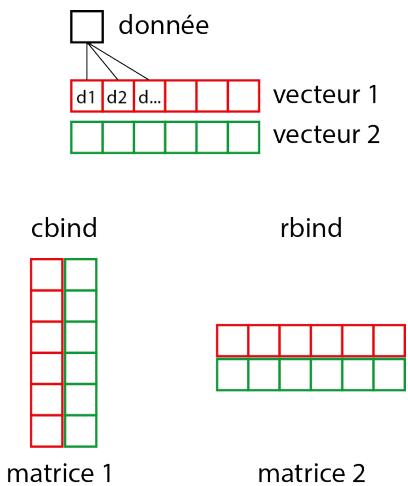


FIG. 1.12 : Du vecteur à la matrice

```
t(matrix1) == matrix2
```

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

Le résultat souligne bien que nous avons 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 avec des matrices, il est encore plus rare de manipuler 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 nous pourrions représenter comme un cube, ou une série de matrices de mêmes dimensions et empilées. Au-delà de trois dimensions, il devient difficile de les représenter mentalement. 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 et la profondeur délimiterait les différentes bandes composant l'image (figure 1.12).

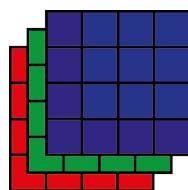


FIG. 1.13 : Un array avec trois dimensions

Créons un *array* en combinant trois matrices avec la fonction `array`. Chacune de ces matrices est composée respectivement de 1, de 2 et de 3 et a une dimension de 5×5 . L'*array* final a donc une dimension 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 la plus souvent utilisée. Dans cette structure, chaque ligne du tableau représente un individu et chaque colonne représente une caractéristique de cet individu. Ces colonnes ont des noms qui permettent facilement d'accéder à leurs valeurs. Créons un *DataFrame* (tableau 1.4) à partir de nos quatre vecteurs et de la fonction `data.frame`.

```

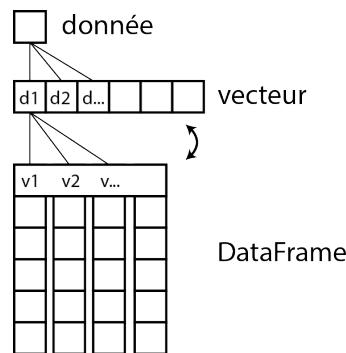
df <- data.frame(
  "age" = ages,
  "taille" = tailles,
  "adresse" = adresses,
  "proprietaire" = proprietaires
)

```

TAB. 1.4 : 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

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"

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-leur 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 devriez 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. Vous allez apprendre désormais à manipuler des données dans des *DataFrames* !

1.4 Manipulation de données

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

1.4.1 Chargement d'un *DataFrame* depuis un fichier

Il est rarement nécessaire de créer vos *DataFrames* manuellement. 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 et 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 *Shape-File*;
- *.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 ayez à charger des fichiers provenant de logiciels propriétaires :

- *.sas7bdat* (SAS);
- *.sav* (SPSS);
- *.dta* (STATA).

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

1.4.1.1 Lecture d'un fichier *csv*

Pour le format *csv*, il n'est pas nécessaire 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 **délimiteur** (ou **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 le nom et la description 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 non de type facteur.

1.4.1.2 Lecture d'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 Lecture d'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 observations et 48colonnes
```

1.4.1.4 Lecture d'un fichier *sav* (SPSS)

Pour importer un fichier *sav* provenant du logiciel statistique SPSS, utilisez la fonction `read.spss` du 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 observations et 48colonnes
```

1.4.1.5 Lecture d'un fichier *sas7bdat* (SAS)

Pour importer un fichier *sas7bdat* provenant du logiciel statistique SAS, utilisez la fonction `read.sas7bdat` du package *sas7bdat*. Installez préalablement le package (`install.packages("sas7bdat")`) et chargez-le (`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 observations et 48 colonnes
```

1.4.1.6 Lecture d'un fichier *xlsx* (Excel)

Lire un fichier Excel dans R n'est pas toujours une tâche facile. Généralement, nous recommandons d'exporter le fichier en question au format *csv* dans un premier temps, puis de le lire avec la fonction `read.csv` dans un second temps (section 1.4.1.1).

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*. Sous windows, il est possible que vous deviez également installer manuellement le package *rJava* et indiquer à R où se trouve JAVA sur votre ordinateur. La procédure est détaillée ici¹¹.

```
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 = "")
```

¹¹ <https://cimentadaj.github.io/blog/2018-05-25-installing-rjava-on-windows-10/installing-rjava-on-windows-10/>

```
## le DataFrame df6 a 951 observations et 48 colonnes
```

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 de type 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 observations et 48colonnes
```

Si nous comparons les temps d'exécution (tableau 1.5), nous constatons que la lecture des fichiers *xlsx* peut être extrêmement longue si nous ne spécifions pas le type des colonnes, ce qui peut devenir problématique pour des fichiers volumineux. Notez également que la lecture d'un fichier *csv* devient de plus en plus laborieuse à mesure que sa taille 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.

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

Durée (secondes)	Fonction
0,46	read.csv
0,14	read.dbf
0,08	read.spss
0,06	read.dta
0,97	read.sas7bdat
19,92	read.xlsx
0,53	read.xlsx2

1.4.2 Manipulation d'un *DataFrame*

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

1.4.2.1 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 suivant¹²). 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 rétrocompatibilité. En d'autres termes, des changements importants peuvent être introduits d'une version à l'autre rendant potentiellement obsolète votre ancien 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 rétrocompatibilité.

1.4.2.2 Gestion des colonnes d'un *DataFrame*

Repartons du *DataFrame* que nous avions chargé précédemment en important un fichier *csv*.

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

1.4.2.2.1 Sélection d'une colonne

Rappelons qu'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
```

¹²<https://blog.ephorie.de/why-i-dont-use-the-tidyverse>

TAB. 1.6 : Avantages et inconvénients du `tidyverse`

Avantage du <code>tidyverse</code>	Problème posé par le <code>tidyverse</code>
Simplicité d'écriture et d'apprentissage	Nouvelle syntaxe à apprendre
Ajout de l'opérateur <code>%>%</code> permettant d'enchaîner les traitements	Perte de lisibilité avec l'opérateur <code>-></code>
La meilleure librairie pour réaliser des graphiques : <code>ggplot2</code>	Remplacement de certaines fonctions de base par d'autres provenant du <code>tidyverse</code> lors de son chargement, pouvant créer des erreurs.
Crée un écosystème cohérent <i>Package</i> en développement et de plus en plus utilisé	Ajout d'une dépendance dans le code Philosophie d'évolution agressive, aucune assurance de rétrocompatibilité

1.4.2.2.2 Sélection de plusieurs colonnes

Il est possible de sélectionner plusieurs colonnes d'un *DataFrame* et de filtrer ainsi les colonnes inutiles. Pour cela, nous pouvons utiliser un vecteur contenant soit les positions des colonnes (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 Suppression de colonnes

Il est parfois plus intéressant et rapide de supprimer directement des colonnes plutôt que de recréer un nouveau *DataFrame*. Pour ce faire, nous attribuons 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 Modification du nom des colonnes

Il est possible de changer le nom d'une colonne. Cette opération est importante pour faciliter la lecture du *DataFrame* ou encore s'assurer que l'exportation du *DataFrame* dans un format particulier (tel que `.dbf` qui ne supporte que les noms de colonnes avec moins de 10 caractères) 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 Calcul 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 1.3.5. Prenons un exemple concret : calculons la densité de population par secteur de recensement dans notre *DataFrame*, puis 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 applique le calcul à chaque ligne de votre jeu de données et stocke 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 la valeur absolue de chaque valeur d'un vecteur;
- `sqrt` calcule la racine carrée de chaque valeur d'un vecteur;
- `log` calcule le logarithme de chaque valeur d'un vecteur;

- `exp` calcule l'exponentielle de chaque valeur d'un vecteur;
- `factorial` calcule la factorielle de chaque valeur d'un vecteur;
- `round` arrondit la valeur d'un vecteur;
- `ceiling`, `floor` arrondit à l'unité supérieure ou inférieure de chaque valeur d'un vecteur;
- `sin`, `asin`, `cos`, `acos`, `tan`, `atan` sont des fonctions de trigonométrie;
- `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 des chaînes de caractères

Outre les données numériques, vous aurez à travailler avec des données de type texte (`string`). 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 Sheet*¹³. 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 mettons, dans un premier temps, des majuscules à la première lettre des prénoms et des 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"
```

Nous pourrions également tout mettre en minuscules ou tout en majuscules.

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

¹³ <https://github.com/rstudio/cheatsheets/blob/master/strings.pdf>

```
## [1] "jérémy 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 Remplacement du texte

Les adresses comprennent des caractères accentués. Ce type de caractères cause régulièrement des problèmes d'encodage. Nous pourrions alors 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"
```

La même fonction peut être utilisée 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 Découpage du texte

Il est parfois nécessaire de découper du texte pour en extraire des éléments. Nous devons alors choisir un caractère de découpage. Dans notre exemple, nous pourrions vouloir extraire les numéros civiques des adresses en sélectionnant le premier espace comme caractère de découpage, et 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, soit 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 découpe 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 garder seulement la première des deux colonnes.

Il est également possible d'extraire des parties de texte et de ne garder par exemple que les N premiers caractères ou les N derniers caractères :

```
# Ne garder que les 5 premiers caractères
substr(df$adresses_3,start = 1, stop = 5)

## [1] "15 ru" "413 B" "3606" "2457"

# Ne garder que les 5 derniers caractères
n_caract <- nchar(df$adresses_3)
substr(df$adresses_3, start = n_caract-4, stop = n_caract)

## [1] " Levy" "urent" " Duke" "Marys"
```

Notez que les paramètres `start` et `stop` de la fonction `substr` peuvent accepter un vecteur de valeurs. Il est ainsi possible d'appliquer une sélection de texte différente à chaque chaîne de caractères dans notre vecteur en entrée. Nous pourrions par exemple vouloir récupérer tout le texte avant le second espace pour garder uniquement le numéro civique et le type de rue.

```
# Étape 1 : récupérer les positions des espaces pour chaque adresses
positions <- str_locate_all(df$adresses_3, " ")

# Étape 2 : récupérer les positions des seconds espaces
sec_positions <- sapply(positions, function(i){
  i[2,1]
})

# Étape 3 : appliquer le découpage
substr(df$adresses_3, start = 1, stop = sec_positions-1)

## [1] "15 rue"          "413 Boulevard" "3606 rue"      "2457 route"
```

1.4.2.5.4 Concaténation du texte

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

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

## [1] "Jérémy 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 d'indiquer 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] "Jérémy 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"
```

Si vous souhaitez concaténer des éléments de texte sans séparateur, la fonction `paste0` peut être plus simple à utiliser.

```
paste0("Please conca", "tenate me!")
```

```
## [1] "Please concatenate me!"
```

1.4.2.6 Manipulation 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 différents *packages*. Nous abordons ici les types date et heure (*date and time*). Pour les manipuler, nous privilégions l'utilisation du *package* `lubridate` du `tidyverse`. Pour illustrer le tout, nous l'utilisons avec un jeu de données ouvertes de la Ville de Montréal représentant les collisions routières impliquant au moins un cycliste survenues après le 1^{er} 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 la collision.

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). Ensuite, Nous concaténons l'heure et la date avec la fonction `paste`, puis nous convertissons 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 d'indiquer celle à utiliser. 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 la barre oblique (/) pour séparer les éléments de la date et le deux points (:) pour l'heure, et une 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, pour éviter que la fonction ne retourne la valeur `NA` lorsqu'elle ne peut lire le format. Après toutes ces opérations, rejetons 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 Extraction 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 et la seconde avec les fonctions `year()`, `month()`, `day()`, `hour()`, `minute()` et `second()`.

1.4.2.6.3 Calcul d'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 7 h comme une variable dans une analyse visant à déterminer le moment critique des collisions routières 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 est 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 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)
```

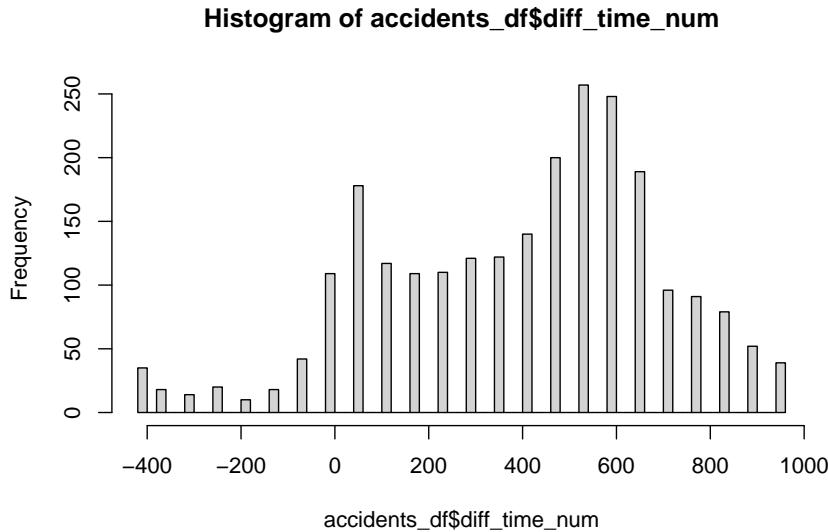


FIG. 1.15 : Répartition temporelle des accidents à vélo

Nous observons clairement deux pics, un premier entre 0 et 100 (entre 7 h et 8 h 30 environ) et un second plus important entre 550 et 650 (entre 16 h et 17 h 30 environ), ce qui correspond sans surprise aux heures de pointe (figure 1.15). Il est intéressant de noter que plus d'accidents se produisent à l'heure de pointe du soir qu'à celle du matin.

1.4.2.6.4 Fuseau horaire

Lorsque nous travaillons avec des données provenant de différents endroits dans le monde ou que nous devons tenir compte des heures d'été et d'hiver, il convient de tenir compte du fuseau horaire. Pour créer une date avec un fuseau horaire, il est possible d'utiliser le paramètre `tz` dans la fonction `as_datetime` et d'utiliser l'identifiant du fuseau approprié. Dans notre cas, les données d'accident ont été collectées à Montréal, qui a un décalage de -5 heures par rapport au temps de référence UTC (+1 heure en été). Le code spécifique de ce fuseau horaire est *EDT*; il est facile de trouver ces codes avec le site web [timeanddate.com](https://www.timeanddate.com/time/map/)¹⁴.

```
accidents_df$datetime <- as_datetime(accidents_df$date_heure,
                                       format = "%Y/%m/%d %H:%M:%S",
                                       tz = "EDT")
```

1.4.2.7 Recodage des variables

Recoder une variable signifie changer ses valeurs selon une condition afin d'obtenir une nouvelle variable. Si nous reprenons le jeu de données précédent sur les accidents à vélo, nous pourrions vouloir créer une nouvelle colonne nous indiquant si la collision a eu lieu en heures de pointe ou non. Nous obtiendrions ainsi une nouvelle variable avec seulement deux catégories plutôt que la variable numérique originale.

¹⁴ <https://www.timeanddate.com/time/map/>

Nous pourrions aussi définir quatre 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 Cas binaire avec `ifelse`

Si nous ne souhaitons créer que deux catégories, le plus simple est d'utiliser la fonction `ifelse`. Cette fonction évalue une condition (section 1.3.5) pour chaque ligne d'un *DataFrame* et produit un nouveau vecteur. Créons donc une variable binaire indiquant si une collision 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 7 h (0) ET 9 h (120), OU entre 16 h 30 (570) ET 18 h 30 (690)?

```
table(is.na(accidents_df$diff_time_num))
```

```
##  
## FALSE TRUE  
## 2414    40
```

Notons dans un premier temps que nous avons 40 observations sans valeur pour la colonne `diff_time_num`. Il s'agit d'observations pour lesquelles nous ne disposons pas de dates au départ.

```
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 FALSE;
- la valeur à renvoyer si la condition est TRUE.

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

```
table(accidents_df$moment_bin)
```

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

```
# Vérifier si nous avons toujours seulement 40 NA  
table(is.na(accidents_df$moment_bin))
```

```
##  
## FALSE TRUE  
## 2414    40
```

Les heures de pointe représentent quatre heures de la journée, ce qui nous laisse neuf heures hors heures de pointe entre 7 h et 20 h.

```
# Ratio de collisions routières en heures de pointe
(841 / 2414) / (4 / 13)
```

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

```
# Ratio de collisions routières hors heure de pointe
(1573 / 2414) / (9 / 13)
```

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

En rapportant les collisions aux durées des deux périodes, nous observons une nette surreprésentation des collisions 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 Cas multiple avec la `case_when`

Lorsque nous souhaitons 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` 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 7 h et 20 h);
- la nuit (entre 21 h et 7 h).

```
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$diff_time_num < 780 ~ "journée",
  accidents_df$diff_time_num > 120 & accidents_df$diff_time_num < 570 ~ "journée",
  accidents_df$diff_time_num < 0 | accidents_df$diff_time_num >= 780 ~ "nuit"
)
```

```
table(accidents_df$moment_multi)
```

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

```
# Vérifions encore les NA
table(is.na(accidents_df$moment_multi))
```

```
##
## FALSE   TRUE
## 2414     40
```

La syntaxe de cette fonction est un peu particulière. Elle accepte un nombre illimité (ou presque) 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 teste d'abord la première condition et assigne ses valeurs, puis recommence pour les prochaines conditions. Ainsi, si une observation (ligne du tableau de données) obtient TRUE à plusieurs conditions, elle obtient au final la valeur de la dernière condition validée. Dans l'exemple précédent, si la première condition est `accidents_df$diff_time_num >= 0 | accidents_df$diff_time_num <= 120`, alors nous obtenons pour seule valeur en résultat "pointe matin" puisque chaque observation a une valeur supérieure à 0 et que nous avons remplacé l'opérateur & (ET) par l'opérateur | (OU).

1.4.2.8 Sous-sélection d'un *DataFrame*

Dans cette section, nous voyons 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 leur index. Pour cela, nous utilisons un jeu de données fourni avec R : le jeu de données **iris** décrivant des fleurs du même nom.

```
data("iris")

# Nombre de lignes et de colonnes
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_virginica1 <- iris[iris$Species == "virginica",]
iris_virginica2 <- subset(iris, iris$Species == "virginica")

# Vérifions que les deux DataFrames ont le même nombre de lignes
nrow(iris_virginica1) == nrow(iris_virginica2)
```

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

Vous pouvez utiliser, dans les deux cas, tous les opérateurs vus dans les sections 1.3.5.2 et 1.3.5.3. L'enjeu est d'arriver à créer un vecteur booléen final permettant d'identifier les observations à conserver.

1.4.2.8.2 Sous-sélection des colonnes

Nous avons déjà vu comment sélectionner des colonnes en utilisant leur nom ou leur index dans la section 1.4.2.2.1. Ajoutons ici un cas particulier où nous souhaitons sélectionner des colonnes selon une condition. Par exemple, nous pourrions vouloir conserver que les colonnes comprenant le mot *Length*. Pour cela,

nous utilisons 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 Sélection 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 simultanément. Pour ce faire, il faut indiquer en premier les index ou la condition permettant de sélectionner une ligne, puis les index ou la condition pour sélectionner les colonnes : `[index_lignes , index_colonnes]`. Sélectionnons cinq premières lignes et les trois premières colonnes 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)[grepl("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 Fusion de *DataFrames*

Terminons cette section avec la fusion de *DataFrames*. Nous distinguons deux méthodes répondant à des besoins différents : par ajout ou par jointure.

1.4.2.9.1 Fusion de *DataFrames* par ajout

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

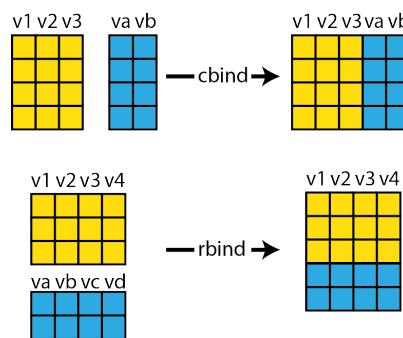


FIG. 1.16 : Fusion de DataFrames

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 commençons par le séparer en trois sous-jeux de données comprenant chacun une espèce d'iris. Puis, nous fusionnons 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 Jointure de *DataFrames*

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. Nous distinguons plusieurs types de jointure :

- les jointures internes permettant de combiner les éléments communs entre deux *DataFrames* 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 n'ont pas de correspondance dans B.

Ces trois jointures sont présentées à la figure 1.17 ; pour ces trois cas, la colonne commune se nomme *id*.

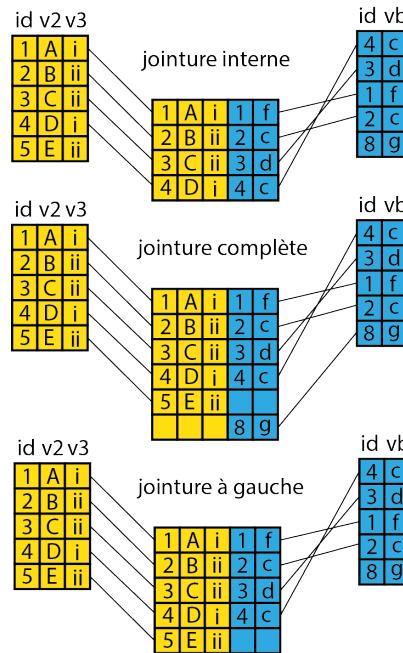


FIG. 1.17 : Jointure de DataFrames

Vous retiendrez que les deux dernières jointures peuvent produire des valeurs manquantes. Pour réaliser ces opérations, nous utilisons 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"))
```

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 forçons ainsi la fonction à garder tous les livres et à 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>

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 Code R bien structuré

Terminons ici avec quelques conseils sur la rédaction d'un code R. Bien rédiger son code est essentiel pour trois raisons :

1. Pouvoir relire et réutiliser son code dans le futur.
2. Permettre à d'autres personnes de bien lire et de réutiliser votre code.
3. Minimiser les risques d'erreurs.

Ne négligez pas l'importance d'un code bien rédigé et bien documenté, vous vous éviterez ainsi des migraines lorsque vous devrez exhumer du code écrit il y a plusieurs mois.

Voici quelques lignes directrices peu contraignantes, mais qui devraient vous être utiles :

1. **Privilégier la clarté à la concision** : il vaut mieux parfois scinder une ligne de code en plusieurs sous-étapes afin de faciliter la lecture de l'opération réalisée. Par exemple, si nous reprenons une ligne de code d'une section précédente où nous sélectionnions l'ensemble des colonnes du jeu de données `iris` comprenant le mot `Length` :

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

Nous pouvons simplifier la lecture de ce code en détaillant les différentes étapes comme suit :

```
noms_cols <- names(iris)
sel_noms <- noms_cols[grep("Length", noms_cols, fixed = TRUE)]
iris_l <- iris[sel_noms]
```

2. **Documenter et commenter son code le plus possible** : il est possible d'ajouter du texte dans un code R qui ne sera pas exécuté, ce que nous appelons des commentaires. Typiquement, une ligne commençant par un `#` n'est pas interprétée par le logiciel. Utilisez des commentaires le plus souvent possible pour décrire les actions que vous souhaitez effectuer avec votre code. Il sera ainsi plus facile de le relire, de naviguer dedans, mais également de repérer d'éventuelles erreurs. Si nous reprenons l'exemple précédent :

```
# Récupération du nom des colonnes dans le DataFrame iris
noms_cols <- names(iris)

# Sélection des colonnes avec les caractères "Length"
sel_noms <- noms_cols[grep("Length", noms_cols, fixed = TRUE)]

# Extraction des colonnes sélectionnées dans un nouveau DataFrame
iris_l <- iris[sel_noms]
```

3. **Éviter le code à rallonge...** : typiquement, essayez de vous limiter à des lignes de code d'une longueur maximale de 80 caractères. Au-delà de ce seuil, il est judicieux de découper votre code en plusieurs lignes.
4. **Adopter une convention d'écriture** : une convention d'écriture est un ensemble de règles strictes définissant comment un code doit être rédigé. À titre d'exemple, il est parfois recommandé d'utiliser le *lowerCamelCase*, le *UpperCamelCase*, ou encore de séparer les mots par des tirets bas *upper_camel_case*. Un mélange de ces différentes conventions peut être utilisé pour distinguer les variables, les fonctions et les classes. Il peut être difficile de réellement arrêter une telle convention, car les différents *packages* dans R utilisent des conventions différentes. Dans vos propres codes, il est surtout important d'avoir une certaine cohérence et ne pas changer de convention.
5. **Indenter le code** : l'indentation du code permet de le rendre beaucoup plus lisible. Indenter son code signifie d'insérer, au début de chaque ligne de code, un certain nombre d'espaces permettant d'indiquer à quel niveau de profondeur nous nous situons. Typiquement, lorsque des accolades ou des parenthèses sont ouvertes dans une fonction, une boucle ou une condition, nous rajoutons deux ou quatre espaces en début de ligne. Prenons un exemple très concret : admettons que nous écrivons une fonction affichant un résumé statistique à chaque colonne d'un jeu de données si cette colonne est de type numérique. L'indentation dans cette fonction joue un rôle crucial dans sa lisibilité. Sans indentation et sans respecter la règle des 80 caractères, nous obtenons ceci :

```
summary_all_num_cols <- function(dataset){for(col in names(dataset)){if(class(dataset[[col]] == "numeric")){print(su
```

Avec de l'indentation et des commentaires, la syntaxe est beaucoup plus lisible puisqu'elle permet de repérer facilement trois niveaux/paliers dans le code :

```
# Définition d'une fonction
summary_all_num_cols <- function(dataset){
  # Itération sur chaque colonne de la fonction
  for(col in names(dataset)){
    # A chaque itération, testons si la colonne est de type numérique
    if(class(dataset[[col]] == "numeric")){
      # Si oui, nous affichons un résumé statistique pour cette colonne
      print(summary(dataset[[col]]))
    } # Ici nous sortons de la condition (niveau 3)
  } # Ici nous sortons de la boucle (niveau 2)
}# Ici nous sortons de la fonction (niveau 1)
```

6. **Adopter une structure globale pour vos scripts** : un code R peut être comparé à une recette de cuisine. Si tous les éléments sont dans le désordre et sans structure globale, la recette risque d'être très difficile à suivre. Cette structure risque de changer quelque peu en fonction de la recette ou de l'auteur(e), mais les principaux éléments restent les mêmes. Dans un code R, nous pouvons distinguer plusieurs éléments récurrents que nous vous recommandons d'organiser de la façon suivante :
 - a. Charger les différents *packages utilisés* par le script. Cela permet dès le début du code de savoir quelles sont les fonctions et méthodes qui seront employées dans le script. Cela limite aussi les risques d'oublier des *packages* qui seraient chargés plus loin dans le code.
 - b. Définir les fonctions dont vous aurez besoin en plus de celles présentes dans les *packages*. Idem, placer nos fonctions en début de code évite d'oublier de les charger ou de les chercher quand nous en avons besoin.
 - c. Définir le répertoire de travail avec la fonction `setwd` et charger les données nécessaires.

- d. Effectuer au besoin les opérations de manipulation sur les données.
 - e. Effectuer les analyses nécessaires en scindant si possible les différentes étapes. Notez également que l'étape de définition des fonctions complémentaires peut être effectuée dans une feuille de code séparée, et l'ensemble de ces fonctions chargées à l'aide de la fonction source. De même, si la manipulation des données est conséquente, il est recommandé de l'effectuer avec un code à part, d'enregistrer les données structurées, puis de les charger directement au début de votre code dédié à l'analyse.
7. **Exploiter les commentaires délimitant les sections dans RStudio** : il est possible d'écrire des commentaires d'une certaine façon pour que l'IDE les détecte comme des délimiteurs de sections. L'intérêt principal est que nous pouvons ensuite facilement naviguer entre ces sections en utilisant RStudio comme montré à la figure 1.18, mais aussi masquer des sections afin de faciliter la lecture du reste du code. Pour délimiter une section, il suffit d'ajouter une ligne de commentaire comprenant quatre fois les caractères -, = ou # à la suite :

```
# Voici ma section 1 -----
# Voici ma section 2 =====
# Voici ma section 3 ######
# Autre exemple pour mieux marquer la rupture dans un code :
#####
### Titre de ma section 4 #####
#####
```

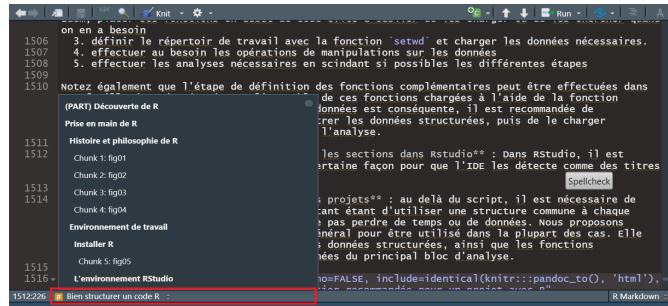


FIG. 1.18 : Navigation dans des sections de codes avec RStudio

8. **Adopter une structure globale pour vos projets** : au-delà du script, il est nécessaire de bien structurer vos projets, le plus important étant d'utiliser une structure commune à chaque projet pour vous faciliter le travail. Nous proposons à la figure 1.19 un exemple de structure assez générale pouvant être utilisée dans la plupart des cas. Elle sépare notamment les données originales des données structurées, ainsi que les fonctions complémentaires et la structuration des données du principal bloc d'analyse.

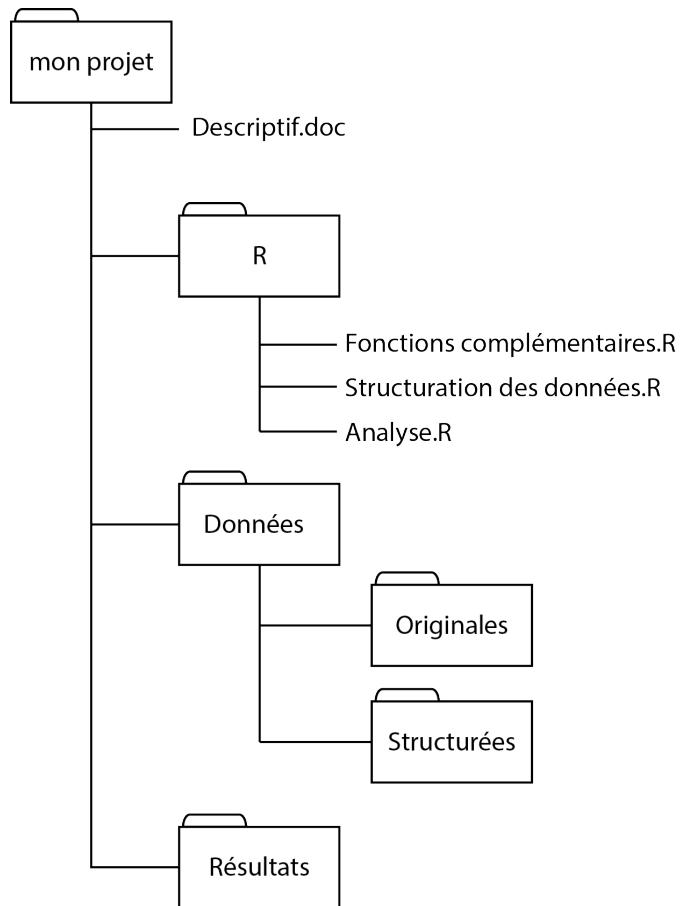


FIG. 1.19 : Structure de dossier recommandée pour un projet avec R

Ne négligez jamais l'importance d'un code bien écrit et documenté!

1.6 Enregistrement des résultats

Comme nous l'avons indiqué précédemment, l'ensemble des objets actuellement chargés dans votre session R sont perdus si vous la fermez. Cela peut être problématique si certains résultats nécessitent de longs temps de calcul ou si vous avez besoin de partager les objets obtenus avec d'autres personnes, mais pas le code pour les obtenir. Il est possible de retrouver les résultats d'une session précédente si ceux-ci ont été enregistrés sur votre disque dur puisque l'action d'enregistrer permet de faire passer vos objets présents dans votre mémoire vive dans des fichiers stockés sur votre disque dur. Vous pouvez pour cela utiliser la fonction `save.image` ou `save`.

`save.image` enregistre une copie exacte de votre session actuelle avec tous les objets présents dans votre environnement dans un fichier `RData`. La fonction `save` permet d'être plus sélectif et de ne garder que certains objets spécifiques.

Voici la syntaxe pour enregistrer toute votre session :

```
save.image(file = 'chemin/vers/mon/fichier/session.RData', compress = TRUE)
```

Vous pouvez aussi utiliser le bouton d'enregistrement dans l'onglet *Environnement* dans RStudio (figure 1.20).

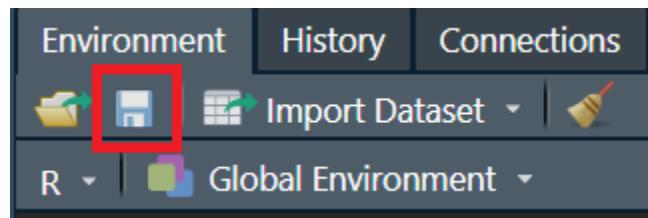


FIG. 1.20 : Bouton enregistrer la session

Il est recommandé de compresser ces fichiers (`compress = TRUE`) pour minimiser leur taille. Pour n'enregistrer que certains objets (ici `iris` et `noms_cols`), vous pouvez adapter cette syntaxe :

```
save(iris, noms_cols, file = 'chemin/vers/mon/fichier/mes_objet.RData', compress = TRUE)
```

Pour récupérer ces objets dans une autre session, il suffit d'utiliser la fonction `load` :

```
load(file = 'chemin/vers/mon/fichier/mes_objet.RData')
```

ou d'utiliser le bouton *ouvrir* de l'onglet *Environnement* dans RStudio (figure 1.21).

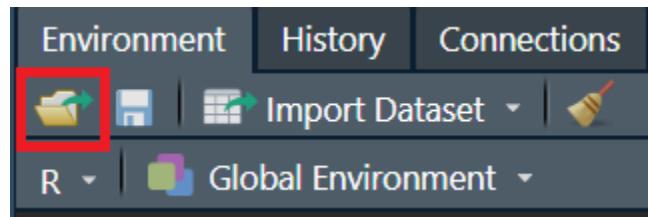


FIG. 1.21 : Bouton charger un fichier RDA

1.7 Session de travail

Comme vous avez pu le constater dans les sections 1.6 et 1.4.1, il est nécessaire de connaître les chemins vers les fichiers que vous souhaitez utiliser dans votre code R. Si tous ces fichiers sont organisés dans un même dossier (ce que nous vous recommandons à la figure 1.19), il est possible de définir un répertoire de travail avec la fonction `setwd`. Il est recommandé d'effectuer cette étape au début de votre code R, après le chargement des *packages*. Ainsi, vous n'aurez pas besoin de réécrire à chaque fois le chemin complet pour accéder à vos fichiers.

```
# Chemin complet
mes_donnees <- read.csv("C:/projets/articles/2022/mon_projet/data/mes_donnees.csv")

# Utilisation de setwd
setwd("C:/projets/articles/2022/mon_projet")
mes_donnees <- read.csv("data/mes_donnees.csv")
```

La fonction `getwd` permet d'afficher le répertoire de travail utilisé actuellement par R.

Si vous utilisez RStudio, il est possible d'utiliser une petite astuce pour définir comme répertoire de travail le dossier dans lequel se trouve le fichier de code R que vous utilisez actuellement :

```
setwd(dirname(rstudioapi:::getActiveDocumentContext()$path))
```

Admettons que votre code R se trouve dans un sous dossier appelé *CodeR* de votre répertoire de travail, vous pouvez remonter d'un niveau dans votre arborescence en utilisant la syntaxe suivante :

```
setwd(paste0(dirname(rstudioapi:::getActiveDocumentContext()$path), "/.."))
```

Le double point (...) indique que nous souhaitons remonter dans le dossier parent du dossier dans lequel nous nous trouvons actuellement.

Il existe deux solutions de rechange à l'utilisation de `setwd` que certains jugent un peu démodé.

- La première est le package `here` permettant de spécifier plus facilement des chemins relatifs et de définir un *top-level directory* pour votre projet.
- La seconde est l'utilisation de la fonctionnalité `projects`¹⁵ de RStudio.

1.8 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! Quelques ressources pertinentes qui pourraient vous être utiles sont aussi reportées au tableau 1.7.

¹⁵<https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects>

TAB. 1.7 : Ressources pertinentes 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 œil sur les nouvelles fonctionnalités de vos 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

1.9 Quiz de révision du chapitre

Questions

- Avec quelle fonction peut-on sélectionner son répertoire de travail ?

- get.wd
- set.wd
- setWd
- setwd

Relisez au besoin la section [1.7](#).

- Installer RStudio est suffisant pour pouvoir utiliser R.

- Vrai
- Faux

Relisez au besoin le début de la section [1.2](#).

- Un package doit être réinstallé à chaque fois que l'on souhaite l'utiliser.

- Vrai
- Faux

Relisez au besoin la section [1.2.3](#).

- La brique de données élémentaire dans R est :

- Le vecteur
- Le DataFrame
- La liste
- La matrice

Relisez au besoin la section [1.3.6](#).

- Un vecteur peut contenir :

- uniquement des valeurs numériques
- uniquement des valeurs textuelles
- uniquement des valeurs booléennes
- des valeurs de types différents

Relisez au besoin la section [1.3.6](#).

- La jointure et la concaténation de DataFrames désignent exactement la même opération.

- Vrai
- Faux

Relisez au besoin la section [1.4.2.9](#).

- Comparativement à une jointure complète, une jointure interne génère un DataFrame avec :

- nécessairement moins d'observations
- nécessairement plus d'observations
- au moins autant d'observations
- autant ou moins d'observations
- le même nombre d'observation

Relisez le deuxième encadré à la section [1.4.2.9.2](#).

Réponses

- Avec quelle fonction peut-on sélectionner son répertoire de travail?
 - setwd
- Installer RStudio est suffisant pour pouvoir utiliser R.
 - Faux
- Un package doit être réinstallé à chaque fois que l'on souhaite l'utiliser.
 - Faux
- La brique de données élémentaire dans R est :
 - Le vecteur
- Un vecteur peut contenir :
 - uniquement des valeurs numériques
 - uniquement des valeurs textuelles
 - uniquement des valeurs booléennes
- La jointure et la concaténation de DataFrames désignent exactement la même opération.
 - Faux
- Comparativement à une jointure complète, une jointure interne génère un DataFrame avec :
 - autant ou moins d'observations

Deuxième partie

Analyses univariées et graphiques dans R

Chapitre 2

Statistiques descriptives univariées

Comprendre la notion de variable et de ses différents types est essentiel en statistiques. En effet, en fonction du type de variable à l'étude, les méthodes de statistique exploratoire ou inférentielle sont différentes. Nous distinguons ainsi cinq types de variables : nominale, ordinale, discrète, continue et semi-quantitative. Aussi, nous abordons un concept central de la statistique : les distributions. Finalement, dans ce chapitre, nous présentons les différentes statistiques descriptives univariées qui peuvent s'appliquer à ces types de variables.



Dans ce chapitre, nous utilisons principalement les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2`, le seul, l'unique!
 - * `ggsurvplot` pour combiner des graphiques et réaliser des diagrammes quantiles-quantiles.
- Pour créer des distributions :
 - * `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` et `moments` 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 et types de variable

2.1.1 Notion de variable

D'un point de vue empirique, une variable est une propriété, une caractéristique d'une unité statistique, d'une observation. Il convient alors de bien saisir à quelle unité d'analyse (ou unité d'observation) s'appliquent les valeurs d'une variable : des personnes, des ménages, des municipalités, des entreprises, etc. Par exemple, pour des individus, l'*âge*, le *genre* ou encore le *revenu* sont autant de caractéristiques qui peuvent être mesurées à partir de variables. Autrement dit, une variable permet de mesurer un phénomène (dans un intervalle de valeurs, c'est-à-dire de manière quantitative) ou de le qualifier (avec plusieurs catégories, c'est-à-dire de manière qualitative).

D'un point de vue plus théorique, une variable permet d'opérationnaliser un concept en sciences sociales (Gilles et Maranda 1994, 30), 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). En effet, la construction d'un modèle théorique suppose d'opérationnaliser différents concepts et d'établir les relations qu'ils partagent entre eux. Or, 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.



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

Ne pas maîtriser la définition d'une variable revient à ne pas bien saisir la caractéristique ou encore 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 souhaitez 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 officielles, langue parlée à la maison, langue maternelle, première langue officielle parlée, connaissance des langues non officielles et langue de travail (<https://www12.statcan.gc.ca/census-recensement/2016/ref/guides/003/98-500-x2016003-fra.cfm>). 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, nous sélectionnerions 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 que nous utiliserions 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 sont 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 les variables mobilisés. Il convient alors de lire attentivement la section méthodologique (pas uniquement la section des résultats ou pire, celle du résumé), sans quoi vous risquez d'aboutir à une revue de littérature approximative. Ayez aussi un **regard critique** sur les variables utilisées en lien avec le cadre théorique. Certains concepts sont très difficiles à traduire en variables ; leurs opérationnalisations (mesures) peuvent ainsi faire l'objet de vifs débats au sein de la communauté scientifique. Très succinctement, c'est notamment le cas du concept de capital social. D'une part, les définitions et ancrages sont bien 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 mesurer le capital social efficacement (de manière quantitative).

Variable de substitution (*proxy variable* en anglais)

Nous faisons la moins pire des recherches ! En effet, les données disponibles sont parfois imparfaites pour

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

répondre avec précision à une question de recherche ; nous pouvons toujours les exploiter, tout en signalant honnêtement leurs faiblesses et limites, et ce, tant pour les données que pour les variables utilisées.

- Des bases de données peuvent être en effet imparfaites. Par exemple, en criminologie, lorsqu'une étude est basée sur l'exploitation de données policières, la limite du **chiffre noir** est souvent signalée : 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 ne soit pas disponible ou qu'elle n'ait tout simplement pas été mesurée. Nous cherchons 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 les cyclistes risquent 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 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 Types de variables

Nous distinguons habituellement les variables qualitatives (nominale ou ordinale) des variables quantitatives (discrète ou continue). Tel qu'illustré à la figure 2.1, plusieurs mécanismes différents visent à qualifier, à classer, à compter ou à mesurer afin de caractériser les unités statistiques (observations) d'une population ou d'un échantillon.

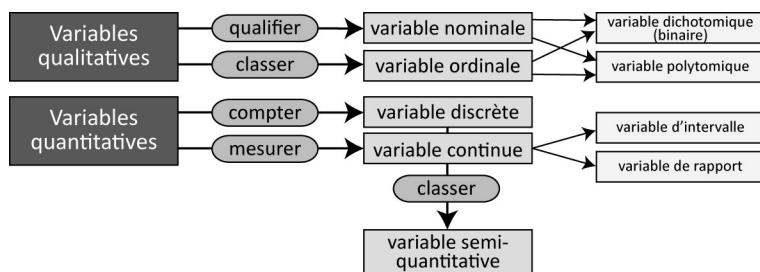


FIG. 2.1 : Types de variables

2.1.2.1 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*, *noir* tandis que le *type de famille* comprendrait les modalités *couple marié*, *couple en union libre* et *famille monoparentale*.

Une **variable ordinale** 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 **dichotomique (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, nous utilisons 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"))
```

Nous calculons 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 Variables quantitatives

Une variable discrète permet de **compter** un phénomène dans un ensemble fini de valeurs, comme le nombre d'accidents impliquant un ou une 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 la disponibilité d'un vélo ou non à la station pour laquelle nous opérons un comptage. Habituellement, une variable discrète ne peut prendre que des valeurs entières (sans décimale), 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 ou d'une 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 dépend de la précision et de la fiabilité de la mesure. Pour un capteur de distance de dépassement, le nombre de décimales dépend de la précision du lidar ou du sonar de l'appareil ; aussi, l'utilisation de trois décimales – soit une précision au millimètre – est largement suffisante pour mesurer la distance de dépassement. De plus, 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 variable ne sont pas constants ; elles n'ont pas de vrai zéro. Autrement dit, ce type de variable a une échelle relative avec un zéro arbitraire. 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), nous pouvons affirmer qu'il y a 20 °C ou 36 °F d'écart entre les deux villes, mais ne pouvons 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 ni 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 va-

riable 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 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, il s'agit de sélectionner le ou les types de données les plus appropriés à votre étude.

2.2.1 Données secondaires *versus* données primaires

Les **données secondaires** sont des données qui existent déjà au début de votre projet de recherche : nul 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 personnes qui font de la recherche, qui étudient ou qui enseignent 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 Réseau canadien des Centres de données de recherche (<https://crdcn.org/fr/donn%C3%A9es>) 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 (<https://www.ciqss.org/>).

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 doit réaliser une collecte de données avec idéalement plusieurs personnes participantes (équipées de différents capteurs), et ce, sur plusieurs jours. Une collecte de données primaires 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 et 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 affiliée la personne responsable du projet de recherche.

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 de 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. 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 (type de rue, 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;
- les pixels des bandes spectrales d'une image satellite.

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 recensements ou d'enquêtes, par exemple, sont des données individuelles pour lesquelles toute une série de variables sont disponibles. 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 Canada²). Comme signalé plus haut, ces microdonnées de recensements 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 de résidence 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 Canada³).

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.

 **Erreur écologique et erreur atomiste** : attention aux interprétations abusives.

Il convient d'être prudent dans l'analyse des données agrégées. Très fréquente en géographie, l'**erreur écologique** (*ecological fallacy* en anglais) est une mauvaise interprétation des résultats. Elle consiste à attribuer des constats obtenus à partir de données agrégées pour un territoire aux individus qui forment la population de ce territoire. À l'inverse, attribuer des résultats à partir de données individuelles à des territoires est une **erreur atomiste**.

Prenons un exemple concret tiré d'une étude récente sur la localisation des écoles primaires et le bruit aérien dans la région métropolitaine de Toronto (Audrin, Apparicio et Séguin 2021). Un des objectifs de cette étude est de vérifier si les écoles primaires ($n_s = 1420$) avec des niveaux de bruit aérien élevés présentent des niveaux de réussite scolaire plus faibles. Les résultats de leur étude démontrent que les enfants scolarisés dans les

²<https://www150.statcan.gc.ca/n1/pub/12-002-x/2012001/article/11642-fra.htm>

³<https://www.statcan.gc.ca/fra/idd/trousse/section5#a4>

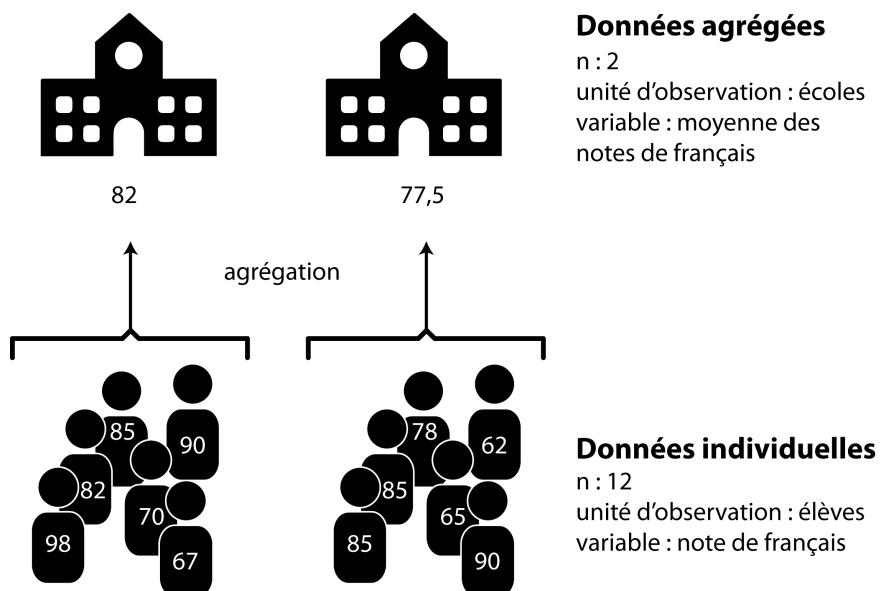


FIG. 2.2 : Exemple d'agrégation de données individuelles

écoles primaires exposées à des niveaux élevés de bruit aérien sont issus de milieux plus défavorisés et ont plus souvent une langue maternelle autre que la langue d'enseignement. Aussi, les écoles avec des niveaux de bruit aérien élevés présentent des niveaux de réussite scolaire plus faibles.

Toutefois, étant donné que les variables sur la réussite scolaire sont mesurées au niveau de l'école (soit les pourcentages d'élèves ayant atteint ou dépassé la norme provinciale en lecture, en écriture et en mathématique, respectivement pour la 3^e année et la 6^e année) et non au niveau individuel, nous ne pouvons pas conclure que le bruit aérien a un impact significatif sur la réussite scolaire des élèves :

« Nous avons pu démontrer que les écoles primaires localisées dans la zone NEF 25 présentent des taux de réussite plus faibles. Rappelons toutefois qu'une association obtenue avec des données agrégées ne peut pas nous permettre de conclure à une influence directe au niveau individuel, car l'agrégation des données entraîne une perte d'information. Cette erreur d'interprétation dite erreur écologique (*ecological fallacy*) tend à laisser penser que les associations entre les groupes s'appliquent à chaque individu (Robinson, 1950). Nos résultats gagneraient à être corroborés à partir d'analyses reposant sur des 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, de gérer et de 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 pouvez consulter :

- la hiérarchie des régions géographiques normalisées pour la diffusion (https://www12.statcan.gc.ca/census-recensement/2016/ref/dict/figures/f1_1-fra.cfm)
- le glossaire illustré (<https://www150.statcan.gc.ca/n1/pub/92-195-x/92-195-x2016001-fra.htm>) des régions géographiques
- les différents profils du recensement de 2016 à télécharger pour les différentes régions géographiques (https://www12.statcan.gc.ca/census-recensement/2016/dp-pd/prof/details/download-telecharger/comp/page_dl-tc.cfm?Lang=F).



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 personnes étudiant dans une université. Bien entendu, elle ne peut interroger toutes les personnes étudiantes de son université. Elle devra donc 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 personnes interrogées. 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. Consultez par exemple cette publication de Statistique Canada (<https://www150.statcan.gc.ca/n1/edu/power-pouvoir/ch13/prob/5214899-fra.htm>).

Autre exemple, une autre chercheure souhaite comprendre les facteurs influençant le sentiment de sécurité des cyclistes dans un quartier. De nouveau, elle ne peut pas enquêter sur l'ensemble des 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

Nous distinguons habituellement deux grandes familles de méthodes statistiques : la statistique descriptive et exploratoire et la statistique inférentielle et confirmatoire. Il existe de nombreuses définitions de ces deux branches de la statistique, celles proposées de Lebart et al. (1995) étant parmi les plus abouties :

- « **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'inflimer, 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, Morineau et Piron 1995, 209).

2.4 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 Définition générale

En probabilité, nous nous intéressons aux résultats d'expériences. Du point de vue de la théorie des probabilités, lancer un dé, mesurer la pollution atmosphérique, compter le nombre de collisions à une intersection, et 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 un modèle mathématique permettant d'associer pour chaque résultat possible d'une expérience la probabilité d'obtenir ce résultat. D'un point de vue pratique, si nous disposons de la distribution régissant l'expérience : « mesurer la concentration d'ozone à Montréal à 13 h en été », nous pouvons calculer la probabilité de mesurer une valeur inférieure à $15 \mu\text{g}/\text{m}^3$.



Loi de probabilité et distribution

L'utilisation que nous faisons ici du terme « distribution » est un anglicisme (éhonté diront certaines personnes). En effet, en français, la définition précédente est plus proche du terme « loi de probabilité ». Cependant, la quasi-totalité de la documentation sur R est en anglais et, dans la pratique, ces deux termes ont tendance à se confondre. Nous avons donc fait le choix de poursuivre avec ce terme dans le reste du livre.

Une distribution est toujours définie dans un intervalle en dehors duquel elle n'est définie ; 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 nous réalisons l'expérience, nous obtenons nécessairement un résultat, et que la somme des probabilités est répartie entre tous les résultats possibles de l'expérience. En langage mathématique, nous disons que l'intégrale de la fonction de densité d'une distribution est 1 dans son 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, nous pouvons 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 de 1/6.

Chacune des deux expériences précédentes est régie par une distribution appartenant à la famille des distributions **discrètes**. Elles servent à représenter 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 en principe infini. Par exemple, mesurer la taille d'une personne adulte sélectionnée au hasard peut produire en principe un nombre infini de valeurs. Les distributions sont utiles pour décrire les résultats potentiels 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).

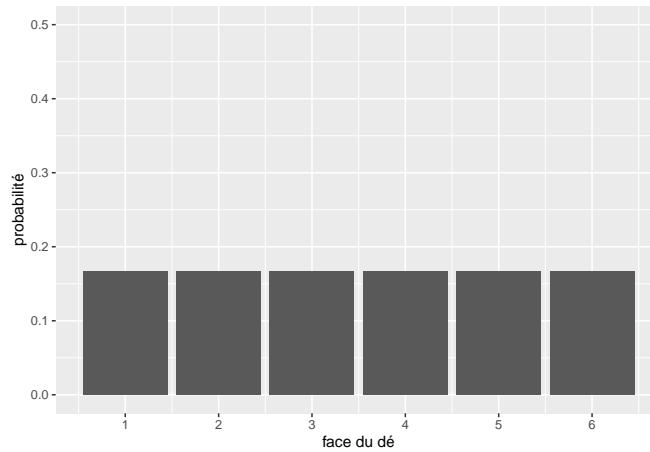


FIG. 2.3 : Distribution théorique d'un lancer de dé

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

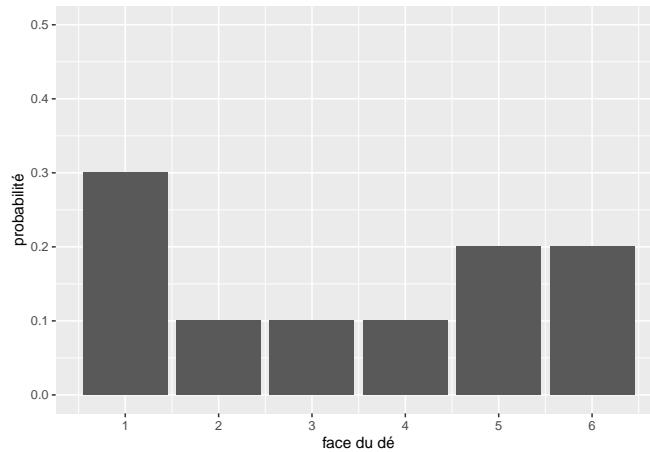


FIG. 2.4 : Distribution empirique d'un lancer de dé (n=10)

Il s'agit de la distribution **empirique**. Chaque échantillon aura sa propre distribution empirique. Cependant, comme le prédit la loi des grands nombres : 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. Du point de vue de la théorie des probabilités, chaque échantillon correspond à un ensemble de tirages aléatoires effectués à partir de la distribution théorique du phénomène étudié.

Pour nous en convaincre, collectons trois échantillons de lancer de dé de respectivement 30, 100 et 1000 observations (figure 2.5). Comme le montre la figure 2.4, nous connaissons la distribution théorique qui régit cette expérience.

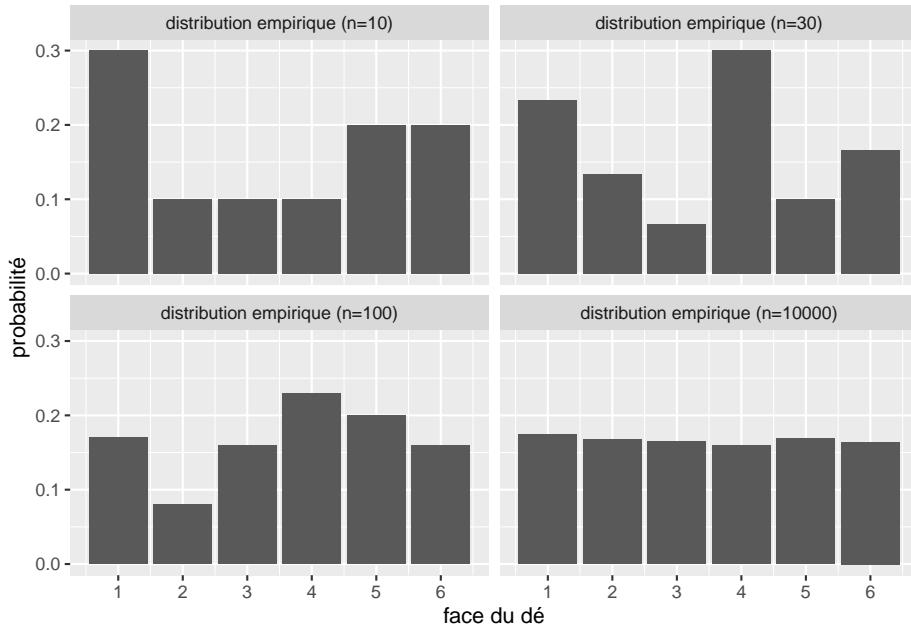


FIG. 2.5 : Distribution empirique d'un lancer de dé

Nous constatons bien qu'au fur et à mesure que la taille de l'échantillon augmente, nous tendons vers la distribution théorique.

Cette relation a été étudiée pour la première fois au XVIII^e siècle par le mathématicien Daniel Bernoulli, qui a montré que la probabilité que la moyenne d'une distribution empirique soit éloignée de la moyenne de la distribution théorique dont elle est tirée diminuait lorsque nous augmentons le nombre des tirages et donc la taille de l'échantillon. Un autre mathématicien, Siméon-Denis Poisson, a fait connaître cette relation sous le nom de « loi des grands nombres ».

Les distributions théoriques sont 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 nous nous posons 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, sélectionner des distributions théoriques pouvant vraisemblablement correspondre au phénomène mesuré.
- Comparer visuellement ou à l'aide de tests statistiques la distribution empirique de la variable et diverses distributions théoriques pour trouver la plus adaptée.

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

2.4.2 Anatomie d'une distribution

Une distribution (ou loi de probabilité) 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) associant chaque résultat possible de l'expérience régie par la distri-

bution à la probabilité d'observer ce résultat. 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 , nous pouvons 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 appliquée, la distribution de Bernoulli est utilisée en analyse spatiale pour étudier la concentration d'accidents de la route ou de crimes en milieu urbain. À chaque endroit du territoire, il est possible de calculer la probabilité qu'un tel évènement ait lieu ou non en modélisant les données observées au moyen de la loi de Bernoulli. 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 de densité de la distribution 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 est infinitésimale. Il sert uniquement à représenter la valeur de la fonction de densité de la distribution pour chaque valeur de x .

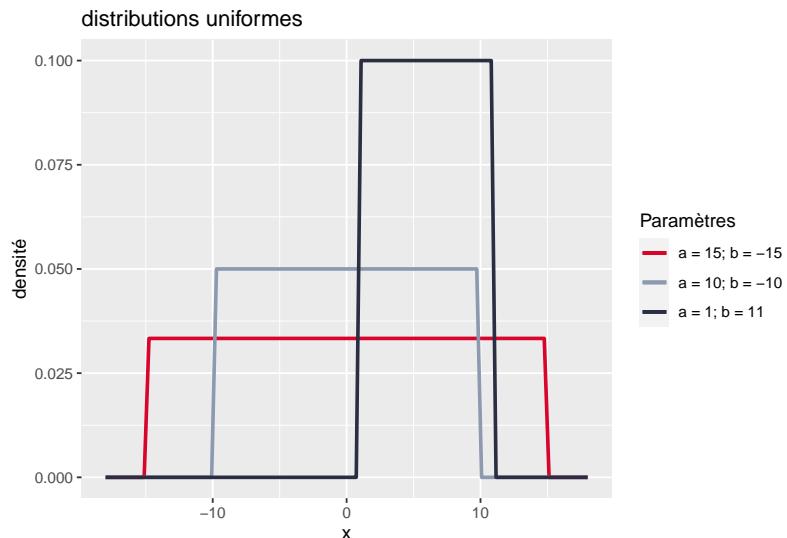


FIG. 2.6 : Distributions uniformes continues

Nous observons 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 est 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 j'ai de récupérer au moins 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 changent la localisation de la distribution (la déplacer vers la droite ou la gauche de l'axe des X), d'autres changent 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.



Fonction de répartition, de survie et d'intensité

Si les fonctions de densité ou de masse d'une distribution sont le plus souvent utilisées pour décrire une distribution, d'autres types de fonctions peuvent également être employées et disposent de propriétés intéressantes.

1. La fonction de répartition : il s'agit d'une fonction décrivant le cumul de probabilités d'une distribution. Cette fonction a un minimum de zéro qui est obtenu pour la plus petite valeur de l'espace d'échantillonnage de la distribution, et un maximum d'un pour la plus grande valeur de ce même espace. Formellement, la fonction de répartition (F) est l'intégrale de la fonction de densité (f).

$$F(x) = \int_{-\infty}^x f(u)du$$

2. La fonction de survie : soit l'inverse additif de la fonction de répartition (R)

$$R(x) = 1 - F(x)$$

3. La fonction de d'intensité, soit le quotient de la fonction de densité et de la fonction de survie (D).

$$D(x) = \frac{f(x)}{F(x)}$$

Ces fonctions jouent notamment un rôle central dans la modélisation des phénomènes qui régissent la survenue des événements, par exemple la mort, les accidents de la route ou les bris d'équipement.

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 Relationships*⁴, qui liste près de 80 distributions.

Nous nous concentrons ici sur une sélection de dix-huit distributions très répandues en sciences sociales. La figure 2.7 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 est possible de produire une figure très différente. Les distributions discrètes sont représentées avec des graphiques en barre, et les distributions continues avec des graphiques de densité.

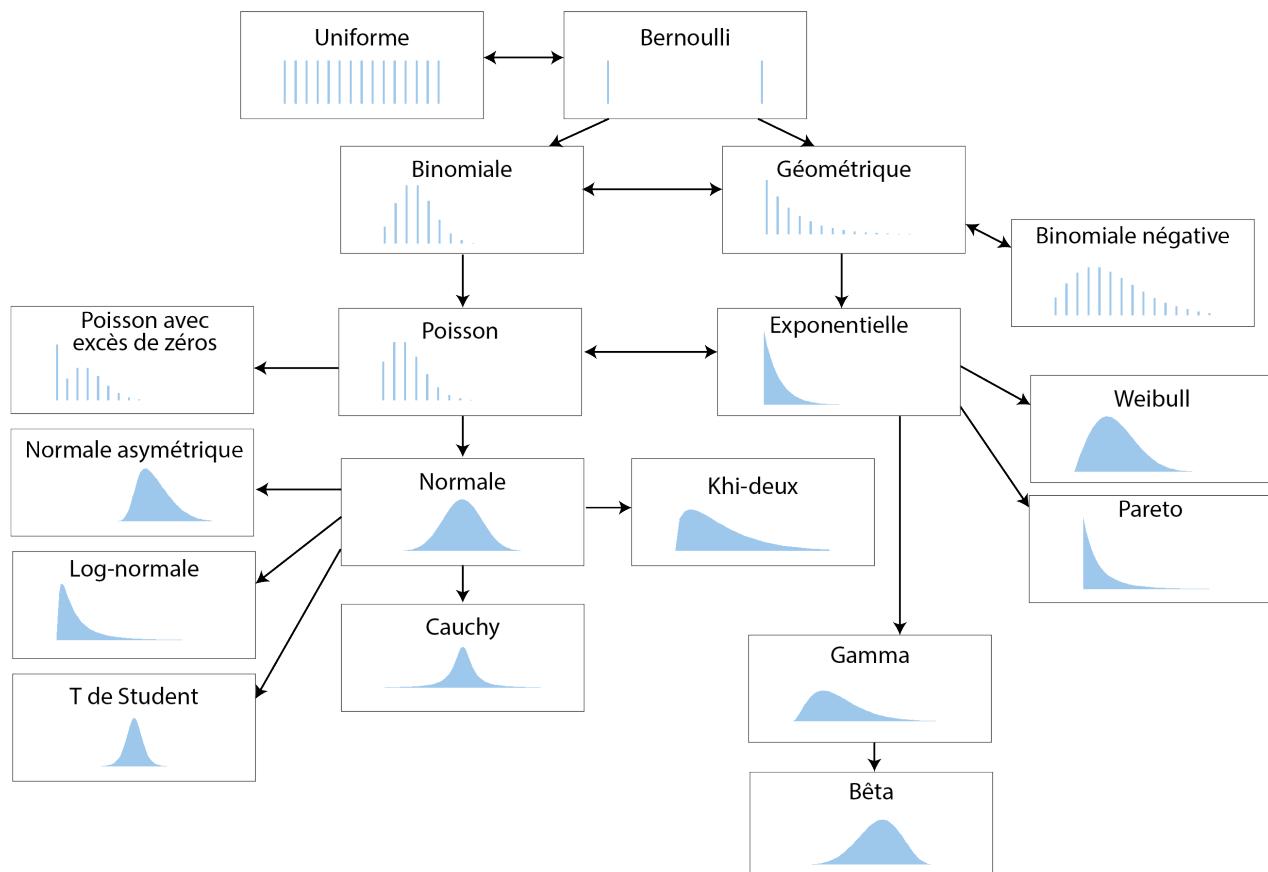


FIG. 2.7 : Dix-huit distributions essentielles, figure inspirée de Sean (2018)

⁴ <http://www.math.wm.edu/~leemis/chart/UDR/UDR.html>

2.4.3.1 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 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 est la modélisation de la survie d'un ou d'une cycliste (1 pour survie, 0 pour décès) lors d'une collision avec un véhicule motorisé, selon une vitesse donnée.

2.4.3.3 Distribution binomiale

La distribution binomiale est utilisée pour caractériser la somme de variables aléatoires (expériences) suivant chacune une distribution de Bernoulli. Un exemple simple est l'accumulation des lancers d'une pièce de monnaie. Si nous comptons le nombre de fois où nous obtenons 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 (figure 2.8). Pour reprendre l'exemple précédent concernant les accidents de la route, une distribution binomiale permettrait de représenter la distribution du nombre de cyclistes ayant survécu sur dix personnes à vélo impliquées dans un accident avec une voiture à une intersection.

2.4.3.4 Distribution géométrique

La distribution géométrique permet de représenter le nombre de tirages qu'il faut faire 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 (figure 2.9). Cette distribution est notamment utilisée en marketing pour modéliser le nombre d'appels nécessaires avant de réussir une vente.

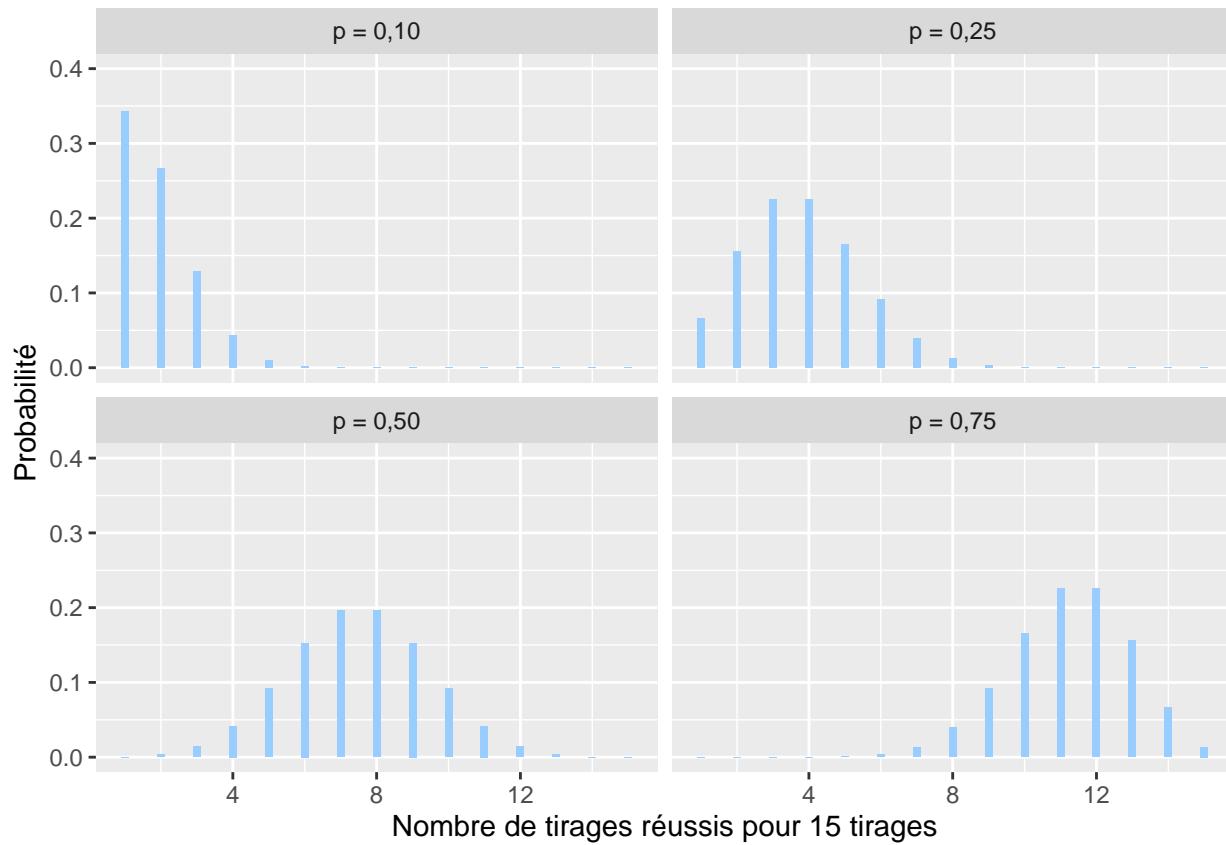


FIG. 2.8 : Distribution binomiale

2.4.3.5 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 (figure 2.10). Cette distribution pourrait être utilisée pour modéliser le nombre de questionnaires x à envoyer pour une enquête pour obtenir au moins n réponses, sachant que la probabilité d'une réponse est p .

2.4.3.6 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 est possible de noter chaque rencontre entre une voiture et un ou une cycliste et de considérer que leur collision est une « réussite» (0 : pas d'accidents, 1 : accident). Cependant, ce type de données est fastidieux à collecter comparativement au simple comptage des accidents. La distribution de Poisson a une fonction de densité avec un seul paramètre généralement noté λ (lambda) et est décrite par la formule suivante :

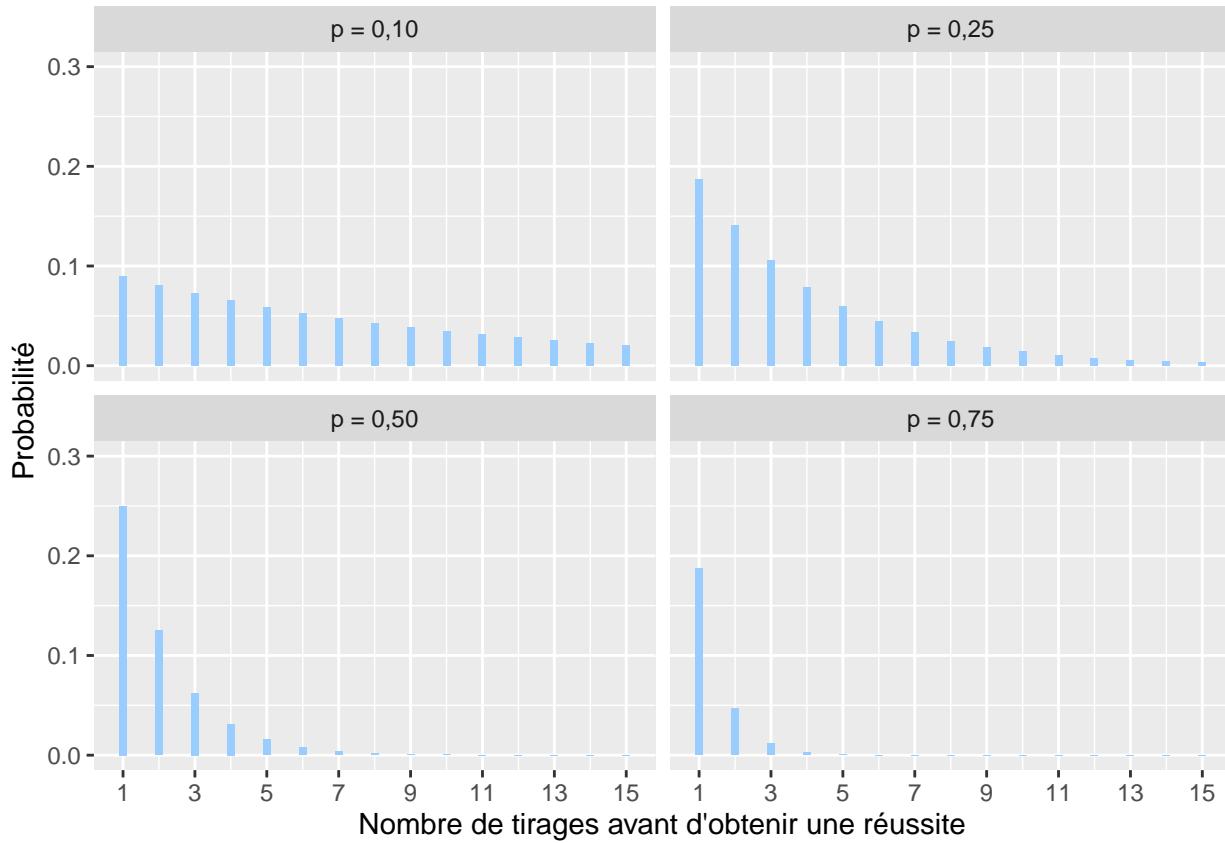


FIG. 2.9 : Distribution géométrique

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

avec x le nombre de cas, $f(x)$ la probabilité d'obtenir x sachant λ . λ peut être vu 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 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. Nous lui préfèrons alors une autre distribution : la distribution de Poisson avec excès de zéros (ou distribution de Pólya) qui inclut un paramètre contrôlant la forte présence de zéros. Sa fonction de densité est la suivante :

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

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

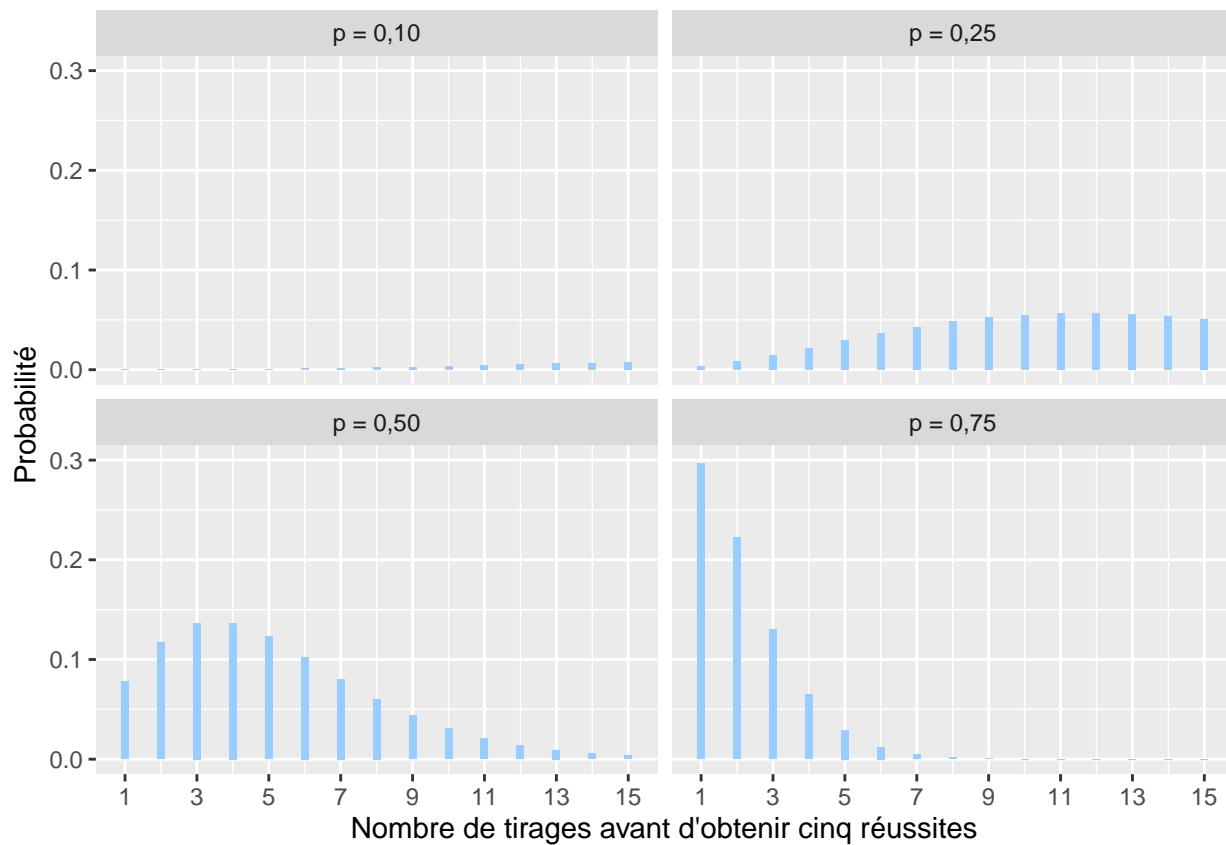


FIG. 2.10 : Distribution binomiale négative

Poisson proprement dite (aussi appelé vrai zéro) ou alors par le processus générant les zéros excédentaires dans le jeu de données, capturé par la probabilité p (faux zéro). p est donc le paramètre contrôlant la probabilité d'obtenir un zéro, indépendamment du phénomène étudié.

2.4.3.8 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. Selon la formule consacrée, cette distribution résulte de la superposition d'un très grand nombre de petits effets fortuits indépendants. C'est ce qu'exprime formellement le théorème central limite qui montre que la somme d'un grand nombre de variables aléatoires 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. Cela s'explique par le fait qu'en moyenne, chaque répétition de l'expérience produit le même résultat, mais qu'un ensemble de petits facteurs aléatoires viennent ajouter de la variabilité dans les données collectées. Prenons un exemple concret : si nous plantons une centaine d'arbres simultanément dans un parc avec un degré d'ensoleillement identique et que nous leur apportons 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 leur taille finale. Cette dernière est cependant davantage affectée par des paramètres majeurs (comme l'espèce, l'ensoleillement, l'arrosage, etc.), et est donc centrée autour d'une moyenne. La fonction de densité de la distribution normale est la suivante :

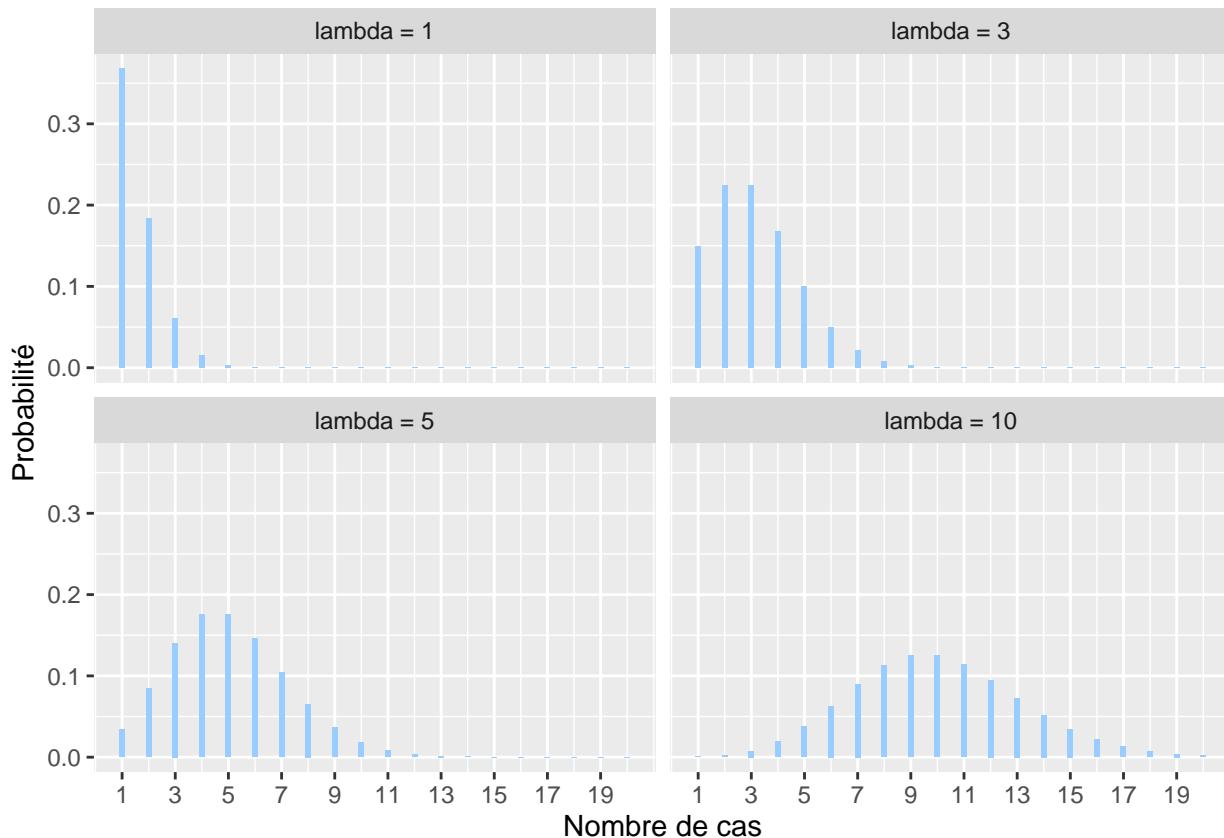


FIG. 2.11 : Distribution de Poisson

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

avec x une valeur dont nous souhaitons 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). Cette fonction suit une courbe normale ayant 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. Ces différentes égalités sont vraies **quelles que soient les valeurs de la moyenne et de l'écart-type**. Notez ici que lorsque $\mu = 0$ et $\sigma = 1$, nous obtenons la loi normale générale (ou centrée réduite) (section 2.5.5.2).

2.4.3.9 Distribution gaussienne asymétrique

La distribution normale asymétrique (*skew-normal*) est une extension de la distribution gaussienne permettant de lever la contrainte de symétrie de la simple distribution gaussienne. Son espace d'échantillonnage est donc $]-\infty; +\infty[$. Sa fonction de densité est la suivante :

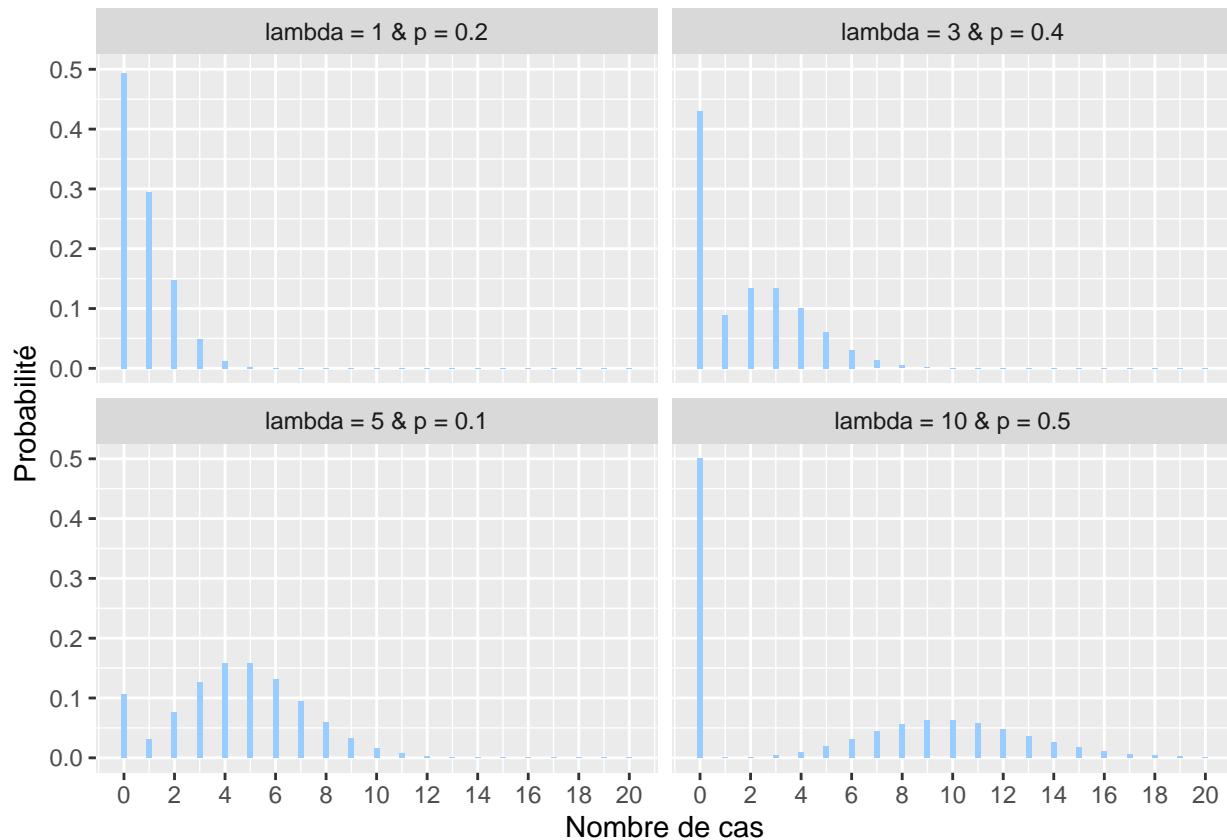


FIG. 2.12 : Distribution de Poisson avec excès de zéros

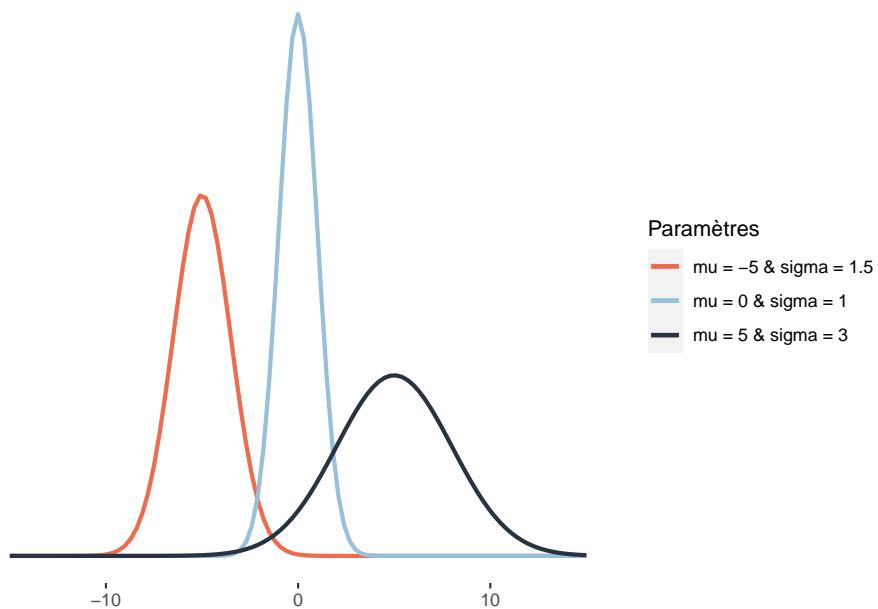


FIG. 2.13 : Distribution gaussienne

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

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 normale asymétrique est une distribution normale ordinaire. Ce type de distribution est très utile lorsque nous souhaitons modéliser une variable pour laquelle nous savons 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.

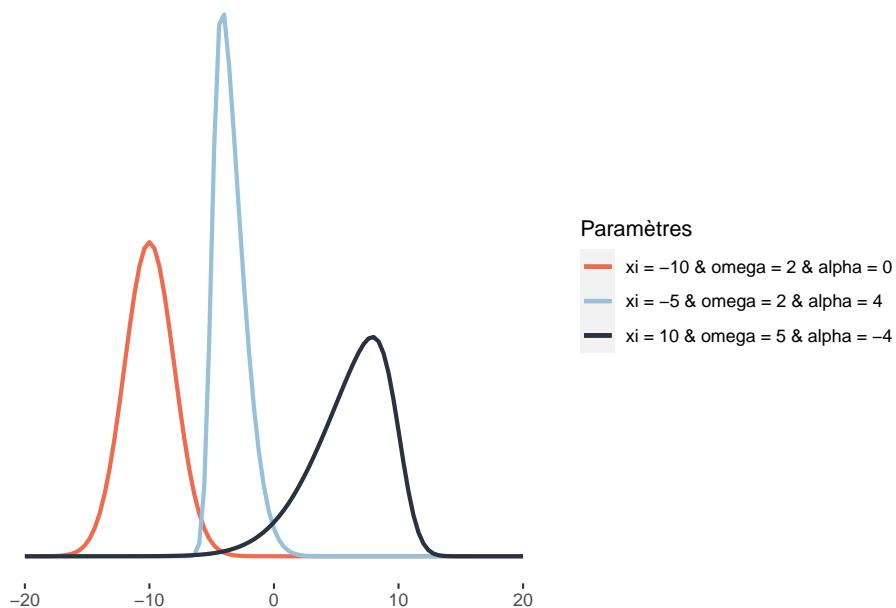


FIG. 2.14 : Distribution gaussienne asymétrique

2.4.3.10 Distribution log-normale

Au même titre que la distribution normale asymétrique, la distribution log-normale 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-normale 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-normale peut 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).

Plus spécifiquement, la distribution log-normale est une transformation de la distribution normale.

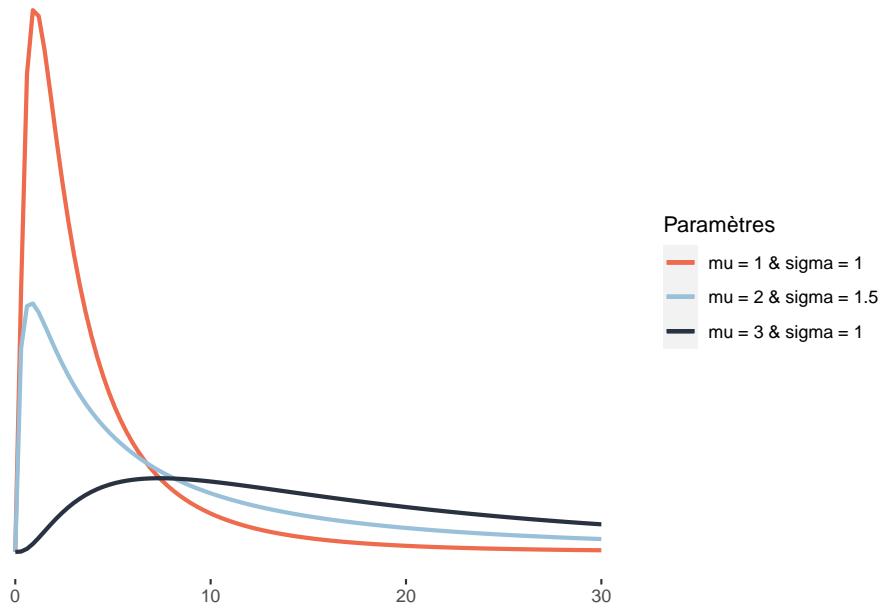


FIG. 2.15 : Distribution log-gaussienne

Comme son nom l'indique, elle permet de décrire le logarithme d'une variable aléatoire suivant une distribution normale.

2.4.3.11 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 se distingue de la distribution normale principalement par le rôle que joue son troisième paramètre, ν : le nombre de degrés de liberté, contrôlant le poids des queues de la distribution. Une petite valeur de ν signifie que la distribution a des « queues plus lourdes » (*heavy tails* en anglais). Entendez par-là que les valeurs extrêmes ont une plus grande probabilité d'occurrence :

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

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 degrés de liberté. Plus ν est grand, plus la distribution de Student tend vers une distribution normale. Ici, la lettre grecque Γ représente la fonction mathématique gamma (à ne pas confondre avec la distribution Gamma). Un exemple d'application en études urbaines est l'exposition au bruit environnemental de cyclistes. Cette distribution s'approche 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.), plus souvent que ce que prévoit une distribution normale, justifiant le choix d'une distribution de Student.

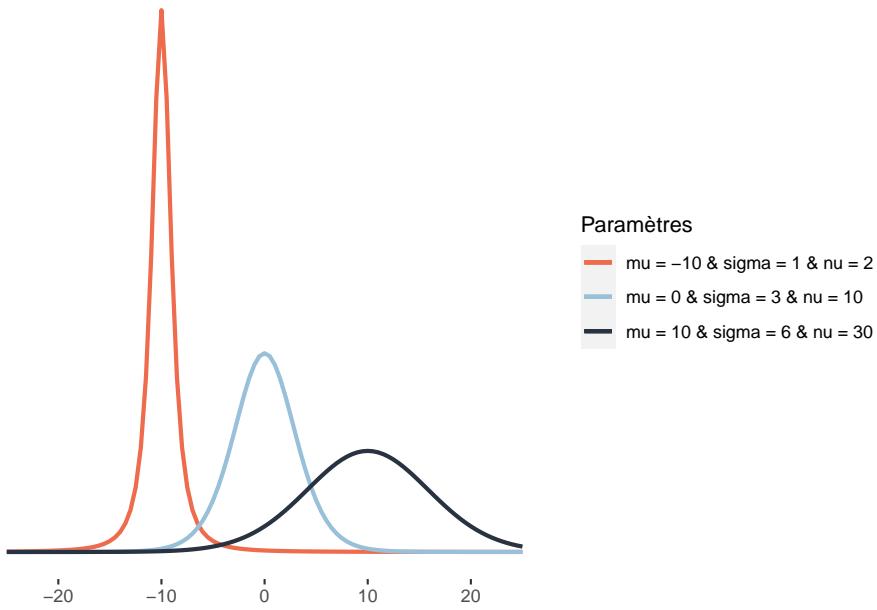


FIG. 2.16 : Distribution de Student

2.4.3.12 Distribution de Cauchy

La distribution de Cauchy est également une distribution symétrique définie sur l'intervalle $]-\infty; +\infty[$. Elle a comme particularité d'être plus aplatie que la distribution de Student (d'avoir des queues potentiellement plus lourdes). Elle est notamment utilisée pour modéliser des phénomènes extrêmes comme les précipitations maximales annuelles, les niveaux d'inondations maximaux annuels ou les seuils critiques de perte 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 Distribution du khi-deux

La distribution du khi-deux est utilisée dans de nombreux tests statistiques. Par exemple, le test du khi-deux de Pearson est utilisé pour comparer les écarts au carré entre des fréquences attendues et observées de deux variables qualitatives. La distribution du khi-deux décrit plus généralement la somme 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 khi-deux, 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 :

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

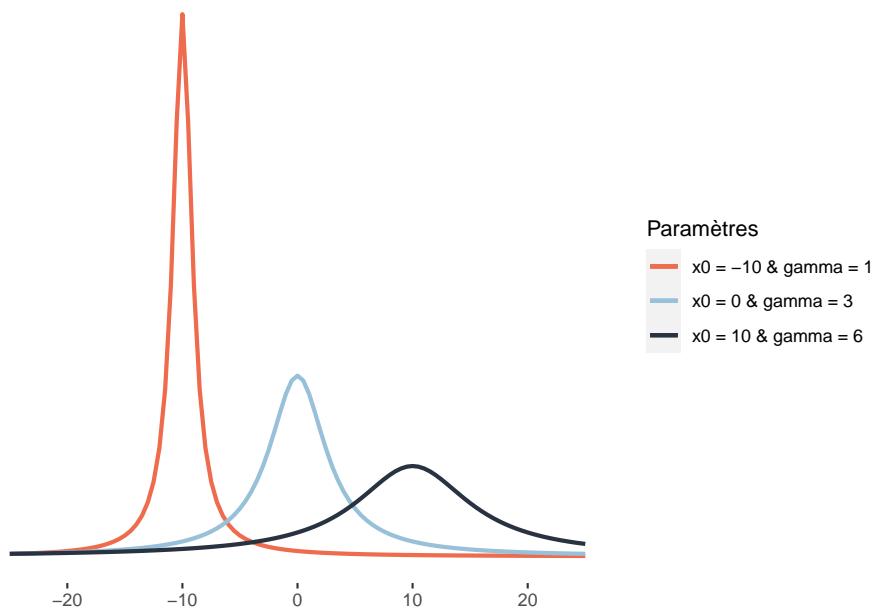


FIG. 2.17 : Distribution de Cauchy

La distribution du khi-deux n'a qu'un paramètre k , représentant donc le nombre de variables mises au carré et dont nous faisons la somme pour obtenir la distribution du khi-deux.

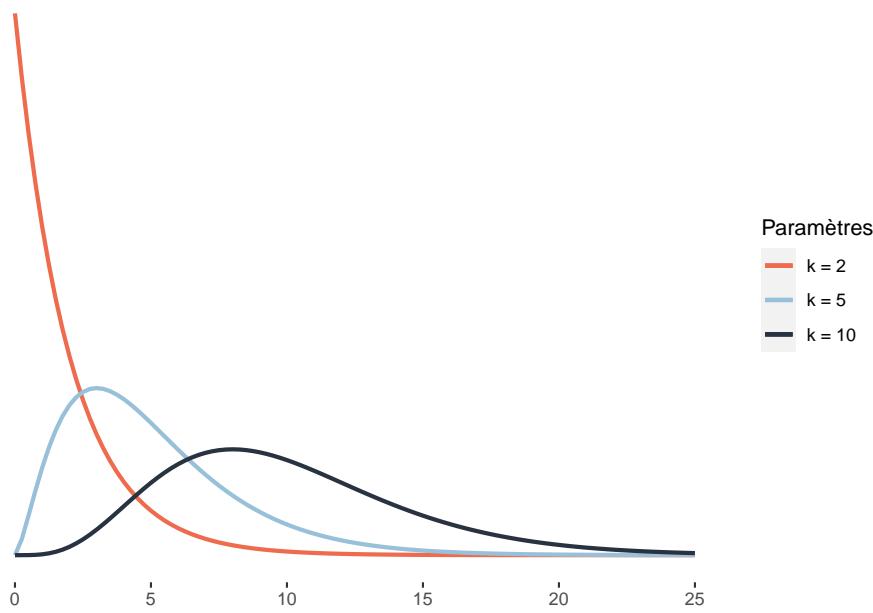


FIG. 2.18 : Distribution du khi-deux

2.4.3.14 Distribution exponentielle

La distribution exponentielle est une version continue de la distribution géométrique. Pour cette dernière, nous nous intéressons 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 sert à modéliser le temps d'attente nécessaire pour qu'un évènement se produise. Il peut aussi s'agir d'une force que nous appliquons jusqu'à ce qu'un matériau cède. Cette distribution est donc définie sur l'intervalle $[0; +\infty[$ et a pour fonction de densité :

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

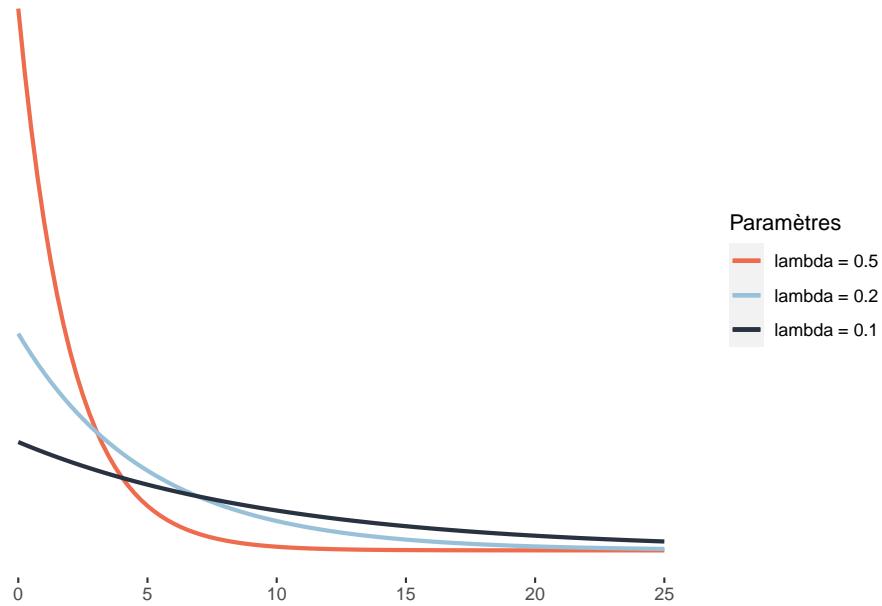


FIG. 2.19 : Distribution exponentielle

La distribution exponentielle est conceptuellement proche de la distribution de Poisson. La distribution de Poisson régit le nombre des événements qui surviennent au cours d'un laps de temps donné. La distribution exponentielle peut servir à modéliser le temps qui s'écoule entre deux événements.

2.4.3.15 Distribution Gamma

La distribution Gamma peut être vue comme la généralisation d'un grand nombre de distributions. Ainsi, la distribution exponentielle et du khi-deux peuvent être vues comme des cas particuliers de la distribution 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 est 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, de la taille des réclamations d'assurance, des quantités de précipitations, etc.

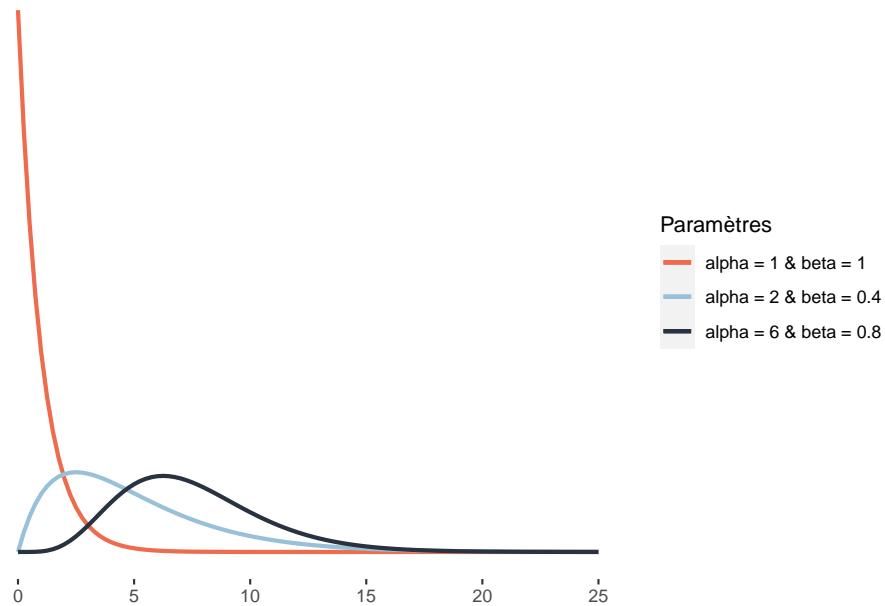


FIG. 2.20 : Distribution Gamma

2.4.3.16 Distribution bêta

La distribution bêta est définie sur l'intervalle $[0; 1]$, elle est donc énormément utilisée pour modéliser des variables étant des proportions ou des probabilités.

La distribution bêta a été élaborée pour modéliser la superposition d'un très grand nombre de petits effets fortuits qui ne sont pas indépendants et notamment pour étudier l'effet de la réalisation d'un événement aléatoire sur la probabilité des tirages subséquents. Elle a aussi une utilité pratique en statistique, car elle peut être combinée avec d'autres distributions (distribution bêta-binomiale, bêta-negative-binomiale, 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. Cela 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 bêta. Un exemple concret est, par exemple, la fraction du temps effectué à pied dans un déplacement multimodal. La distribution de bêta 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 correspond à la fonction mathématique Beta : ne pas la confondre avec la distribution Beta et le paramètre Beta (β) de cette même distribution.

2.4.3.17 Distribution de Weibull

La distribution de Weibull est directement liée à la distribution exponentielle, cette dernière étant en fait un cas particulier de distribution Weibull. Elle sert donc souvent à modéliser 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 :

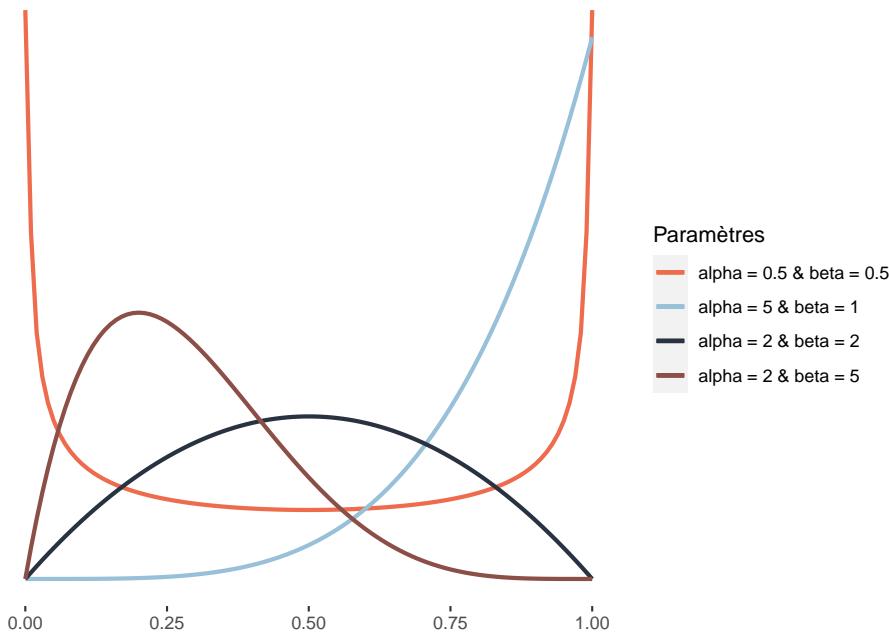


FIG. 2.21 : Distribution bêta

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

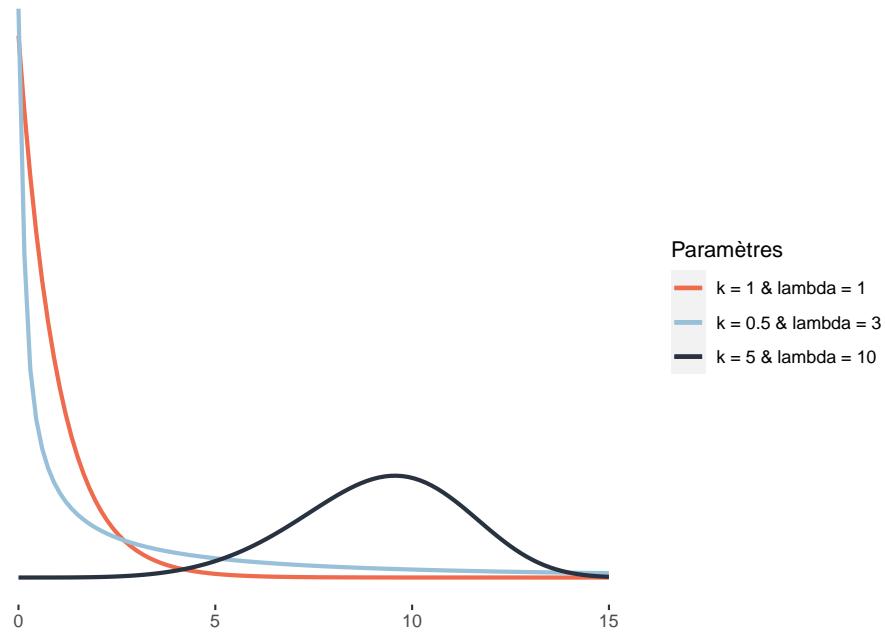
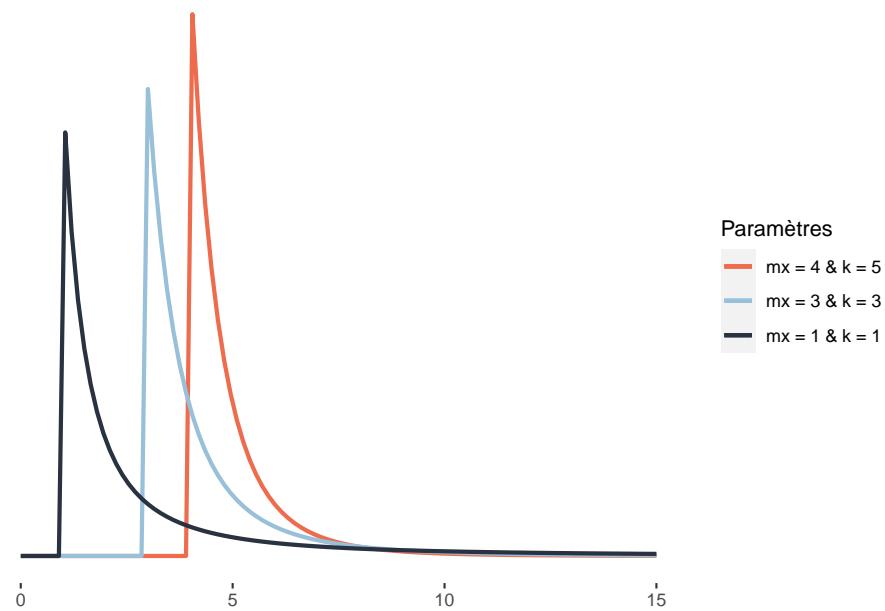
λ est le paramètre de dispersion (analogue à celui d'une distribution exponentielle classique) et k le paramètre de forme. Pour bien comprendre le rôle de k , prenons un exemple : la propagation d'un champignon d'un arbre à son voisin. Si $k < 1$, le risque instantané que l'évènement modélisé se produise diminue avec le temps (en d'autres termes, plus le temps passe, plus petite devient la probabilité d'être contaminé). Si $k = 1$, alors le risque instantané que l'évènement se produise reste identique dans le temps (la loi de Weibull se résume alors à une loi exponentielle). Si $k > 1$, alors le risque instantané que l'évènement se produise augmente avec le temps (la probabilité pour un arbre d'être contaminé s'il ne l'a pas déjà été — pas seulement le risque cumulé — augmente en fonction du temps). 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 Distribution Pareto

Cette distribution a été élaborée par Vilfredo Pareto pour donner une forme mathématique à ce qui porte aujourd'hui le nom de principe de Pareto et que nous exprimons souvent de manière imagée — dans une société donnée, 20 % des individus possèdent 80 % de la richesse —, mais qui est plus justement exprimée en écrivant que, de manière générale, dans toute société, la plus grande partie du capital est détenue par une petite fraction de la population. 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.

**FIG. 2.22 :** Distribution de Weibull**FIG. 2.23 :** Distribution de Pareto

Au-delà de la question de la répartition de la richesse, la distribution de Pareto peut également être utilisée pour décrire la répartition de la taille des villes (Reed 2002), la popularité des hommes sur Tinder⁵ ou la taille des fichiers échangés sur Internet (Reed et Jorgensen 2004). Pour ces trois exemples, nous avons les situations suivantes : 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.

La loi de Pareto est liée à la loi exponentielle. Si une variable aléatoire suit une loi de Pareto, le logarithme du quotient de cette variable et de son paramètre de localisation est une variable aléatoire qui suit une loi exponentielle.

2.4.3.19 Cas particuliers

Sachez également qu'il existe des distributions « plus exotiques » 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 distributions composées (*mixture distributions*), permettant de modéliser des phénomènes issus de la superposition de plusieurs distributions. Par exemple, la distribution de la taille de l'ensemble des êtres humains est en réalité une superposition de deux distributions gaussiennes, une pour chaque sexe, puisque ces deux 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 par une distribution gaussienne bivariée plutôt que deux distributions distinctes. Ce choix serait pertinent si nous présumons que ces deux variables sont corrélées plutôt qu'indépendantes.
- Les distributions censurées décrivant des variables pour lesquelles les données sont issues d'un tirage « censuré ». En d'autres termes, la variable étudiée varie sur une certaine étendue, mais du fait du processus de tirage (collecte des données), les valeurs au-delà de certaines limites sont censurées. Un bon exemple est 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 descendent plus bas que ce seuil, 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 (elles sont au-delà de la limite).
- Les distributions tronquées, souvent confondues avec les distributions censurées, décrivent des situations où des données au-delà d'une certaine limite sont impossibles à collecter et 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. Vous aurez certainement besoin de le relire cette section avant d'aborder le chapitre 8 portant sur les modèles linéaires généralisés (GLM). Wikipédia dispose d'informations très détaillées sur chaque distribution si vous avez besoin d'informations complémentaires.

⁵<https://medium.com/@worstonlinedater/tinder-experiments-ii-guys-unless-you-are-really-hot-you-are-probably-better-off-not-wasting-your-2ddf370a6e9a>

Pour un tour d'horizon plus exhaustif des distributions, vous pouvez aussi faire un tour sur les projets ProbOnto⁶ et *the ultimate probability distribution explorer*⁷.

2.5 Statistiques descriptives sur des variables quantitatives

2.5.1 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 (prononcé *mu*) pour des données pour une population et \bar{x} (prononcé *x barre*) pour un échantillon.
- 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 personne à vélo par un personne conduisant 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 295 099 \$; la moitié de ces 8685 ménages avaient un revenu inférieur à 100 153 \$ 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 important, 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 (58 594 \$ versus 50 318 \$) souligne 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.), nous privilégions habituellement l'utilisation des revenus médians.

2.5.2 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.
- 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 %).

⁶<https://sites.google.com/site/probonto/screenshots>

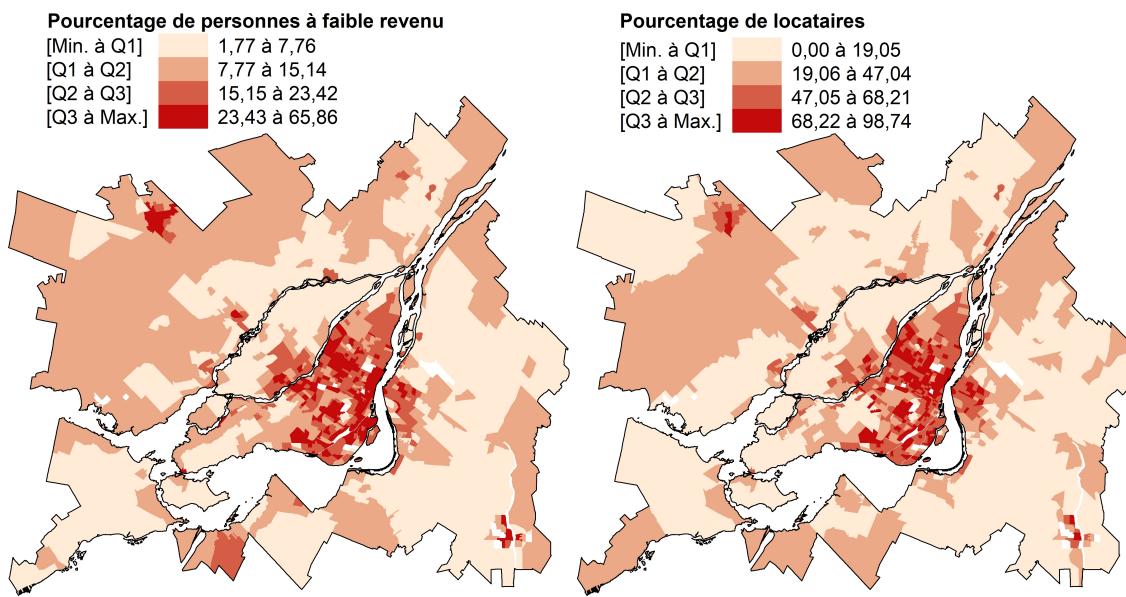
⁷<https://blog.wolfram.com/2013/02/01/the-ultimate-univariate-probability-distribution-explorer/>

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 **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 dix classes (déciles). Avec les quartiles, les bornes des classes qui comprennent chacune 25 % des unités spatiales sont 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 de repérer, en un coup d'œil, à quelle tranche de 25 % ou de 20 % des données appartiennent 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 et 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 et 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 participantes et participants ont été exposés à des niveaux de bruit soit inférieurs à 69,1 dB(A) (premier quartile), soit supérieurs à 74 dB(A) (troisième quartile). Quant à l'analyse des centiles, elle permet de constater que durant 5 % et 10 % du temps, les participantes et participants étaient exposés à des niveaux de bruit très élevés, dépassant 75 dB(A) ($C_{90} = 76$ et $C_{99} = 77,2$).

2.5.3 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 est grand, plus la boîte est allongée (figure 2.25)
- **La variance** est la somme des déviations à la moyenne au carré (numérateur) divisée par le nombre

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

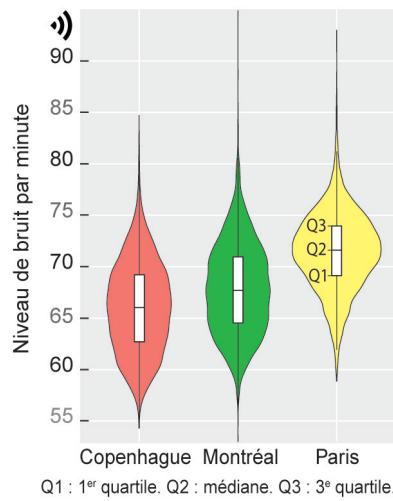


FIG. 2.25 : Graphique en violon, boîte à moustaches et intervalle interquartile

d’observations pour une population (σ^2) ou divisée par le nombre d’observations moins une (s^2) pour un échantillon (équation (2.20)). Puisque les déviations à la moyenne sont mises au carré, la valeur de la variance (tout comme celle de l’écart-type) est 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 (équation (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 la même unité 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 écarts-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 est important, plus l’écart entre ces deux mesures de dispersion pour une population et un échantillon est 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 (équation (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, nous utilisons le coefficient de variation : une variable est plus dispersée qu'une autre si la valeur de son CV est plus élevée. Certaines personnes préfèrent 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 égal à la valeur de la deuxième observation (soit 25), la **médiane** à celle de la quatrième (30), le **troisième quartile** à celle de la sixième (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 (équation (2.20)). En divisant cette somme par le nombre d'observations, nous obtenons 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 (équation (2.21)), soit $\sigma = \sqrt{33,5} = 5,79$ et $s = \sqrt{38,29} = 6,19$.
- Finalement, les valeurs des coefficients de variation (équation (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 , où B étant simplement A multipliée par 10. Pour A , les valeurs de la moyenne, de l'étendue et de l'intervalle interquartile sont respectivement 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, étant donné que le coefficient de variation (CV) est le rapport de l'écart-type avec la moyenne, il a la même valeur pour A et B , ce qui démontre que le 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

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		

contre 25,33) et le coefficient de variation (0,19 contre 0,65). Par contre, comme l'intervalle interquartile est calculé sur 50 % des observations centrales ($Q_3 - Q_1$), il n'est pas affecté par cette valeur extrême.

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

Statistique	Unité de mesure	Valeurs extrêmes
Moyenne	X	X
Étendue	X	X
Intervalle interquartile	X	
Variance	X	X
Écart-type	X	X
Coefficient de variation		X

2.5.4 Paramètres de forme

2.5.4.1 Vérification de la normalité d'une variable quantitative

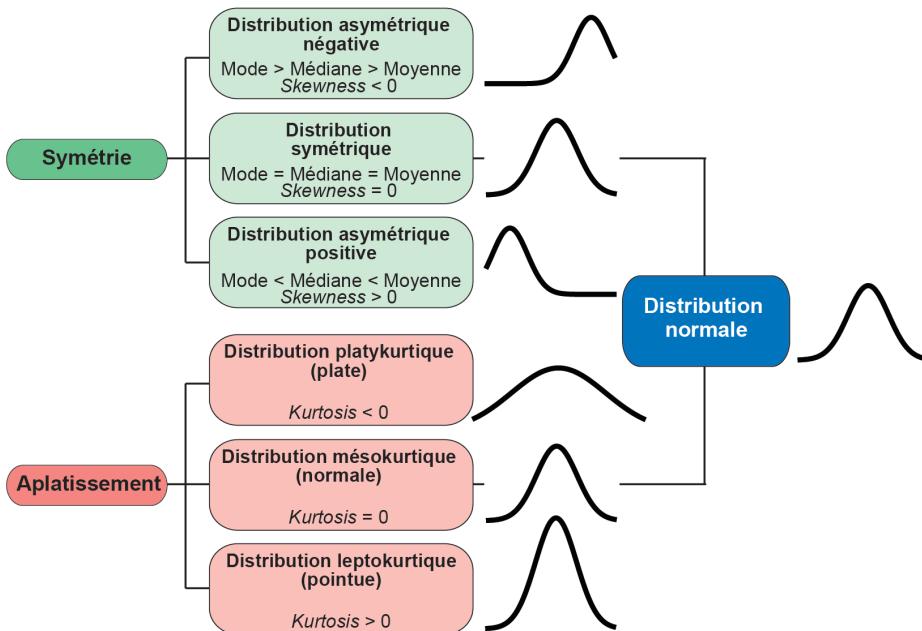


De nombreuses méthodes statistiques qui sont abordées dans les chapitres suivants – entre autres, la corrélation de Pearson, les test t et l'analyse de variance, les régressions simple et multiple – requièrent que la variable quantitative suive une **distribution normale** (nommée aussi **distribution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

Dans cette sous-section, nous décrivons 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 et diagramme quantile-quantile), les tests de normalité (tests de Shapiro-Wilk, de Kolmogorov-Smirnov, de Lilliefors, d'Anderson-Darling et de Jarque-Bera).

Il est vivement recommandé de réaliser les trois démarches !

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

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

2.5.4.1.1 Vérification de 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 est alors égale

à la médiane et au mode (nous utilisons uniquement le mode pour une variable discrète et non pour une variable continue). Pour évaluer l'asymétrie, nous utilisons habituellement le coefficient d'asymétrie (*skewness* en anglais).

Sachez toutefois qu'il existe trois façons (formules) pour le calculer (Joanes et Gill 1998) : g_1 est la formule classique (équation (2.23)), disponible dans R avec la fonction *skewness* du package *moments*, G_1 est une version ajustée (équation (2.24)), utilisée dans les logiciels SAS et SPSS notamment) et b_1 est une autre version ajustée (équation (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 et Gill 1998). Quelle que soit la formule utilisée, le coefficient d'asymétrie 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. Nous employons 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 est égale à la moyenne). Pour une variable discrète, les valeurs du mode, de la moyenne et de la médiane sont 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 et la moyenne est supérieure à la médiane. En sciences sociales, les variables de revenu (taux 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 la 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.

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

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

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

Pour évaluer l'aplatissement d'une distribution, nous utilisons le coefficient d'aplatissement (*kurtosis* en anglais). Là encore, il existe trois formules pour le calculer (équation (2.26), (2.27), (2.28)) qui renvoient des valeurs très semblables pour de grands échantillons (Joanes et 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.

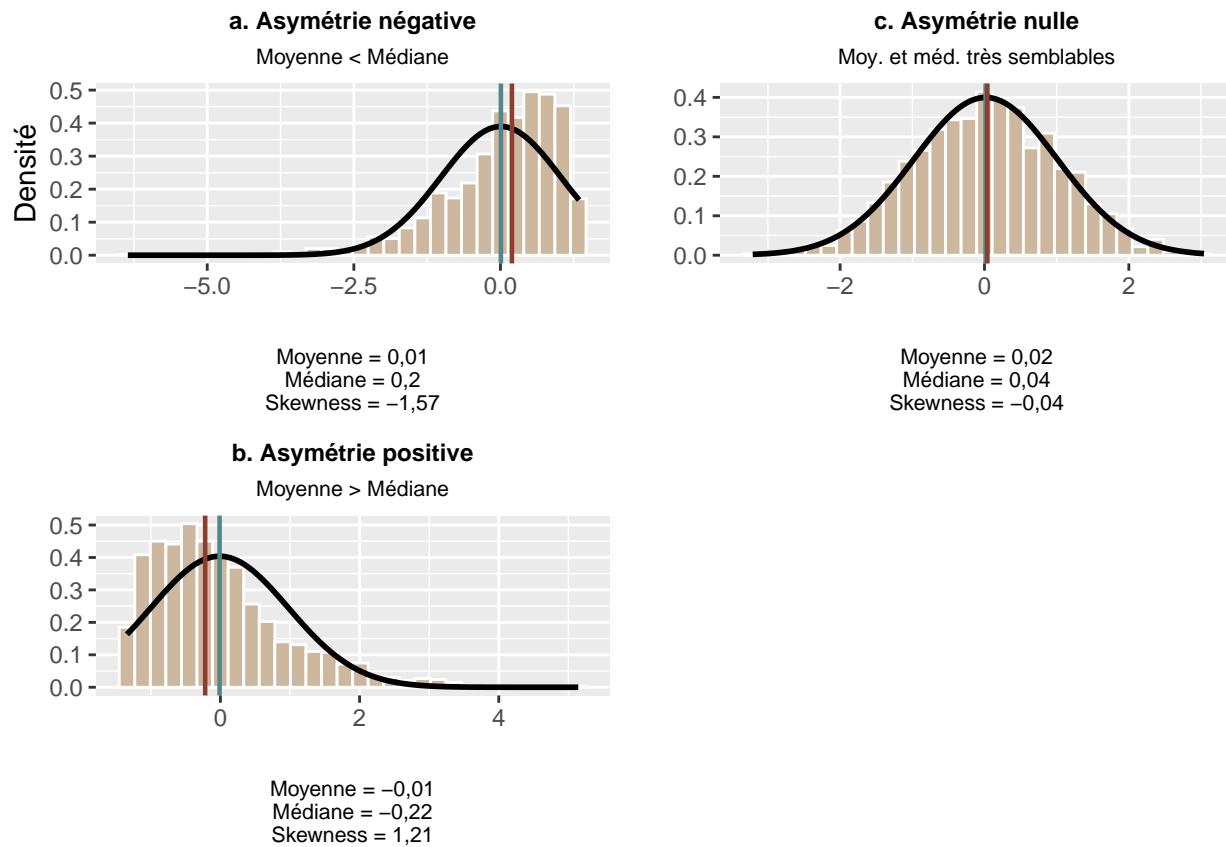


FIG. 2.27 : Asymétrie d'une distribution

- 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.

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

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

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



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 . Nous parlons alors de *kurtosis* normalisé (*excess kurtosis* en anglais). Pour une distribution normale, il prend la valeur de 0, comparativement à la valeur de 3 pour un *kurtosis* non normalisé. Par conséquent, avant de calculer le *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.

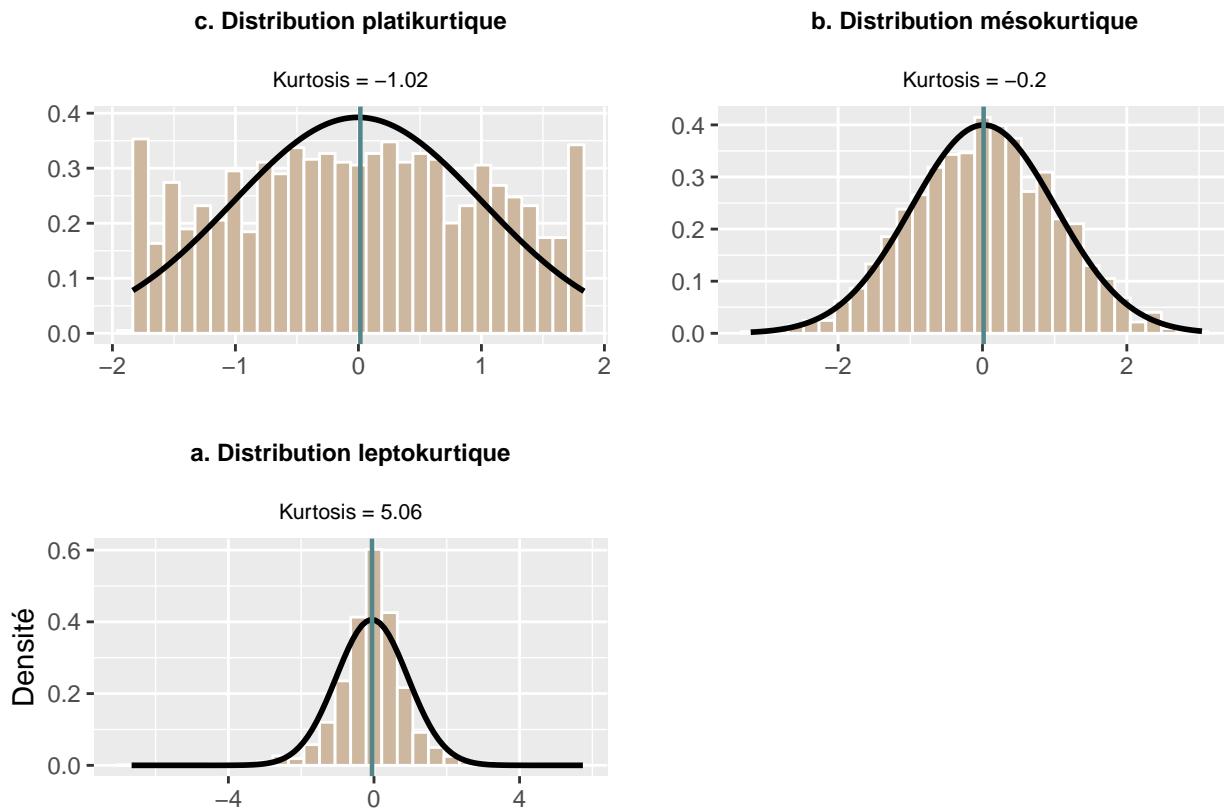


FIG. 2.28 : Applatissement d'une distribution

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

## [1] -0.141

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

## [1] 2.863
```

2.5.4.1.2 Vérification de 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 comme qu'illusté 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 95 % de la distribution normale (en marron sur la figure).

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

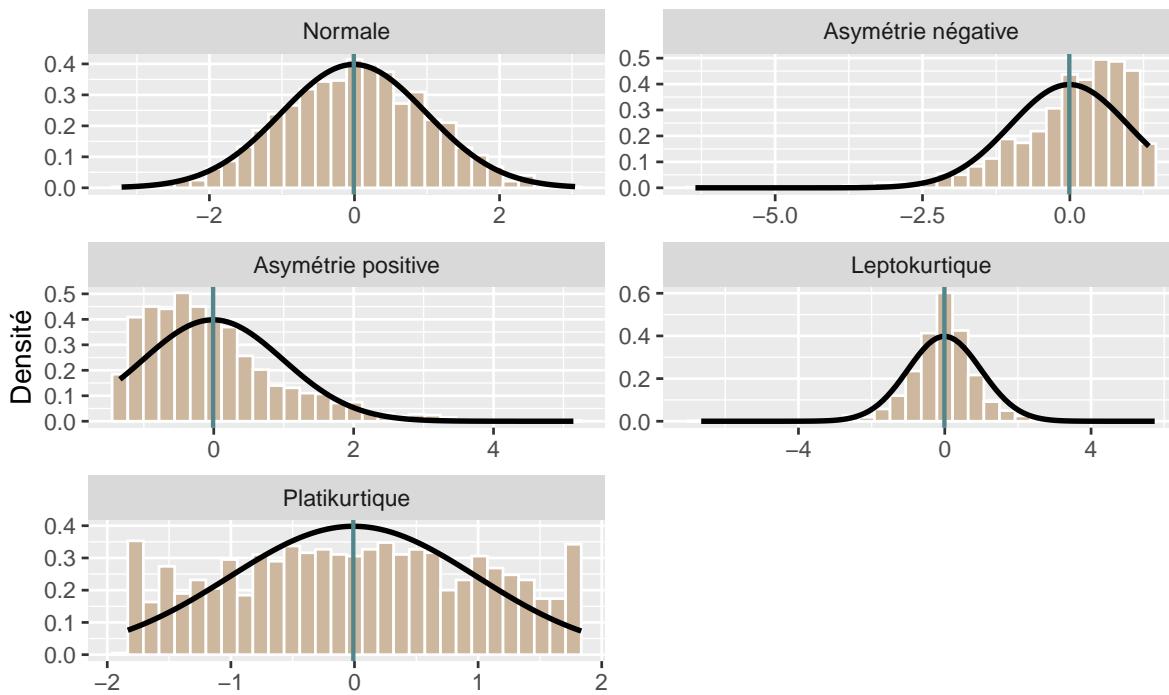


FIG. 2.29 : Histogrammes et courbe normale

2.5.4.1.3 Vérification de 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), de **Lilliefors** (LF), de **Shapiro-Wilk** (SW), d'**Anderson-Darling** et de **Jarque-Bera** (JB). Sachez toutefois qu'il y en a d'autres non discutés ici (tests de d'Agostino-Pearson, de Cramer-von Mises, de Ryan-Joiner, de 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 auteur(e)s 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 Wah 2011 ; Yap et Sim 2011). Selon Razali et al. (2011), le meilleur test semble être celui de Shapiro-Wilk, puis ceux d'Anderson-Darling, de Lilliefors et de 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 inférieure ou égale au seuil alpha choisi (habituellement $\alpha = 0,05$), la distribution est anormale. À l'inverse, si $p > 0,05$, la distribution est normale.

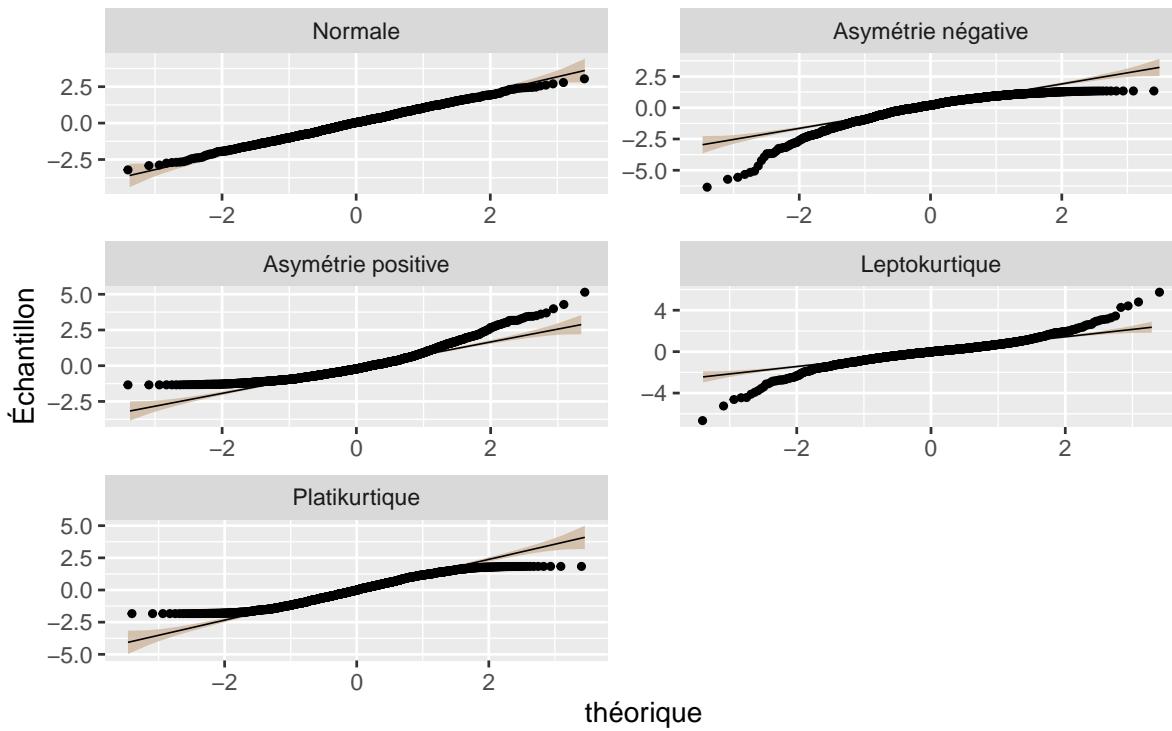


FIG. 2.30 : Diagrammes quantile-quantile

Dans le tableau 2.7 sont reportées les valeurs des différents tests pour les cinq types de distribution générés à la figure 2.29. Sans surprise, pour l'ensemble des tests, la valeur de p est inférieure à 0,05 pour la distribution normale.



La plupart des auteurs s'entendent sur le fait que ces tests sont très restrictifs : plus la taille de votre échantillon 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 de grands échantillons ($n > 200$).

TAB. 2.6 : 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; 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, ce test 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 les équations 2.23 et 2.26).	JarqueBeraTest du package DescTools

TAB. 2.7 : Tests de normalité pour différentes distributions

	Normale	Asymétrie négative	Asymétrie positive	Leptokurtique	Platikurtique
Skewness	0,004	1,360	-1,094	0,142	-0,058
Kurtosis	-0,267	2,438	1,034	4,356	-1,076
Kolmogorov-Smirnov (KS)	0,028	0,104	0,110	0,096	0,062
Lilliefors (LF)	0,028	0,104	0,110	0,096	0,062
Shapiro-Wilk (SW)	0,996	0,901	0,918	0,930	0,967
Anderson-Darling (AD)	0,550	11,678	10,656	8,584	3,817
Jarque-Bera (JB)	1,136	374,558	170,249	915,688	18,960
KS (valeur p)	0,837	0,000	0,000	0,000	0,041
LF (valeur p)	0,461	0,000	0,000	0,000	0,000
SW (valeur p)	0,217	0,000	0,000	0,000	0,000
AD (valeur p)	0,156	0,000	0,000	0,000	0,000
JB (valeur p)	0,567	0,000	0,000	0,000	0,000

2.5.4.2 Tests pour d'autres formes de distribution

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 peut 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 en tenant compte 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 permettent d'adopter une forme la plus proche possible des données. Nous appelons 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 `dnomadeladistribution` (`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 tentons 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 , la 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 la moyenne et l'écart-type de la variable, 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
```

Nous observons ainsi que les valeurs 7 et 8 sont très probables selon le modèle alors que les valeurs 4 et 10 sont très improbables.

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 , nous calculons 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 infinitésimement petits et donc à complexifier grandement le calcul. Nous préférons 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 ajoutant +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 des données ouvertes de la Ville de Toronto⁸. 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 (2 heures 30) n'est tout simplement pas passé (personne ne risque d'attendre 2 heures 30 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) +
  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 (en minutes)',
       y = '')
```

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

- La distribution Gamma, strictement positive et asymétrique, est aussi une généralisation de la distribution exponentielle utilisée pour modéliser des temps d'attente. Pour des raisons similaires, nous pouvons aussi retenir la distribution de Weibull et la distribution log-normale. Nous écartons ici la distribution normale asymétrique 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 représentons-les graphiquement pour éliminer les moins bons candidats. Nous utilisons également le package `actuar` pour la fonction de densité de Pareto (`dpareto`).

⁸<https://open.toronto.ca/catalogue/?search=bus%20delay&sort=score%20desc>

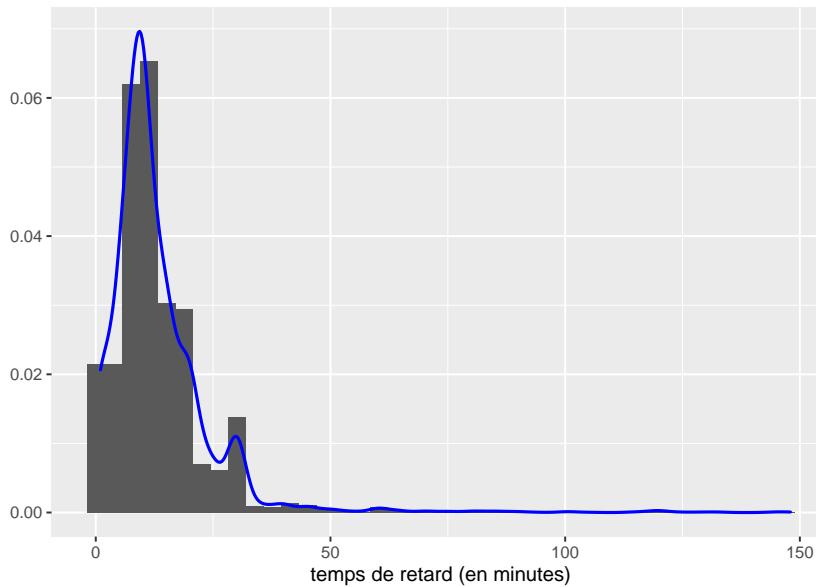


FIG. 2.31 : Distribution empirique des temps de retard des bus à Toronto en janvier 2019

```

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$estimate)) +
  labs(x = 'temps de retard (en minutes)',
       y = '',
       subtitle = "Modèle Gamma")
plot2 <- ggplot(data = data_trt_bus) +
  geom_histogram(aes(x=Min.Delay, y = ..density..), bins = 40) +
  stat_function(fun = dweibull, color = 'red', size = 0.8,
                args = as.list(model_weibull$estimate)) +
  labs(x = 'temps de retard (en minutes)',
       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 (en minutes)',
       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 (en minutes)',
       y = '',
       subtitle = "Modèle Pareto")
ggarrange(plotlist = list(plot1, plot2, plot3, plot4),
          ncol = 2, nrow = 2)

```

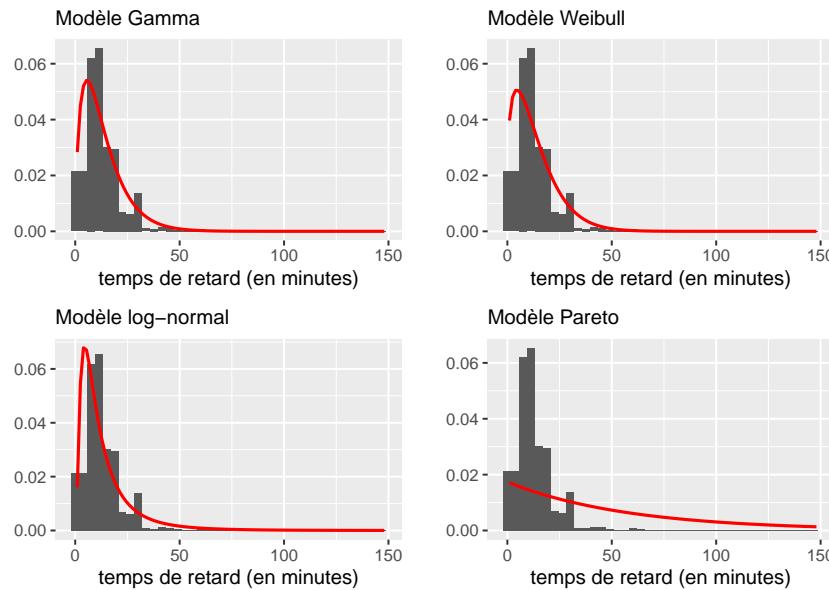


FIG. 2.32 : Comparaison des distributions ajustées aux données de retard des bus

Visuellement, nous constatons que la distribution de Pareto est un mauvais choix. Pour les trois autres distributions, la comparaison des valeurs de *loglikelihood* s'impose (tableau 2.8).

```

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("Distribution", "LogLikelihood"),
           caption = 'Comparaison des LogLikelihood des trois distributions',
           align = c("l", "r"))
)

```

TAB. 2.8 : Comparaison des LogLikelihood des trois distributions

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

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

Comme la valeur de p est inférieure à 0,05, nous ne pouvons 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 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
```

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 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 (2.33). Le jeu de données complet est disponible sur le site

des données ouvertes⁹ 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')
```

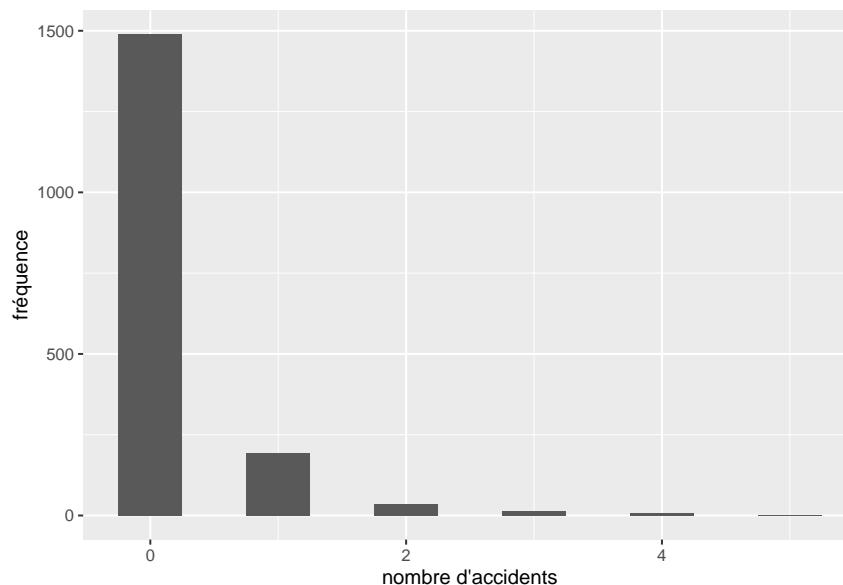


FIG. 2.33 : Distribution empirique du nombre d'accidents par intersection impliquant un ou une cycliste à Montréal en 2017 dans les quartiers centraux

Nous avons effectivement de nombreux zéros, alors essayons d'ajuster nos deux distributions à ce jeu de données. Dans la figure 2.34, 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
```

⁹<http://donnees.ville.montreal.qc.ca/dataset/collisions-routieres>

```

optim.method = "L-BFGS-B", # méthode d'optimisation recommandée dans la documentation
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),
                         y=dpois(0:10, model_poisson$estimate)
                         )
plot1 <- ggplot() +
  geom_bar(aes(x=nb_accident, weight = prop), width = 0.6, data = counts) +
  geom_bar(aes(x=x, weight = y), width = 0.15, data = dfpoisson, fill = "red") +
  scale_x_continuous(limits = c(-0.5,7), breaks = c(0:7)) +
  labs(subtitle = "Modèle de Poisson",
       x = "nombre d'accidents",
       y = "")
dfpoissonzi <- data.frame(x=c(0:10),
                           y=dZIP(0:10, model_poissonzi$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 de Poisson avec excès de zéro",
       x = "nombre d'accident",
       y = "")
ggarrange(plotlist = list(plot1,plot2), ncol = 2)

```

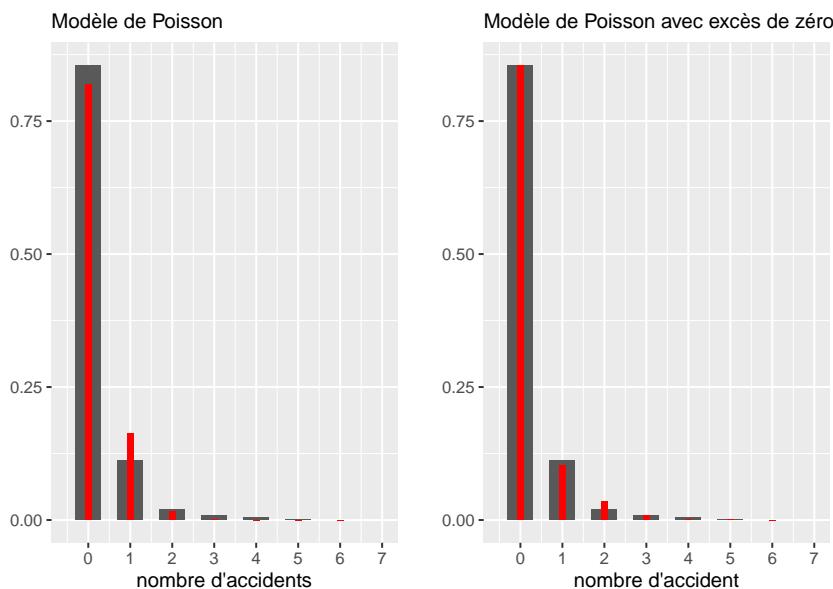


FIG. 2.34 : Ajustement des distributions de Poisson et Poisson avec excès de zéros

Visuellement, comme nous pouvons l'observer à la figure 2.34, 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

print(model_poissonzi$loglik)

## [1] -931.8778

#afficher les paramètres ajustés
model_poisson$estimate

##      lambda
## 0.1991963

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, nous devons 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 Transformation des variables

2.5.5.1 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 fournissent 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 auteur(e)s (notamment, les statisticiennes Barbara G. Tabachnick et Linda S. Fidell (2007, 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)$.



Pour une valeur égale ou inférieure à 0, nous ne pouvons 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 négatives ou égales à 0 :

- `sqrt(df$x - min(df$x+1))` 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 statisticien(ne)s, et les personnes utilisatrices débutantes ou avancées des méthodes quantitatives. Field et al. (2012, 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 de 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 15 000 \$, 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, Miles et Field 2012, 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 nous souhaitons utiliser, il n'est donc pas nécessaire de toujours se limiter à la distribution normale. Par exemple, dans les modèles de régression généralisés (GLM), nous pourrions indiquer que la 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 analysé 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 simplement pour 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 pas toujours le cas, parfois c'est pire ! Prenez soin de comparer les histogrammes, les valeurs de *skewness*, de *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 qui sont 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) (équation (2.29)). Par conséquent, nous parlons aussi de variable centrée réduite qui a comme propriétés intéressantes une moyenne égale à 0 et un écart-type égal 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 13) et les méthodes factorielles (chapitre 12).

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

- **La transformation en rang** 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 4.3.3) et certains tests non paramétriques (sections 6.1.2 et 6.2.2).
- **La transformation sur une échelle de 0 à 1** (ou de 0 à 100) qui consiste à soustraire à chaque observation la valeur minimale et à diviser le tout par l'étendue (équation (2.30)).

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

Pour un *DataFrame*, nommé df , comprenant une variable X , la syntaxe ci-dessous illustre comment obtenir quatre transformations (côte z , rang, 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)
# ou encore avec la fonction scale
df2$zX <- scale(df2$X, center = TRUE, scale = TRUE)

# Transformation en rang avec la fonction rank
df2$rx <- rank(df2$X)

# Transformation de 0 à 1 ou de 0 à 100
df2$x01 <- (df2$X-min(df2$X))/(max(df2$X)-min(df2$X))
df2$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'aplatissement
- `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, consultez 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(x="personnes à faible revenu (%)", y = "densité")+
  stat_function(fun = dnorm, args = list(mean = moyenne, sd = ecart_type),
                color = "#e63946", size = 1.2, linetype = "dashed")

```

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 des variables (*VarsSelect* dans la syntaxe ci-dessous);
- d'utiliser ensuite les fonctions *sapply* et *apply*.

```
# Noms des variables du DataFrame
names(dataMTL)
```

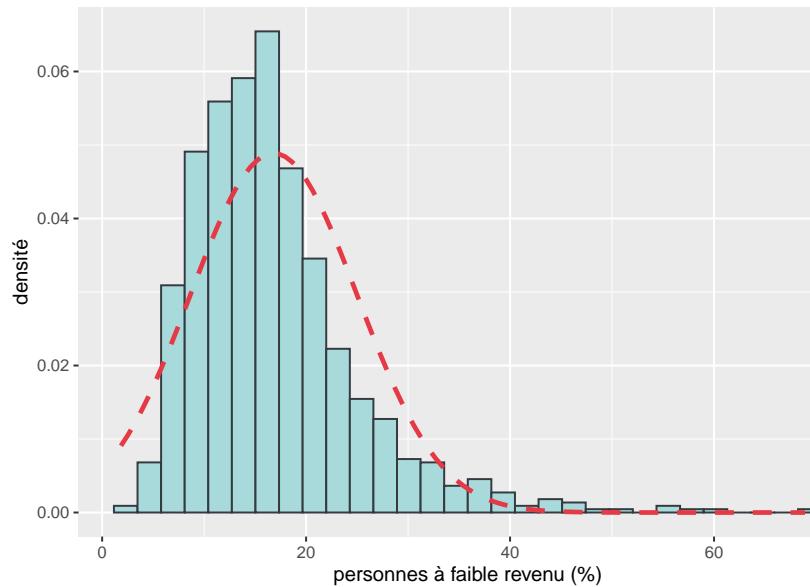


FIG. 2.35 : Histogramme avec courbe normale

```
## [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 Transformation d'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 *skewness* et de *kurtosis* et des 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"            "PctMenageelpers"         "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=expression("Habitants au km"\^2),
       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=expression("Logarithme d'habitants au km"\^2),
       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=expression("Racine carrée d'habitants au km"\^2),
       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=expression("Inverse d'habitants au km"\^2),
       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. Les valeurs des statistiques de forme et du test de Shapiro-Wilk ainsi 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.

2.6 Statistiques descriptives sur des variables qualitatives et semi-qualitatives

2.6.1 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 subdivi-

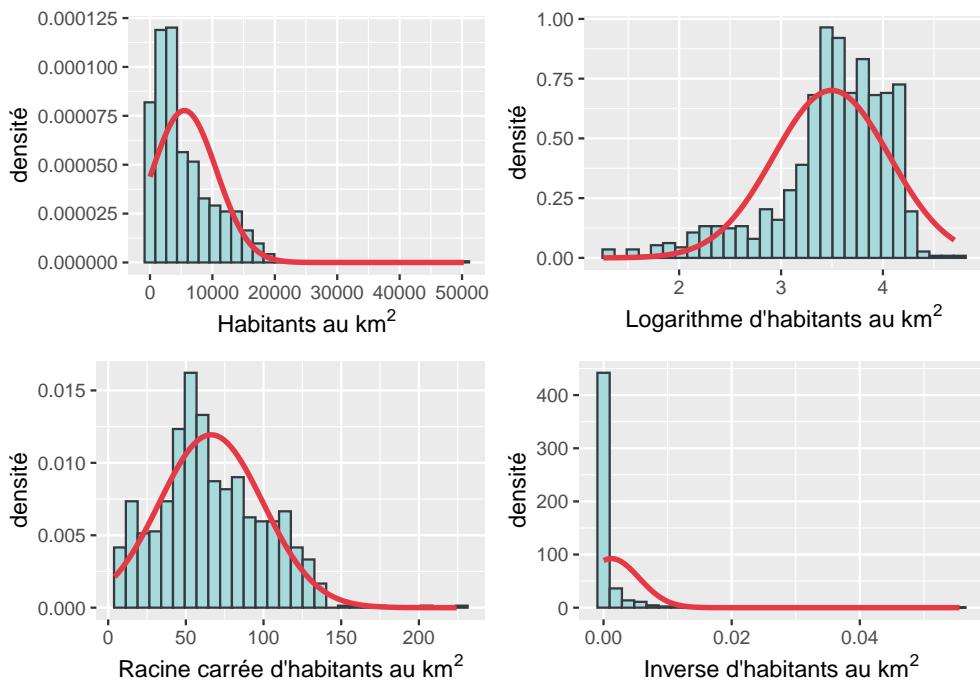


FIG. 2.36 : Histogramme des transformations

sion 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 personnes réalisant des trajets 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 ($30\ 645/857\ 540 = 0,036$); leur somme est égale à 1. Elles peuvent bien entendu être exprimées en pourcentage ($30\ 645/857\ 540 \times 100 = 3,57$); leur somme est alors égale à 100 %. Par exemple, 3,7 % de ces personnes utilisent le vélo comme mode de transport principal.
- Les fréquences absolues cumulées (**FAC**) représentent la fréquence observée (FAS) de la modalité à laquelle sont additionnées celles qui la précédent. 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 ($453\ 930/857\ 540 = 0,529$) et en pourcentage ($453\ 930/857\ 540 \times 100 = 52,93$). Par exemple, plus de la moitié des personnes utilisent l'automobile comme mode de transport principal (passagère ou conductrice).

⚠️ 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 navetteur(ve)s se déplacent en véhicule motorisé (conducteur(trice) ou passager(ère)) ou en transport en commun. Par contre, si la troisième modalité avait été à pied, le pourcentage aurait été de

TAB. 2.10 : Différents types de fréquences sur une variable qualitative ou semi-qualitative

Mode de transport	FAS	FRS	FRS (%)	FAC	FRC	FRC (%)
Véhicule motorisé (conducteur(trice))	427 530	0,499	49,86	427 530	0,499	49,86
Véhicule motorisé (passager(ère))	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			

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 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, plus du quart de la population à moins de 25 ans (27,91 %) et 83,33 %, moins de 65 ans.

Différents graphiques peuvent être construits pour illustrer la répartition des observations : les graphiques en barre (verticale et horizontale) avec les fréquences absolues et les diagrammes circulaires ou en anneau pour les fréquences relatives (figure 2.37). Ces graphiques seront présentés plus en détail 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")

# Vecteur pour les fréquences absolues simples (FAS)
Navetteurs <- c(304470,237555,582150,494205,271560,52100)

# Somme des FAS
```

TAB. 2.11 : Différents types de fréquences sur une variable semi-qualitative

Groupes d'âge	FAS	FRS	FRS (%)	FAC	FRC	FRC (%)
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			

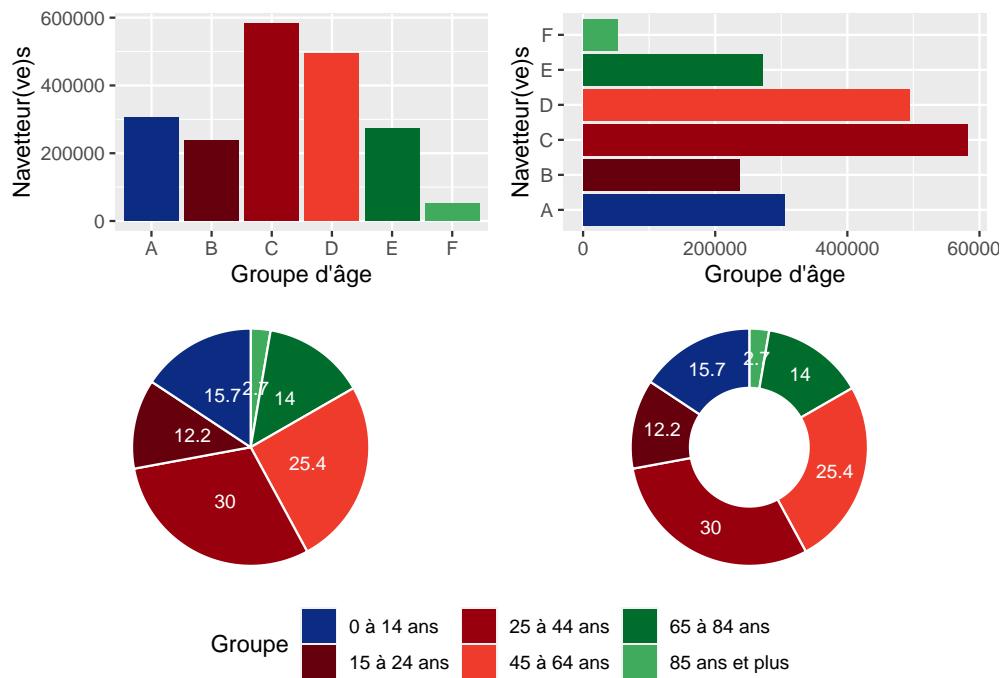


FIG. 2.37 : Différents graphiques pour représenter les fréquences absolues et relatives

```
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 Statistiques descriptives pondérées : pour aller plus loin

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 ne reportons pas ici les formules des versions pondérées de toutes les statistiques descriptives. Revenez 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 est identique à 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 pied 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, Apparicio et Séguin 2018, 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 ;
- w_i , la pondération, soit le nombre d'enfants de moins de dix ans.

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

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

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, 236).

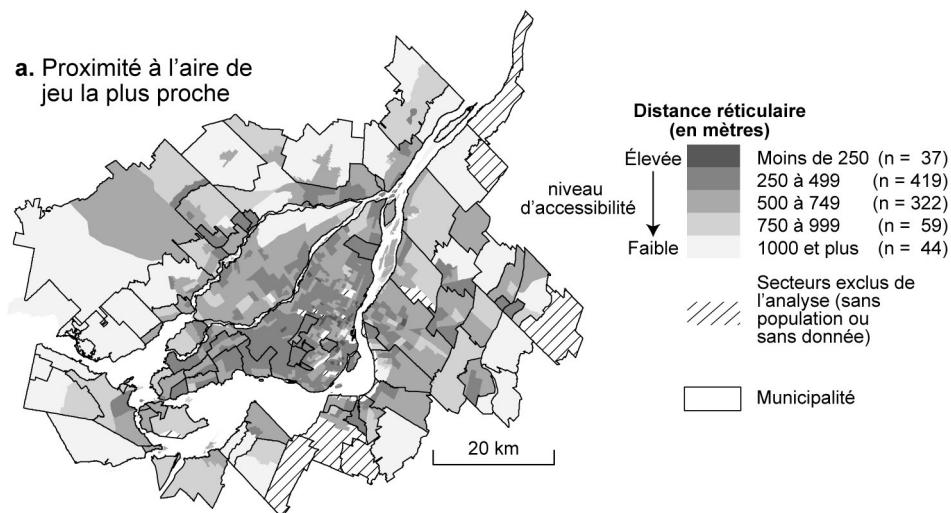


FIG. 2.38 : Accessibilité aux aires de jeux par secteur de recensement, Communauté métropolitaine de Montréal, 2016

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

# xi (variable) et wi (pondération)
x <- df$AireJeux
w <- df$PopMoins10

# Calcul des paramètres de position
```

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

```

# Moyenne
Hmisc::wtd.mean(x, w)

## [1] 559.8026

Weighted.Desc.Stat::w.mean(x, w)

## [1] 559.8026

# Quartiles et percentile
Hmisc::wtd.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

```

2.8 Quiz de révision du chapitre

Questions

- **Le mode de transport (auto, vélo, marche, transport en commun) est :**

- une variable ordinaire
- une variable continue
- une variable discrète
- une variable nominale

Relisez au besoin la section [2.1.2](#).

- **Laquelle de ces deux options est la bonne ?**

- Variables qualitatives (nominales ou ordinaires) ; variables quantitatives (continues et discrètes)
- Variables qualitatives (nominales ou discrètes) ; variables quantitatives (ordinaires et discrètes)

Relisez au besoin le début de la section [1.2](#).

- **Des données collectées sur le terrain avec un GPS sont des données primaires spatiales.**

- Vrai
- Faux

Relisez au besoin la section [2.2](#).

- **L'erreur écologique consiste à :**

- attribuer des résultats à partir de données individuelles à des territoires
- attribuer des constats obtenus à partir de données agrégées pour un territoire aux individus

Relisez au besoin la section [2.2.4](#).

- **Laquelle de ces deux options est la bonne ?**

- Fonctions de masse pour les distributions discrètes, fonctions de densité pour les distributions continues
- Fonctions de masse pour les distributions continues, fonctions de densité pour les distributions discrètes

Relisez au besoin la section [2.4](#).

- **Quels sont les trois paramètres de tendance centrale ?**

- Moyenne
- Médiane
- Variance
- Intervalle interquartile
- Mode

Relisez au besoin la section [2.5.1](#).

- **Quels sont les paramètres position ?**

- Quartiles
- Déciles
- Variance
- Centiles

Relisez au besoin la section [2.5.2](#).

- **Quels sont les paramètres de dispersion ?**

- Étendue
- Intervalle interquartile
- Variance
- Écart-type

Relisez au besoin la section [2.5.3](#).

- **Comment vérifie-t-on qu'une variable est normalement distribuée ?**

- Coefficients d'asymétrie et d'aplatissement (skewness et kurtosis)
- Graphiques : histogramme avec courbe normale et diagramme quantile-quantile
- Tests de normalité (comme celui de Shapiro-Wilk)
- Examen des valeurs minimale et maximale

Relisez au besoin la section [2.5.4](#).

- **Le centrage et la réduction d'une variable consistent à :**

- soustraire à chaque valeur sa moyenne (soit un centrage), puis à la diviser par son écart-type.
- simplement à trier une variable en ordre croissant, puis à affecter le rang de chaque observation de 1 à n
- soustraite à chaque observation la valeur minimale et à diviser le tout par l'étendue.

Relisez au besoin la section [2.5.5.2](#).

Réponses

- Le mode de transport (auto, vélo, marche, transport en commun) est :
 - une variable nominale
- Laquelle de ces deux options est la bonne ?
 - Variables qualitatives (nominales ou ordinaires) ; variables quantitatives (continues et discrètes)
- Des données collectées sur le terrain avec un GPS sont des données primaires spatiales.
 - Vrai
- L'erreur écologique consiste à :
 - attribuer des constats obtenus à partir de données agrégées pour un territoire aux individus
- Laquelle de ces deux options est la bonne ?
 - Fonctions de masse pour les distributions discrètes, fonctions de densité pour les distributions continues
- Quels sont les trois paramètres de tendance centrale ?
 - Moyenne
 - Médiane
 - Mode
- Quels sont les paramètres position ?
 - Quartiles
 - Déciles
 - Centiles
- Quels sont les paramètres de dispersion ?
 - Étendue
 - Intervalle interquartile
 - Variance
 - Écart-type

- Comment vérifie-t-on qu'une variable est normalement distribuée ?
 - Coefficients d'asymétrie et d'aplatissement (skewness et kurtosis)
 - Graphiques : histogramme avec courbe normale et diagramme quantile-quantile
 - Tests de normalité (comme celui de Shapiro-Wilk)
- Le centrage et la réduction d'une variable consistent à :
 - soustraire à chaque valeur sa moyenne (soit un centrage), puis à la diviser par son écart-type.

Chapitre 3

Magie des graphiques

Dans ce chapitre, nous découvrons les incroyables capacités graphiques de R. Pour ce faire, nous couvrons en profondeur les fonctionnalités du *package ggplot2* du *tidyverse*. Selon nous, il s'agit de loin du meilleur *package* pour réaliser des graphiques.



Dans ce chapitre, nous utilisons les *packages* suivants :

- Pour créer des graphiques :
 - * *ggplot2*, le seul, l'unique!
 - * *ggridges* pour combiner des graphiques.
 - * *ggthemes* pour utiliser des thèmes complémentaires pour les graphiques.
- Pour les couleurs :
 - * *RColorBrewer* pour accéder à des palettes de couleurs.
- Pour les graphiques spéciaux :
 - * *chorddiag* pour construire des graphiques d'accord.
 - * *fmsb* pour construire des graphiques en radar.
 - * *treemap* pour construire une carte proportionnelle.
 - * *wordcloud2* et *textrank* pour construire un nuage de mots.
- 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 collections*.



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.

La 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.

Le second graphique² représente quatre informations pour 234 villes à travers le monde :

- la croissance démographique (axe des abscisses),
- la vulnérabilité au changement climatique (axe des ordonnées),
- la taille des villes (taille des cercles),
- le continent sur lequel est localisée chaque ville (couleur des cercles).

Le graphique est à la fois très accrocheur et esthétique. En un coup d'œil, nous constatons 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 une croissance démographique plus faible, qu'elles soient de petites, de moyennes ou de grandes villes.

Souvent négligée, la visualisation de données est perçue comme une tâche triviale : il s'agit simplement de représenter une donnée sous forme d'un graphique, car c'est l'option la plus pratique ou qui prend le moins d'espace. Pourtant, les avantages de la visualisation des données sont nombreux. Par exemple, la visualisation de données intègre aujourd'hui des supports dynamiques comme des animations, des figures interactives ou des applications web. R offre d'ailleurs des possibilités très intéressantes en la matière avec des *packages* comme `shiny`, `plotly` ou `leaflet`. Toutefois, nous ne couvrons pas ici ces méthodes plus récentes en visualisation des données qui devraient faire l'objet d'un autre livre.

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 livrer 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, ni 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 visualisations de données, sinon l'essence du message sera perdue.
- **Aide à la décision**. Une illustration (graphique ou carte) peut être un outil facilitant la prise de décisions.

3.1 Philosophie du ggplot2

Le *package* `ggplot2` fait partie du `tidyverse` et dispose 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`), 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 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 3.1) :

- Les **géométries** sont les formes utilisées pour représenter les données. Il peut s'agir de points, de lignes, de cercles, de rectangles, d'arcs de cercle, etc.
- Les **échelles et systèmes de coordonnées** permettent de contrôler la localisation des éléments dans

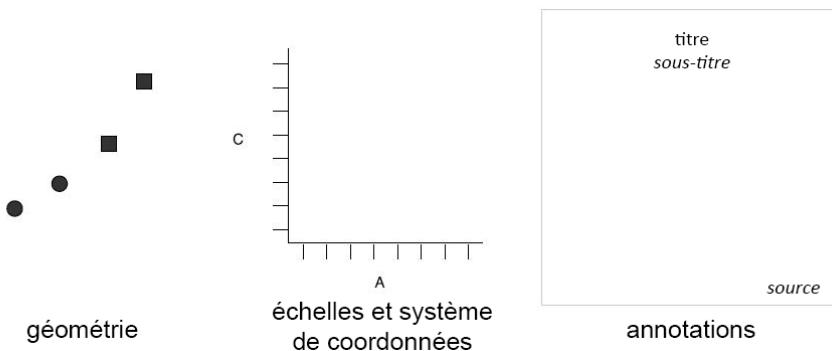


FIG. 3.1 : Trois composantes d'un graphique, adapté de @wickham2010layered

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.

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 nous combinons des données, leur assignation à des dimensions, un type de géométries, des échelles et un système de coordonnées, nous obtenons un **calque** (*layer* en anglais). 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 représentons la relation qui existe entre la longueur et la largeur des sépales 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 (figure 3.2). 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 sépales, et les coordonnées Y pour représenter la longueur des sépales. 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 (figure 3.3). 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

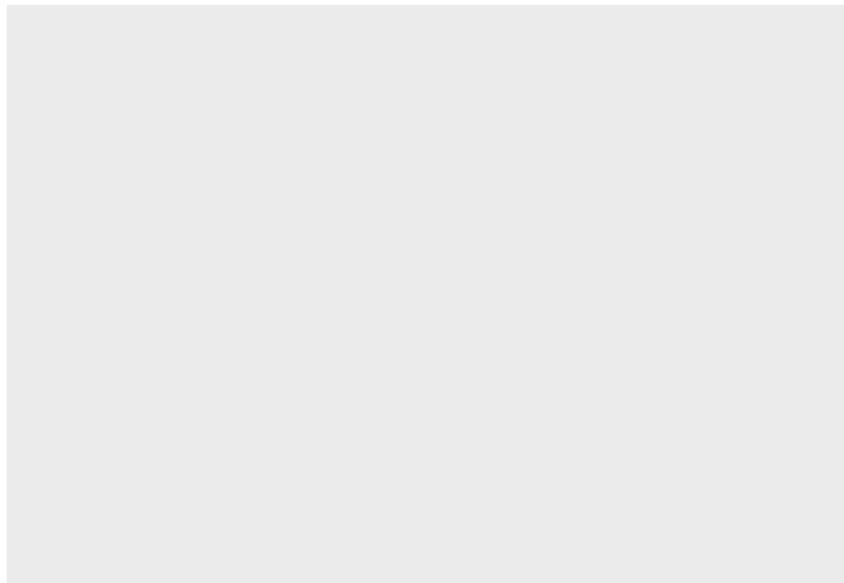


FIG. 3.2 : Base d'un graphique

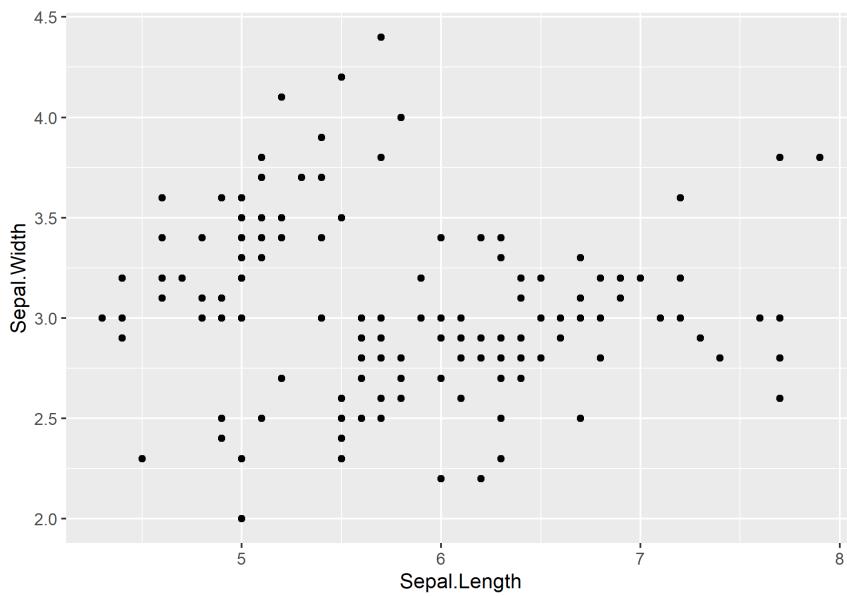


FIG. 3.3 : Ajout des dimensions au graphique

appliqués à tous les calques qui composent le graphique. Il est aussi possible de définir `mapping` et `data` au sein des fonctions des géométries :

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris)
```

La troisième étape consiste à ajouter au graphique des 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` (figure 3.5).

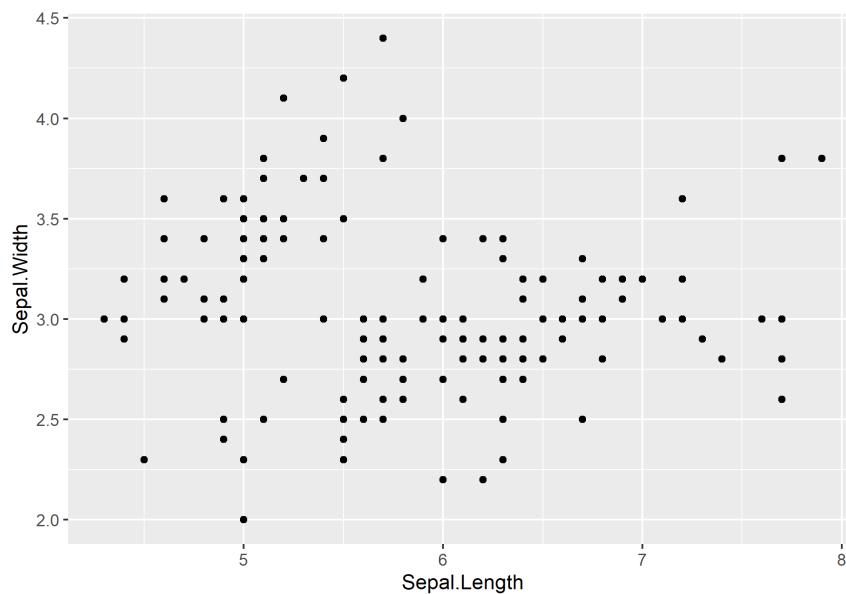


FIG. 3.4 : Autre spécification des arguments mapping et data

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales")
```

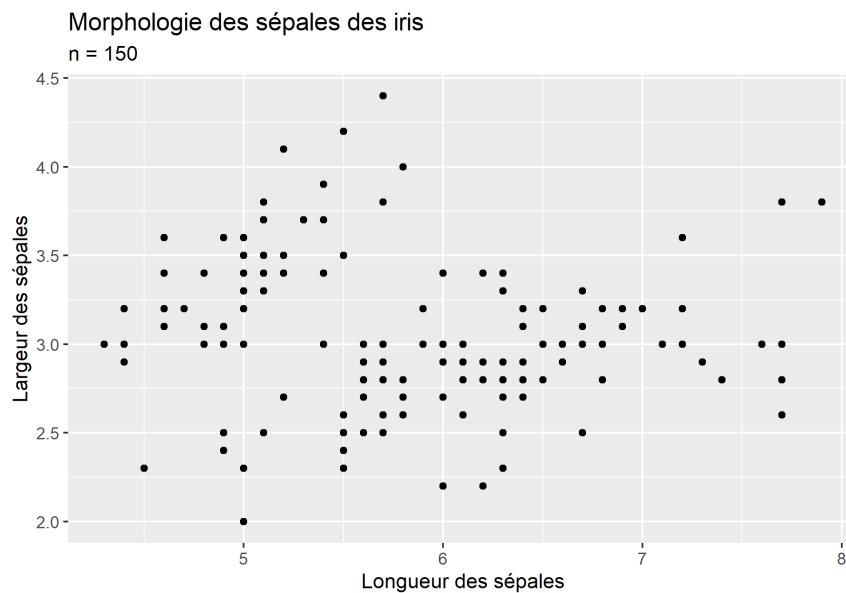


FIG. 3.5 : Ajout de titres

3.1.2 Types de géométries

Le package `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 proposées par ggplot2

Géométrie	Fonction
point	geom_point
ligne	geom_line
chemin	geom_path
boîte à moustaches	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 Habillage

Dans le premier exemple, nous avons montré comment ajouter le titre, le sous-titre et les titres des axes sur un graphique. Il est aussi possible d'ajouter du 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 (coordonnées `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 sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  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() +
```

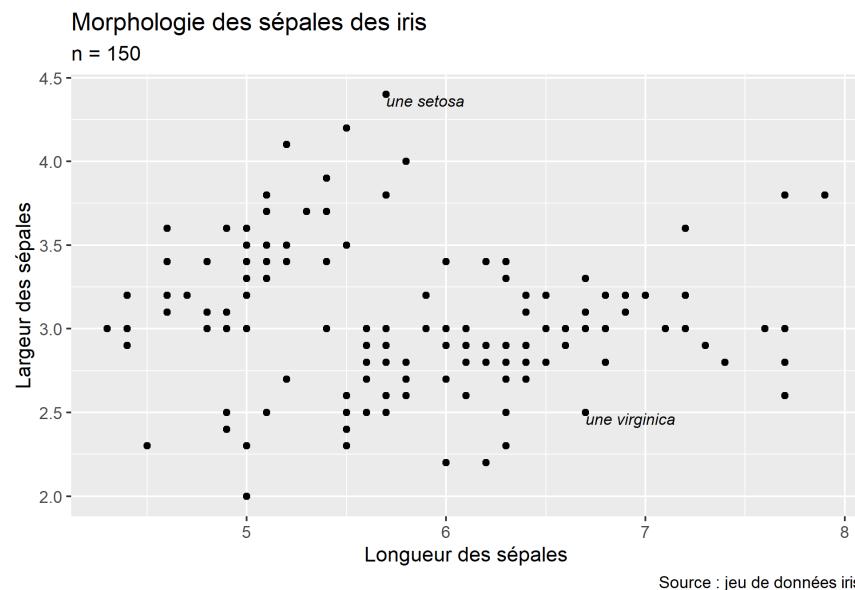


FIG. 3.6 : Ajout d'annotations textuelles

```
geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  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(0.7,0.7,0.7,.5), # remplissage transparent à 50%
           color = "black") # contour de couleur verte
```

Comme le dernier calque ajouté au graphique est le rectangle, vous noterez qu'il recouvre tous les calques existant, y compris les précédentes annotations. Pour corriger cela, il suffit de changer l'ordre des calques.

```
ggplot() +
  annotate("rect",
           ymin = sepal.width_extent[[1]],
           ymax = sepal.width_extent[[2]],
           xmin = sepal.length_extent[[1]],
           xmax = sepal.length_extent[[2]],
           fill = rgb(0.7,0.7,0.7,.5), # remplissage transparent à 50%
           color = "green") + # contour de couleur verte
```

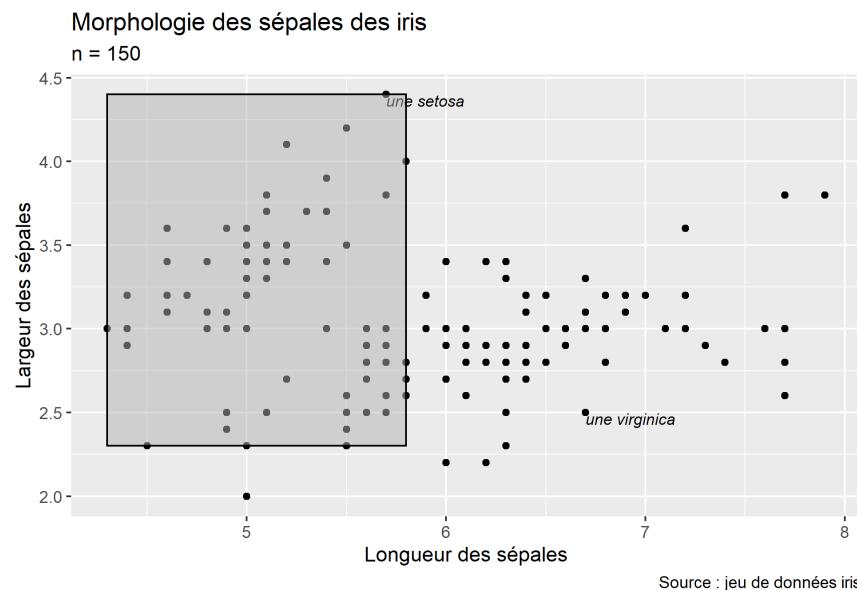


FIG. 3.7 : Ajout d'annotations géométriques

```
geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  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")
```

3.1.4 Utilisation 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 paramétrer tous ces éléments. Une option intéressante est d'utiliser des thèmes déjà préconstruits. Le package `ggplot2` propose une dizaine de thèmes : constatons leur impact sur le graphique précédent.

- Le thème classique (`theme_classic`) (figure 3.9)

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  theme_classic()
```

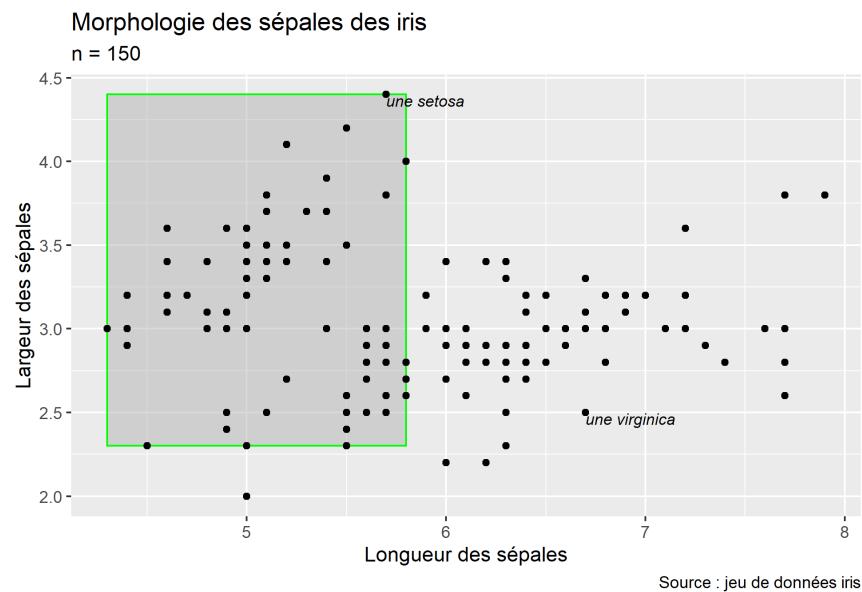


FIG. 3.8 : Gestion de l'ordre des annotations

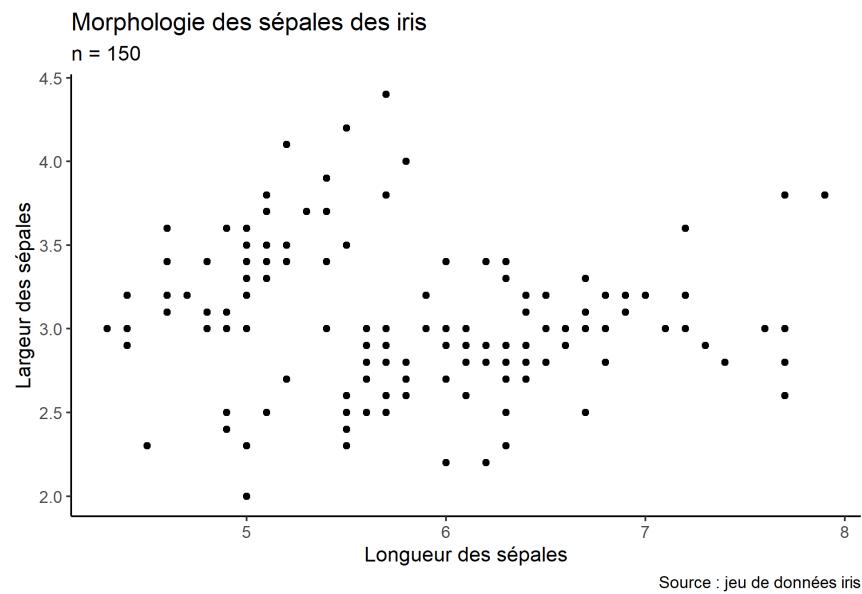


FIG. 3.9 : Thème classique

- Le thème gris (theme_gray) (figure 3.10)

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  theme_gray()
```

- Le thème noir et blanc (theme_bw) (figure 3.11)

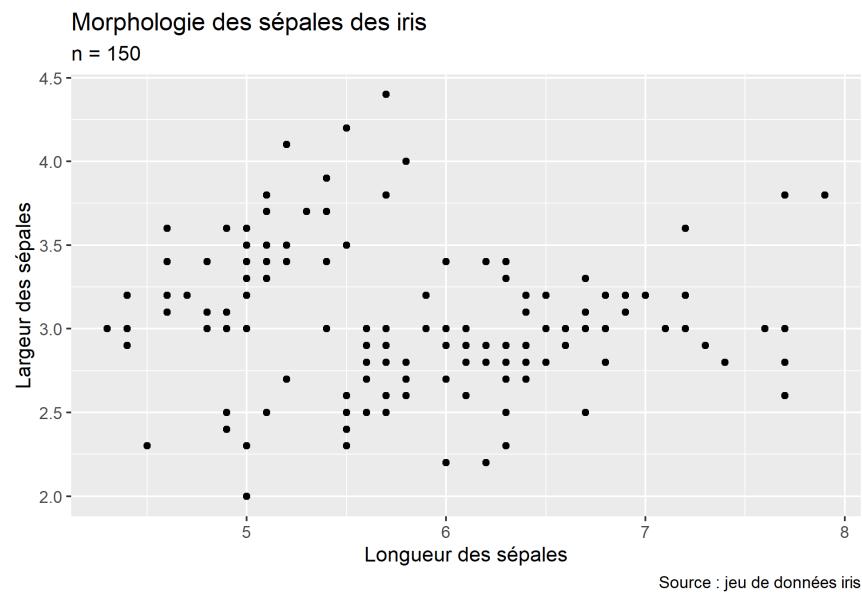


FIG. 3.10 : Thème gris

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  theme_bw()
```

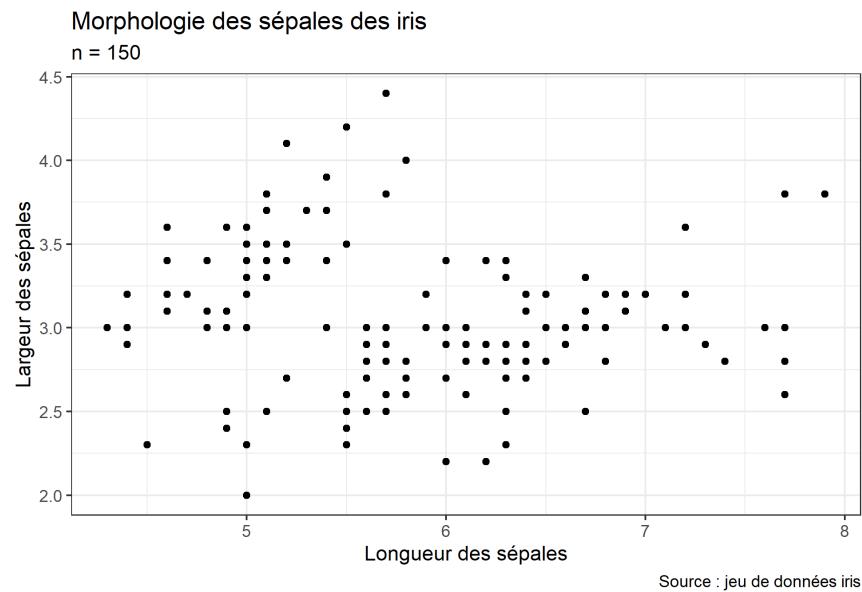


FIG. 3.11 : Thème noir et blanc

- Le thème minimal (`theme_minimal`) (figure 3.12)

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  theme_minimal()
```

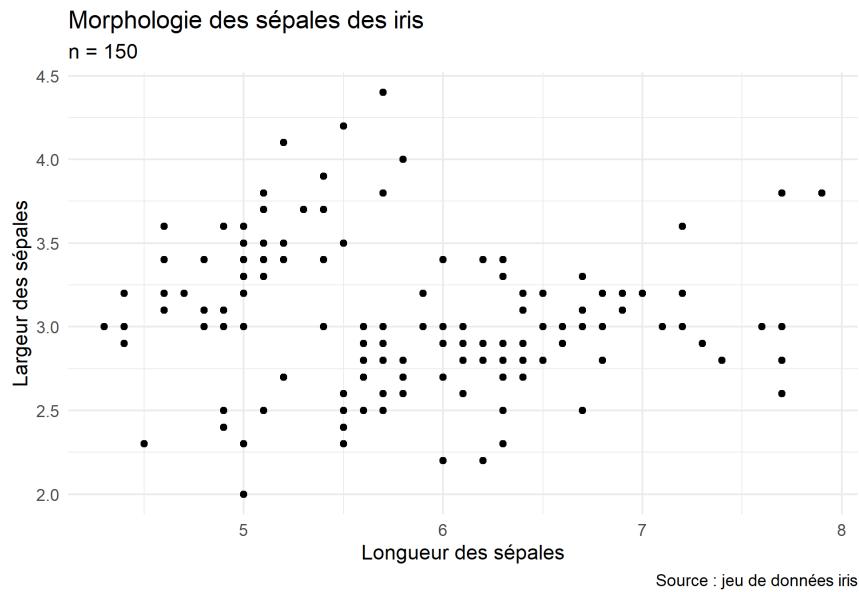


FIG. 3.12 : Thème minimal

Il est aussi possible d'utiliser le package `ggthemes` qui apporte des thèmes complémentaires intéressants dont :

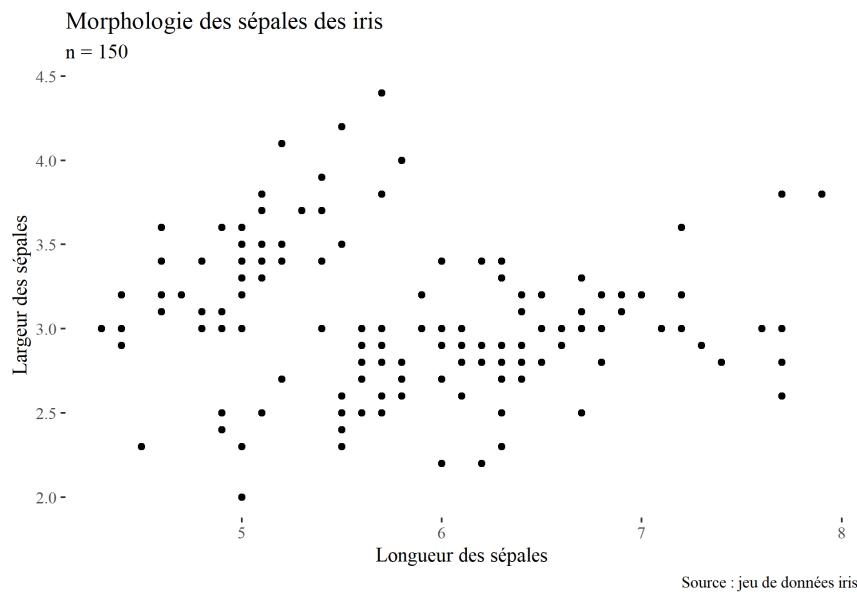
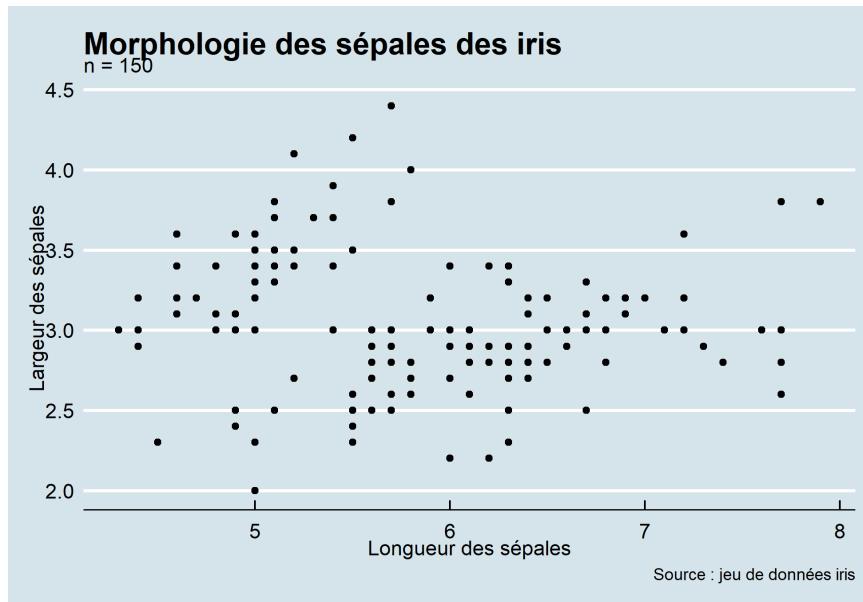
- Le thème *tufte* (`theme_tufte`, à l'ancienne...) (figure 3.13)

```
library(ggthemes)

ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  theme_tufte()
```

- Le thème *economist* (`theme_economist`, inspiré de la revue du même nom) (figure 3.14)

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  theme_economist()
```

**FIG. 3.13 :** Thème tufte**FIG. 3.14 :** Thème economist

- Le thème *solarized* (`theme_solarized`, plus original) (figure 3.15)

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  theme_solarized()
```

Il en existe bien d'autres et vous pouvez composer vos propres thèmes. N'hésitez pas à explorer la docu-

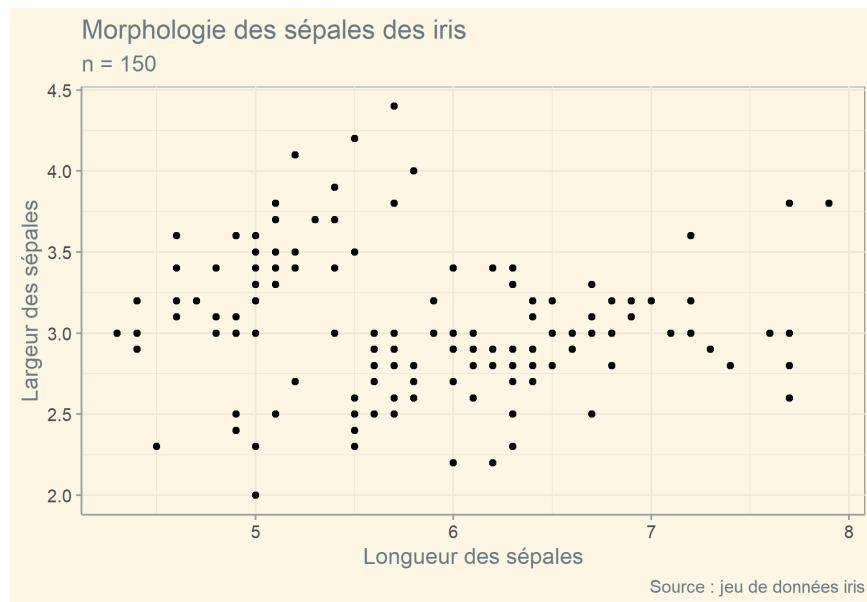


FIG. 3.15 : Thème solarized

mentation de `ggplot2` et de `ggthemes` pour en apprendre plus !

3.1.5 Composition d'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 variable 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 l'espèce des iris (figure 3.16). 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 sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  facet_wrap(vars(Species_fac), ncol=2)
```

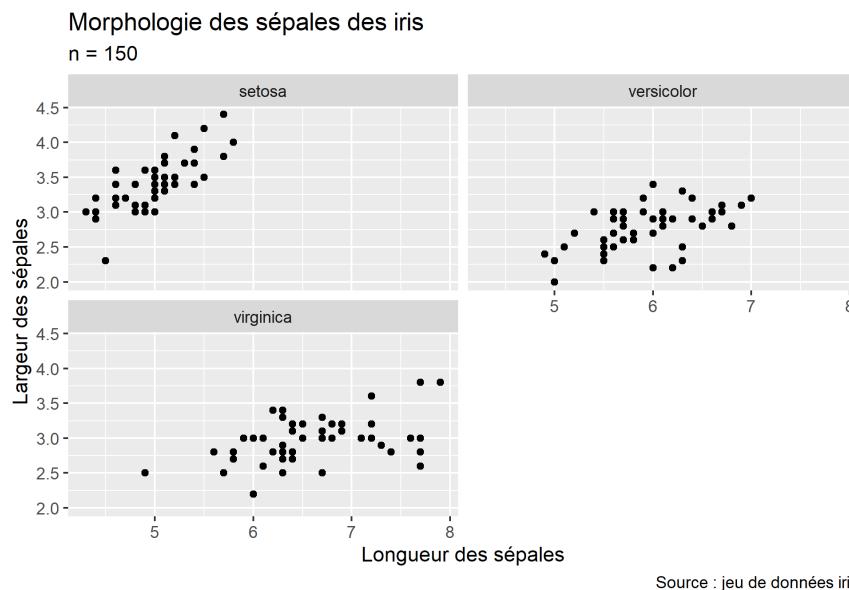


FIG. 3.16 : Graphique à facettes

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` (figure 3.17).

```
ggplot() +
  geom_point(mapping = aes(x = Sepal.Length, y = Sepal.Width), data = iris) +
  labs(title = "Morphologie des sépales des iris", subtitle = "n = 150",
       x = "Longueur des sépales",
       y = "Largeur des sépales",
       caption = "Source : jeu de données iris") +
  facet_wrap(vars(Species_fac), ncol=3)
```

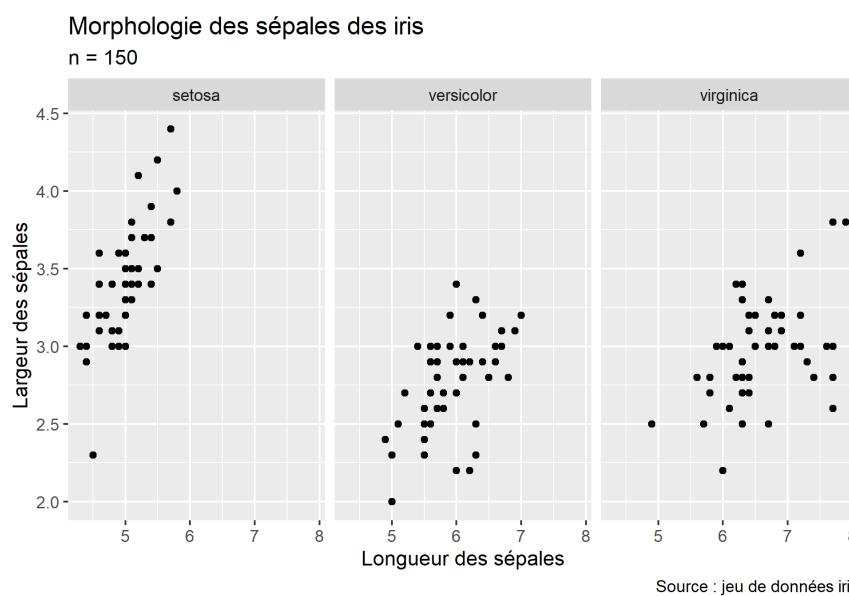


FIG. 3.17 : Graphique à facettes en une ligne

3.1.5.2 Arrangement 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 deux nuages de points comparant plusieurs variables en fonction de l'espèce des iris, puis combinons-les (figure 3.18). Attribuons également 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 sépales",
       x = "Longueur",
       y = "Largeur",
       color = "Espèce")

plot2 <- ggplot(data = iris) +
  geom_point(aes(x = Petal.Length, y = Petal.Width, color = Species)) +
  labs(subtitle = "Caractéristiques des pétales",
       x = "Longueur",
       y = "Largeur",
       color = "Espèce")

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 sépales et pétales des iris",
                               face = "bold", size = 12, just = "center"),
               bottom = text_grob("Source : jeu de données iris",
                                  face = "italic", size = 8, just = "left")
               )
```

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 Couleur

Dans un graphique, la couleur peut être utilisée à la fois pour représenter une variable quantitative (dégradé de couleur ou mise en classes), ou une variable qualitative (couleur par catégorie). Dans `ggplot2`, il

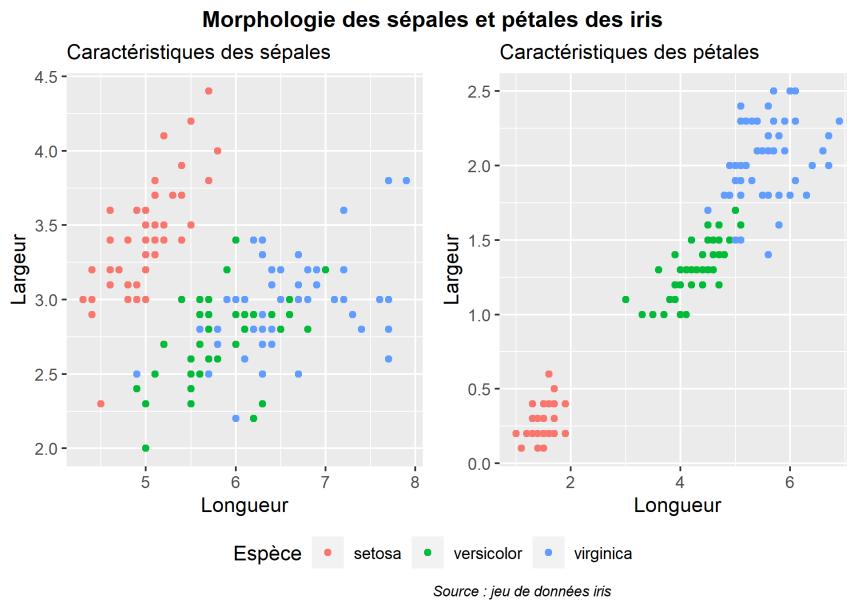


FIG. 3.18 : Figure avec plusieurs graphiques avec ggarrange

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 :

- En utilisant le nom de la couleur dans une chaîne de caractère : "chartreuse4". R dispose de 657 noms de couleurs prédéfinis. Pour tous les afficher, utilisez la fonction `colors()`, qui permet de les visualiser (figure 3.19).
- 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, transparence). Cette notation doit contenir quatre nombres entre 0 et 1 (0 % et 100 %), indiquant respectivement la quantité de rouge, de vert, de bleu et la transparence. Ces quatre 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 (figure 3.20). 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 colors.co⁴ qui propose de nombreuses palettes à portée de clic.

³<https://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3>

⁴<https://colors.co/palettes/trending>

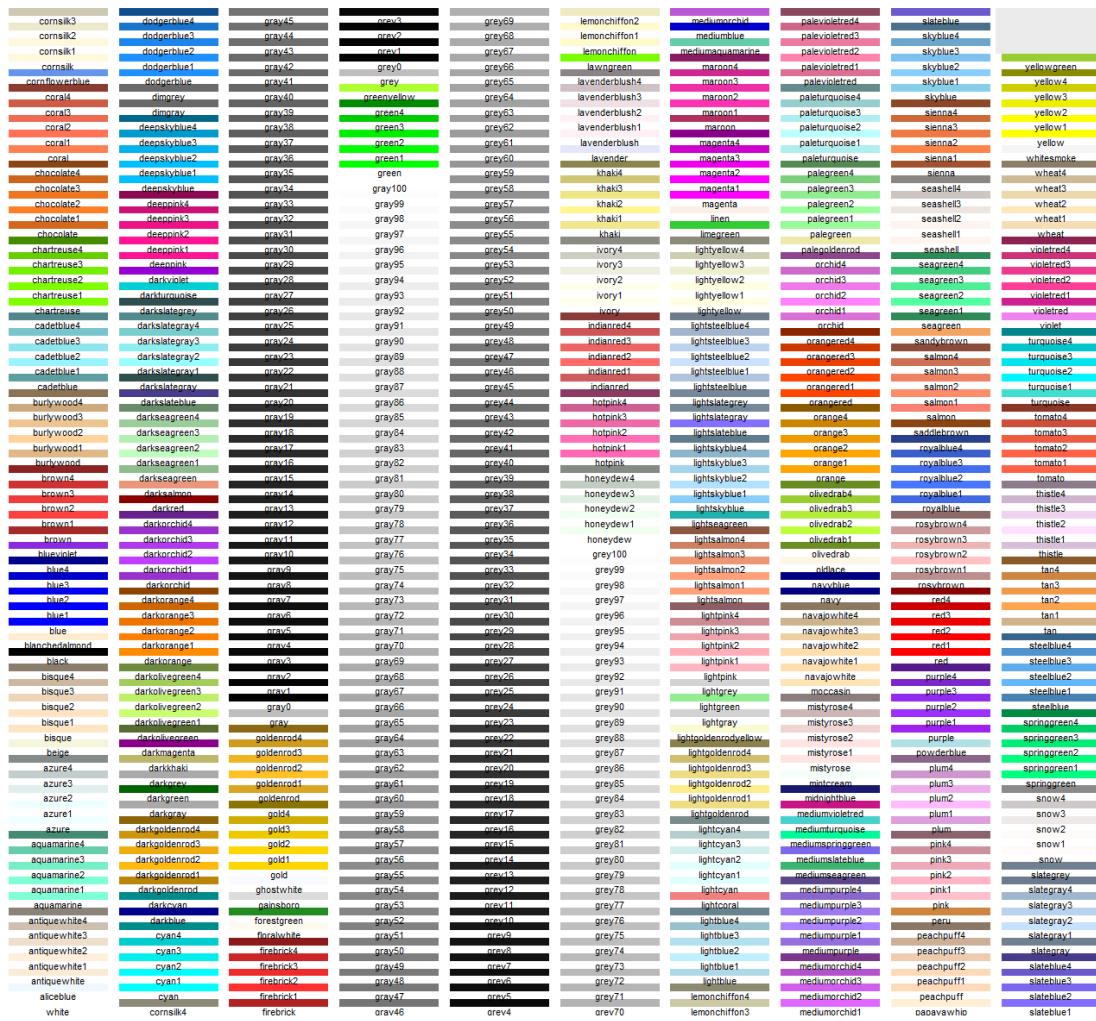


FIG. 3.19 : Couleurs de base

3.2 Principaux graphiques



Puisque vous avez désormais une certaine connaissance des 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 ou que vous présenterez 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, nous utilisons 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 l'histogramme risque d'avoir plein de trous.

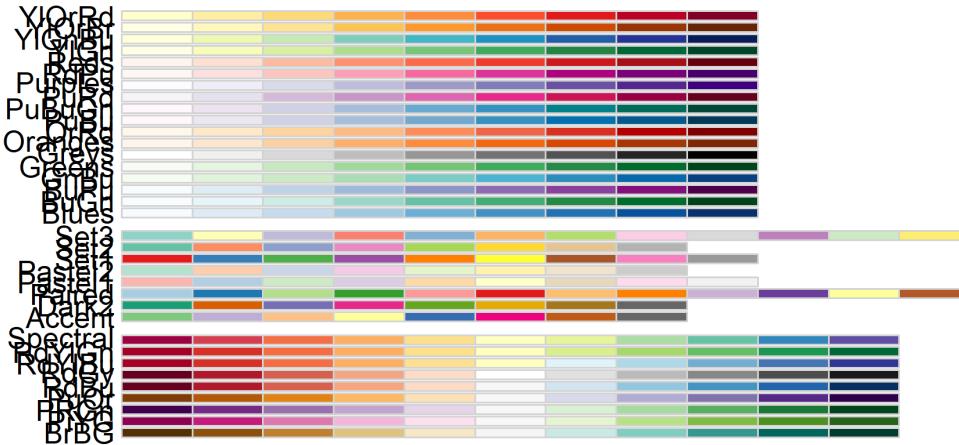


FIG. 3.20 : Palette de couleurs de ColorBrewer

3.2.1.1 Histogramme simple

Générons quatre variables ayant respectivement une distribution gaussienne, Student, Gamma et bêta, puis réalisons un histogramme pour chacune de ces variables et combinons-les avec la fonction ggarrange (figure 3.21).

```
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") +
  labs(y="fréquences")+ylim(c(0,130))

plot2 <- ggplot(data = distrib) +
  geom_histogram(aes(x = gamma), bins = 50, color = "#343a40", fill = "#f1faee") +
  labs(y="fréquences")+ylim(c(0,130))

plot3 <- ggplot(data = distrib) +
  geom_histogram(aes(x = beta), bins = 50, color = "#343a40", fill = "#a8dadcc") +
  labs(y="fréquences")+ylim(c(0,130))

plot4 <- ggplot(data = distrib) +
  geom_histogram(aes(x = student), bins = 50, color = "#343a40", fill = "#1d3557") +
```

```
labs(y="fréquences")+ylim(c(0,130))

histogrammes <- list(plot1, plot2, plot3, plot4)

ggarrange(plotlist = histogrammes, ncol = 2, nrow = 2)
```

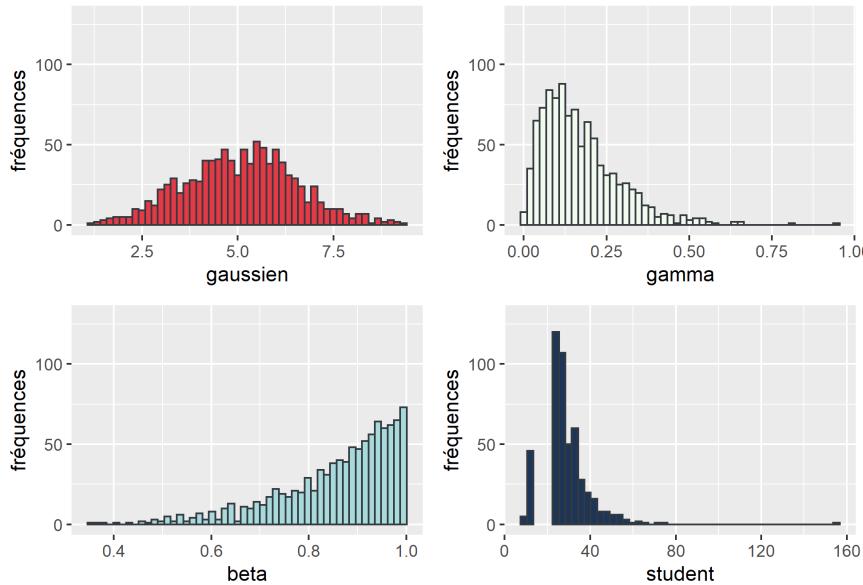


FIG. 3.21 : Histogrammes

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*, *bêta* et *student*) et 1000 observations, pour qu'il n'ait plus que deux colonnes (la valeur originale et le nom de l'ancienne colonne) et 4000 observations. La figure 3.22 décrit graphiquement ce processus qui peut être effectué avec la fonction `melt` du package `reshape2`.

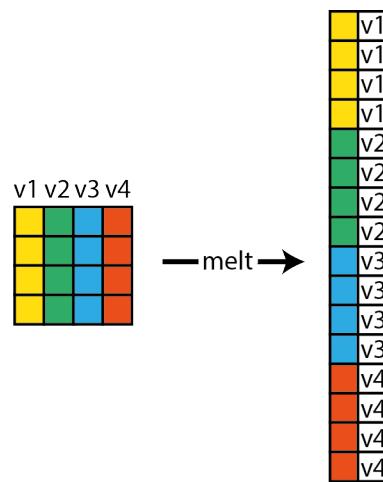


FIG. 3.22 : Empiler les données d'un DataFrame

```

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 = "fréquences")+
  scale_fill_manual(values = c("#e63946","#f1faee","#a8adac","#1d3557"))+
  facet_wrap(vars(distribution), ncol=2, scales = "free")+
  theme(legend.position = "none")

```

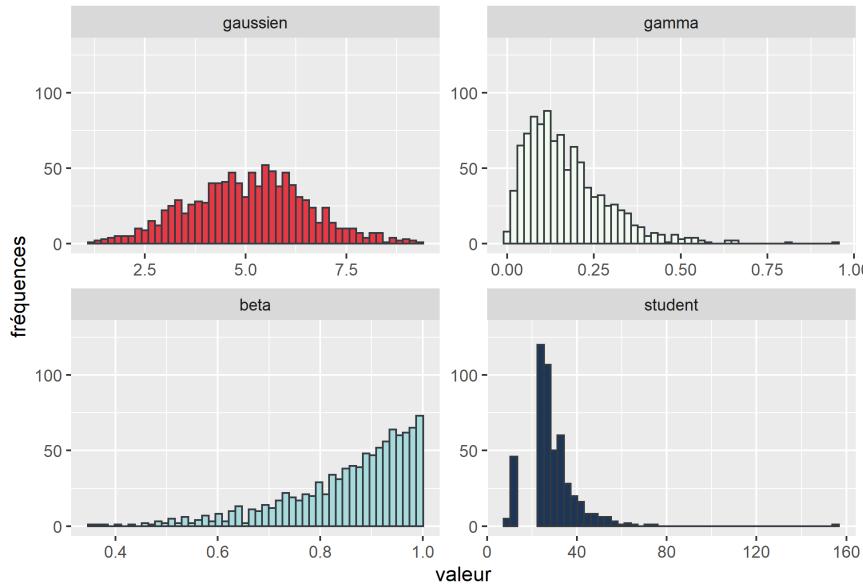


FIG. 3.23 : Histogrammes à facettes

3.2.1.2 Histogramme de densité

Les histogrammes que nous venons de construire utilisent la fréquence 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 = distribrs) +
  geom_histogram(aes(x = gaussien, y = ..density..),
                bins = 30, color = "#343a40", fill = "#1d3557")+

```

```

  labs(x = "gaussien", y = "densité")

plot2 <- ggplot(data = distribs) +
  geom_histogram(aes(x = gaussien, y = ..count..),
                 bins = 30, color = "#343a40", fill = "#a8dadc") +
  labs(x = "gaussien", y = "fréquences")

ggarrange(plotlist = list(plot1, plot2), ncol = 2)

```

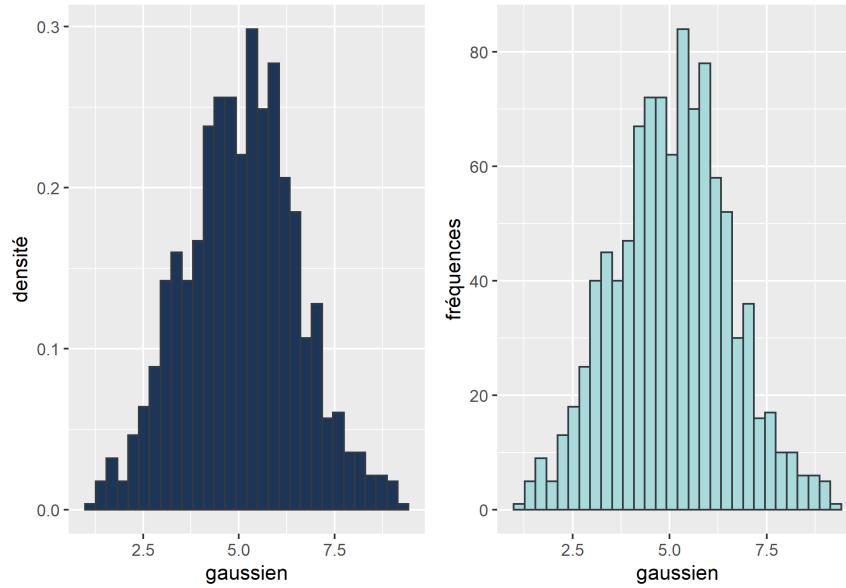


FIG. 3.24 : Histogrammes de densité

Le graphique de droite (fréquence) nous indique donc que plus de 60 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.

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, nous ajoutons 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. Il est aussi possible d'ajouter une ligne verticale (`geom_vline`) pour indiquer la moyenne de la distribution.

```

moyenne <- mean(distribbs$gaussien)
ecart_type <- sd(distribbs$gaussien)

ggplot(data = distribbs) +
  geom_histogram(aes(x = gaussien, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#a8dadc") +
  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=''))

```

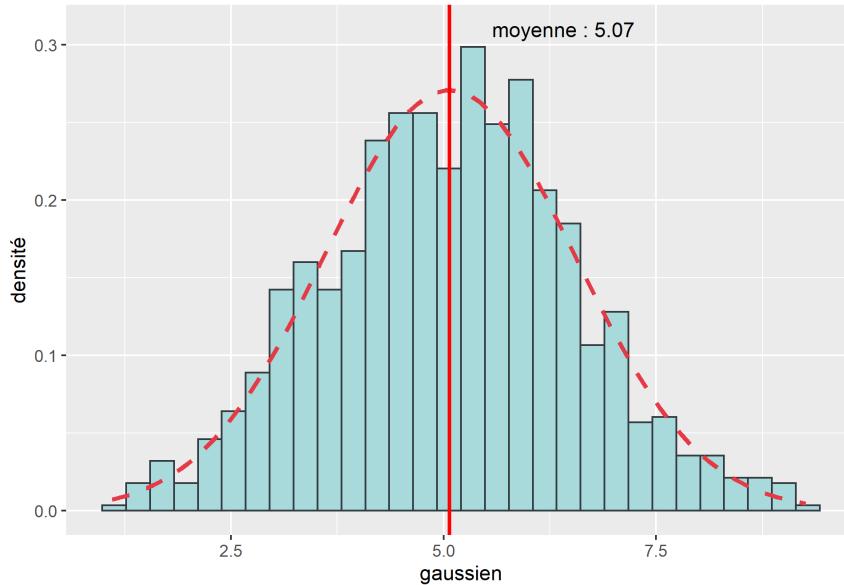


FIG. 3.25 : Histogramme et courbe normale

Dans notre cas, nous savons que notre variable est normalement distribuée (car produite avec la fonction `rnorm`), et nous pouvons 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 sur des valeurs extrêmes. Si nous reprenons notre distribution de Student, nous pouvons clairement distinguer un ensemble de valeurs fortes à droite de la distribution. Nous pourrions, dans notre cas, considérer que des valeurs au-delà de 50 constituent des cas extrêmes que nous souhaitons représenter dans une autre couleur. Pour cela, nous devons créer une variable catégorielle nous permettant de distinguer ces cas particuliers.

```

distrib$cas_extreme <- ifelse(distrib$student >=50, "extrême", "normale")

ggplot(data = distrib) +
  geom_histogram(aes(x = student, y = ..count.., fill = cas_extreme),
                 bins = 30, color = "#343a40") +
  scale_fill_manual("", values = c("#a8dad0", "#e63946")) +
  labs(title = 'Distribution de Student', x = "valeur", y = "fréquence")

```

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 option intéressante est d'utiliser une version lissée de l'histogramme, soit le graphique de densité. Cette opération de lissage est réalisée le plus

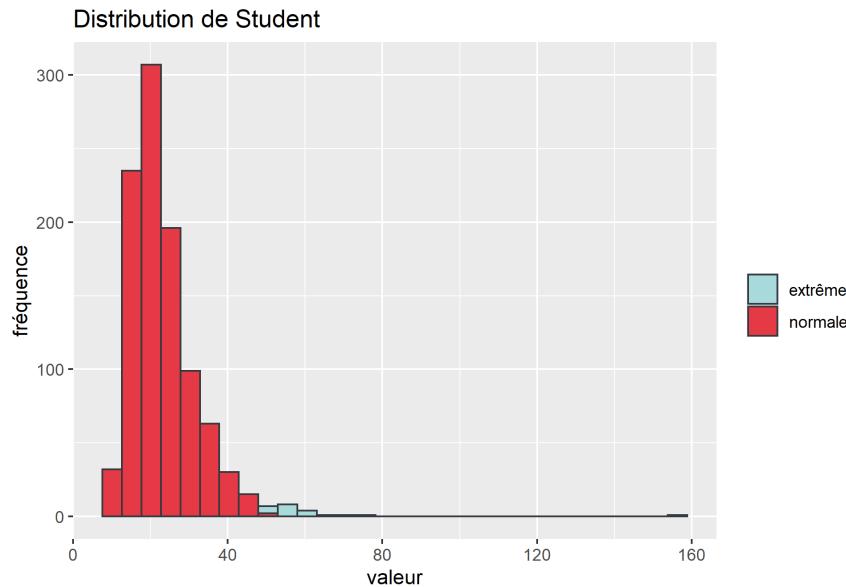


FIG. 3.26 : Histogramme coloré

souvent à partir de fonctions kernel. Reconstruisons notre figure avec les quatre distributions, mais en utilisant cette fois-ci des graphiques de densité.

```
ggplot(data = melted_distribrs) +
  geom_density(aes(x = valeur, fill = distribution), color = "#343a40") +
  scale_fill_manual(values = c("#e63946", "#f1faee", "#a8adac", "#1d3557")) +
  facet_wrap(vars(distribution), ncol=2, scales = "free") +
  theme(legend.position = "none")
```

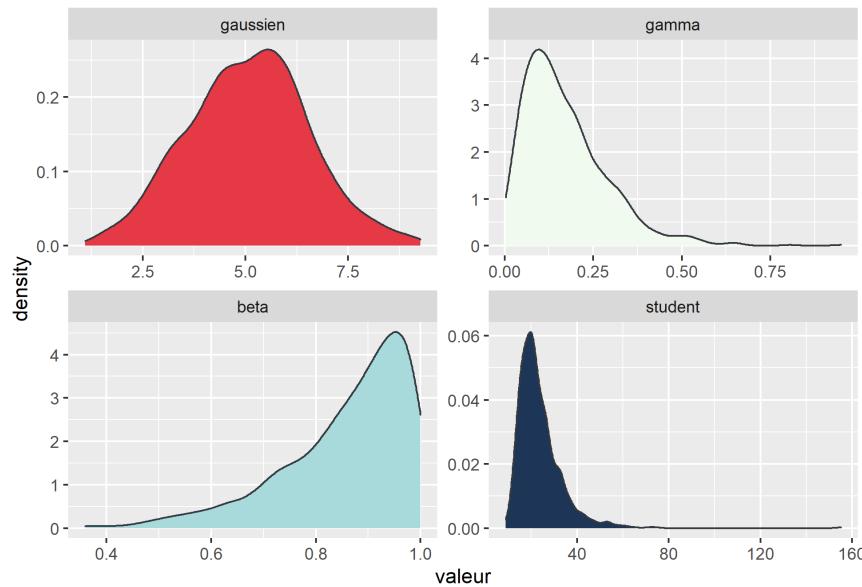


FIG. 3.27 : Graphiques de densité à facette

Les graphiques de densité sont souvent utilisés pour comparer la distribution d'une variable pour plusieurs sous-groupes d'une population. Si nous reprenons le jeu de données *iris*, nous pouvons comparer

les longueurs de sépales en fonction des espèces. Nous constatons ainsi que les setosas ont une nette tendance à avoir des sépales plus courts et qu'à l'inverse, les virginicas ont les sépales généralement les plus longs.

```
ggplot(data = iris)+  
  geom_density(aes(x = Sepal.Length, fill = Species),  
               color = "#343a40", alpha = 0.4)+  
  labs(x = 'Longueur de sépales',  
       y = '',  
       fill = 'Espèce')
```

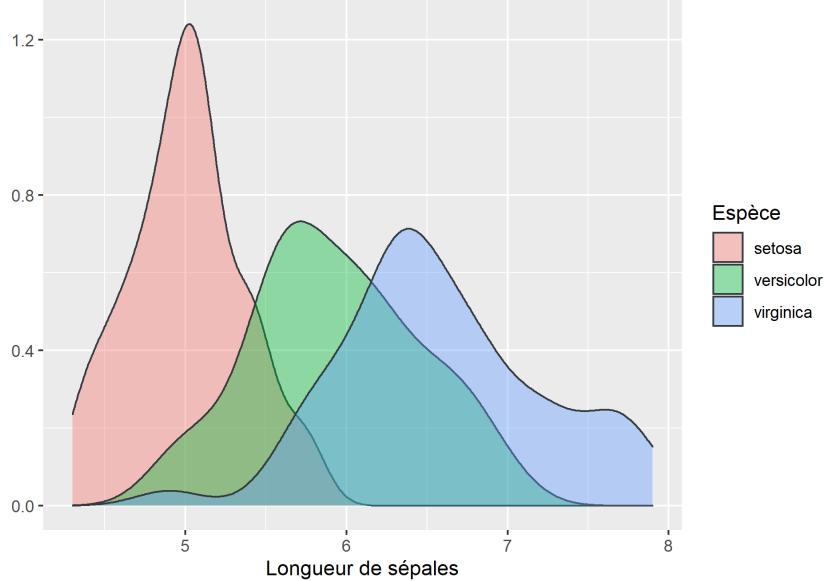


FIG. 3.28 : Graphiques de densité superposés

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 en comparaison avec le niveau d'urbanisation dans l'ensemble des pays à travers le monde. Nous avons extrait ces données sur le site web de la Banque mondiale⁵, puis nous les avons structurés dans un fichier *csv*.

```
data_co2 <- read.csv("data/graphique/world_urb_co2.csv", encoding = "UTF-8")  
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.

⁵<https://donnees.banquemondiale.org/indicateur>

```
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')
```

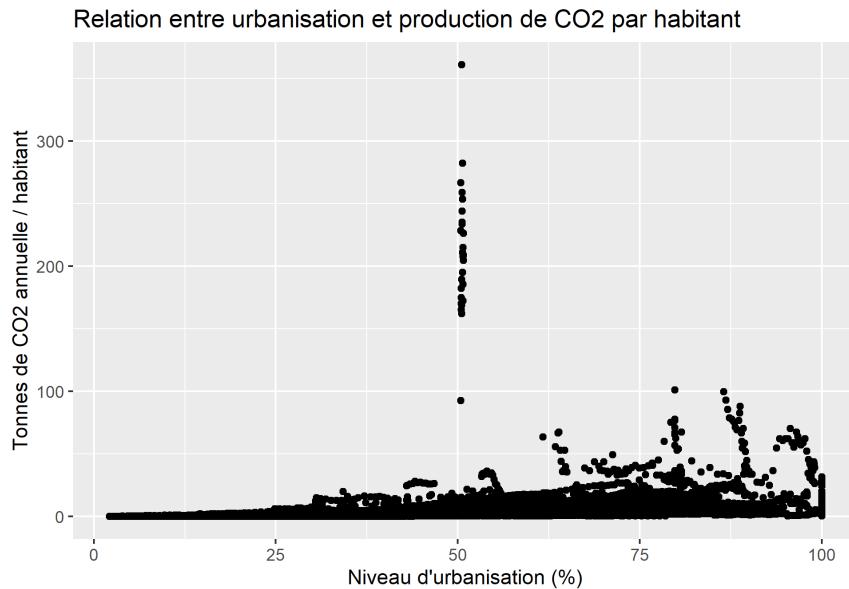


FIG. 3.29 : Nuage de points simple

À la première lecture de ce graphique, nous observons immédiatement un ensemble de points étranges dont le volume de CO₂ par habitant annuel est au-dessus de 150 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"  
## [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.

```
graphique <- 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')
```

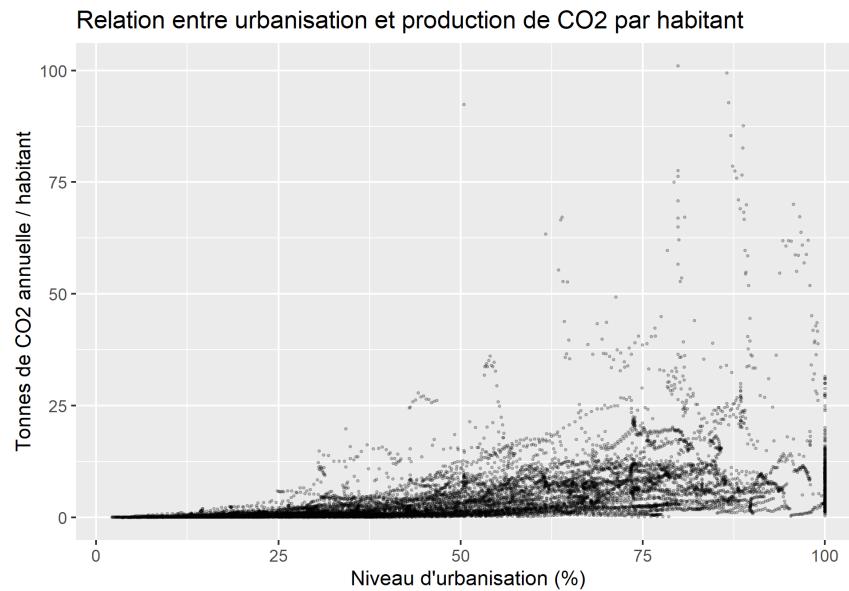


FIG. 3.30 : Nuage de points simple avec transparence

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 = CO2t_hab), bins = 50) +
  scale_fill_continuous(type = "viridis") +
  labs(x = "Niveau d'urbanisation (%)",
       y = 'Tonnes de CO2 annuelle / habitant',
       fill = "Effectif",
       title = 'Relation entre urbanisation et production de CO2 par habitant')
```

Nous observons ainsi une forte concentration dans le bas du graphique; les pays avec des rejets annuels

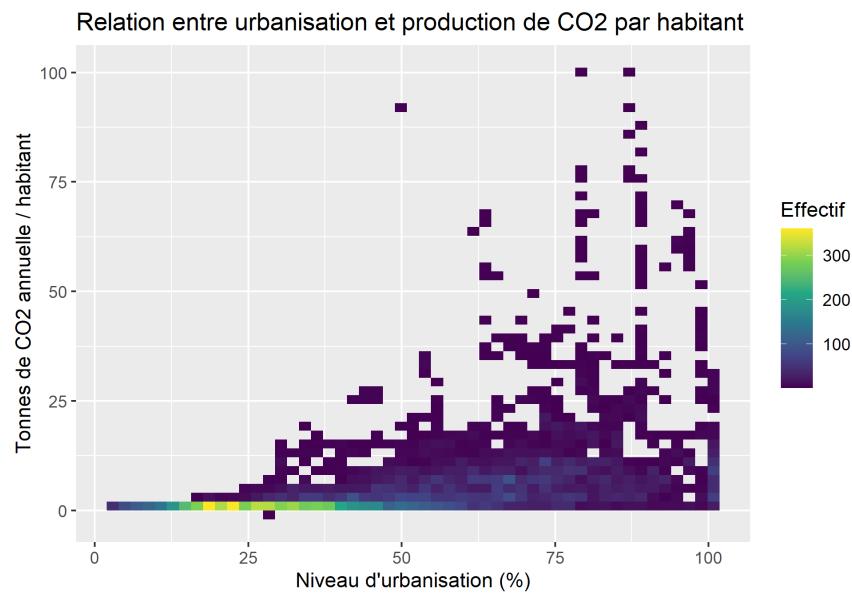


FIG. 3.31 : Nuage de points simple

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',
       fill = "Effectif",
       title = 'Relation entre urbanisation et production de CO2 par habitant')
```

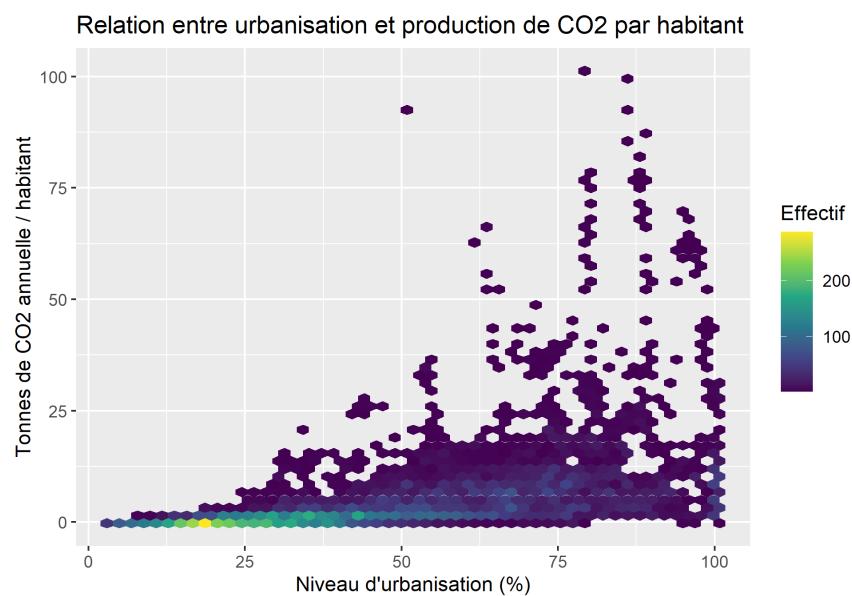


FIG. 3.32 : Densité en deux dimensions par hexagones

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 = C02t_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',
       fill = "densité",
       title = 'Relation entre urbanisation et production de CO2 par habitant') +
  ylim(0,25)
```

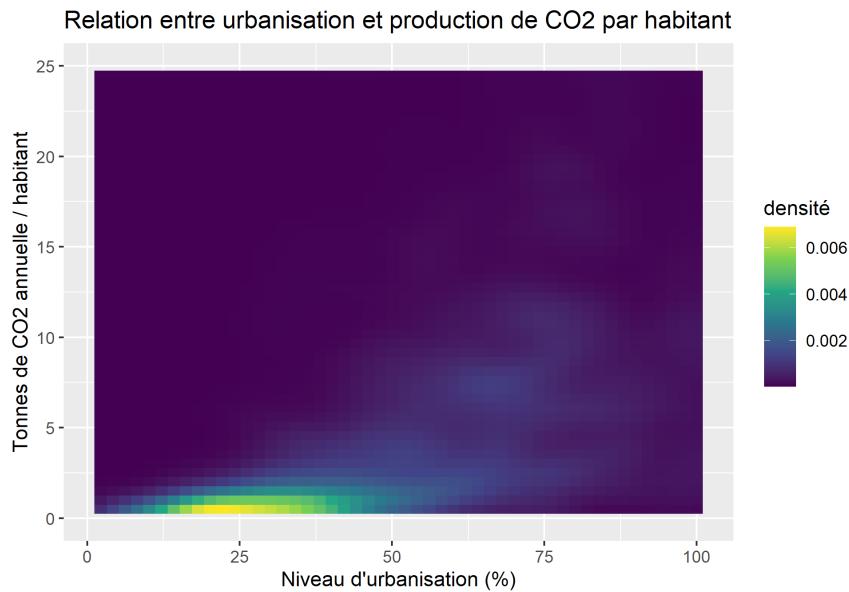


FIG. 3.33 : Densité lissée en deux dimensions

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`.

```
graphique <- 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')
```

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 à une augmentation de la production annuelle de CO₂ par habitant. Nous pourrions 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 = CO2t_hab), alpha = 0.2, size = 0.7)+  
  geom_smooth(aes(x = Urbanisation, y = CO2t_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')
```

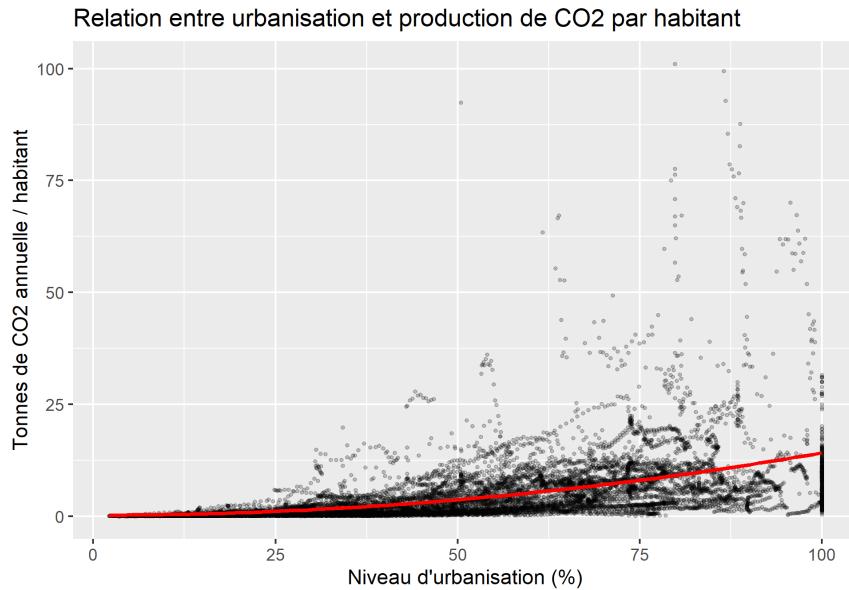


FIG. 3.34 : Nuage de points avec droite de régression quadratique

La régression quadratique (avec x au carré) nous indique ainsi que l'impact du niveau d'urbanisation est plus important à mesure que ce niveau 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*.

```
graphique <- ggplot(data = data_co2)+  
  geom_point(aes(x = Urbanisation, y = CO2t_hab), alpha = 0.2, size = 0.5)+  
  geom_smooth(aes(x = Urbanisation, y = CO2t_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')
```

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 ligne

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 == "Europe de l'Ouest")
# choix des valeurs pour l'axe des x
x_ticks <- seq(1960,2020,10)

ggplot(data = data_europe) +
  geom_path(aes(x = an, y = C02t_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()
```

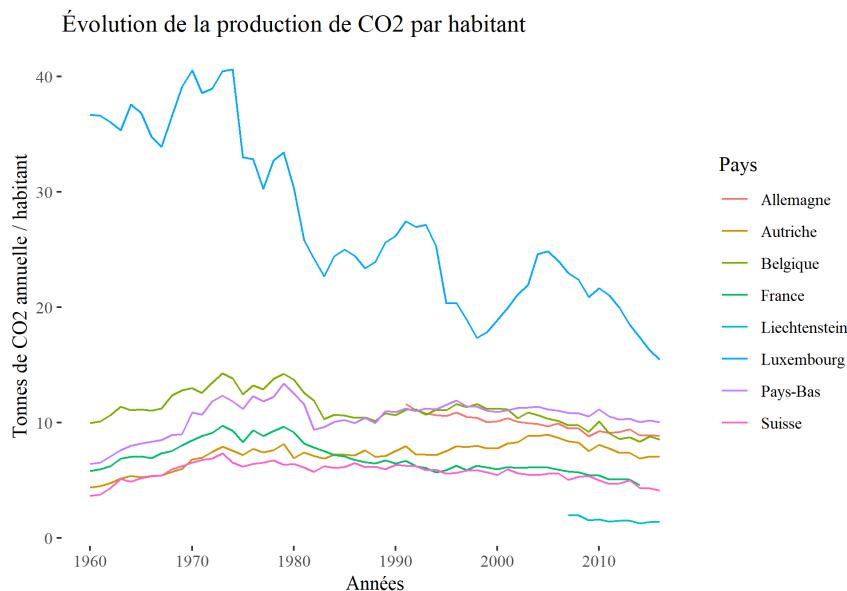


FIG. 3.35 : Graphique en ligne

Nous remarquons 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'erreur 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 fournie 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()
```

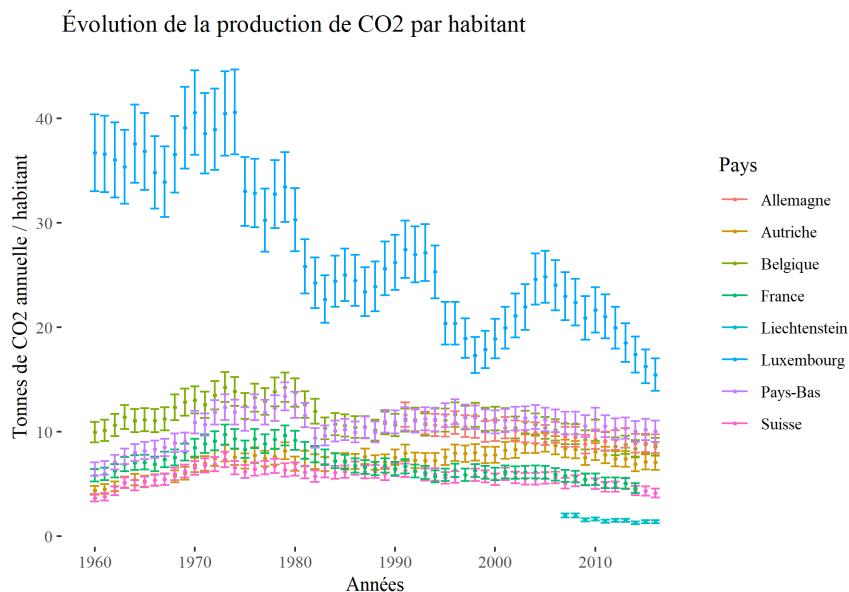


FIG. 3.36 : Graphique en ligne avec barres d'erreur

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)
```

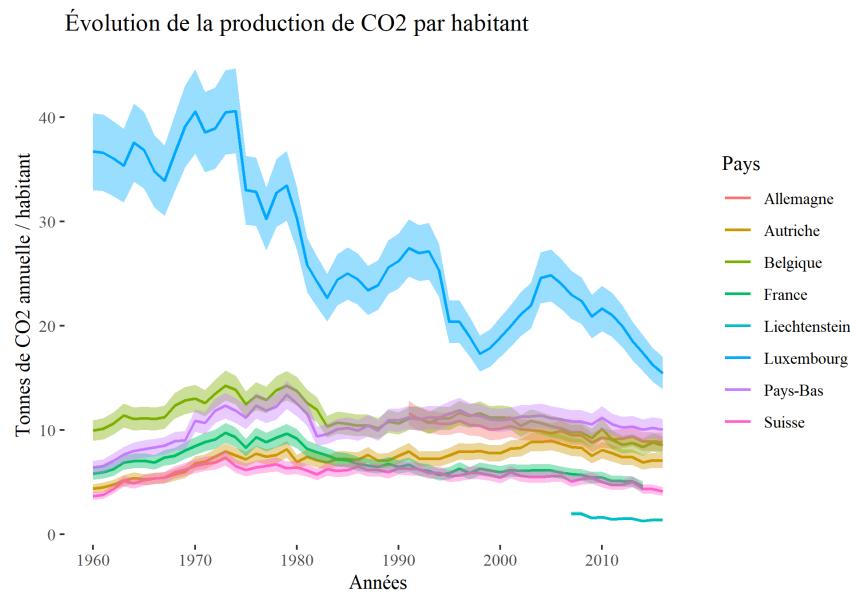


FIG. 3.37 : Graphique en ligne avec marge d'erreur

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'erreur. 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 cette incertitude est crucial pour que le lecteur puisse délimiter la portée des conclusions de vos analyses.

3.2.5 Boîte à moustaches

Les boîtes à moustaches (*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 nous reprenons 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 boîte représente la moyenne. Les extrémités de la boîte représentent le premier et le troisième quartile. Plus une boîte est allongée, plus les situations sont diversifiées pour les observations appartenant au groupe représenté par la boîte. Au contraire, une boîte étroite indique un groupe homogène. Notez qu'en inversant les variables dans les axes X et Y, nous obtiendrions des boîtes à moustaches 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)) +
```

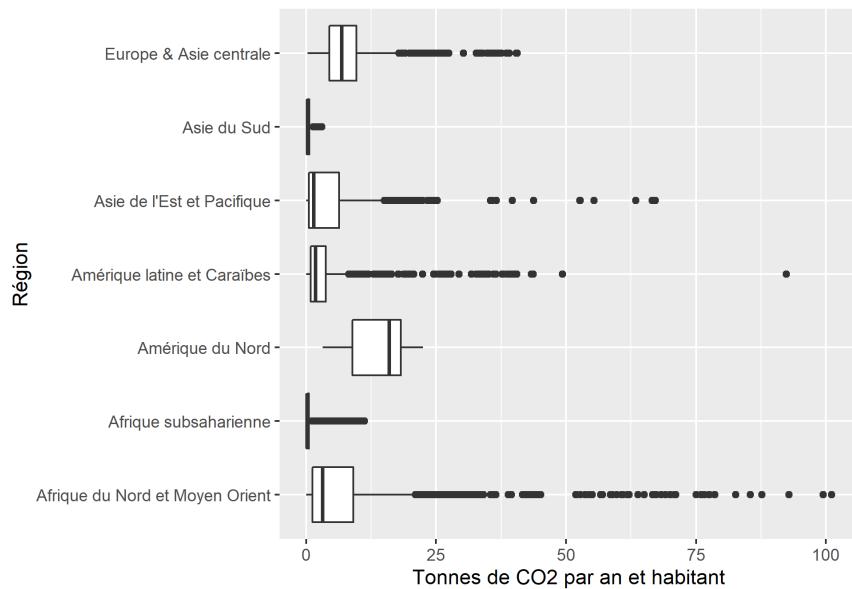


FIG. 3.38 : Boîtes à moustaches

```
xlim(c(0,50)) +
  labs(x = "Tonnes de CO2 par an et habitant",
       y = 'Région')
```

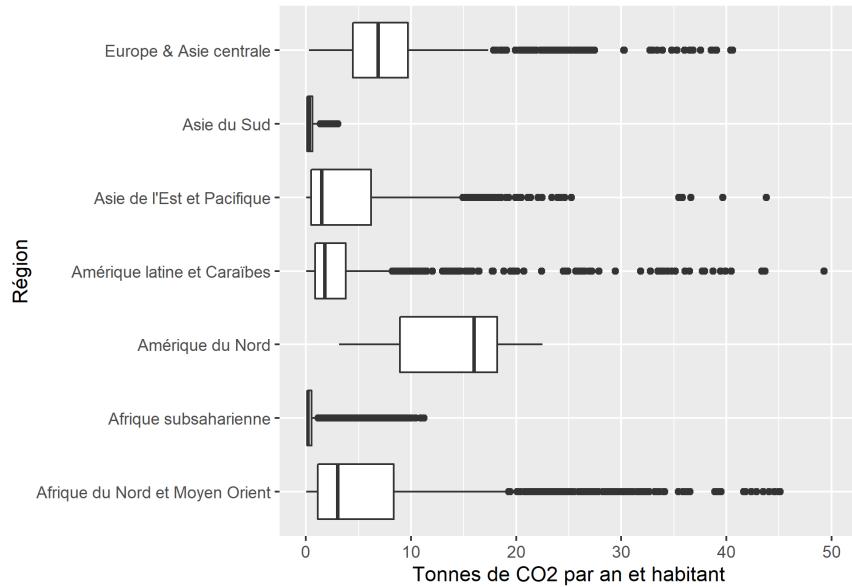


FIG. 3.39 : Boîtes à moustaches améliorées

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 moustaches, il est possible de les représenter directement avec la fonction `geom_jitter`.

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`

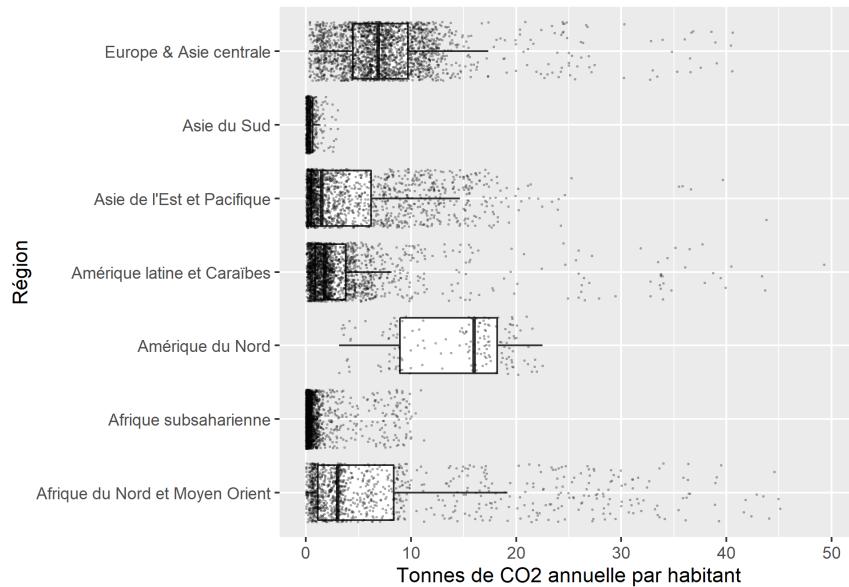


FIG. 3.40 : Boîtes à moustaches avec observations

= NA.

3.2.6 Graphique en violon

Les boîtes 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 boîte. Une solution est de remplacer la simple boîte par la distribution de la variable étudiée. Nous obtenons ainsi des graphiques en violon (`geom_violin`). Considérant les très grands écarts que nous avons observés entre les régions avec les boîtes à moustaches, 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)==FALSE)

ggplot(data = data_co2_comp)+
  geom_violin(aes(y = region7,x = CO2t_hab))+
  xlim(c(0,50))+
  labs(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 inférieures à 12 tonnes (ligne bleue) et l'autre pour lequel elles sont supérieures. En explorant les données, nous constatons que les Bermudes appartiennent au 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 un graphique en violon et une boîte à moustaches pour bénéficier des avantages des deux.

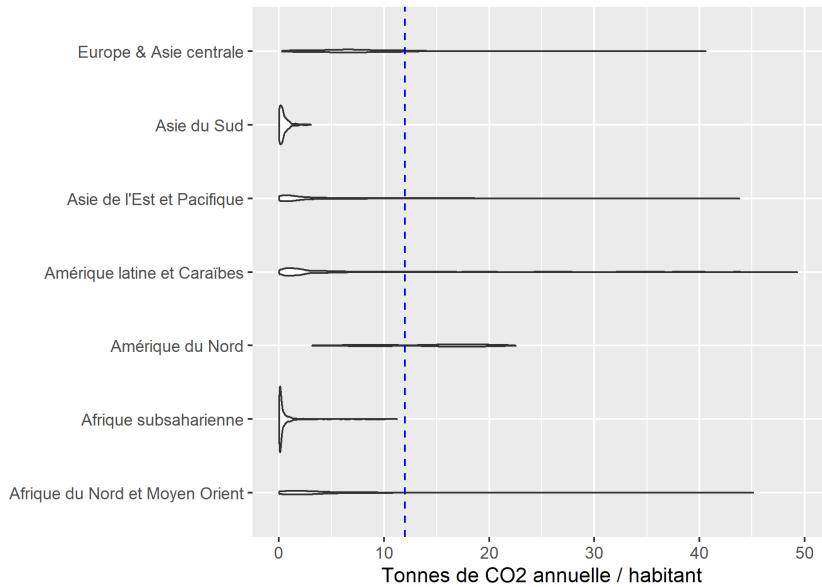


FIG. 3.41 : Graphiques en violon

```
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(x = "Tonnes de CO2 annuelle / habitant",  
       y = '')
```

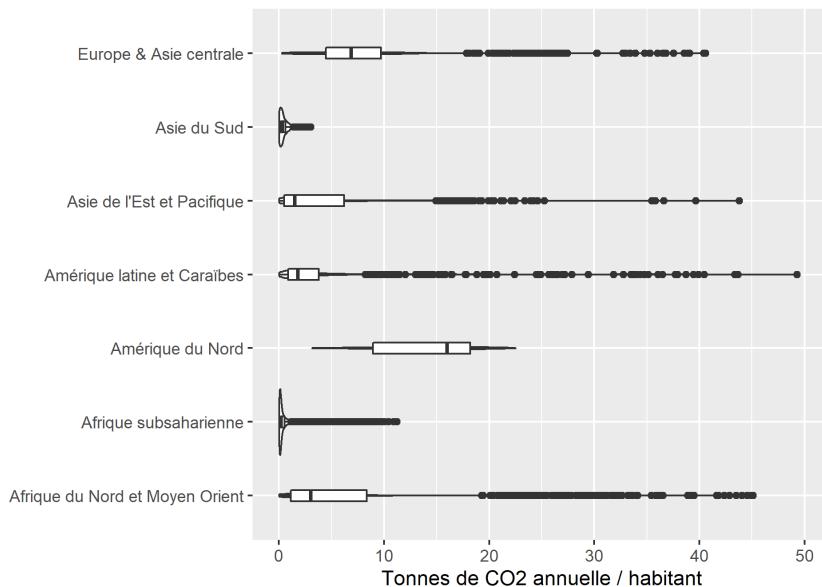


FIG. 3.42 : Graphiques en violon et boîtes à moustaches

3.2.7 Graphique en barre

Les graphiques en barre 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 2017 - Région Québec-Lévis*, au niveau des grands secteurs. La figure 3.42, tirée du rapport⁶ intitulé *La mobilité des personnes dans la région de Québec-Lévis (Volet Enquête-ménages : faits saillants)* délimite ces grands secteurs.

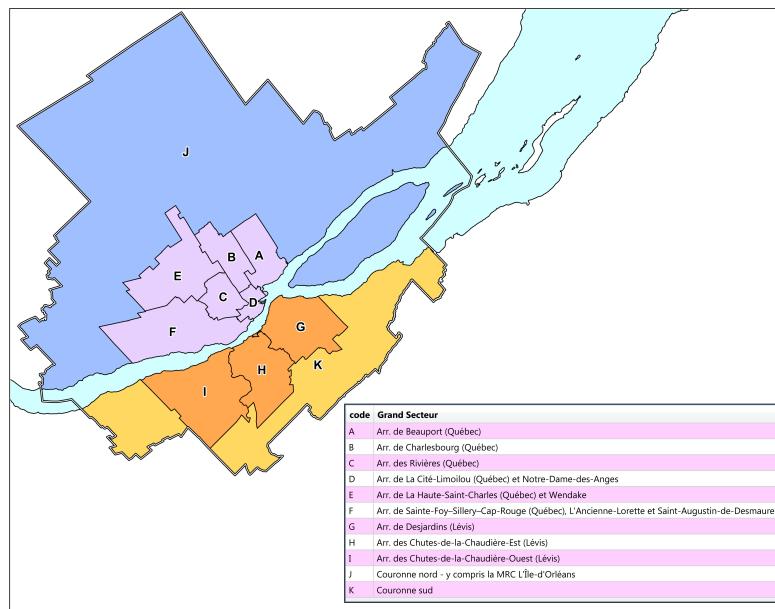


FIG. 3.43 : Grands secteurs de Québec

Nous représentons pour chaque secteur le nombre moyen de déplacements 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 au 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 au secteur A. En sommant les valeurs de chaque ligne, nous obtenons 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',
                      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)',
```

⁶https://www.transports.gouv.qc.ca/fr/ministere/Planification-transports/enquetes-origine-destination/quebec/2017/Documents/EOD17_faits_saillants_VF.pdf

```

'Arr. de Charlesbourg (Québec)',
'Arr. des Rivières (Québec)',
'Arr. de la Cité-Limoilou (Québec)',
'Arr. de la Haute-Saint-Charles (Québec)',
'Arr. de Sainte-Foy-Sillery-Cap-Rouge (Québec)',
'Arr. de Desjardins (Lévis)',
'Arr. des Chutes-de-la-Chaudière-Est (Lévis)',
'Arr. des 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 barre
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 nous concaténons en premier les lettres et les noms.
# Nous fusionnons 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"
# 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
)

```

Plutôt que de représenter les arrivées et les départs dans deux graphiques séparés, il est possible de les empiler dans un même graphique en barre. 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', 'départs', 'arrivées')

```

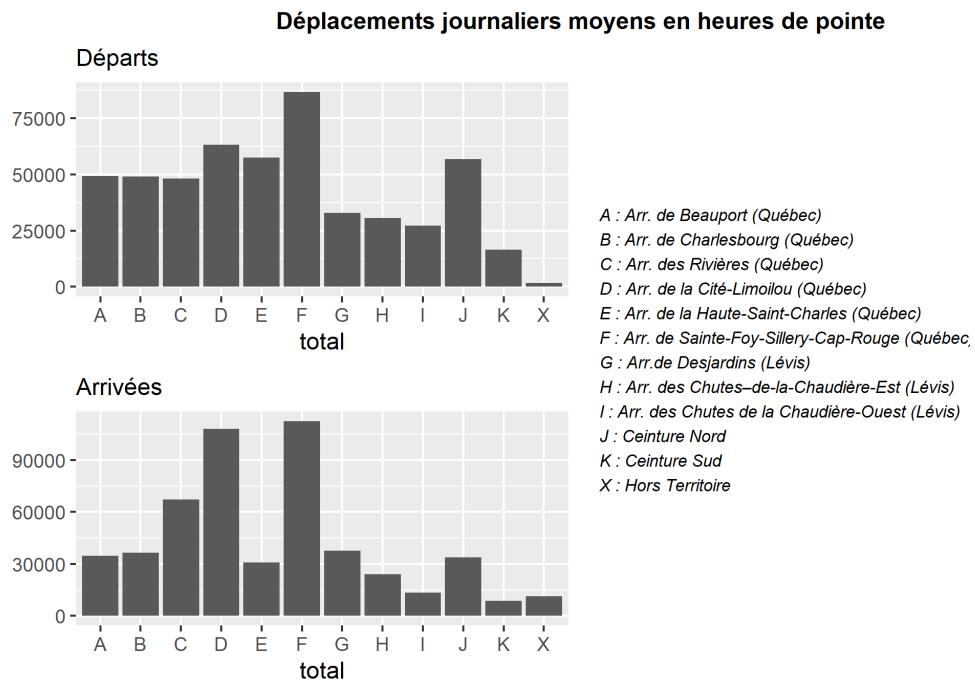


FIG. 3.44 : Graphiques en barre simples

```
# Comparaison du format original et du format "fondu"
head(df)
```

```
##      depart arrivee
## V1    49241   34777
## V2    48909   36344
## V3    48044   67198
## V4    63132  108138
## V5    57367   30859
## V6    86504  112379
                                secteur code
Arr. de Beauport (Québec)     A
Arr. de Charlesbourg (Québec)  B
Arr. des Rivières (Québec)    C
Arr. de la Cité-Limoilou (Québec) D
Arr. de la Haute-Saint-Charles (Québec) E
Arr. de Sainte-Foy-Sillery-Cap-Rouge (Québec) F
```

```
head(melted_df)
```

```
##      code deplacement effectif
## 1      A    départs    49241
## 2      B    départs    48909
## 3      C    départs    48044
## 4      D    départs    63132
## 5      E    départs    57367
## 6      F    départs    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)

```

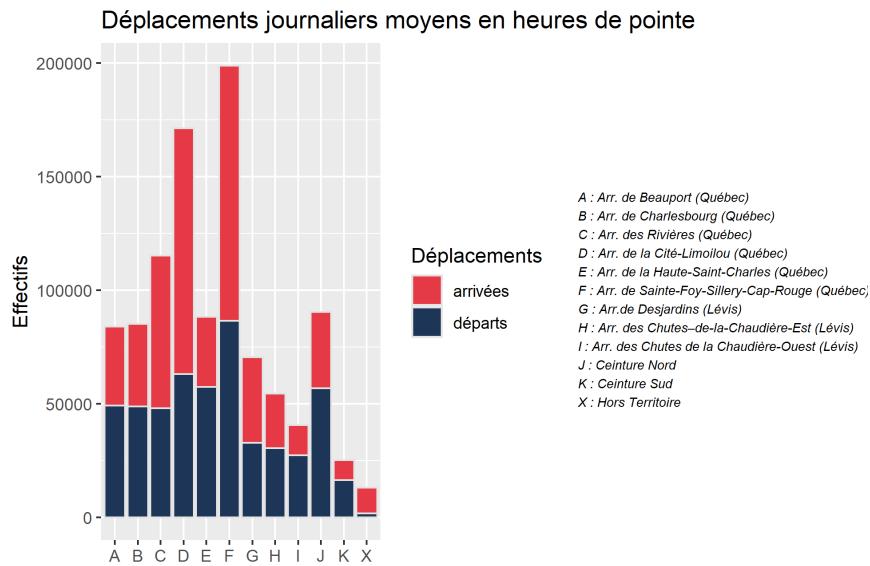


FIG. 3.45 : Graphique en barre empilée

3.2.8 Graphique circulaire

Une option directe au graphique en barre est le graphique ou diagramme circulaire, appelé aussi graphique en tarte (pour les personnes à la dent sucrée) ou en camembert (pour celles amatrices de fromage). Il est suffisamment connu et utilisé pour qu'aucune présentation ne s'impose. Pour être exact, un graphique en tarte n'est rien d'autre qu'un graphique en barre dont le système de coordonnées a été modifié. Cela impose cependant de calculer à l'avance la position des étiquettes que nous souhaitons 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 nous connaissons 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

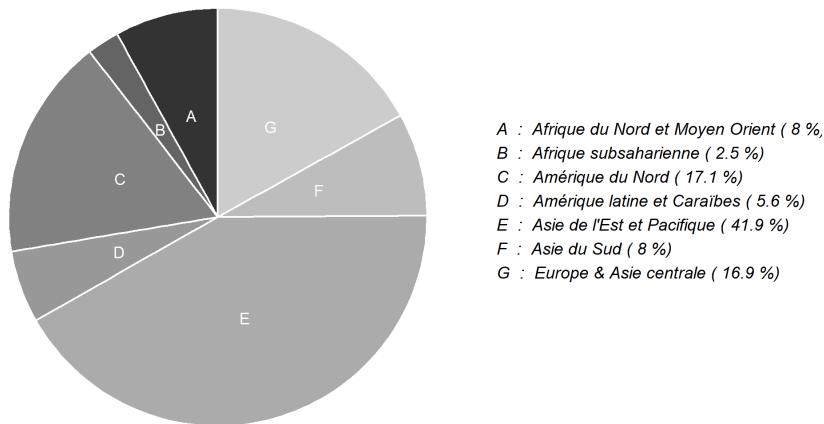


FIG. 3.46 : Graphique en tarte

Si à la place de la géométrie `geom_bar`, vous utilisez `geom_rect`, vous pouvez convertir votre graphique en tarte en graphique en anneau (ou en beigne, pour les personnes à la dent sucrée) :

```

# Calculer la limite inférieure et supérieure du beigne
df$ymax <- cumsum(df$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)

```

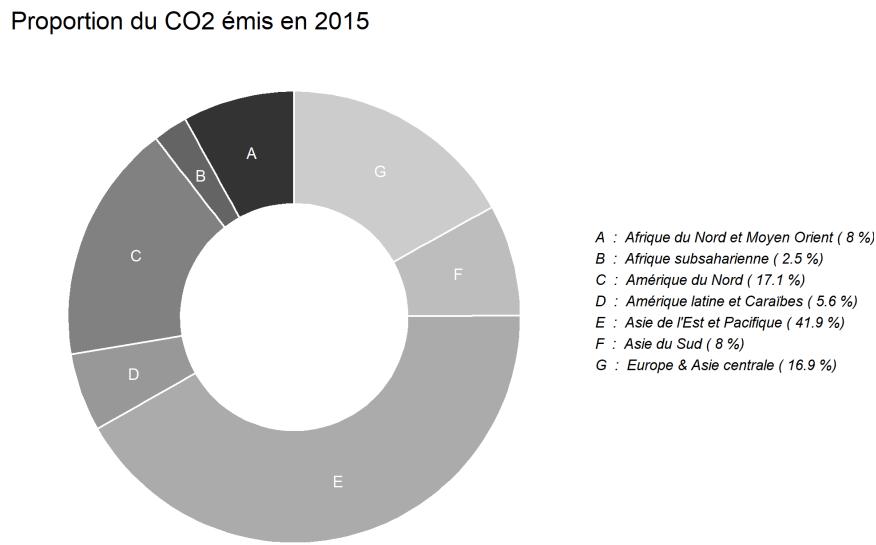


FIG. 3.47 : Graphique en anneau

3.3 Graphiques spéciaux

Dans cette dernière section, nous abordons 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 groupes d'observations. Chaque variable est associée à un axe et chaque observation est représentée avec un polygone. Prenons l'exemple de données relatives aux logements par secteur de recensement dans la région métropolitaine de Montréal en 2016. Nous pourrions 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 devons donc utiliser le package `fmsb`.

```

library(fmsb)

data <- read.csv('data/bivariee/sr_rmr_mtl_2016.csv', header = T, encoding = 'UTF-8')

# Agréger 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
                    )

# Choisir les 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)
)

# Choisir les 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)
)

# Dessiner du graphique
radarchart(data_chart,
            title = "Comparaison des types de logements dans la RMR",
            pcol = couleurs_contour, pfcol = couleurs,
            plwd = 2, plty=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"))

```

```

)
# Ajouter 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

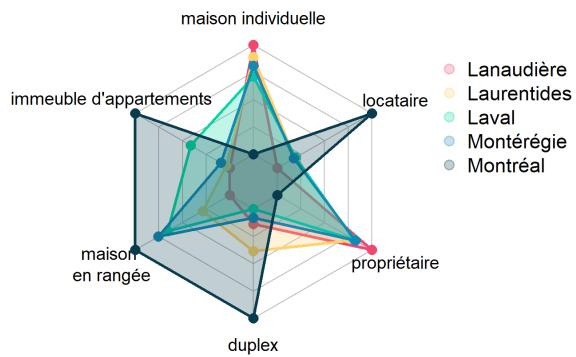


FIG. 3.48 : Graphique en anneau

À la lecture du graphique, nous constatons rapidement que l'île de Montréal a une situation très différente 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 multidimensionnel, mais il présente 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 solution 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 qui est 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 personnes via un réseau social, de flux de population, etc. Reprenons nos données de l'*Enquête origine-destination 2017 - Région Québec-Lévis* pour illustrer le tout. Nous utilisons le package *chorddiag*, très facile d'utilisation et produisant des graphiques interactifs, pour faciliter grandement la lecture de ce type de graphique. Cependant, ce package ne fait pas partie du répertoire CRAN, nous devons 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

# Nous supprimons 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 HTLM
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). Nous pouvons cependant constater que les secteurs des Rivières, de la Cité-Limoilou et de la Haute-Saint-Charles at-

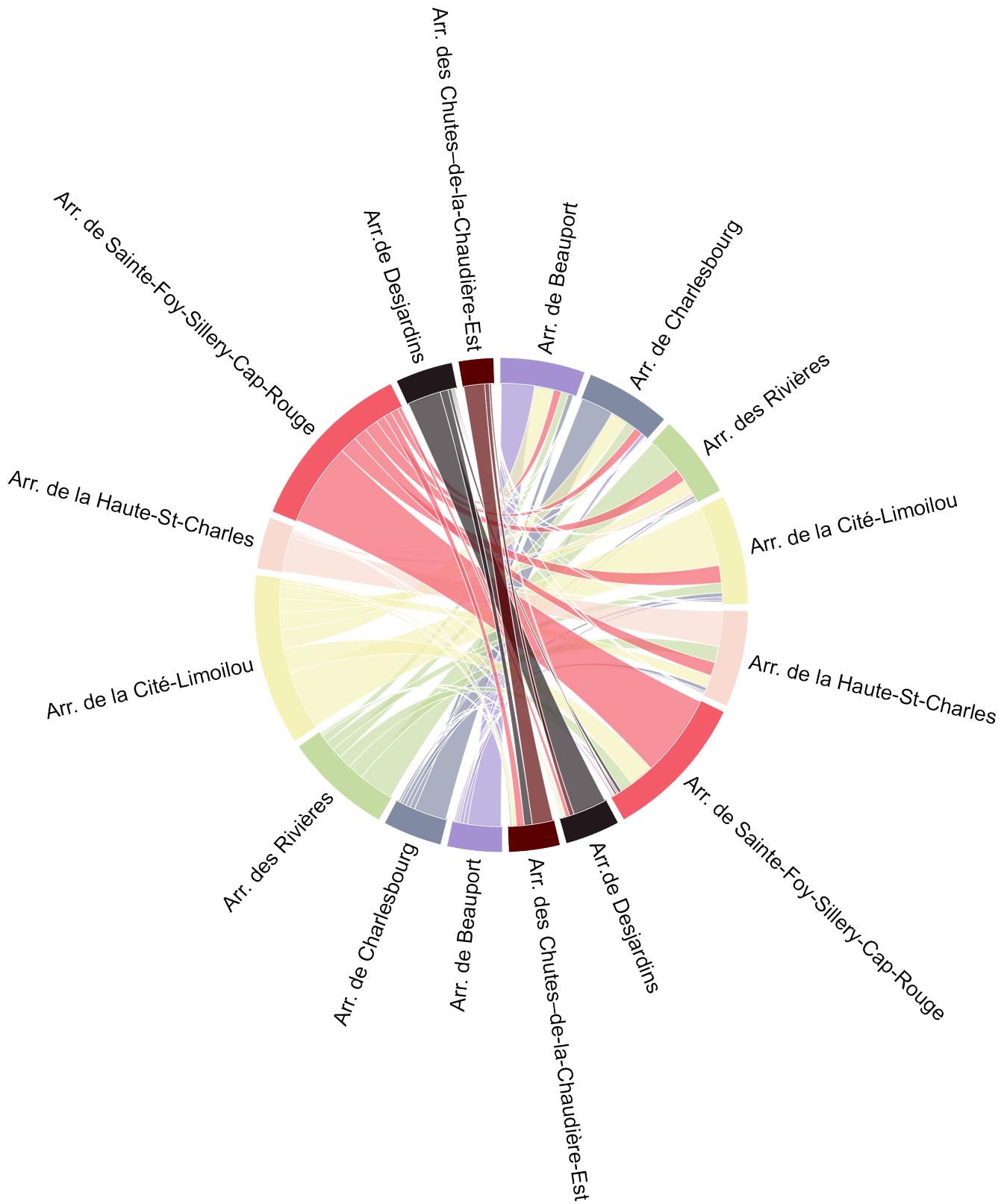


FIG. 3.49 : Diagramme d'accord

tirent 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 graphes) proposée par Mihalcea et Tarau (2004) et implémentée dans le package `textrank`. Nous avons 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 de 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 (nom, adjetif, verbe, 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, comme 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)

# 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_montral_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 sauts 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)

# Étape 5 : analyse de la nature des mots du texte avec le dictionnaire fr
# Nous obtenons des DataFrames décrivant les mots des textes
annotate_qc <- udpipe_annotate(model, text_qc)
df_qc <- data.frame(annotate_qc)

annotate_mtl <- udpipe_annotate(model, text_mtl)
df_mtl <- data.frame(annotate_mtl)

# Étape 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)

# Étape 7 : nettoyer les coquilles dans les mots clefs
# Note : 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)

# Étape 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)

```

Notez qu'à chaque génération du nuage de mots, vous obtiendrez une disposition différente. N'hésitez



FIG. 3.50 : Nuage de mots pour le SAD de Montréal

pas à en essayer plusieurs jusqu'à ce que vous trouviez celle qui vous semble optimale.

3.3.4 Carte proportionnelle

Une carte proportionnelle ou carte à cases (*treemap* en anglais) 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 *treemap*. La variable de quantité est bien sûr les émissions de CO₂ par pays ; ces pays sont regroupés dans un premier ensemble de régions (découpage en 23 régions), qui elles-mêmes sont regroupées dans des régions plus larges (découpage en sept régions). Pour construire un *treemap*, nous allons utiliser le package *treemap*.

```
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",
```



FIG. 3.51 : Nuage de mots pour le SAD de Québec

```

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 boîtes
overlap.labels=0.5, # tolérance de superposition
inflate.labels=F, # agrandir la taille des étiquettes ou non
palette = brewer.pal(7,'Paired')
)
  
```

3.4 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

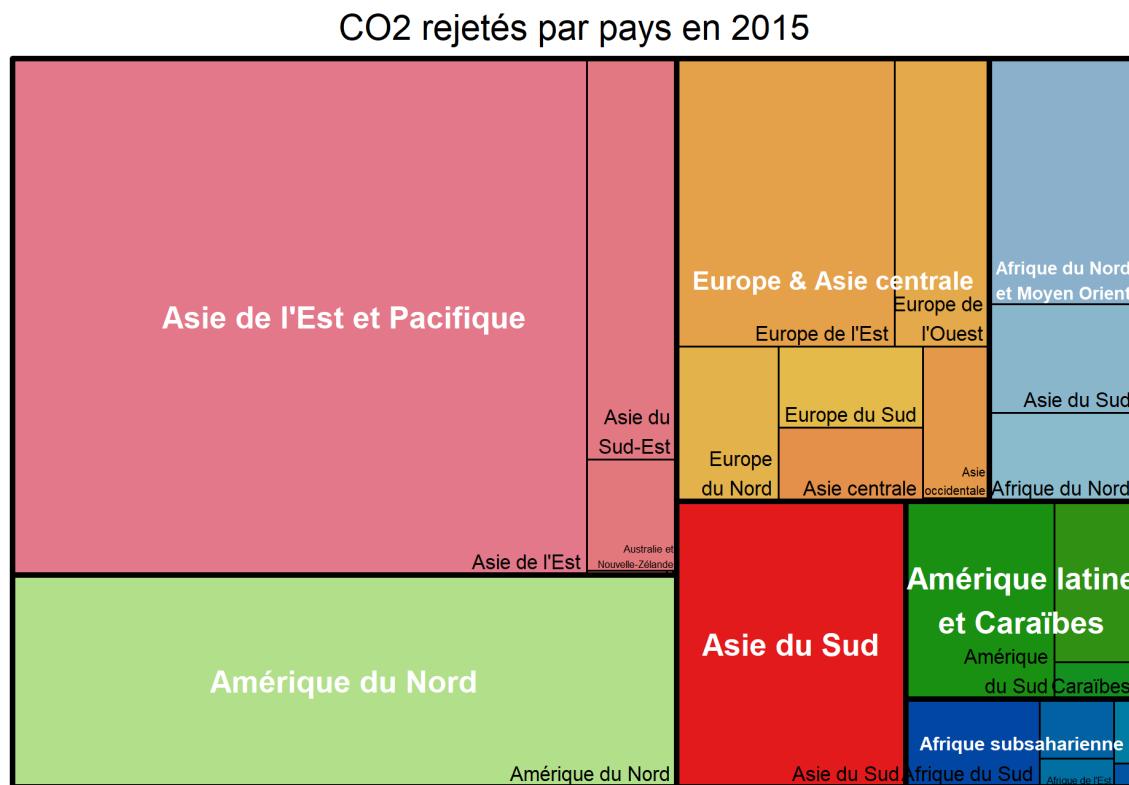


FIG. 3.52 : Treemap

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 de *DataFrame*, mais respectivement un *SpatialDataFrame* et un objet `sf` (*spatial feature collection*). Sans entrer dans les détails, sachez que deux *packages* permettent de manipuler des objets spatiaux dans R : le traditionnel `sp` (avec les *SpatialDataFrames*) et le plus récent `sf` (avec les *spatial feature collections*). 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 `sp` et `sf` peuvent être utilisés sans distinction. En revanche, pour cartographier directement avec `ggplot2`, il est plus facile d'utiliser un objet de type `sf`. Toutefois, nous vous recommandons fortement d'utiliser le package `sf`, puisque `sp` (et son format *SpatialDataFrame*) est progressivement délaissé dans R.

Une carte thématique permet de représenter la répartition spatiale de variables qualitatives ou quantitatives. Nous la distinguons des cartes topographiques, dont l'objectif est de représenter la localisation d'objets spécifiques (route, habitation, rivière, lac, etc.). La première est relativement facile à construire dans R, car elle se limite à quelques symboles relativement simples. Pour la seconde, nous préférons généralement utiliser un logiciel comme QGIS⁷.

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 avons aussi besoin des *packages* `classInt` pour calculer les intervalles des classes et `ggsn` pour afficher une échelle.

⁷<https://qgis.org/en/site/>

```

library(sf)
library(classInt)
library(ggsn)

# chargement des données
spatialdf <- st_read("data/bivariee/IlotsVeg2006.shp")

## Reading layer `IlotsVeg2006' from data source
## `D:\Articles et colloque\Livre en cours\AnalysesQuanti\Livre\livre_statistique_Phil_Jere\data\bivariee\'
##   using driver `ESRI Shapefile'
## Simple feature collection with 10213 features and 12 fields
## Geometry type: MULTIPOLYGON
## Dimension:      XY
## Bounding box:  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 îlots 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)

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)) +

```

Végétation dans les îles de recensement

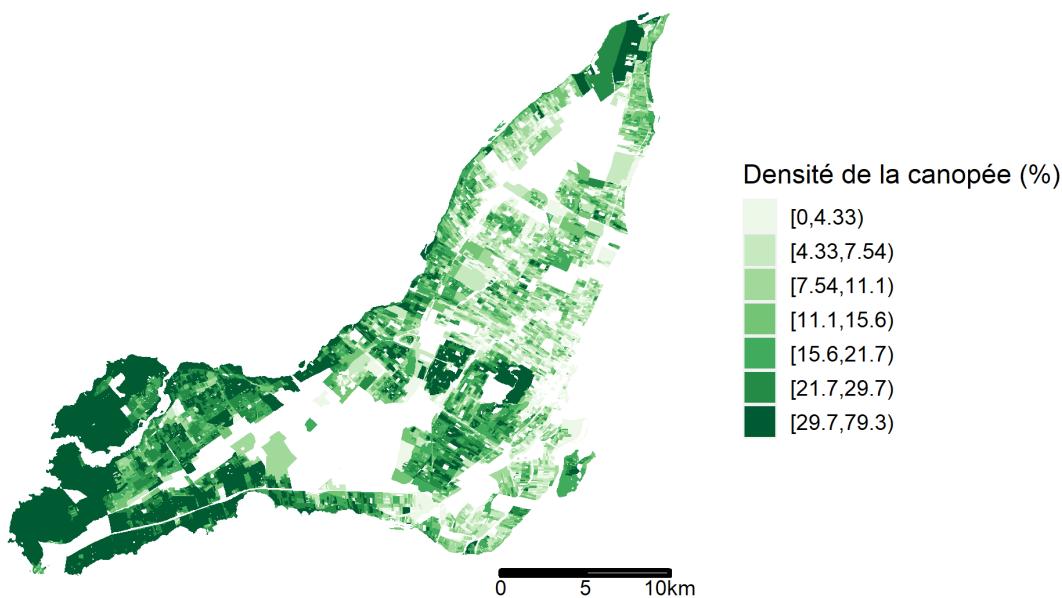


FIG. 3.53 : Carte thématique avec ggplot2

```
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) +
  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)
```

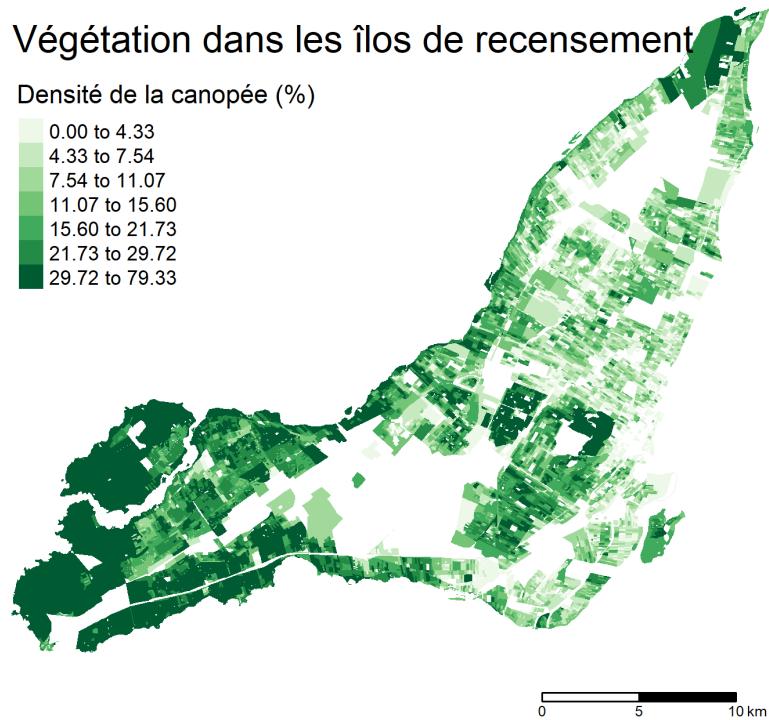


FIG. 3.54 : Carte thématique avec tmap

```
tmap_arrange(carte1, carte2, ncol = 2)
```

3.5 Exportation 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 3.56) pour enregistrer votre figure au format image ou au format *pdf* (vectoriel). Notez qu'avec la seconde option, vous pourrez retoucher votre graphique avec un logiciel externe comme *Inkscape* ou *Illustrator*.

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.

```
data(iris)

plot1 <- ggplot() +
  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*, la solution de remplacement à la fonction *ggsave*

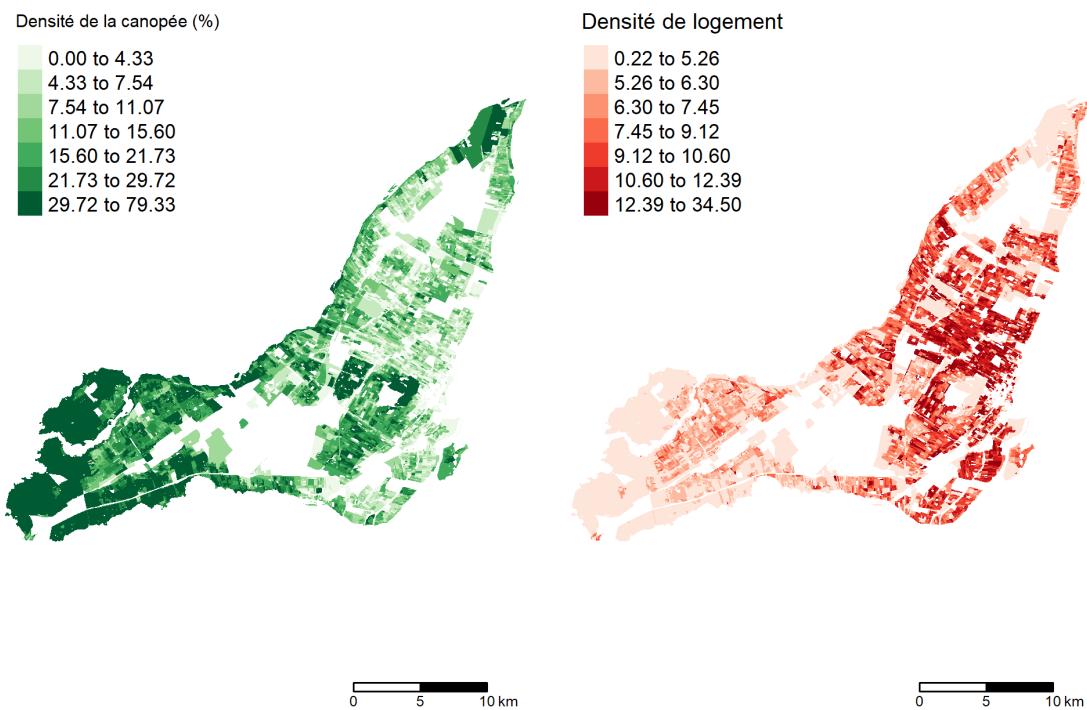


FIG. 3.55 : Combiner des cartes avec tmap

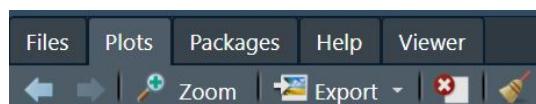


FIG. 3.56 : Exporter un graphique dans RStudio

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 nous souhaitions 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 qu'observé la partie émergée de l'iceberg dans ce chapitre. Il est également possible de réaliser une visualisation en 3D dans R (`plot3D`, `rgl`), d'animer des graphiques pour en faire des *GIF* ou des vidéos (`gganimate`), 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 Gallery⁸, probablement LE site web proposant le plus de matériel sur la réalisation des graphiques dans R.
- Data to viz⁹, 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.

⁸<https://www.r-graph-gallery.com/>

⁹<https://www.data-to-viz.com/>

Troisième partie

Analyses bivariées

Chapitre 4

Relation linéaire entre deux variables quantitatives

Dans le cadre de ce chapitre, nous présentons les trois principales méthodes permettant d'explorer la relation linéaire entre deux variables quantitatives, soit la covariance, la corrélation et la régression linéaire simple.



Dans ce chapitre, nous utilisons les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2`, le seul, l'unique!
 - * `ggpubr` pour combiner des graphiques et réaliser des diagrammes quantiles-quantiles.
- Pour manipuler des données :
 - * `dplyr` notamment pour les fonctions `group_by`, `summarize` et les pipes `%>%`.
- Pour les corrélations :
 - * `boot` pour réaliser des corrélations avec *bootstrap*.
 - * `correlation`, de l'ensemble de packages `easy_stats`, offrant une large panoplie de mesures de corrélation.
 - * `corrplot` pour créer des graphiques de matrices de corrélation.
 - * `Hmisc` pour calculer des corrélations de Pearson et Spearman.
 - * `ppcor` pour calculer 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 de beaux tableaux d'une matrice de corrélation en HTML, en LaTeX ou en ASCII.
- Autres *packages* :
 - * `foreign` pour importer des fichiers externes.
 - * `MASS` pour générer des échantillons normalement distribués.
 - * `stargazer` pour imprimer des tableaux.



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, l'autre augmente également ou diminue systématiquement.

En études urbaines, nous pourrions 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, nous pourrions vouloir vérifier si le revenu médian des ménages et le coût moyen du loyer varient dans le même sens que la couverture végétale ; ou encore s'ils varient 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 4.2) et la **corrélation** (section 4.3).

4.1 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 4.1 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, nous observons 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.

Prenons un exemple concret pour illustrer le cas **c**. 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). Nous pouvons supposer que les secteurs récemment construits, surtout ceux dans les banlieues, présentent des niveaux de végétation plus faibles. Au fur et à mesure 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 4.2).

Dans les sous-sections suivantes, nous décrivons deux statistiques descriptives et exploratoires – la covariance (section 4.2) et la corrélation (section 4.3) – utilisées pour évaluer la **relation linéaire** entre deux variables continues (cas **a** et **b** à la figure 4.1). 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

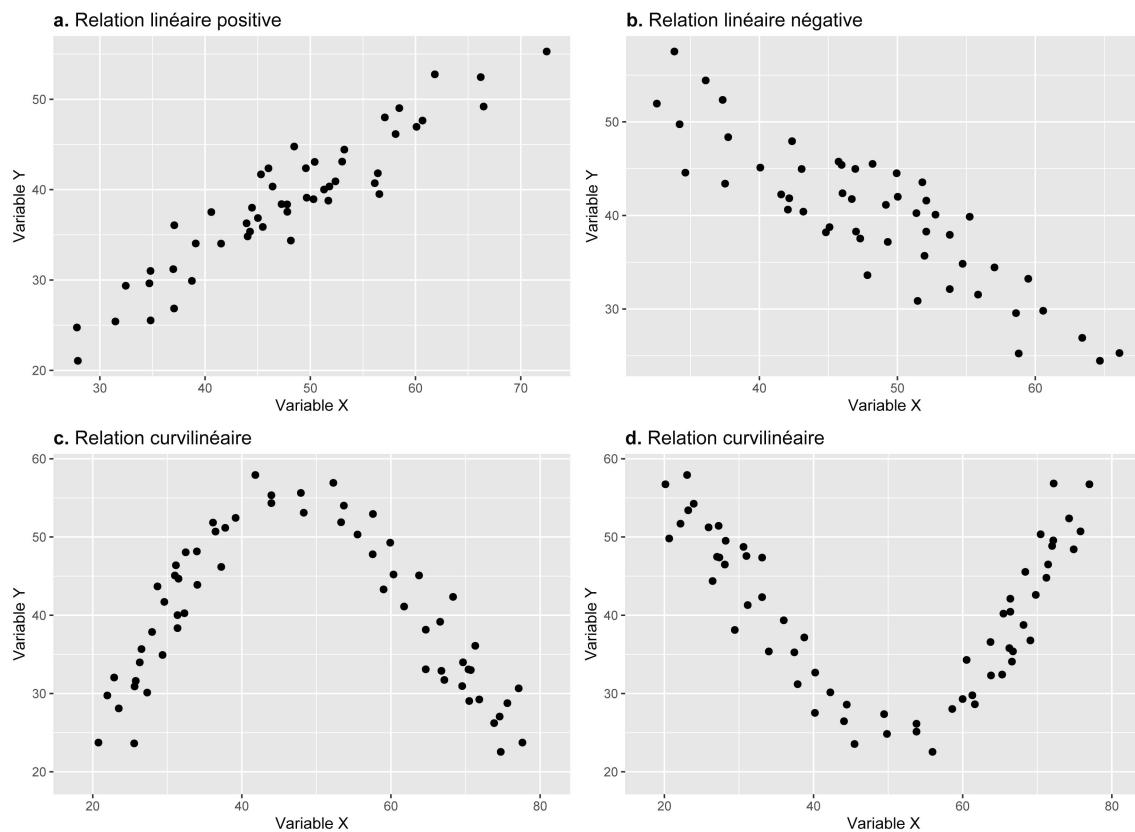


FIG. 4.1 : Relations linéaires et curvilinéaires entre deux variables continues

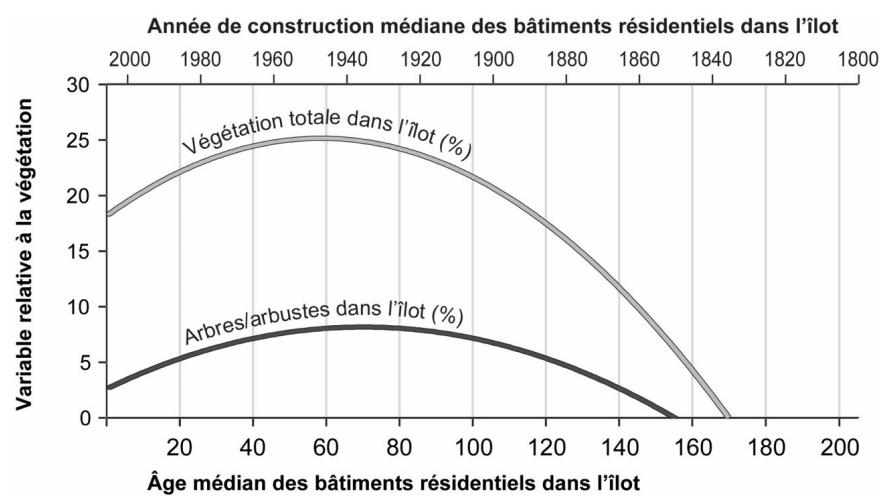


FIG. 4.2 : Exemples de relations curvilinéaires

variable indépendante (variable explicative). Puis, nous décrivons la régression linéaire simple (section 4.4) qui permet justement de prédire une variable dépendante (Y) à partir d'une variable indépendante (X).

4.2 Covariance

4.2.1 Formulation

La covariance (équation 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 :

$$cov(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1} = \frac{\text{covariation}}{n - 1} \quad (4.1)$$

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 .

4.2.2 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 lorsque les valeurs de la variable X s'éloignent de la moyenne, les valeurs de Y s'éloignent aussi dans le même sens; et elle est 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 km^2) au niveau des secteurs de recensement de la région métropolitaine de Montréal, nous obtenons une valeur de covariance de 33 625. En revanche, si la densité de population est exprimée en milliers d'habitants au km^2 , la valeur de la covariance sera de 33,625, alors que la relation linéaire entre les deux variables reste la même comme illustré à la figure 4.3. Pour remédier à ce problème, nous privilégions l'utilisation du coefficient de corrélation.

4.3 Corrélation

4.3.1 Formulation

Le coefficient de corrélation de Pearson (r) est égal à la covariance (numérateur) divisée par le produit des écarts-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 qu'il faut soustraire de chaque valeur sa moyenne correspondante et la diviser par son écart-type. Il correspond ainsi à la moyenne du produit des deux variables centrées réduites. Il s'écrit alors :

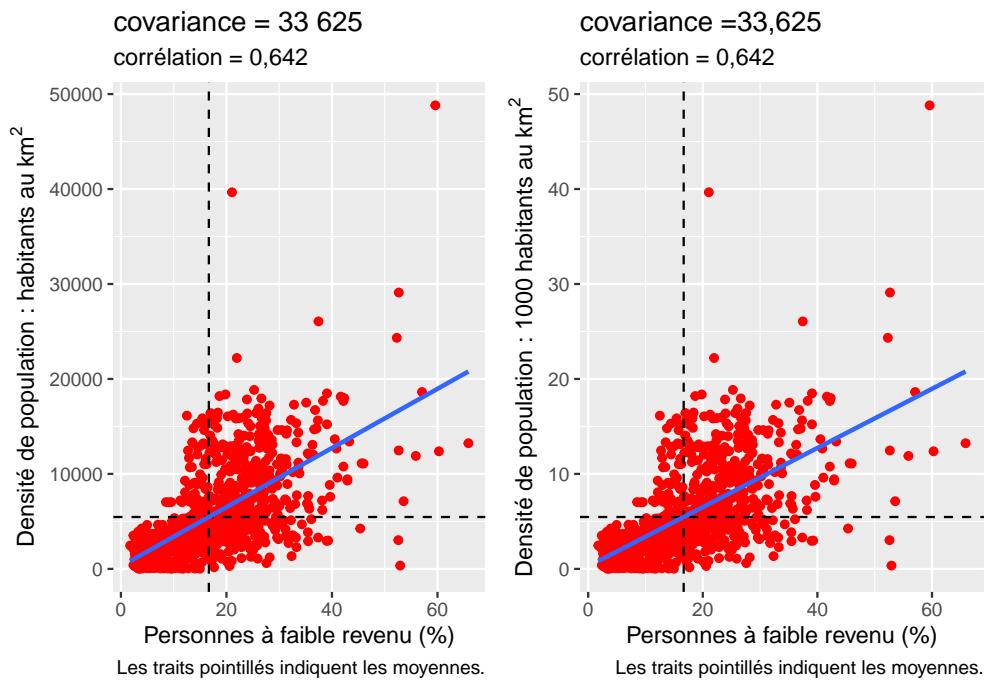


FIG. 4.3 : Covariance et unités de mesure

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2(y_i - \bar{y})^2}} = \sum_{i=1}^n \frac{Zx_i Z y_i}{n-1} \quad (4.2)$$

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
df1 <- as.data.frame(mvrnorm(N, c(moy_x, moy_y), cov))
# Centrage et réduction des deux variables
df1$zV1 <- scale(df1$V1, center = TRUE, scale = TRUE)
df1$zV2 <- scale(df1$V2, center = TRUE, scale = TRUE)
# Corrélation de Pearson
cor1 <- cor(df1$V1, df1$V2)
cor2 <- sum(df1$zV1*df1$zV2) / (nrow(df1)-1)
cat("Corrélation de Pearson = ", round(cor1,5),
    "\nMoyenne du produit des variables centrées-réduites =", round(cor2,5))
## Corrélation de Pearson = 0.81297
```

```
## Moyenne du produit des variables centrées-réduites = 0.81297
```

4.3.2 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 relation linéaire négative parfaite;
- 1 quand il y a une relation linéaire positive parfaite (figure 4.4).

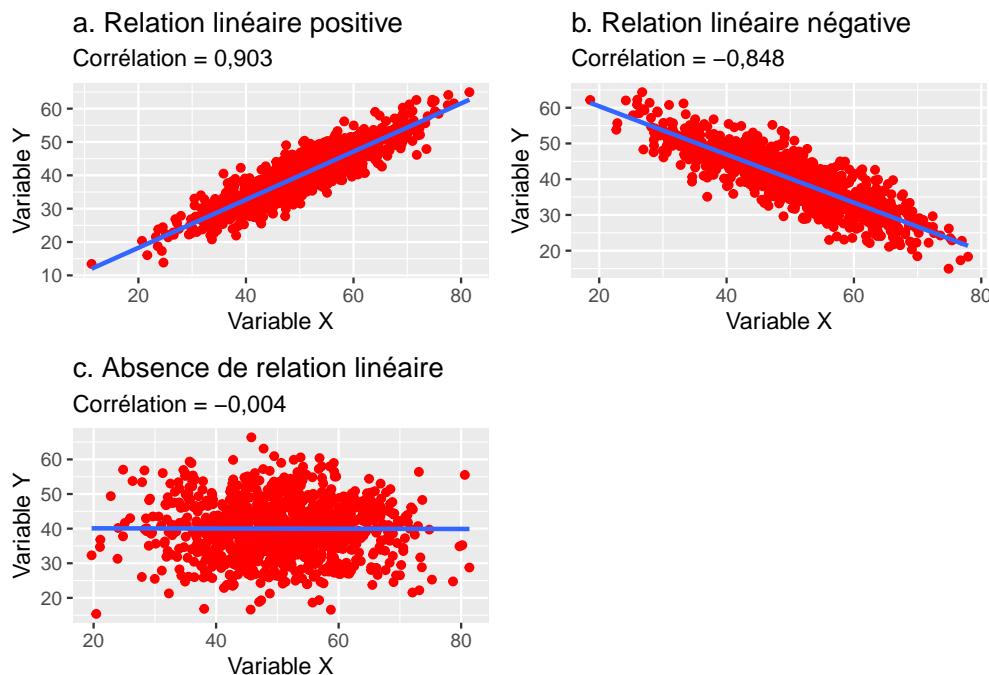


FIG. 4.4 : Relations entre deux variables continues et coefficients de corrélation de Pearson

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, nous utilisons habituellement les intervalles de valeurs reportés 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 $0,2$ est 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.

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 $1,0$

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 4.5).

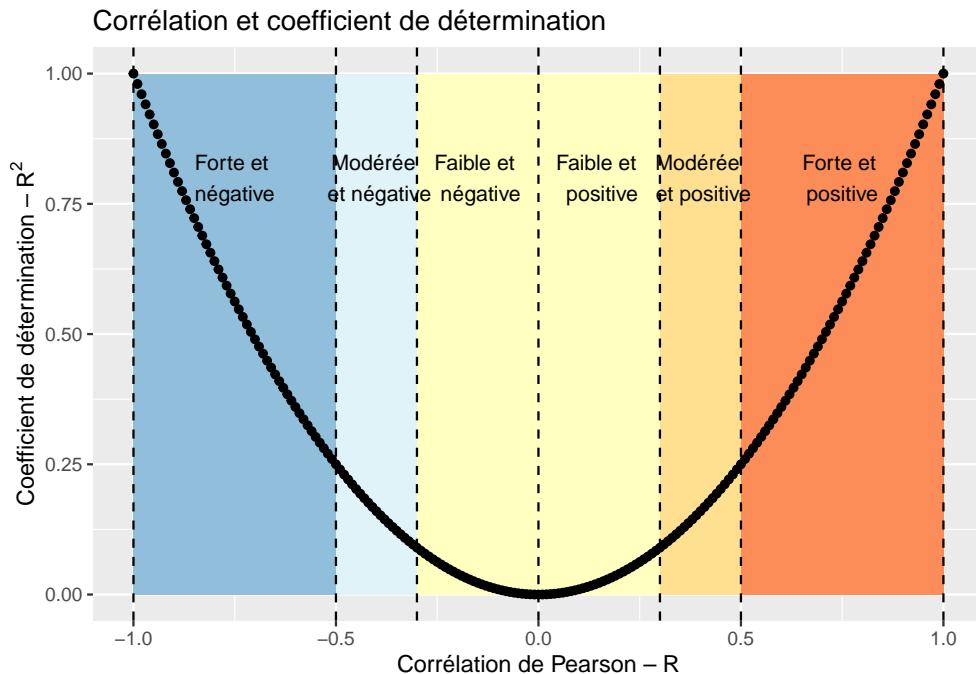


FIG. 4.5 : Coefficient de corrélation et proportion de la variance expliquée

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 ou extrêmes. D'ailleurs, plus le nombre d'observations est réduit, plus la présence de valeurs extrêmes a une répercussion importante sur le résultat du coefficient de corrélation de Pearson. En guise d'exemple, dans le nuage de points à gauche de la figure 4.6, il est possible d'identifier des valeurs extrêmes qui se démarquent nettement dans le jeu de données : six observations avec une densité de population supérieure à $20\,000$ habitants au km^2 et deux observations avec un pourcentage de 65% et plus supérieur à 55% . Si l'on supprime ces observations (ce qui est défendable dans ce contexte) – soit moins d'un pour cent 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.

4.3.3 Corrélations pour des variables anormalement distribuées (coefficient 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 (ρ) et secondairement, le tau (τ) de Kendall, 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 covarianc e des deux variables de rangs sur les écarts-types des variables de rangs. En d'autres termes, il représente simplement le coefficient de Pearson calculé sur les rangs des deux variables :

$$r_{xy} = \frac{\text{cov}(rg_x, rg_y)}{\sigma_{rg_x} \sigma_{rg_y}} \quad (4.3)$$

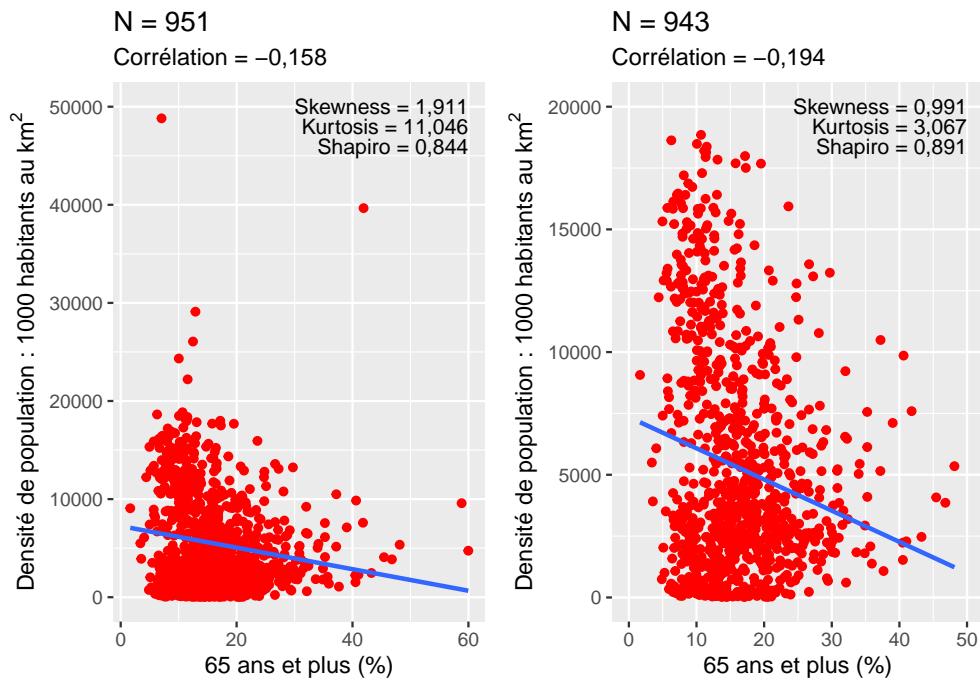


FIG. 4.6 : Illustration de l'effet des valeurs extrêmes sur le coefficient de Pearson

La syntaxe ci-dessous démontre clairement que le coefficient de Spearman est bien le coefficient de Pearson calculé sur les rangs (4.3.1).

```
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 écarts-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 écarts-types sur les rangs : -0.11953
```

Le **tau 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 total de paires d'observations. Des paires sont dites concordantes 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 du tau Kendall peut être chronophage : plus le nombre d'observations est élevé, plus les temps de calcul et la mémoire utilisée sont importants. En effet, avec $n=1000$, le nombre de paires d'observations ($0,5 \times n(n - 1)$) est de 499 500, contre près de 50 millions avec $n=10\ 000$ (49 995 000).

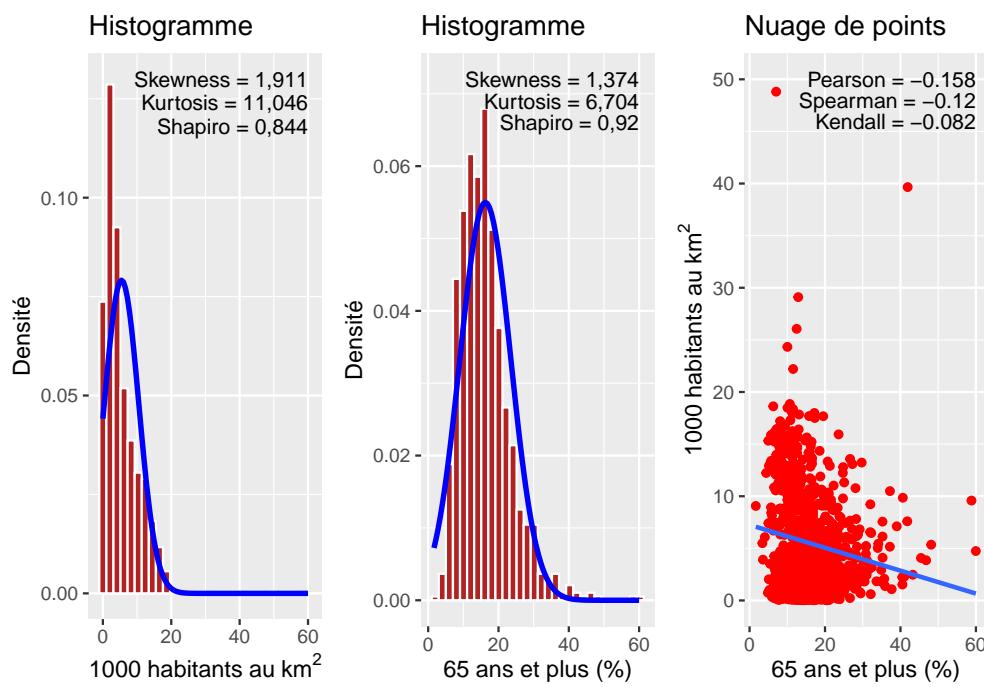


FIG. 4.7 : Comparaison des coefficients de Pearson, Spearman et Kendall sur deux variables anormalement distribuées

À la lecture des deux histogrammes à la figure 4.7, il est clair que les 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 existant entre les deux variables. Notez que les coefficients de Spearman et de Kendall sont tous les deux plus faibles.

4.3.4 Corrélations robustes (*Biweight midcorrelation*, *Percentage bend correlation* et la corrélation *pi* de Shepherd)

Dans l'exemple donné à la figure 4.6, nous avions identifié des valeurs extrêmes et les 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 pour des jeux de données présentant potentiellement des valeurs extrêmes. Elle est calculée comme suit :

$$\begin{aligned}
 u_i &= \frac{x_i - \text{med}(x)}{9 \times (\text{med}(|x_i - \text{med}(x)|))} \text{ et } v_i = \frac{y_i - \text{med}(y)}{9 \times (\text{med}(|y_i - \text{med}(y)|))} \\
 w_i^{(x)} &= (1 - u_i^2)^2 I(1 - |u_i|) \text{ et } w_i^{(y)} = (1 - v_i^2)^2 I(1 - |v_i|) \\
 I(x) &= \begin{cases} 1, \text{ si } x = 1 \\ 0, \text{ sinon} \end{cases} \\
 \tilde{x}_i &= \frac{(x_i - \text{med}(x))w_i^{(x)}}{\sqrt{(\sum_{j=1}^m)(x_j - \text{med}(x))w_j^{(x)}]} \text{ et } \tilde{y}_i = \frac{(y_i - \text{med}(y))w_i^{(y)}}{\sqrt{(\sum_{j=1}^m)(y_j - \text{med}(y))w_j^{(y)}}} \\
 \text{bicor}(x, y) &= \sum_{i=1}^m \tilde{x}_i \tilde{y}_i
 \end{aligned} \tag{4.5}$$

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 % sont généralement recommandés) dont la valeur est éloignée de la médiane. Pour une description complète de la méthode, vous pouvez lire l'article de Wilcox (1994).

Enfin, une autre option est l'utilisation de la corrélation *pi* de Sherphred (Schwarzkopf, Haas et Rees 2012). Il s'agit simplement d'une méthode en deux étapes. Premièrement, les valeurs extrêmes sont identifiées à l'aide d'une approche par *bootstrap* utilisant la distance de Mahalanobis (calculant les écarts multivariés entre les observations). Deuxièmement, 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 corrélation.

```

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$CI_high))
})
dfCorr <- data.frame(do.call(rbind, rs))
names(dfCorr) <- c("r", "IC_2.5", "IC_97.5")
dfCorr$method <- methods

# Impression du tableau avec le package stargazer
library(stargazer)
stargazer(dfCorr, type="text", summary=FALSE, rownames=FALSE, align = FALSE, digits = 3,
          title="Comparaison de différentes corrélations pour les deux variables")

```

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 la communauté scientifique. 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.

TAB. 4.2 : Comparaison de différentes corrélations pour les deux variables

r	IC 2,5 %	IC 97,5 %	Méthode
-0,158	-0,219	-0,095	Pearson
-0,120	-0,184	-0,055	Spearman
-0,137	-0,199	-0,074	Biweight
-0,174	-0,235	-0,111	Percentage
-0,119	-0,185	-0,052	Shepherd

4.3.5 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'échantillonnage, 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 permet de quantifier le niveau de certitude quant à l'existence d'une corrélation entre les 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 respectivement le coefficient de corrélation et le nombre d'observations. De manière classique, nous utiliserons 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 degrés 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 serait inférieure à 5 %. Dans ce contexte, nous pouvons raisonnablement rejeter l'hypothèse nulle (corrélation de 0).

La courte syntaxe ci-dessous illustre comment calculer la valeur de t , le nombre de degrés de liberté et la valeur de p pour une corrélation donnée.

```
df <- read.csv("data/bivariee/sr_rmr_mtl_2016.csv")
r <- cor(df$A6plus, df$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 , comme 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 à 95 % de -0,219 à -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), "\n", 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 %. L'intervalle de confiance et le seuil de significativité doivent être définis avant l'étude. Leur choix 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. Il est possible d'estimer la corrélation en mobilisant la notion de *bootstrap*, soit des méthodes d'inférence statistique basées sur des réplications des données initiales par rééchantillonnage. Concrètement, la méthode du *bootstrap* permet une mesure de la corrélation avec un intervalle de confiance à partir de r réplications, comme 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)

```

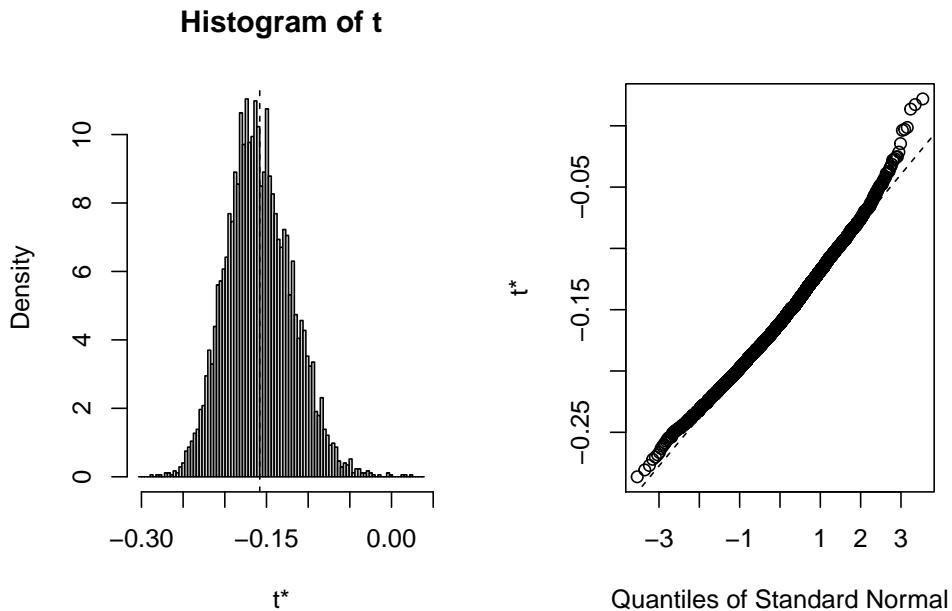


FIG. 4.8 : Histogramme pour les valeurs de corrélation issues du Bootstrap

```

# Corrélation "bootstrapée"
corBootstraped

## 
## ORDINARY NONPARAMETRIC BOOTSTRAP
## 
## Call:
## boot(data = df, statistic = correlation, R = 5000, X = "A65plus",
##       Y = "HabKm2", cor.type = "pearson")
## 
## 
## Bootstrap Statistics :
##      original      bias    std. error
## t1* -0.1580801 -0.0004329577  0.03964719

```

```

# 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%   (-0.2354, -0.0799 )  (-0.2386, -0.0866 )
##
## Level      Percentile        BCa
## 95%   (-0.2296, -0.0776 )  (-0.2175, -0.0499 )
## 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 obtenus 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 %.

4.3.6 Corrélation partielle



Quelle est la relation entre deux variables continues une fois prise en compte une autre variable dite de contrôle ? En études urbaines, nous pourrions vouloir vérifier si deux variables sont ou non associées après avoir contrôlé 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, après avoir contrôlées une ou plusieurs autres variables quantitatives (dites variables de contrôle).

Le coefficient de corrélation partielle peut être calculé pour plusieurs mesures de corrélation (notamment, 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 nous souhaitons 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 est forte, plus la couverture végétale est faible (r de Pearson = -0,563). La valeur du r de Pearson s'élève à -0,513 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, elle chute à -0,316. Pour calculer la corrélation partielle, nous pouvons 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 avec la fonction pcor.test 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

```

4.3.7 Mise en œuvre dans R

Comme vous l'aurez compris, il est possible d'arriver au même résultat 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 de Spearman (mais non celle de Kendall) avec les valeurs de p .
- `psych`, dont la fonction `corr.test` permet d'obtenir une matrice de corrélation (Pearson, Spearman et Kendall), les intervalles de confiance et les valeurs de p .
- `stargazer` pour créer de beaux tableaux d'une matrice de corrélation en *HTML*, 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 bayésiennes, robustes, non linéaires ou multiniveaux.

```

df1 <- 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(df1$RevMedMen, df1$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(df1$RevMedMen, df1$Locataire, type="spearman")

##          x      y
## x  1.00 -0.91
## y -0.91  1.00
##
## 
## n= 951
##
## 
## P
##   x   y
## x    0
## y    0

# Matrice de corrélation avec la fonction rcorr de Hmisc pour plus de variables
# Nous créons au préalable un vecteur avec les noms des variables à sélectionner
Vars <- c("RevMedMen", "Locataire", "LogTailInc", "A65plus", "ImgRec", "HabKm2", "FaibleRev")
Hmisc:::rcorr(df1[, 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 du package 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' si souhaité
p <- cor(df1[, 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
## 

# Créer un tableau avec la matrice de corrélation
# dans un fichier Word (.doc)
apaTables::apa.cor.table(df1[, 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 4.9 et 4.10). L'intérêt de ce type de figure est de repérer rapidement des associations intéressantes lorsque nous calculons les corrélations entre un grand nombre de variables.

```

library("corrplot")
library("ggpubr")
df1 <- read.csv("data/bivariee/sr_rmr_mtl_2016.csv")
Vars <- c("RevMedMen", "Locataire", "LogTailInc", "A65plus", "ImgRec", "HabKm2", "FaibleRev")
p <- cor(df1[, Vars], method="pearson")
couleurs <- colorRampPalette(c("#053061", "#2166AC", "#4393C3", "#92C5DE",
                                "#D1E5F0", "#FFFFFF", "#FDDBC7", "#F4A582",
                                "#D6604D", "#B2182B", "#67001F"))
corrplot::corrplot(p, addrect = 3, method="number",
                   diag=FALSE, col=couleurs(100))

```

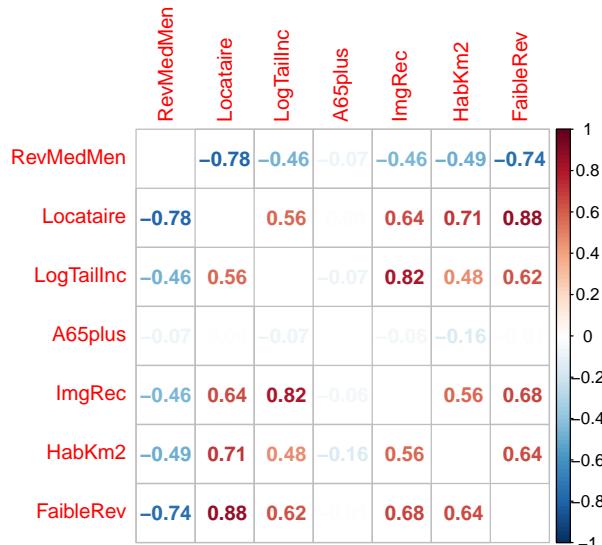


FIG. 4.9 : Matrice de corrélation avec corrplot (chiffres)

```
fig2 <- corrplot.mixed(p, lower="number", lower.col = "black",
                        upper = "ellipse", upper.col=couleurs(100))
```

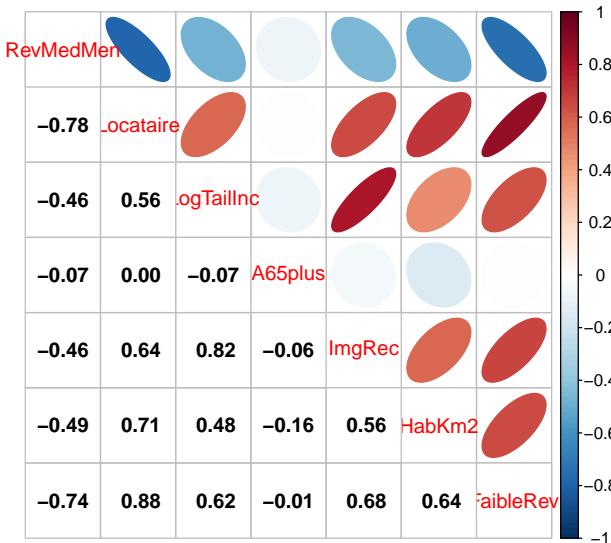


FIG. 4.10 : Matrice de corrélation avec corrplot (chiffres et ellipses)

4.3.8 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 nous pouvons rapidement le constater avec le signe du coefficient.
- Mettre le r et le p en italique et en minuscules.
- 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 locataires* est fortement négative ($r = -0,78, p < 0,001$).
- La corrélation entre les variables *revenu médian des ménages* et *pourcentage de locataires* est forte ($r(949) = -0,78, p < 0,001$).
- La corrélation entre les variables *densité de population* et *revenu médian des ménages* est modérée ($r = -0,49, p < 0,001$).
- 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>.

4.4 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 . Nous appelons cette équation un modèle, car elle cherche à représenter la réalité de façon simplifiée.

La régression linéaire simple relève ainsi de la statistique inférentielle et se distingue ainsi de la **covariance** (section 4.2) et de la **corrélation** (section 4.3) qui relèvent quant à eux 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 et d'étudiantes à un examen (variable dépendante Y) en fonction du nombre d'heures consacrées 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 personnes inscrites au cours la session suivante en fonction du temps qu'ils ou qu'elles prévoient passer à étudier, et ce, avant 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). Nous supposons 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, nous distinguons la **régression linéaire simple** (une seule variable indépendante) de la **régression linéaire multiple** (plusieurs variables indépendantes); cette dernière est largement abordée au chapitre 7.

Dans cette section, nous décrivons succinctement la régression linéaire simple. Concrètement, nous voyons comment déterminer la droite de régression, interpréter ses différents paramètres du modèle et évaluer la qualité d'ajustement du modèle. Nous n'abordons ni les hypothèses liées au modèle de régression linéaire des moindres carrés ordinaires (MCO) ni les conditions d'application. Ces éléments sont expliqués au chapitre 7, 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 qu' X cause Y . Autrement dit, la corrélation, soit le degré d'association entre deux variables, ne signifie pas 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, sans pour autant que Messerli (2012) et les journalistes concluent à un lien de causalité entre les deux variables :
 - * Radio Canada (<https://ici.radio-canada.ca/nouvelle/582457/chocolat-consommateurs-nobels>)
 - * La Presse (<https://www.lapresse.ca/vivre/sante/nutrition/201210/11/01-4582347-etude-plus-un-pays-mange-de-chocolat-plus-il-a-de-prix-nobel.php>)
 - * Le Point (https://www.lepoint.fr/insolite/le-chocolat-dope-aussi-l-obtention-de-prix-nobel-12-10-2012-1516159_48.php).

Les chercheurs et les chercheuses savent bien que la consommation de chocolat ne permet pas d'obtenir des résultats intéressants et de les 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 pouvez visionner cette courte vidéo ludique de vulgarisation (https://www.youtube.com/embed/A_naeATJ6o).

L'association entre deux variables peut aussi être simplement le fruit du hasard. Si nous explorons de très grandes quantités de données (avec un nombre impressionnant d'observations et de variables), soit une démarche relevant du forage ou de la fouille de données (*data mining* en anglais), le hasard fera que nous risquons d'obtenir des corrélations surprenantes entre certaines variables. Prenons un exemple concret : admettons que nous ayons collecté 100 variables et que nous calculons les corrélations entre chaque paire de variables. Nous obtenons une matrice de corrélation de 100×100 , à laquelle nous pouvons enlever la diagonale et une moitié de la matrice, ce qui nous laisse un total de 4950 corrélations différentes. Admettons que nous choisissons un seuil de significativité de 5 %, nous devons alors nous 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 illustrer ce fait 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 enregistré dans Results
Results <- c()
# itérons 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=0, y = 0.028,
           label = paste("Nombre moyen de corrélations significatives\n"
                         "sur 1000 répliques : ", 
                         round(mean(df$Values),0), sep=""), hjust="left")+
  xlab("Nombre de corrélations significatives")+
  ylab("densité")

```

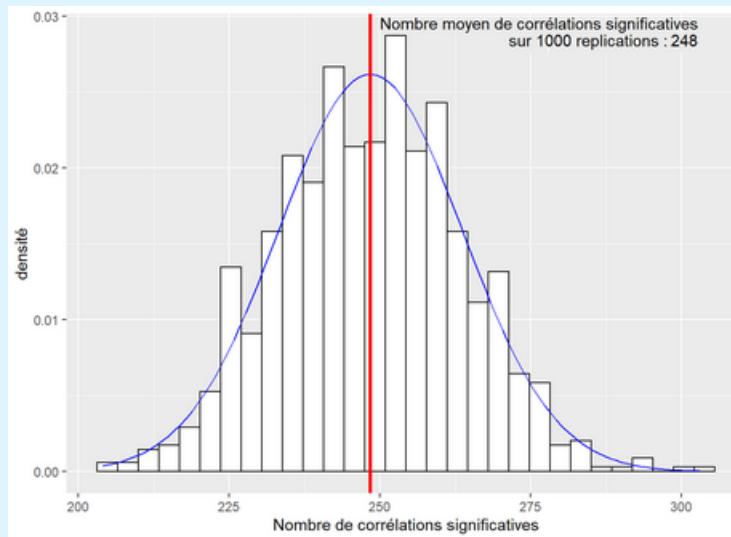


FIG. 4.11 : Corrélations significatives obtenues aléatoirement

4.4.1 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 dépendante et la valeur de la variable indépendante pour l'observation i . β_0 est la constante (*intercept* en anglais) et représente 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 ; s'il est proche de 0, la relation est nulle (la droite est 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 de personnes occupées se rendant au travail principalement à vélo et la distance entre chaque municipalité et le centre-ville de la région métropolitaine (tableau 4.3).

D'emblée, à la lecture du nuage de points (figure 4.12), nous décelons une forte relation linéaire négative entre les deux variables : plus la distance entre la municipalité et le centre-ville de la région métropolitaine augmente, plus le pourcentage de cyclistes est faible, ce qui est confirmé par le coefficient de corrélation ($r = -0,90$). La droite de régression (en rouge à la figure 4.12) qui résume le mieux la relation entre Vélo (variable dépendante) et KmCV (variable indépendante) s'écrit alors : $\text{Vélo} = 30,603 - 1,448 \times \text{KmCV}$.

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 entre la municipalité et le 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 serait 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 :

- $\hat{y}_i = 30,603 + (-1,448 \times 5 \text{ km}) = 23,363$
- $\hat{y}_i = 30,603 + (-1,448 \times 10 \text{ km}) = 8,883$
- $\hat{y}_i = 30,603 + (-1,448 \times 20 \text{ km}) = 1,643$

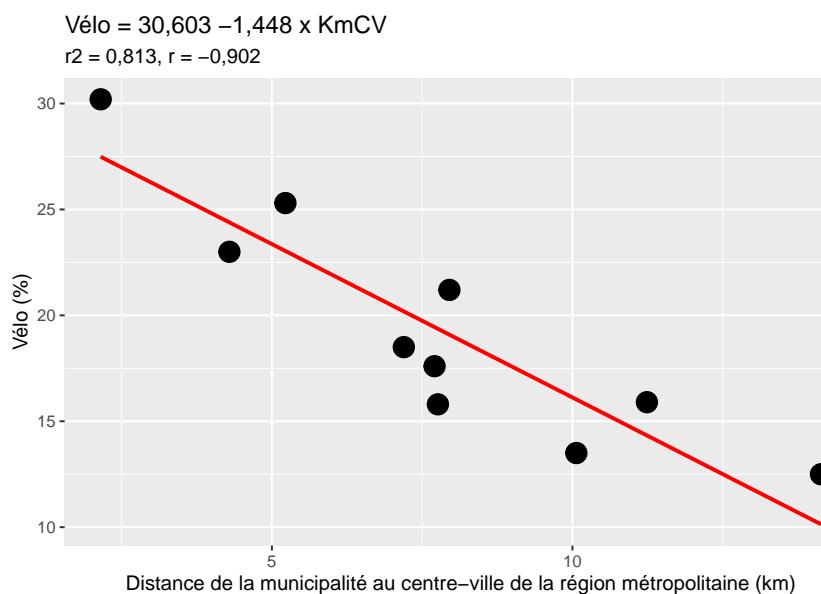


FIG. 4.12 : Relation linéaire entre l'utilisation du vélo et la distance au centre-ville

4.4.2 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 4.13, 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 le terme d'erreur de i). Si un point est au-dessus de la droite de régression, la valeur observée est 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)).

$$\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{cov(X, Y)}{var(X)} \quad (4.9)$$

$$\beta_0 = \bar{Y} - \beta_1 \bar{X} \quad (4.10)$$

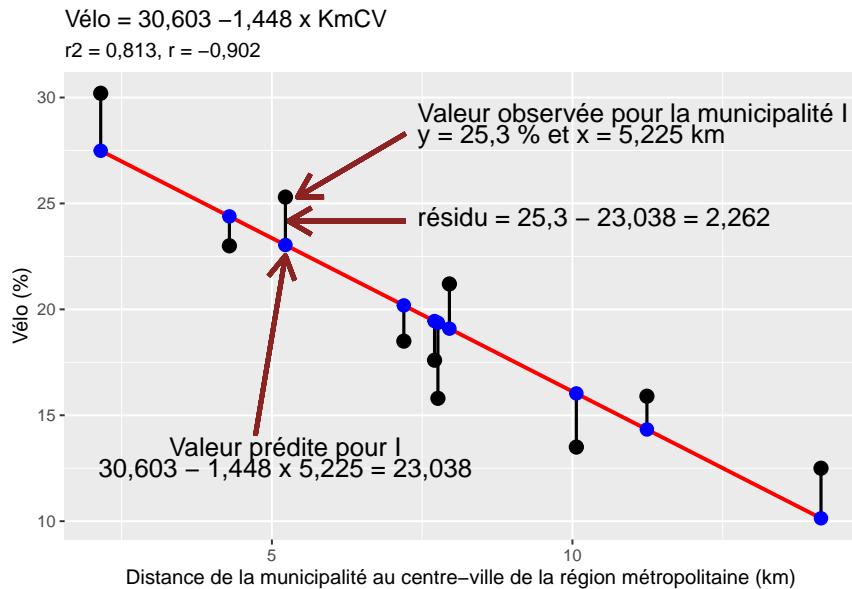


FIG. 4.13 : Droite de régression, valeurs observées, prédictes et résidus

4.4.3 Mesure de 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 (*root-mean-square error* en anglais, *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, prédictes 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

$$y_i = \beta_0 + \beta_1 x_1 + \epsilon_i \Rightarrow Y = \beta_0 + \beta_1 X + \epsilon \quad (4.11)$$

Elle comprend ainsi une partie de Y qui est expliquée par le modèle et une autre partie non expliquée, soit ϵ , appelée 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 (*total sum of squares* en anglais)
- SCE est la somme des écarts au carré des valeurs prédictes à la moyenne (*regression sum of squares* en anglais)
- SCR est la somme des carrés des résidus (*sum of squared errors* en anglais).

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 est faible, plus le modèle est efficace pour prédire la variable indépendante. L'erreur

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

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, nous avons : $\sqrt{5,23} = 2,29$. 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, nous avons : $R^2 = 227,04 / 279,3 = 0,8129$, ce qui signale que 81,3 % de la variance du pourcentage de cyclistes est expliquée par la distance entre la municipalité et le centre-ville de la région métropolitaine. Tel que signalé dans la section 4.3.2, 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 la table des valeurs critiques de F , section 14.2). Notez que nous utilisons rarement la table F puisqu'avec la fonction `pf(fobtenu, 1, n-2, lower.tail = FALSE)`, nous obtenons 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 est élevée, plus le modèle est 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.

4.4.4 Mise en œuvre dans R

Comment calculer une régression linéaire simple dans R. Rien de plus simple avec la fonction `lm(formula = y ~ x, data= DataFrame)`.

```

df1 <- read.csv("data/bivariee/Reg.csv", stringsAsFactors = F)
## Création d'un objet pour le modèle
monmodele <- lm(Velo ~ KmCV, df1)
## Résultats du modèle avec la fonction summary
summary(monmodele)

```

```

## 
## Call:
## lm(formula = Velo ~ KmCV, data = df1)
##
## 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(monmodele$residuals^2)
RMSE <- sqrt(MSE)
cat("MSE=", round(MSE, 2), "; RMSE=", round(RMSE, 2), sep="")

## MSE=5.23; RMSE=2.29
```

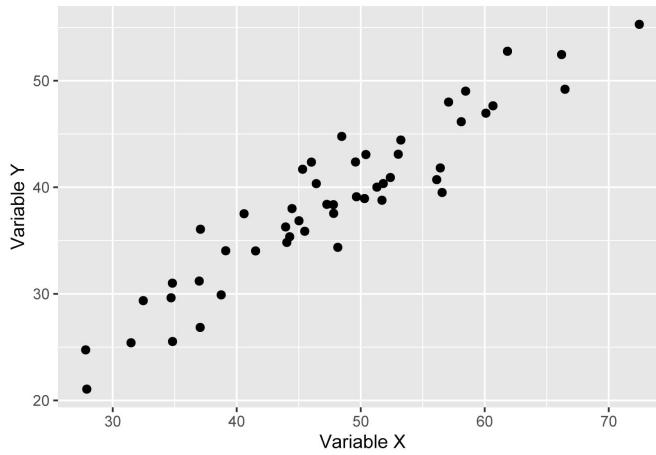
4.4.5 Comment rapporter une régression linéaire simple

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 entre la municipalité et le centre-ville de la région métropolitaine (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 : $30,603 - 1,448 \times (\text{distance au centre-ville en km})$.

4.5 Quiz de révision du chapitre

Questions

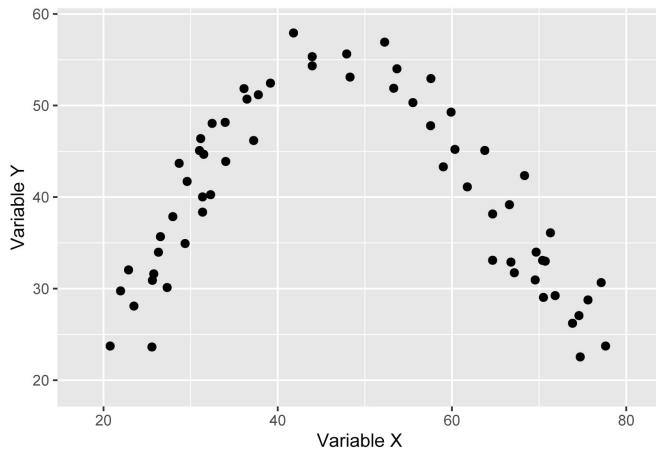
- D'après le nuage de points, les variables X et Y partagent une relation :



- linéaire positive
- linéaire négative
- curvilinearéaire
- absence de relation

relisez au besoin la section 4.1.

- D'après le nuage de points, les deux variables partagent une relation :



- linéaire positive
- linéaire négative
- curvilinearéaire
- absence de relation

Relisez au besoin la section 4.1.

- La valeur de la covariance peut être positive ou négative. Plus sa valeur absolue est élevée :

$$\text{cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1} = \frac{\text{covariation}}{n - 1}$$

- plus la relation linéaire entre les deux variables est importante
- plus la relation linéaire entre les deux variables est faible
- curvilinéaire
- absence de relation

Relisez au besoin la section [4.2.2](#).

- **Les coefficients de corrélation (Pearson, Spearman, etc.) varient de à :**

- 0 à 1
- moins l'infini à plus l'infini
- 0 à 100
- -1 à 1

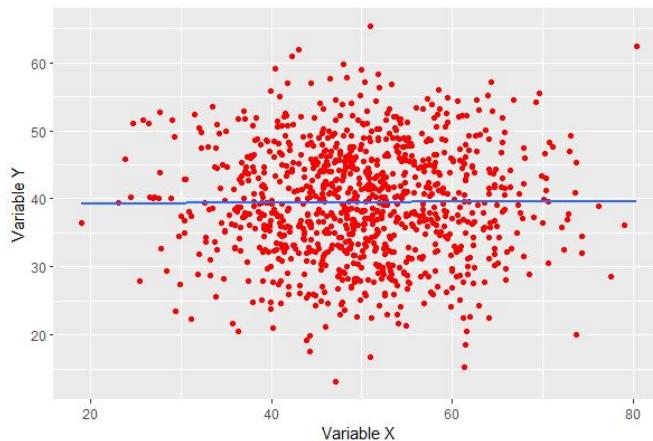
Relisez au besoin la section [4.3.3](#).

- **Cette statistique est tributaire des unités de mesure des deux variables. Cet inconvénient s'applique à :**

- la corrélation de Pearson
- la corrélation de Spearman
- la covariance
- la corrélation de Kendall

Relisez au besoin la section [4.2.2](#).

- **À la lecture du nuage de points, la corrélation de Pearson entre les deux variables est proche de :**



- 0
- -1
- 1

Relisez au besoin la section [4.1](#).

- **Le coefficient de Spearman est le coefficient de Pearson calculé sur des variables transformées en :**

- rangs
- scores Z (variables centrées réduites)
- sur une échelle 0 à 100

Relisez au besoin la section [4.3.3](#).

- **Dans une régression linéaire, le résidu pour une observation est :**

- la différence entre la variance de Y et la variance expliquée
- la différence entre la valeur observée et la valeur prédictive
- le carré de la différence entre la valeur observée et la valeur prédictive

Relisez au besoin la section [4.4.2.](#)

- **Le coefficient de détermination (R2) varie de :**

- -1 à 1
- -100 à 100
- 0 à 1

Relisez au besoin la section [4.4.3.](#)

- **Un facteur confondant est une sorte de fondant au chocolat ?**

- Vrai
- Faux

Relire le deuxième encadré à la section [4.4.](#)

Réponses

- D'après le nuage de points, les variables X et Y partagent une relation :
 - linéaire positive
- D'après le nuage de points, les deux variables partagent une relation :
 - curvilinéaire
- La valeur de la covariance peut être positive ou négative. Plus sa valeur absolue est élevée :
 - plus la relation linéaire entre les deux variables est importante
- Les coefficients de corrélation (Pearson, Spearman, etc.) varient de à :
 - -1 à 1
- Cette statistique est tributaire des unités de mesure des deux variables. Cet inconvénient s'applique à :
 - la covariance
- À la lecture du nuage de points, la corrélation de Pearson entre les deux variables est proche de :
 - 0
- Le coefficient de Spearman est le coefficient de Pearson calculé sur des variables transformées en :
 - rangs
- Dans une régression linéaire, le résidu pour une observation est :
 - la différence entre la valeur observée et la valeur prédictive
- Le coefficient de détermination (R2) varie de :
 - 0 à 1
- Un facteur confondant est une sorte de fondant au chocolat ?
 - Faux

Chapitre 5

Relation entre deux variables qualitatives

Dans le cadre de ce chapitre, nous présentons les deux principales méthodes permettant d'explorer les associations entre deux variables qualitatives : la construction d'un tableau de contingence et le test du khi-deux (χ^2 , appelé aussi kih carré).



Dans ce chapitre, nous utilisons les *packages* suivants :

- `gmodels` pour construire des tableaux de contingence.
- `vcd` pour construire un graphique pour un tableau de contingence.
- `DescTools` pour calculer le kih-deux de Mantel-Haenszel.
- `stargazer` pour imprimer des tableaux.



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, nous pourrions supposer que les jeunes se déplacent proportionnellement plus en modes de transport actif (vélo et marche) et en transport en commun. À l'inverse, il est possible que les 45 à 64 ans se déplacent majoritairement en véhicules motorisés.

Pour vérifier l'existence d'associations significatives entre les modalités de deux variables qualitatives, il est possible de construire un **tableau de contingence** (section 5.1), puis de réaliser le **test du kih-deux** (section 5.2).

5.1 Construction de tableau de contingence

Les données du tableau de contingence suivant décrivent 279 projets d'habitation à loyer modique (HLM) dans l'ancienne ville de Montréal, croisant les modalités de la période de construction (en colonne) et de la taille (en ligne) des projets HLM (Apparicio 2002). Les différents éléments du tableau sont décrits ci-dessous.

- **Les fréquences observées** (*Count* au tableau ci-dessous), nommées communément f_{ij} , correspondent aux observations appartenant à la fois à la i^e modalité de la variable en ligne et à la j^e

modalité de la variable en colonne. À titre d'exemple, nous comptons 14 projets 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 entendu, 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; *Total Percent*, *Row Percent* et *Column Percent* au tableau ci-dessous). 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 construites 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 construites 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** (*Expected Values* au tableau ci-dessous), représentent les valeurs que l'on devrait observer théoriquement s'il y avait indépendance entre les modalités des deux variables; autrement dit, si la répartition des deux modalités des deux variables était dû au hasard. Pour le croisement de deux modalités, la fréquence théorique est égale au produit des marges divisé par le nombre total d'observations ($ft_{ij} = (n_{i.} n_{.j})/n_{ij}$). 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). Nous avons donc moins de projets HLM de moins de 25 logements avant 1975 à quoi nous pourrions nous attendre du hasard.
- **La déviation** (*Residual* au tableau ci-dessous) est la différence entre la fréquence observée et la fréquence théorique ($f_{ij} - ft_{ij}$). Plus la déviation est grande, plus nous nous écartons 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, nous observons six habitations de moins de 25 logements construits avant 1975 et une fréquence théorique de 16,06. La déviation est donc -10,06, soit une sous-représentation du phénomène.
- **La contribution au khi-deux** (*Chi-square contribution* au tableau ci-dessous) 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-deux représente le khi-deux total pour l'ensemble du tableau de contingence (ici à 63,54), que nous aborderons 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
```

5.2 Test du khi-deux

Avec le test du khi-deux, nous postulons qu'il y a indépendance entre les modalités des deux variables qualitatives, soit l'hypothèse nulle (H_0). Puis, nous calculons 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 degrés de liberté et d'un seuil critique de significativité (prenons 5 % ici), nous pouvons trouver la valeur critique de khi-deux dans la table des valeurs critiques du khi-deux, soit 21,03 (voir section 14.1). Puisque la valeur du khi-deux calculée dans le tableau de contingence (63,54) est bien supérieure à celle obtenue dans le tableau des valeurs critiques (21,03), nous pouvons 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 de 32,91.

Bien entendu, une fois que nous connaissons le nombre de degrés de liberté, nous pouvons directement calculer les valeurs critiques pour différents seuils de signification et éviter ainsi de recourir à la table du khi-deux. Dans la même veine, nous pouvons aussi calculer la valeur de p d'un tableau de contingence en spécifiant le nombre de degrés de liberté et la valeur du khi-deux obtenue.

```
cat("Valeurs critiques du khi-deux 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 khi-deux 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("Valeur de p du khi-deux obtenu (63,54291) avec 12 degrés de liberté :", "\n",
  pchisq(q=63.54291, df=12, lower.tail = FALSE))

## Valeur de p du khi-deux obtenu (63,54291) avec 12 degrés de liberté :
## 5.063101e-09
```



Outre le khi-deux, d'autres mesures d'association permettent de mesurer le degré d'association entre deux variables qualitatives. Les plus courantes sont reportées dans le tableau suivant. À des fins de comparaison, le khi-deux décrit précédemment est aussi reporté sur la première ligne du tableau.

TAB. 5.1 : Autres mesures d'association entre deux variables qualitatives

Statistique	Formule	Propriété et interprétation
Khi-deux	$\chi^2 = \sum \frac{(f_{ij} - ft_{ij})^2}{ft_{ij}}$	Mesure classique du khi-deux calculée à partir des différences entre les fréquences observées et attendues. Valeur de p disponible
Ratio de vraisemblance du khi-deux	$G^2 = 2 \sum f_{ij} \ln \left(\frac{f_{ij}}{ft_{ij}} \right)$	Calculé à partir du ratio entre les fréquences observées et attendues. Valeur de p disponible
khi-deux 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 ordinales . Valeur de p disponible.
Corrélation polychorique	Obtenue itérativement par maximum de vraisemblance	Dans le même esprit que le khi-deux de Mantel-Haenszel, la corrélation polychorique s'applique à deux variables ordinales . 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, nous avons donc recours à des échelles de Likert allant de 1 à 10. Pour cet exemple, il est pertinent d'utiliser la corrélation polychorique. Comme pour une corrélation de Pearson, la corrélation polychorique varie de -1 à 1, une valeur négative indiquant une relation inverse entre les deux variables théoriques et inversement. Une valeur de p peut être obtenue.
Coefficient Phi	$\phi = \sqrt{\frac{\chi^2}{n}}$	Simplement le khi-deux divisé par le nombre d'observations. Si les deux variables qualitatives comprennent deux modalités chacune 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 surtout utile pour les tableaux comprenant deux modalités pour chacune des variables. 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 2x2, la valeur du V de Cramer sera égale à celle du Coefficient Phi. Pas de valeur de p disponible.

5.3 Mise en œuvre dans R

Pour calculer le khi-deux entre deux variables qualitatives, nous utilisons la fonction de base :

`chisq.test(x = ..., y = ...)` qui renvoie le nombre de degrés de liberté, les valeurs du khi-deux et de p .

```
# Importation du csv
dataHLM <- read.csv("data/bivariee/hlm.csv")
# Calcul du khi-deux 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 structure de votre tableau de données. 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, nous créons un facteur afin de spécifier un intitulé à chaque modalité (`factor(levels =c(...), labels = c(..))`). Puis, nous utilisons la fonction `CrossTable` du package `gmodels`. Pour obtenir les fréquences théoriques, les contributions locales au khi-deux et les déviations, nous spécifions 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)
table(dataHLM$Periode)
```

```
##
## 1 2 3 4 5
## 56 48 48 47 80
```

```
# La variable Taille comprend 4 modalités (de 1 à 4)
table(dataHLM$Taille)
```

```
##
## 1 2 3 4
## 80 53 99 47
```

```

# Création d'un facteur pour les cinq modalités de la période de construction
dataHLM$Periode <- factor(dataHLM$Periode,
                            levels = c(1,2,3,4,5),
                            labels = c("<1975",
                                      "1975-1979",
                                      "1980-1984",
                                      "1985-1989",
                                      "1990-1994"))

# Création d'un facteur pour les quatre modalités de la taille des habitations
dataHLM$Taille <- factor(dataHLM$Taille,
                           levels = c(1,2,3,4),
                           labels = c("<25 log.",
                                     "25-49",
                                     "50-99",
                                     "100 et +"))

# Pour construire un tableau de contingence, nous utilisons
# la fonction CrossTable du package gmodels.
# Les deux lignes ci-dessous sont mises en commentaire pour ne pas répéter le tableau.
# CrossTable(x=dataHLM$Taille, y=dataHLM$Periode, digits = 2,
#            expected=TRUE, chisq = TRUE, resid = TRUE, format="SPSS")

```

Deuxième cas de figure : vous disposez déjà d'un tableau de contingence, soit les fréquences observées (f_{ij}). Nous n'utilisons donc pas la fonction `CrossTable`, mais directement la fonction `chisq.test`.

```

# Importation des données
df1 <- read.csv("data/bivariee/data_transport.csv", stringsAsFactors = FALSE)
df1 # 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(df1[, c("Homme", "Femme")])
dimnames(Matrice) <- list(unique(df1$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",
                                         "À pied",
                                         "Bicyclette",
                                         "Autre moyen"),
                           Sexe=c("Homme", "Femme"))

# Test du khi-deux
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 (F_{ij})

```
test$observed
```

	Sexe	
## Transport	Homme	Femme
## Automobile (conducteur)	689400	561830
## Automobile (passager)	21315	40010
## Transport en commun	181435	238330
## À pied	43715	54360
## Bicyclette	24295	13765
## Autre moyen	8395	6970

Fréquences théoriques (FT_{ij})

```
round(test$expected,0)
```

	Sexe	
## Transport	Homme	Femme
## Automobile (conducteur)	643313	607917
## Automobile (passager)	31530	29795
## Transport en commun	215820	203945
## À pied	50425	47650
## Bicyclette	19568	18492
## Autre moyen	7900	7465

Déviations ($F_{ij} - FT_{ij}$)

```
round(test$observed-test$expected,0)
```

	Sexe	
## Transport	Homme	Femme
## Automobile (conducteur)	46087	-46087
## Automobile (passager)	-10215	10215
## Transport en commun	-34385	34385
## À pied	-6710	6710
## Bicyclette	4727	-4727
## Autre moyen	495	-495

Contributions au khi-deux

```
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
## À pied              892.81  944.80
## Bicyclette            1141.71 1208.19
## Autre moyen            31.04   32.85

```

```

# Marges en ligne et en colonne
colSums(Matrice)

```

```

## Homme   Femme
## 968555 915265

```

```

rowSums(Matrice)

```

```

## Automobile (conducteur)  Automobile (passager)  Transport en commun
##                 1251230                  61325          419765
##                 À 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
## À 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
## À 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
## À pied                     4.78  5.94
## Bicyclette                 2.51  1.42
## Autre moyen                0.92  0.76
```

Pour obtenir les autres mesures d'association (tableau 5.2), nous pourrons utiliser la syntaxe suivante :

```
df1 <- 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 khi-deux
  n <- sum(testChi2$observed) # Nombre d'observations
  nc <- ncol(testChi2$observed) # Nombre de colonnes
  l <- nrow(testChi2$observed) # Nombre de lignes
  dl <- (nc-1)*(l-1)          # Nombre de degrés de libertés
  chi2 <- testChi2$statistic # Valeur du khi-deux
  Pchi2 <- testChi2$p.value # P pour le khi-deux

  #Ratio de vraisemblance du khi-deux
  G <- 2*sum(testChi2$observed*log(testChi2$observed/testChi2$expected)) # G2
  PG <- pchisq(G, df=dl, lower.tail = FALSE) # P pour le G22

  # khi-deux de Mantel-Haenszel avec le package DescTools
  MHTest <- DescTools::MHChisqTest(testChi2$observed)
  MH <- MHTest$statistic
  PMH <- MHTest$p.value

  # Coefficient de correlation polychorique
  df1 <- data.frame("x" = as.factor(x),
                    "y" = as.factor(y))
  polychoricCorr <- correlation::cor_test(df1,"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(nc-1,l-1)))

  # Tableau pour les résultats
  dfsortie <- data.frame(
    Statistique = c("Khi-deux",
                  "Ratio de vraisemblance du khi-deux",
                  "Khi-deux de Mantel-Haenszel",
                  "Corrélation Polychorique",
                  "Coefficient de Phi",
                  "V de Cramer"),
```

```

Valeur = round(c(chi2, G, MH, polyR, phi, vc),3),
P = round(c(Pchi2, PG, PMH, polyP , NA, NA),10))
return(dfsortie)
}

dfkhi2 <- AutresMesuresKhi2(df1$Periode, df1$Taille)

# Impression du tableau avec le package stargazer
library(stargazer)
stargazer(dfkhi2, type="text", summary=FALSE, rownames=FALSE, align = FALSE, digits = 3,
          title="Mesures d'association entre les deux variables qualitatives")

```

TAB. 5.2 : Mesures d'association entre deux variables qualitatives

Statistique	Valeur	P
Khi-deux	63,543	0
Ratio de vraisemblance du khi-deux	67,286	0
Khi-deux de Mantel-Haenszel	48,486	0
Corrélation Polychorique	-0,479	0
Coefficient de Phi	0,477	
V de Cramer	0,276	

5.4 Interprétation d'un tableau de contingence

Nous vous proposons une démarche très simple pour vérifier l'association entre deux variables qualitatives avec les deux étapes suivantes :

- Nous posons l'hypothèse nulle (H_0), soit l'indépendance entre les deux variables.
- Si la valeur du khi-deux total du tableau de contingence est inférieure à la valeur critique du khi-deux 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 est 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 a pas d'association significative entre les modalités des deux variables. Vous pouvez simplement signaler que selon les résultats du test du khi-deux, il n'y a pas d'association significative entre les deux variables ($\chi^2 = \dots$ avec $p = \dots$).
- S'il y a dépendance ($khi_{observ}^2 > khi_{critique}^2$), trouvez les cellules ij où les contributions au khi-deux sont les plus fortes, c'est-à-dire où les associations entre les modalités i de la variable en ligne et les modalités j de la variable en colonne sont les plus marquées. 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 ligne ou en colonne pour appuyer vos propos.



Pour repérer rapidement les cellules où les contributions au khi-deux sont les plus fortes, vous pouvez construire un graphique avec la fonction `mosaic` du package `vcg`. À la figure 5.1, la taille des rectangles représente 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.

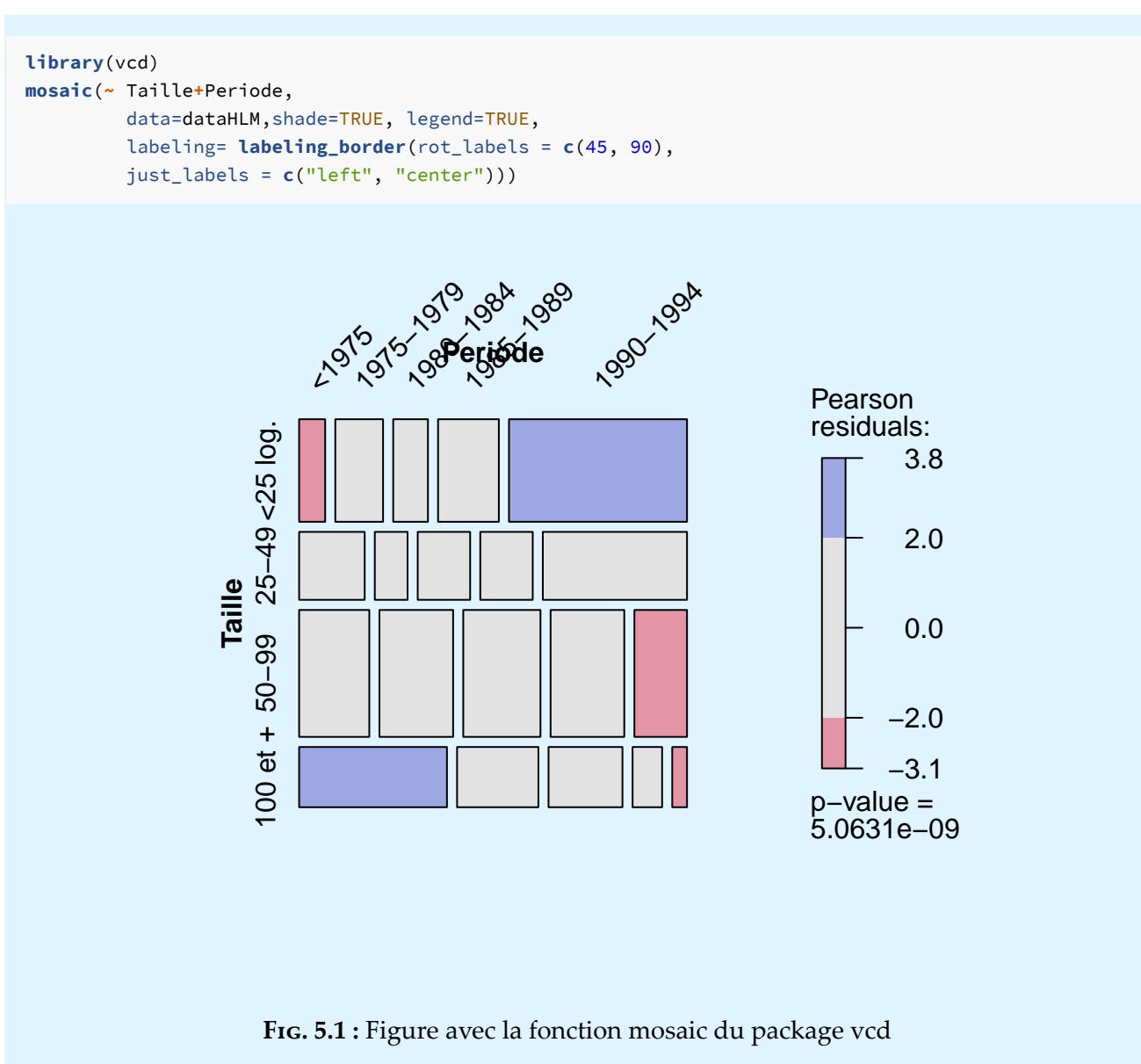


FIG. 5.1 : Figure avec la fonction mosaic du package vcd

Exemple d'interprétation. « Les résultats du test du khi-deux 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 khi-deux 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 a construit de grandes habitations dépassant souvent 100 logements ; du milieu des années 1970 à la fin des années 1980, on privilégie la construction d'habitats 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 habitations érigés à partir de 1990 disposent de moins de 25 logements» (Apparicio (2002), p. 117-118). Notez que cette évolution décroissante est aussi soutenue par le coefficient négatif de la corrélation polychorique.

Vous pouvez aussi construire un graphique pour appuyer vos constats, soit avec les pourcentages en ligne ou en colonne (figure 5.2 tirée de Apparicio (2002)).

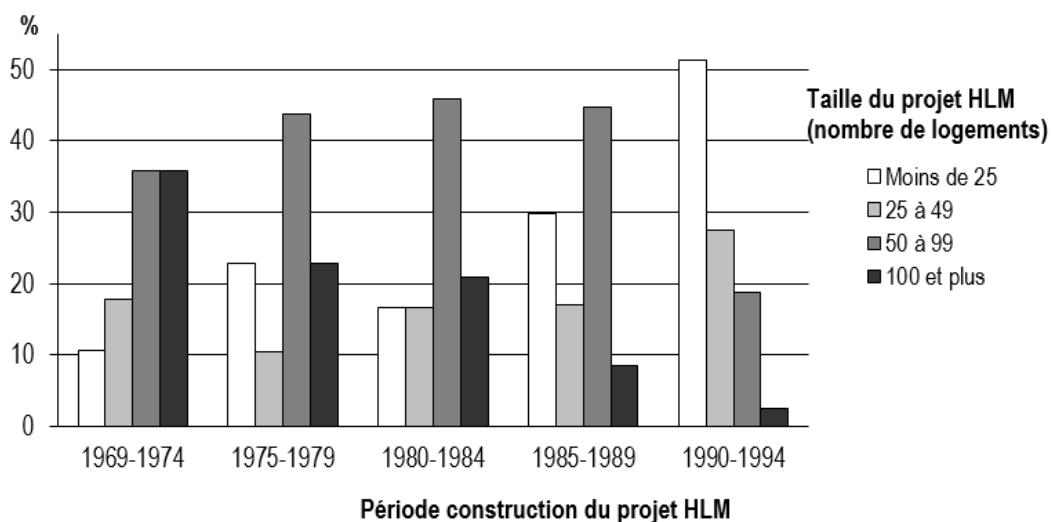


FIG. 5.2 : Taille des projets d'habitation à loyer modique selon la période de construction

Comment rapporter succinctement les résultats d'un test du khi-deux ?

Le test du khi-deux 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, consultez <https://www.socscistatistics.com/tutorials/chisquare/default.aspx>.

5.5 Quiz de révision du chapitre

Questions

- Pour analyser la relation entre deux variables qualitatives, vous utilisez :

- la covariance
- la régression linéaire simple
- le coefficient de corrélation de Pearson
- un tableau de continence et un test du khi2
- le coefficient de corrélation de Spearman
- un compteur Geiger

Relisez au besoin l'introduction du chapitre 5.

- Dans un tableau de contingence, quels sont les éléments disponibles ?

- Fréquences théoriques
- Fréquences observées
- Trois pourcentages (total, en lignes et en colonnes)
- Déviations
- Variance
- Contributions au khi-deux

Relisez au besoin la section 5.1.

- La corrélation polychorique est particulièrement bien adaptée pour mesurer l'association entre deux variables qualitatives :

- ordinaires
- nominales

Relisez au besoin l'encadré à la section 5.2.

- S'il y a indépendance entre les deux variables qualitatives (khi-deux observé inférieur au khi-deux critique), il n'est pas nécessaire d'analyser en détail le tableau de contingence.

- Vrai
- Faux

Relisez au besoin la section 5.4.

Réponses

- Pour analyser la relation entre deux variables qualitatives, vous utilisez :
 - un tableau de continence et un test du khi2
- Dans un tableau de contingence, quels sont les éléments disponibles ?
 - Fréquences théoriques
 - Fréquences observées
 - Trois pourcentages (total, en lignes et en colonnes)
 - Déviations
 - Contributions au khi-deux
- La corrélation polychorique est particulièrement bien adaptée pour mesurer l'association entre deux variables qualitatives :
 - ordinaires
- S'il y a indépendance entre les deux variables qualitatives (khi-deux observé inférieur au khi-deux critique), il n'est pas nécessaire d'analyser en détail le tableau de contingence.

- Vrai

Chapitre 6

Relation entre une variable qualitative et une variable quantitative

Dans le cadre de ce chapitre, nous présentons les principales méthodes permettant d'explorer les associations entre une variable quantitative et une variable qualitative avec deux modalités (tests de Student, de Welch et de Wilcoxon) ou avec plus de deux modalités (ANOVA et test de Kruskal-Wallis).



Dans ce chapitre, nous utilisons les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2`, le seul, l'unique!
 - * `ggbplotr` pour combiner des graphiques.
- Pour manipuler des données :
 - * `dplyr`, avec les fonctions `group_by`, `summarize` et les pipes `%>%`.
- Pour les test t :
 - * `sjstats` pour réaliser des test t pondérés.
 - * `effectsize` pour calculer les tailles d'effet de tests t .
- Pour la section sur les ANOVA :
 - * `**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`.
- Autre *package* :
 - * `foreign` pour importer des fichiers externes.

6.1 Relation entre une variable quantitative et une variable qualitative à deux modalités



Les moyennes de deux groupes de population sont-elles significativement différentes ? Nous souhaitons 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, nous pourrions 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, nous pourrions 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, nous cherchons 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 personnes utilisant habituellement l'automobile pour se rendre au travail sont recrutées. 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. Nous évaluerons 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. Nous pourrions 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.

6.1.1 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, nous souhaitons 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, nous avons 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, nous disposons 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 ; nous parlons alors d'étude, d'expérience ou d'analyse pré-post. Concrètement, nous cherchons à savoir si la moyenne des différences des observations avant et après est significativement différente de 0. Si c'est le cas, nous pouvons 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, nous pourrions questionner le même échantillon de commerçant(e)s ou personnes l'utilisant 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, nous utiliserons le test non paramétrique de Wilcoxon (section 6.1.2). 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, soit $0,5 < (S_{X_A}^2 / S_{X_B}^2) < 2$.
- 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 :

- Nous posons 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 $\bar{X}_A \neq \bar{X}_B$.

- Nous calculons la valeur de t et le nombre de degrés de liberté. La valeur de t est négative quand la moyenne du groupe A est inférieure au groupe B et inversement.
- Nous comparons la valeur absolue de t ($|t|$) avec celle issue de la table des valeurs critiques de T (voir section 14.2) 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, nous pouvons calculer la taille de l'effet.

Cas 1. Test de Student pour des échantillons indépendants avec des variances similaires (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) :

$$t = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{\frac{S_p^2}{n_A} + \frac{S_p^2}{n_B}}} \text{ avec } S_p^2 = \frac{(n_A - 1)S_{X_A}^2 + (n_B - 1)S_{X_B}^2}{n_A + n_B - 2} \quad (6.1)$$

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 dissemblables). 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 :

$$t = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{\frac{S_{X_A}^2}{n_A} + \frac{S_{X_B}^2}{n_B}}} \text{ et } dl = \frac{\left(\frac{S_{X_A}^2}{n_A} + \frac{S_{X_B}^2}{n_B} \right)^2}{\frac{S_{X_A}^4}{n_A^2(n_A-1)} + \frac{S_{X_B}^4}{n_B^2(n_B-1)}} \quad (6.2)$$

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 pouvez repérer comment sont calculés les moyennes, les nombres d'observations et les variances pour les deux groupes, le nombre de degrés de liberté et 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) :

- Au cas 1, 60 cyclistes utilisant un vélo personnel roulant en moyenne à 18 km/h (écart-type de 1,5) et 50 autres utilisant un système de vélopartage 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 autres utilisant un système de vélopartage avec une vitesse moyenne de 15 km/h (écart-type de 1,5). Ce faible écart des moyennes, combiné à une plus forte variance réduit la significativité de la différence entre les deux groupes.

D'emblée, l'analyse visuelle des boîtes à moustaches (figure 6.1) 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 semblables. 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(
  vitesse = c(Velo2A,Velo2B),
  type = c(rep("Vélo personnel",length(Velo2A)), rep("Vélopartage",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)

```

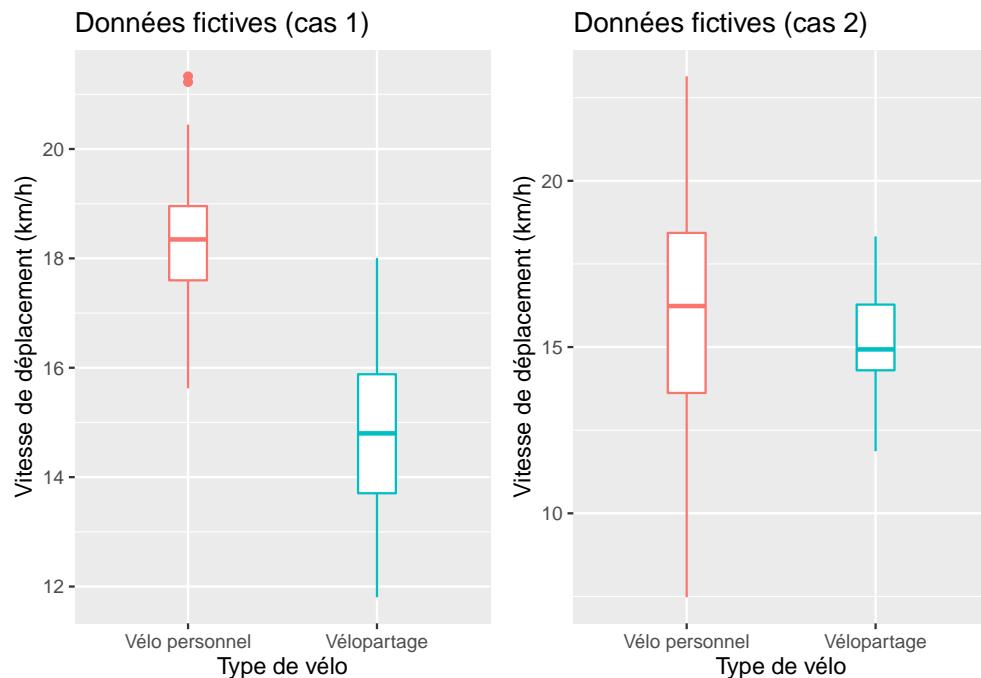


FIG. 6.1 : Boîtes à moustaches sur des échantillons fictifs non appariés

```

# Appel de la fonction pour le cas 1
tstudent_independants(Velo1A, Velo1B)

```

```

## 
##   groupe A (n = 60), moy = 18.3,

```

```

##           variance = 1.5, écart-type = 1.2
## groupe B (n = 50), moy = 14.8,
##           variance = 2.1, écart-type = 1.4
## ratio variance = 0.73
## t-test (variances égales): t(dl = 108) = 13.7819, p = 0
## t-Welch (variances inégales): t(dl = 96.974) = 13.5864, p = 0
## Variances semblables. 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 = 16.2,
##           variance = 10.2, écart-type = 3.2
## groupe B (n = 50), moy = 15.3,
##           variance = 2.1, écart-type = 1.4
## ratio variance = 4.97
## t-test (variances égales): t(dl = 108) = 1.7379, p = 0.085077
## t-Welch (variances inégales): t(dl = 84.978) = 1.8473, p = 0.068184
## Variances dissemblables. Utilisez le test de Welch-Satterwaite!
## Les moyennes des deux groupes ne sont pas significativement différentes.

```

6.1.1.1 Principe de base et formulation pour des échantillons dépendants (appariés)

Nous disposons de plusieurs personnes 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, nous réalisons les étapes suivantes :

- Nous posons 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 nous pouvons tester une autre valeur que 0.
- Nous calculons la valeur de t et le nombre de degrés de liberté. La valeur de t est négative quand la moyenne des différences entre X_{t_1} et X_{t_2} est négative et inversement.
- Nous comparons la valeur absolue de t ($|t|$) avec celle issue de la table des valeurs critiques de T avec le 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 appariés, la valeur de t se calcule comme suit :

$$t = \frac{\bar{D} - \mu_0}{\sigma_D / \sqrt{n}} \quad (6.3)$$

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 nous voulons tester (habituellement 0). Bien entendu, il est possible de fixer une autre valeur pour μ_0 : par exemple, avec $\mu_0 = 10$, nous chercherions ainsi à vérifier si la moyenne des différences est significativement différente de 10. Le nombre de degrés de liberté est 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 pouvez 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 générée 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 elles 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.
- Au cas 2, les valeurs de risque ont une moyenne de 70 avant et de 66 après, avec des écarts-types de 5.

D'emblée, l'analyse visuelle des boîtes à moustaches (figure 6.2) pairées montre que la perception du risque semble avoir nettement diminué 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_dépendants <- function(A, B, mu=0){
  d <- A-B           # différences entre les observations pairees
  moy <- mean(d)      # Moyenne des différences
  e_t <- sd(d)        # Écart-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",
                      xlab="", ylab="Sentiment de sécurité",
```

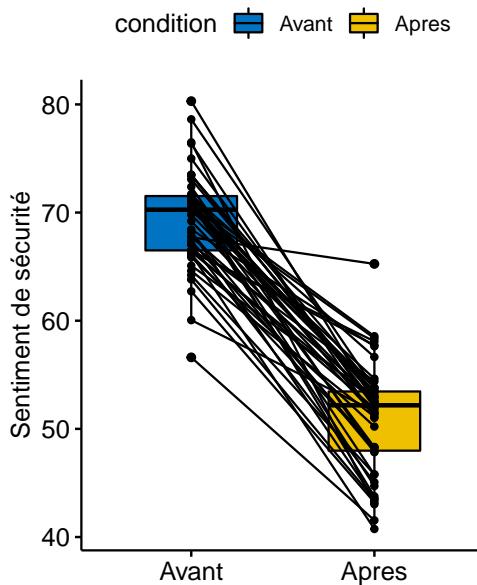
```

        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",
                      xlab="", ylab="Sentiment de sécurité",
                      title = "Données fictives (cas 2)")
ggarrange(boxplot1, boxplot2, ncol = 2, nrow = 1)

```

Données fictives (cas 1)



Données fictives (cas 2)

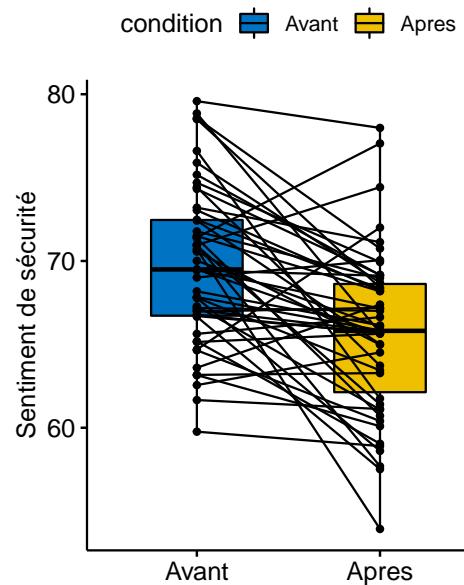


FIG. 6.2 : Boites à moustaches sur des échantillons fictifs appariés

```

# Test t : appel de la fonction tstudent_dependants
tstudent_dependants(Avant1, Apres1, mu=0)

```

```

##
## groupe A : moy = 69.4, var = 19.2, sd = 4.4
## groupe B : moy = 51.1, var = 25.3, sd = 5
## Moyenne des différences = 18.3
## Ecart-type des différences = 5.9
## t(dl = 49) = 22.1, p = 0
## La moyenne des différences entre les échantillons est significative

```

```
tstudent_dependants(Avant2, Apres2, mu=0)
```

```

##
## groupe A : moy = 69.7, var = 22.3, sd = 4.7
## groupe B : moy = 65.7, var = 23.8, sd = 4.9

```

```
## Moyenne des différences = 4
## Ecart-type des différences = 5.1
## t(dl = 49) = 5.46, p = 0
## La moyenne des différences entre les échantillons est significative
```

6.1.1.2 Mesure de la taille de l'effet

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 d'une comparaison de moyennes (avec des échantillons pairés ou non), pour mesurer la taille de l'effet, nous utilisons habituellement le d de Cohen ou encore le g de Hedges ; le second étant un ajustement du premier. Notez que nous analysons la taille de l'effet uniquement si le test de Student ou de Welch s'est révélé significatif ($p < 0,05$).

Pourquoi utiliser le d de Cohen ? 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 de l'effet indépendamment 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 , d 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, le d de Cohen s'écrit :

$$\frac{\bar{X}_A - \bar{X}_B}{\sqrt{\frac{(n_A-1)S_A^2 + (n_B-1)S_B^2}{n_A+n_B-2}}} \quad (6.4)$$

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 s'écrit :

$$d = \frac{\bar{X}_A - \bar{X}_B}{\sqrt{(S_A^2 + S_B^2)/2}} = \frac{\bar{X}_A - \bar{X}_B}{(\sigma_A + \sigma_B)/2} \quad (6.5)$$

avec σ_A et σ_B étant les écarts-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.

$$g = d - \left(1 - \frac{3}{4(n_A + n_B) - 9} \right) \quad (6.6)$$

Moins utilisé en sciences sociales, mais surtout en études cliniques, le delta de Glass est simplement la différence des moyennes des deux groupes indépendants (numérateur) sur l'écart-type du deuxième groupe (dénominateur). Dans une étude clinique, nous avons habituellement un groupe qui subit un traitement (groupe de traitement) et un groupe qui reçoit un placebo (groupe de contrôle ou groupe témoin). L'effet de taille est ainsi évalué par rapport au groupe de contrôle :

$$\Delta = \frac{\bar{X}_A - \bar{X}_B}{\sigma_B} \quad (6.7)$$

Finalement, pour des échantillons dépendants (pairés), le delta de Glass s'écrit : $d = \bar{D}/\sigma_D$ 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 est 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 6.1).

TAB. 6.1 : 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	
2,0 : Énorme	

6.1.1.3 Mise en œuvre dans R

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. Heureusement, 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.

La fonction `t.test` permet ainsi de calculer les tests 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` est utilisé pour spécifier si les échantillons sont dépendants (`paired=TRUE`) ou indépendants (`paired=FALSE`).
- Le paramètre `var.equal` est 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 = DataFrame, paired = FALSE, pooled_sd = TRUE)` ou `cohens_d(x, y, data = DataFrame, paired = FALSE, pooled_sd = TRUE)`
- `hedges_g(x ~ y, data = DataFrame, paired = FALSE, pooled_sd = TRUE)` ou `hedges_g(x, y, data = DataFrame, paired = FALSE, pooled_sd = TRUE)`
- `glass_delta(x ~ y, data = DataFrame, paired = FALSE, pooled_sd = TRUE)` ou `glass_delta(x, y, data = DataFrame, 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 6.3 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.

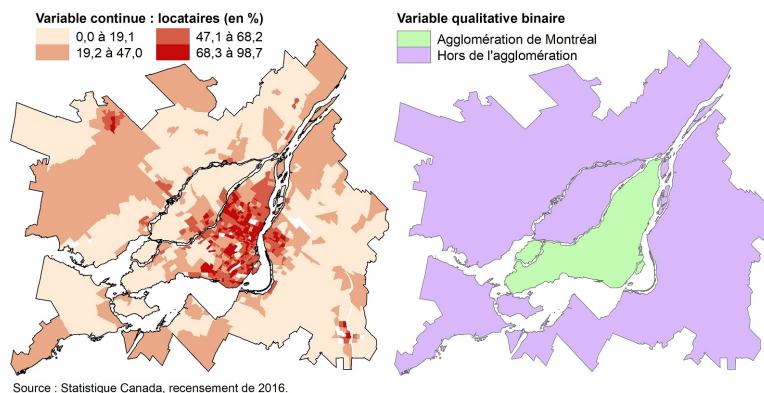
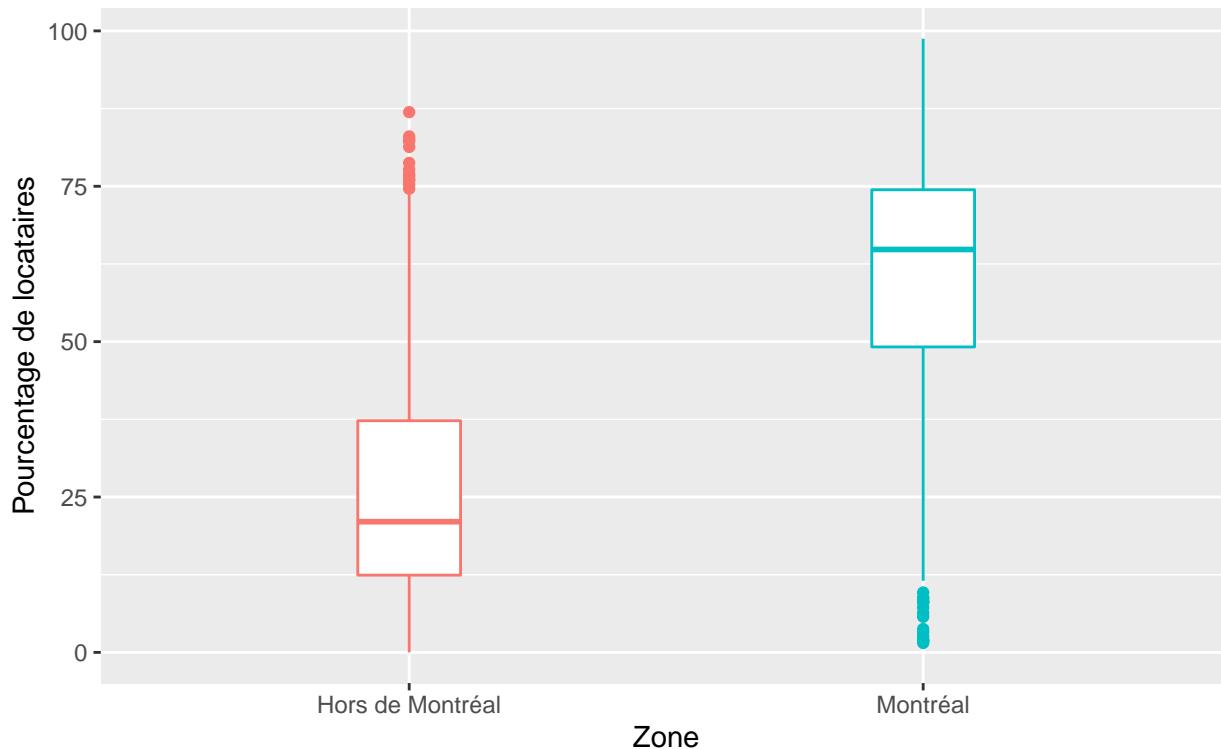


FIG. 6.3 : Pourcentage de locataires par secteur de recensement, région métropolitaine de recensement de Montréal, 2016

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 ($t = -23,95$; $p < 0,001$, d de Cohen = 1,54).

```
library("foreign")
library("effectsize")
library("ggplot2")
library("dplyr")
# Importation du fichier
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 -----
#Boîte 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="région métropolitaine de recensement de Montréal, 2016")
```

Locataires par secteur de recensement région métropolitaine de recensement de Montréal, 2016



```
# nombre d'observations, moyennes et écarts-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
```

```
# Nous vérifions 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; il est 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 Nous utilisons 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 Nous utilisons 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
         paired = FALSE,        # échantillons non pairés (indépendants)
         var.equal=FALSE)       # variances différentes
}

## 
## Les variances ne diffèrent pas!
## Nous utilisons 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]
##
```

```
## - Estimated using pooled SD.
```

```
hedges_g(Locataire ~ Montreal, data = dfRMR, paired = FALSE)
```

```
## Hedges' g | 95% CI
## -----
## -1.56 | [-1.70, -1.41]
##
## - Estimated using pooled SD.
## - Bias corrected using Hedges and Olkin's method.
```

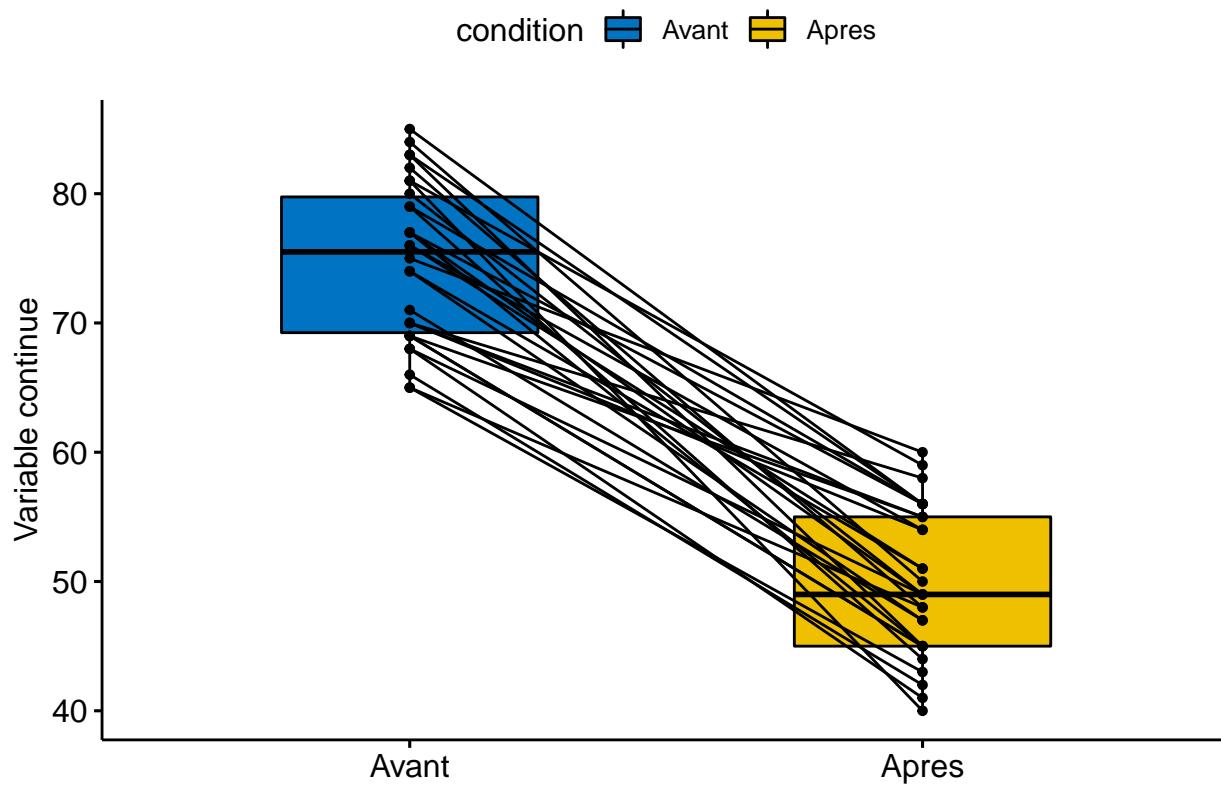
Notez que les valeurs du d de Cohen et du g de Hedge sont très semblables ; rappelons que le second est une correction du premier pour des échantillons de taille réduite. Avec 951 observations, nous disposons d'un échantillon suffisamment grand pour que cette correction soit négligeable.

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",
          xlab="", ylab="Variable continue")
```



```
# 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:
## 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 qualitative
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)
```

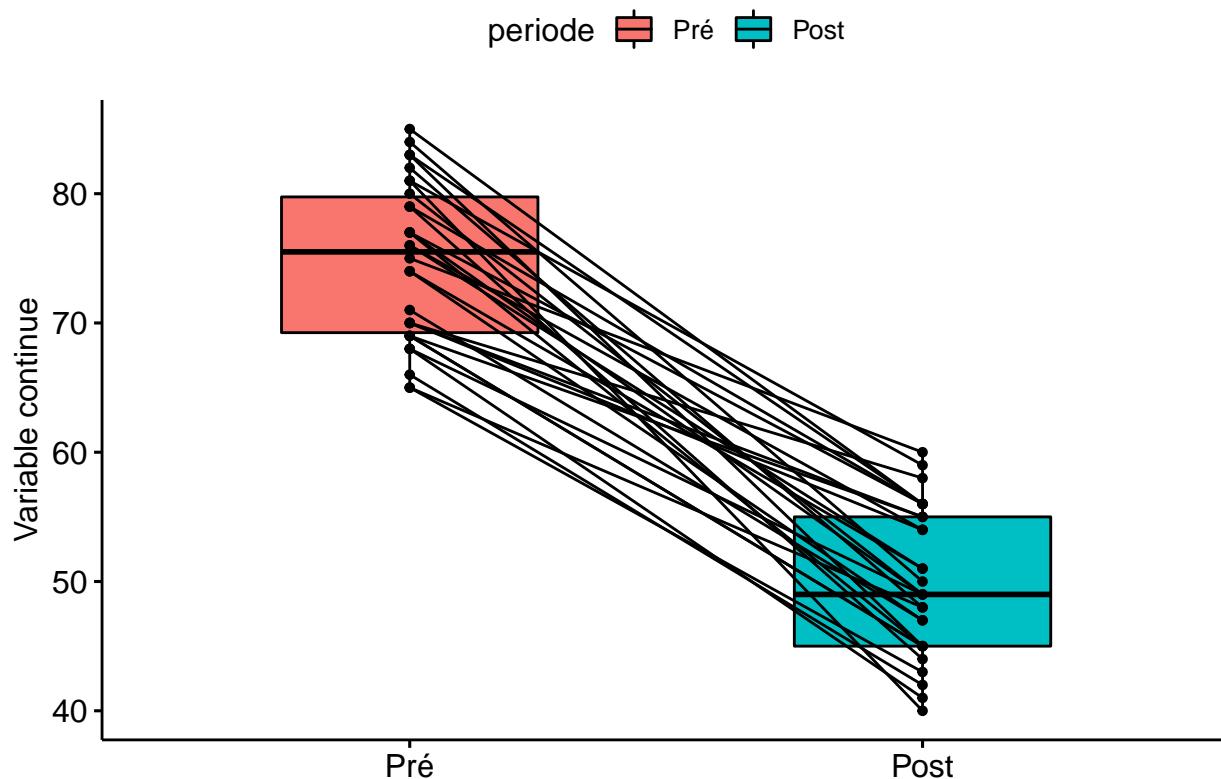
```
##   id participant risque periode
## 1  1          1    79     Pré
## 2  2          2    71     Pré
## 3  3          3    81     Pré
## 4  4          4    83     Pré
## 5  5          5    77     Pré
## 6  6          6    74     Pré
```

```
# nombre d'observations, moyennes et écarts-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",
         xlab="", ylab="Variable continue")
```



```
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
```

6.1.1.4 Comparaison 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). Nous

souhaitons vérifier si les personnes âgées de moins de 15 ans résidant sur l'île de Montréal bénéficient en moyenne d'une meilleure accessibilité au parc.

- Dans une étude sur la concentration de polluants atmosphériques dans l'environnement autour des écoles primaires montréalaises, Carrier *et al.* (2014) souhaitaient vérifier si les élèves fréquentant les écoles les plus défavorisées sont plus exposé(e)s 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 de NO_2 (variable continue), les quintiles extrêmes d'un indice de défavorisation (premier et dernier quintiles, variable qualitative) et le nombre d'élèves 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 pondéré
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]
```

6.1.1.5 Comment rapporter un test de Student ou de Welch ?

Pour les différentes versions du test, il est important de rapporter les valeurs de t et de p , les moyennes et écarts-types 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 = 29 117 dollars; écart-type = 258 022) est bien inférieur à celui des hommes (moyenne = 44 463; écart-type = 588 081). La différence entre les moyennes des deux sexes (-15 345) en faveur des hommes est d'ailleurs significative ($t = -27,09$; $p < 0,001$).
- Il y a un effet significatif selon le sexe ($t = -27,09$; $p < 0,001$), le revenu total des hommes (moyenne = 44 463; écart-type = 588 081) 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)

- Nous constatons 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, consultez <https://www.socscistatistics.com/tutorials/ttest/default.aspx>.

6.1.2 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. Nous privilégions 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, nous utilisons la fonction `wilcox.test` dans laquelle le paramètre `paired` permet 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 recensement 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.

```

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, des moyennes et
# des écarts-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 ceux en dehors 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 = 72, moy = 71.9
## PRE. median = 46.5, moy = 45.7

wilcox.test(df1$Avant, df1$Apres, paired = TRUE)

```

```

##
## Wilcoxon signed rank test with continuity correction
##
## data: df1$Avant and df1$Apres
## V = 1275, p-value = 7.731e-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 nous rapportons 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 ($W = 1223, p < 0,001$).
- Les résultats du test de Wilcoxon signalent que les rangs post-tests sont significativement plus faibles que ceux du pré-test ($W = 1273,5, p < 0,001$).
- 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) ($W = 1273,5, p < 0,001$).

6.2 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, nous avons recours 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.

6.2.1 Analyse de variance

L'analyse de variance (ANOVA) est largement utilisée en psychologie, en médecine et en 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 nous administrons un placebo (un médicament sans substance active), soit le groupe de contrôle ou le groupe témoin;
- un second groupe auquel nous administrons le médicament avec un faible dosage;
- un troisième avec un dosage moyen;
- un quatrième avec un dosage élevé.

La variable continue permet 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, nous nous attendons 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 permettent aussi de repérer le dosage 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 placébo.

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 de régions géographiques. En guise d'exemple, le recours à l'ANOVA permet 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 ?

6.2.1.1 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 (6.8));
- 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 (6.9));
- la **variance intragroupe** (Var_{intra}) ou variance non expliquée (VNE), soit la somme des variances des groupes de la variable indépendante (équation (6.10)).

$$VT = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (6.8)$$

$$Var_{inter} \text{ ou } VE = n_{g_1} \sum_{i \in g_1} (\bar{y}_{g_1} - \bar{y})^2 + n_{g_2} \sum_{i \in g_2} (\bar{y}_{g_2} - \bar{y})^2 + \dots + n_{g_k} \sum_{i \in g_k} (\bar{y}_{g_k} - \bar{y})^2 \quad (6.9)$$

$$Var_{intra} \text{ ou } VNE = \sum_{i \in g_1} (y_i - \bar{y}_{g_1})^2 + \sum_{i \in g_2} (y_i - \bar{y}_{g_2})^2 + \dots + \sum_{i \in g_n} (y_i - \bar{y}_{g_k})^2 \quad (6.10)$$

où \bar{y} est la moyenne de l'ensemble de la population; $\bar{y}_{g_1}, \bar{y}_{g_2}, \bar{y}_{g_k}$ sont respectivement les moyennes des groupes 1 à k (k étant le nombre de modalités de la variable qualitative) et n_{g_1}, n_{g_2} et n_{g_k} sont les nombres d'observations dans les groupes 1 à k .

La variance totale (VT) est égale à la somme de la variance intergroupe (expliquée) et la variance intragroupe (non expliquée) (équation (6.11)). Le ratio entre la variance intergroupe (expliquée) et la variance

totale est dénommé *Eta*² (équation (6.12)). 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.

$$VT = Var_{inter} + Var_{intra} \text{ ou } VT = VNE + VE \quad (6.11)$$

$$\eta^2 = \frac{Var_{inter}}{VT} \text{ ou } \eta^2 = \frac{VE}{VT} \quad (6.12)$$



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 en plusieurs groupes dans le chapitre sur les méthodes de classification (chapitre 13). En ANOVA, nous retenons 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 est performante.

Examinons un premier jeu de données fictives sur la vitesse de déplacement 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 6.2). D'emblée, nous notons 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 celles et ceux avec 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, nous avons $(16,900 - 17,339)^2 = 0,193$ et $(16,900 - 17,580)^2 = 0,462$. Les valeurs des trois variances sont les suivantes :

- la **variance totale** (VT) est donc égale à la somme de la troisième colonne (424,663).
- la **variance intergroupe** (expliquée, VE), elle est égale à $5 \times (17,580 - 17,339)^2 + 7 \times (12,257 - 17,339)^2 + 6 \times (23,067 - 17,339)^2 = 377,904$.
- la **variance intragroupe** (non expliquée, VNE) est égale à $11,228 + 21,537 + 13,993 = 46,758$.

Nous avons 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 6.3). 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** (VT) est égale à 121,756.
- la **variance intergroupe** (expliquée, VE) est égale à $5 \times (19,300 - 18,528)^2 + 7 \times (17,871 - 18,528)^2 + 6 \times (18,650 - 18,528)^2 = 6,087$.
- la **variance intragroupe** (non expliquée, VNE) est égale à $9,140 + 50,254 + 56,275 = 115,669$.

Nous avons 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.

TAB. 6.2 : 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. 6.3 : 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

6.2.1.2 Test de Fisher

Pour vérifier si les moyennes sont statistiquement différentes (autrement dit, si leur différence est significativement différente de 0), nous avons recours au test F de Fisher. Pour ce faire, nous posons l'hypothèse nulle (H_0), soit que les moyennes des groupes sont égales ; autrement dit que la variable qualitative n'a 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 $\overline{y_A} = \overline{y_B} = \overline{y_C}$. La statistique F se calcule comme suit :

$$F = \frac{\frac{Var_{inter}}{k-1}}{\frac{Var_{intra}}{n-k}} \text{ ou } F = \frac{\frac{VE}{k-1}}{\frac{VNE}{n-k}} \quad (6.13)$$

où n et k sont respectivement les nombres d'observations et de modalités de la variable qualitative. L'hypothèse nulle (les moyennes sont égales) est rejetée si la valeur du F calculé est supérieure à la valeur critique de la table F avec les degrés de liberté ($k-1, n-k$) et un seuil α ($p=0,05$ habituellement) (voir la table des valeurs critiques de F , section 14.2). Notez que nous utilisons rarement la table F puisqu'avec la fonction `aov`, nous obtenons 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 est élevée, plus la différence entre les moyennes est 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. Nous en concluons alors que :

- pour le cas A, le F calculé est égal à $(377,904 / 2) / (46,758 / 15) = 60,62$. 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 moyennes était de 0.
- pour le cas B, le F calculé est égal à $(6,087 / 2) / (115,669 / 15) = 0,39$. Il est inférieur à la valeur F critique ; les moyennes ne sont donc pas statistiquement différentes au seuil de 0,05.

6.2.1.3 Conditions d'application de l'ANOVA et solutions de recharge

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, nous avons recours au test de normalité de Shapiro-Wilk (section 2.5.4.1.3). Rappelez-vous toutefois que ce test est très restrictif, surtout pour de grands échantillons.
- **Homoscédasticité.** La variance dans les échantillons doit être la même (homogénéité des variances). Pour vérifier cette condition, nous utilisons 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, nous utiliserons 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, Peckham et Sanders 1972; Lix, Keselman et Keselman 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).
- 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'erreur 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, voire 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 ($VT = V_{intra} + V_{inter}$). Cela signifie que vous pouvez toujours calculer Eta^2 . Par contre, le test de Fisher ne peut pas être utilisé, car il est 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 carrée, 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 Breusch-Pagan) sur la variable continue transformée.
- Détecter les observations qui contribuent le plus à l'anormalité et à l'hétéroscédasticité (valeurs aberrantes ou extrêmes). 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 observations. Si les conditions sont respectées après la suppression et que les résultats de l'ANOVA (Eta^2 et test F de Fisher) sont très semblables, conservez donc les résultats de l'ANOVA initiale et signalez que vous avez procédé 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 en partie de la subjectivité des chercheur(e)s. Il faut s'adapter au jeu de données et à la culture statistique en vigueur dans votre champ d'études. 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.

6.2.2 Test non paramétrique de Kruskal-Wallis

Le test non paramétrique de Kruskal-Wallis est une solution de rechange à l'ANOVA classique lorsque le jeu de données présente de graves problèmes de normalité et d'hétéroscédasticité. Cette méthode représente une ANOVA appliquée à une variable continue transformée préalablement en rangs. Du fait de la transformation en rangs, nous ne vérifions plus si les moyennes sont différentes, mais bel et bien si les médianes de la variable continue sont différentes. Pour ce faire, nous utiliserons la fonction `kruskal.test`.

6.2.3 Mise en œuvre dans R

Dans une étude récente, Apparicio *et al.* (2018) 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 : une personne à vélo, une autre en automobile et une dernière se déplaçant en transport en commun, équipées 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 17 h. Au total, une centaine de trajets ont ainsi été réalisés. Des analyses de variance ont ainsi permis de comparer les trois modes (automobile, vélo et transport en commun) en fonction des temps de déplacement, des niveaux d'exposition au bruit, des niveaux d'exposition au dioxyde d'azote et de la dose totale inhalée de dioxyde d'azote. Nous vous proposons ici d'analyser une partie de ces données.

6.2.3.1 Première ANOVA : différences entre les temps de déplacement

Comme première analyse de variance, nous vérifions si les moyennes des temps de déplacement sont différentes entre les trois modes de transport.

Dans un premier temps, nous calculons les moyennes des différents groupes. Nous pouvons 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 violon (figure 6.4). 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 violon sont situées à la même hauteur. Autrement dit, à la lecture des deux graphiques, il ne semble pas y avoir de différences significatives entre les trois groupes en termes de temps de déplacement.

```

library("ggplot2")
library("ggpubr")
# Graphique de densité
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 trajet (en minutes)")

# Graphique en violon
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 violon",
    x = "Mode de transport",
    y = "Durée du trajet (en minutes)")+
  theme(legend.position = "none")
ggarrange(GraphDens, GraphViolon)

```

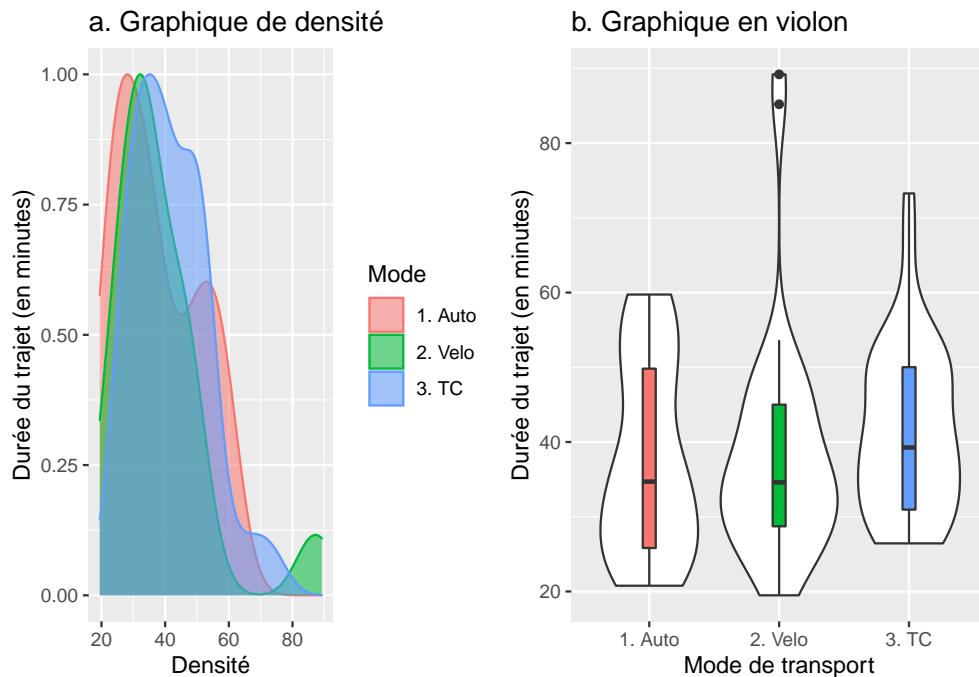


FIG. 6.4 : Graphiques de densité et en violon

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 nous ne pouvons rejeter l'hypothèse d'une distribution normale (autrement dit, la distribution est anormale). À la lecture des résultats ci-dessous, seul le groupe utilisant le 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 6.5). Ces graphiques sont utilisés pour déterminer visuellement si une distribution empirique (observée 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. Nous pouvons 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",
          xlab="Théorique", ylab="Échantillon")
```

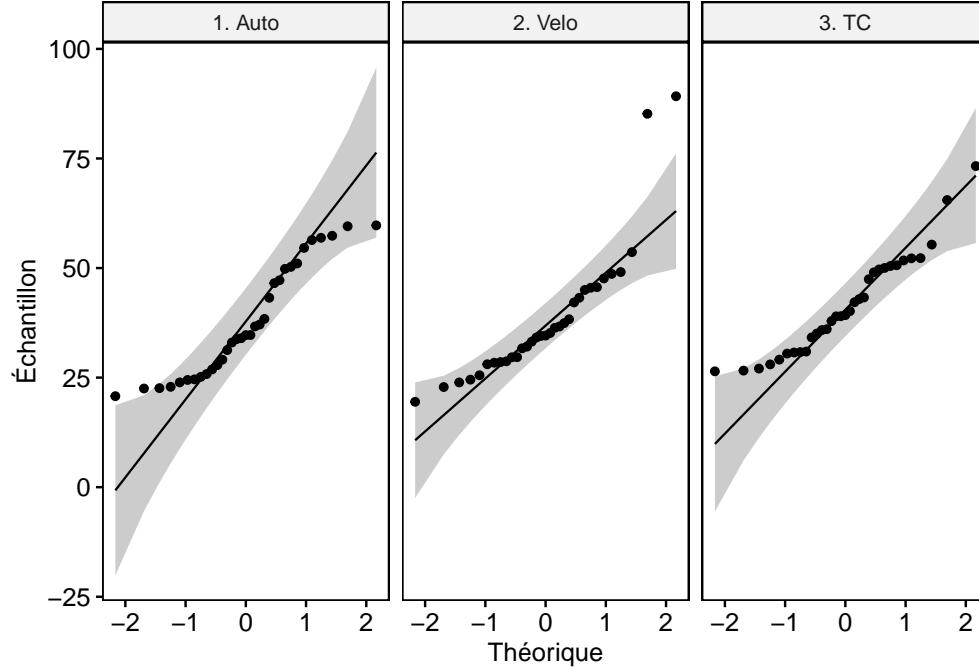


FIG. 6.5 : QQ Plot pour les groupes

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)

## 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 Eta^2 .

```

library("rstatix")
library("car")
library("effectsize")
# 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_squared du package effectsize
effectsize::eta_squared(aov1)

## Parameter | Eta2 |      90% CI
## -----
## Mode      | 0.02 | [0.00, 0.07]

```

```
# 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 déplacement des trois modes de transport.

6.2.3.2 Deuxième ANOVA : différences entre les niveaux d'exposition au bruit

Dans ce second exercice, nous analysons 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 %>%
  # Nom du DataFrame
  group_by(Mode) %>%
  # Variable qualitative
  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 Auto    laeq     1094  66.8   4.56
## 2 Velo    laeq     1124  68.8   4.29
## 3 TC      laeq     1207  74.0   6.79
```

À la lecture des graphiques de densité et en violon (figure 6.6), 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 personnes en 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 plus élevés dans le métro que dans les autobus.

```
library("ggplot2")
library("ggsignif")
# 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 violon
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)) +
```

```

library(ggplot2)
library(dplyr)

# Condition 1 : normalité des échantillons
# Test pour la normalité des échantillons (groupes) : test de Shapiro
df_Bruit %>%
  group_by(Mode) %>%
  shapiro_test(laeq) # Variable continue
```

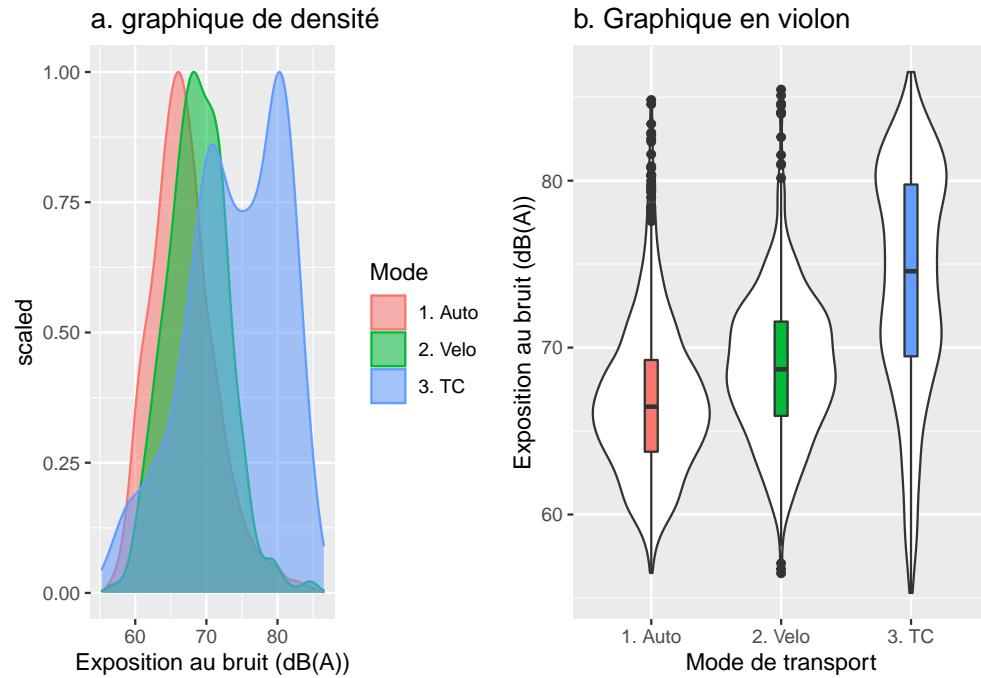


FIG. 6.6 : Graphique de densité et en violon

Le test de Shapiro et les graphiques QQ plot (figure 6.7) 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.

```

library("dplyr")
library("ggpubr")
library("rstatix")

# Condition 1 : normalité des échantillons
# Test pour la normalité des échantillons (groupes) : test de Shapiro
df_Bruit %>%
  group_by(Mode) %>%
  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", xlab="Théorique", ylab="Échantillon")
```

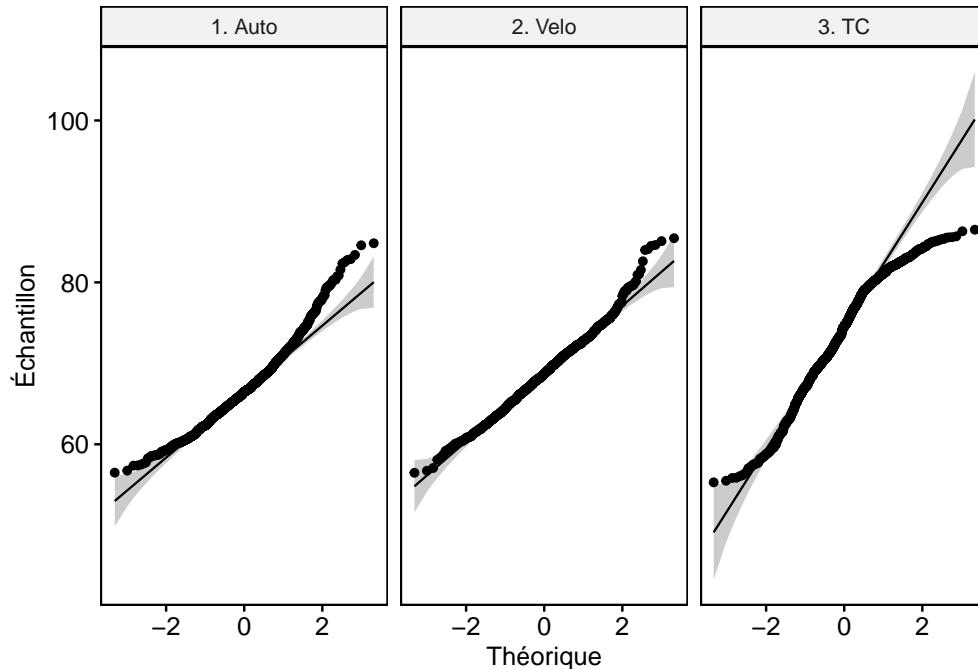


FIG. 6.7 : QQ Plot pour les groupes

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)

## 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 est 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.

```
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, nous constatons 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 personnes ayant pris le 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 du 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 personnes ayant pris le 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
```

6.2.4 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 tests 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 moins de quatre minutes comparativement aux personnes ayant pris le transport en commun (Moy = 41,6, ET = 11,4).
- Les résultats de l'analyse de variance à un facteur démontrent que le mode de 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 personnes en transport en commun (Moy = 74,0, ET = 6,79) et les cyclistes (Moy = 68,8, ET = 4,3) ont des niveaux d'exposition au bruit significativement plus élevés que les automobilistes (Moy = 66,8, ET = 4,56).
- Les résultats du test de Kruskal-Wallis démontrent 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, de 1124 pour le vélo et de 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). Nous évaluons 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, nous incluons 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 de plusieurs variables continues. Par exemple, pour une métropole donnée, nous pourrions vouloir comparer les moyennes de loyers entre la ville-centre et ceux des première et 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, nous souhaiterions 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 une 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 pairés. Nous parlons alors d'**analyse de variance sur des mesures répétées**. Par exemple, nous pourrions évaluer la perception du sentiment de sécurité relativement à la pratique du 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, nous pourrions sélectionner un échantillon d'individus (100, par exemple) pour lesquels nous évaluerions 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) ; nous mesurons alors une variable continue pour plusieurs groupes d'individus à différents moments ou pour des 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 BOSSA-NOVA et le CASANOVA ne sont pas des variantes de l'ANOVA !

6.3 Conclusion sur la troisième partie

Dans le cadre de cette troisième partie du livre, nous avons abordé les principales méthodes exploratoires et confirmatoires bivariées permettant d'évaluer la relation entre deux variables. La figure 6.8 propose un résumé de ces méthodes.

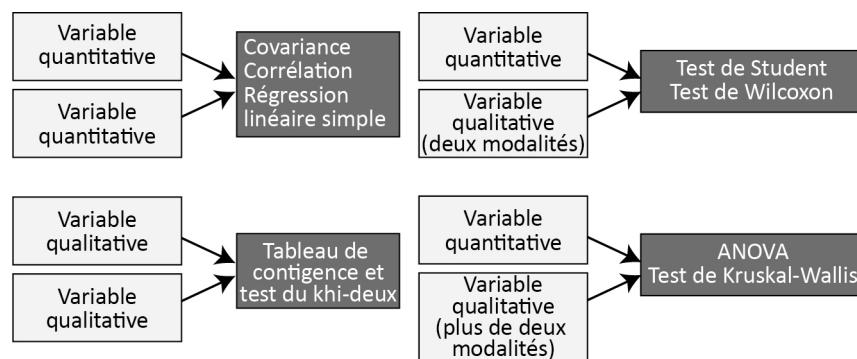


FIG. 6.8 : Les principales méthodes bivariées

6.4 Quiz de révision du chapitre

Questions

- **Comment comparer les moyennes de deux groupes ?**

- t de Student (test t)
- Analyse de variance (ANOVA)
- Covariance et corrélation
- Test de Kruskal-Wallis

Relisez au besoin la section [6.1.1](#).

- **Comment comparer les médianes de plus de deux groupes ?**

- t de Student (test t)
- Analyse de variance (ANOVA)
- Test de Kruskal-Wallis
- Test de Wilcoxon

Relisez au besoin la section [6.1.2](#).

- **Les observations de deux groupes qui n'ont aucun lien entre eux; les tailles des deux échantillons peuvent être différentes. Cette affirmation s'applique à des**

- échantillons indépendants (dits non appariés)
- échantillons dépendants (dits appariés)

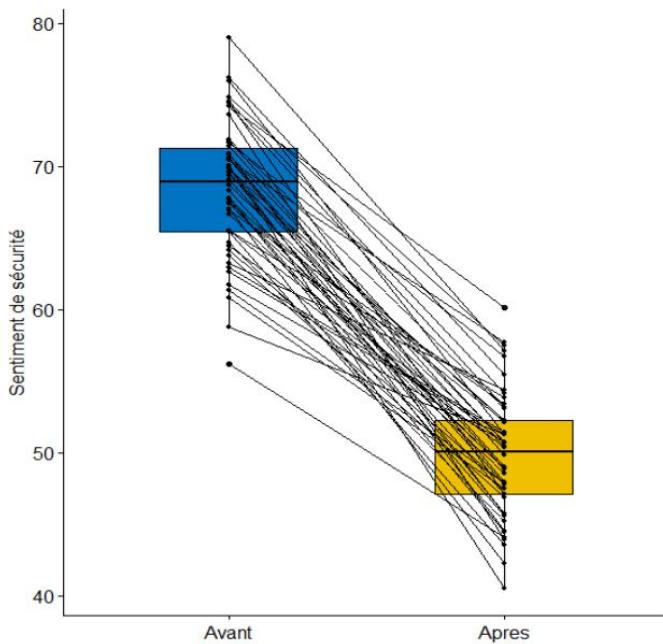
Relisez au besoin le début de la section [6.1.1](#).

- **Lorsque les variances des deux groupes sont dissemblables, quel test utilisez-vous ?**

- Test de Student (test t)
- Test de Welch (appelé aussi Satterthwaite)
- Analyse de variance (ANOVA)

Relisez au besoin la section [6.1.1](#).

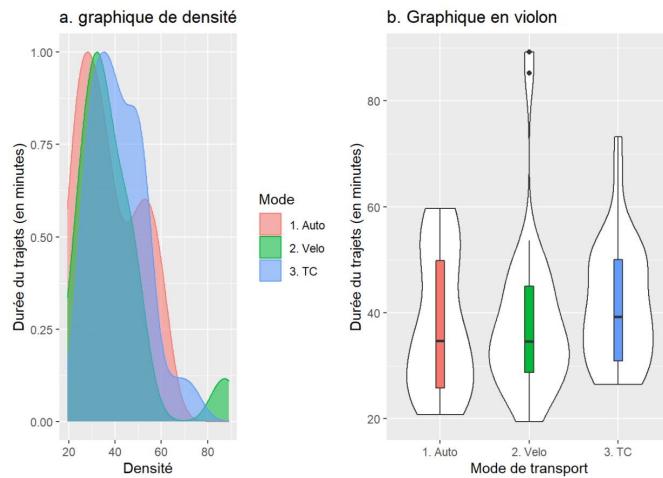
- **Ces boîtes à moustaches s'appliquent à des :**



- échantillons dépendants (dits appariés)
- échantillons indépendants (dits non appariés)

Relisez au besoin la section [6.1.1.1](#).

- Plus la variance intergroupe (dissimilarité des groupes) est maximisée et corolairement plus la variance intragroupe (homogénéité de chacun des groupes) est minimisée, plus les groupes sont clairement distincts et plus l'ANOVA est performante. Selon vous, à la lecture de ces graphiques, l'ANOVA risque-t-elle d'être très performante ?



- Oui
- Non

Relisez au besoin la section [6.2.1.1](#).

- Quelles sont les trois conditions d'application de l'ANOVA ?
 - Normalité des groupes
 - Homogénéité des variances des groupes (homoscédasticité)
 - Indépendance des observations (pseudo-réplication)

- Il faut deux groupes
- Les groupes doivent être de taille égale

Relisez au besoin la section [6.2.1.3](#).

- **Le test non paramétrique de Kruskal-Wallis permet de comparer les médianes de plus de deux groupes.**

- Vrai
- Faux

Relisez au besoin la section [6.2.2](#).

- **Sur quelle(s) variances est basée l'ANOVA ?**

- La variance totale
- La variance intragroupe
- La variance intergroupe

Relire le deuxième encadré à la section [6.2.1.1](#).

- **Quelles sont les variantes de l'ANOVA ?**

- Une ANOVA à deux facteurs
- Une ANOVA multifacteur
- L'analyse de covariance (ANCOVA)
- L'analyse de variance multivariée (MANOVA)
- L'analyse de covariance multivariée (MANCOVA)
- La SUPERNOVA
- La BOSSANOVA
- Le CASANOVA

Relire le deuxième encadré à la section [6.2.4](#).

Réponses

- Comment comparer les moyennes de deux groupes ?
 - t de Student (test t)
- Comment comparer les médianes de plus de deux groupes ?
 - Test de Wilcoxon
- Les observations de deux groupes qui n'ont aucun lien entre eux ; les tailles des deux échantillons peuvent être différentes. Cette affirmation s'applique à des
 - échantillons indépendants (dits non appariés)
- Lorsque les variances des deux groupes sont dissemblables, quel test utilisez-vous ?
 - Test de Welch (appelé aussi Satterthwaite)
- Ces boîtes à moustaches s'appliquent à des :
 - échantillons dépendants (dits appariés)
- Plus la variance intergroupe (dissimilarité des groupes) est maximisée et corolairement plus la variance intragroupe (homogénéité de chacun des groupes) est minimisée, plus les groupes sont clairement distincts et plus l'ANOVA est performante. Selon vous, à la lecture de ces graphiques, l'ANOVA risque-t-elle d'être très performante ?
 - Non
- Quelles sont les trois conditions d'application de l'ANOVA ?
 - Normalité des groupes
 - Homogénéité des variances des groupes (homoscédasticité)
 - Indépendance des observations (pseudo-réPLICATION)

- Le test non paramétrique de Kruskal-Wallis permet de comparer les médianes de plus de deux groupes.
 - Vrai
- Sur quelle(s) variances est basée l'ANOVA ?
 - La variance totale
 - La variance intragroupe
 - La variance intergroupe
- Quelles sont les variantes de l'ANOVA ?
 - Une ANOVA à deux facteurs
 - Une ANOVA multifacteur
 - L'analyse de covariance (ANCOVA)
 - L'analyse de variance multivariée (MANOVA)
 - L'analyse de covariance multivariée (MANCOVA)

Quatrième partie

Modèles de régression

Chapitre 7

Régression linéaire multiple

Dans ce chapitre, nous présentons la méthode de régression certainement la plus utilisée en sciences sociales : la régression linéaire multiple. À titre de rappel, dans la section 4.4, nous avons vu que la régression linéaire simple, basée sur la méthode des moindres carrés ordinaires (MCO), permet d'expliquer et de prédire une variable continue en fonction d'une autre variable. Toutefois, quel que soit le domaine d'étude, il est rare que le recours à une seule variable explicative (X) permette de prédire efficacement une variable continue (Y). La régression linéaire multiple est simplement une extension de la régression linéaire simple : elle permet ainsi de prédire et d'expliquer une variable dépendante (Y) en fonction de plusieurs variables indépendantes (explicatives).

Plus spécifiquement, nous abordons ici les principes et les hypothèses de la régression linéaire multiple, comment mesurer la qualité d'ajustement du modèle, introduire des variables explicatives particulières (variable qualitative dichotomique ou polytomique, variable d'interaction, etc.), interpréter les sorties d'un modèle de régression et finalement la mettre en oeuvre dans R.



Dans ce chapitre, nous utilisons les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2`, le seul, l'unique!
 - * `ggsave` pour combiner les graphiques
- Pour obtenir les coefficients standardisés :
 - * `QuantPsyc` avec la fonction `lm.beta` (section 7.4.2).
- Pour les effets marginaux des variables indépendantes :
 - * `ggeffects` avec la fonction `ggpredict` (section 7.7.4).
- Pour vérifier la normalité des résidus :
 - * `DescTools` avec les fonctions `Skewness` et `Kurtosis` et `JarqueBeraTest` (section 7.6.2).
- Pour vérifier l'homoscédasticité des résidus :
 - * `lmtest` avec la fonction `bptest` pour le test de Breusch-Pagan (section 7.7.3.3).
- Pour vérifier la multicolinéarité excessive :
 - * `car` avec la fonction `vif` (section 7.7.3.4).
- Autre *package* :
 - * `foreign` pour importer des fichiers externes.

7.1 Objectifs de la régression linéaire multiple et construction d'un modèle de régression

Selon Barbara G. Tabachnick et Linda S. Fidell (2007), un modèle de régression permet de répondre à deux objectifs principaux relevant chacun d'une approche de modélisation particulière.

La première approche a pour objectif d'identifier les relations entre une variable dépendante (VD) et plusieurs variables indépendantes (VI). Il s'agit alors de déterminer si ces relations sont positives ou négatives, significatives ou non et d'évaluer leur ampleur. La construction du modèle de régression repose alors sur un cadre théorique et la formulation d'hypothèses, sur les relations entre chacune des VI et la VD.

La seconde approche est exploratoire et très utilisée en forage ou en fouille de données (*data mining* en anglais). Parmi un grand ensemble de variables disponibles dans un jeu de données, elle vise à identifier la ou les variables permettant de prédire le plus efficacement (précisément) une variable dépendante. Parfois, ce type de démarche ne repose ni sur un cadre théorique ni sur la formulation d'hypothèses entre les VI et la VD. Dans des cas extrêmes, on s'intéresse uniquement à la capacité de prédiction du modèle, et ce, sans analyser les associations entre les VI et la VD. L'objectif étant d'obtenir le modèle le plus efficace possible afin de prédire à l'avenir la valeur de la variable dépendante pour des observations pour lesquelles elle est inconnue. Pour ce faire, nous avons recours à des régressions séquentielles (*stepwise regressions*) dans lesquelles les variables peuvent être ajoutées une à une au modèle ou retirées de celui-ci; nous conserverons dans le modèle final uniquement celles qui ont un apport explicatif significatif. Signalons d'emblée que dans le reste du chapitre, comme du livre, nous ne nous étendons pas plus sur cette approche de modélisation, et ce, pour deux raisons. D'une part, cette approche met souvent en évidence des relations significatives entre des variables sans qu'il y ait une relation de causalité entre elles. D'autre part, en sciences sociales, un modèle de régression doit être basé sur un cadre théorique et conceptuel élaboré à la suite à d'une revue de littérature rigoureuse.

Cadre conceptuel et élaboration d'un modèle de régression

Pour bien construire un modèle de régression, il convient de définir un cadre conceptuel élaboré à la suite à une revue de littérature sur le sujet de recherche. Ce cadre conceptuel permet d'identifier les dimensions et les concepts clefs permettant d'expliquer le phénomène à l'étude. Par la suite, pour chacun de ces concepts ou les dimensions, il est alors possible 1) d'identifier les différentes variables indépendantes qui sont introduites dans le modèle et 2) de formuler une hypothèse pour chacune d'elles. Par exemple, pour telle ou telle variable explicative, on s'attendra à ce qu'elle fasse augmenter ou diminuer significativement la variable dépendante. De nouveau, la formulation de cette hypothèse doit s'appuyer sur une interprétation théorique de la relation entre la VI et la VD.

Prenons en guise d'exemple une étude récente portant sur la multiexposition des cyclistes au bruit et à la pollution atmosphérique (Gelb et Apparicio 2020). Dans cet article, les auteurs s'intéressent aux caractéristiques de l'environnement urbain qui contribuent à augmenter ou réduire l'exposition des cyclistes à la pollution de l'air et au bruit routier. Pour ce faire, une collecte de données primaires a été réalisée avec trois cyclistes dans les rues de Paris du 4 au 7 septembre 2017. Au total, 64 heures et 964 kilomètres ont ainsi été parcourus à vélo afin de maximiser la couverture de la ville de Paris et les types d'environnements urbains traversés.

Leur cadre conceptuel est schématisé à la figure 7.1. Les deux variables indépendantes (à expliquer) sont l'exposition au dioxyde d'azote (NO_2) et l'exposition au bruit (mesurée en décibel dB(A)). Avant d'identifier les caractéristiques de l'environnement urbain affectant ces deux expositions, plusieurs facteurs, dits **variables de contrôle**, sont considérés. Par exemple, la concentration de NO_2 varie en fonction des conditions météorologiques (vent, température et humidité) et de la pollution d'arrière-plan (variant selon le moment de la journée, le jour de la semaine et la localisation géographique au sein de la ville). Ces dimensions ne sont pas le centre d'intérêt direct de l'étude. En effet, les auteurs s'intéressent aux impacts des caractéristiques locales de l'environnement urbain. Pour pouvoir les identifier sans biais, il est nécessaire de contrôler (filtrer)

l'ensemble de ces autres facteurs.

Dans leur cadre conceptuel, les auteurs regroupent les caractéristiques locales de l'environnement urbain en trois grandes dimensions : les caractéristiques du segment (type de rues ou de voies cyclables empruntés, intersections traversées, pente et vitesse), celles de la forme urbaine (densité résidentielle, végétation, ouverture de la rue et occupations du sol) et celles du trafic (nombre et types de véhicules croisés, congestion et zones 30 km/h). Une fois ce cadre conceptuel construit, il reste alors à identifier les variables qui permettent d'opérationnaliser chacun de concepts retenus.

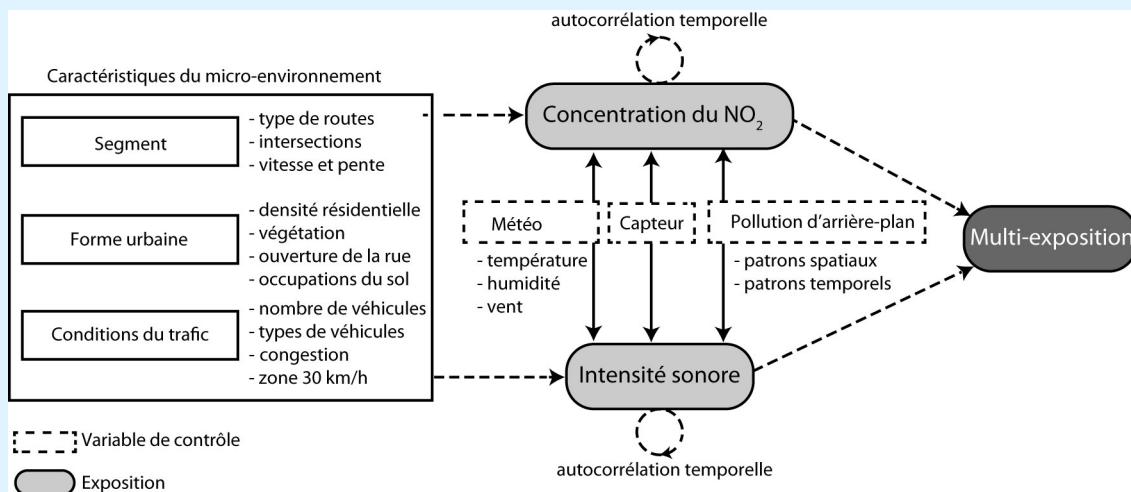


FIG. 7.1 : Exemple de cadre conceptuel

Notion de variables de contrôle *versus* variables explicatives

Dans un modèle de régression, nous distinguons habituellement trois types de variables : la variable dépendante (Y) que nous souhaitons prédire ou expliquer et les variables indépendantes (X) qui peuvent être soit des variables de contrôle (*covariates* en anglais), soit des variables explicatives. Les premières sont des facteurs qu'il faut prendre en compte (contrôler) avant d'évaluer nos variables d'intérêt (explicatives).

Dans l'exemple précédent, les chercheurs voulaient évaluer l'impact des caractéristiques de l'environnement urbain (variables explicatives) sur les expositions des cyclistes au dioxyde d'azote et au bruit, et ce, une fois contrôlés les effets de facteurs reconnus comme ayant un impact significatif sur la concentration de ces polluants (conditions météorologiques et la pollution d'arrière-plan). Autrement dit, si les variables de contrôle n'avaient pas été prises en compte, l'étude des variables d'intérêt serait biaisée par les effets de ces facteurs qui n'auraient pas été contrôlés. À titre d'exemple, il est possible que les zones de circulation limitées à 30 km/h soient concentrées dans les quartiers centraux et denses de Paris. Dans ces quartiers, la pollution d'arrière-plan a tendance à être supérieure. Si nous tenons pas compte de cette pollution d'arrière-plan, nous pourrions arriver à la conclusion que les zones de 30 km/h sont des milieux dans lesquels les cyclistes sont plus exposés à la pollution atmosphérique.

Construction de modèles de régression imbriqués, incrémentiels

En lien avec le cadre conceptuel du modèle, il est fréquent de construire plusieurs modèles emboîtés. Par exemple, à partir du cadre conceptuel (figure 7.1), les auteurs auraient très bien pu construire quatre modèles :

- un premier avec uniquement les variables de contrôle (modèle A);
- un second incluant les variables de contrôle et les variables explicatives de la dimension des caractéristiques du segment (modèle B);
- un troisième reprenant les variables du modèle B dans lequel sont introduites les variables explicatives relatives à la forme urbaine (modèle C);
- un dernier modèle dans lequel sont ajoutées les variables explicatives relatives aux conditions du trafic (modèle D).

L'intérêt d'une telle approche est qu'elle permet d'évaluer successivement l'apport explicatif de chacune des dimensions du modèle ; nous y reviendrons dans la section 5.3.

Nous disons alors que deux modèles sont imbriqués lorsque le modèle avec le plus de variables comprend également **toutes** les variables du modèle avec le moins de variables.

7.2 Principes de base de la régression linéaire multiple

7.2.1 Un peu d'équations...

La régression linéaire multiple vise à déterminer une équation qui résume le mieux les relations linéaires entre une variable dépendante (Y) et un ensemble de variables indépendantes (X). L'équation de régression s'écrit alors :

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \epsilon_i \quad (7.1)$$

avec :

- y_i , la valeur de la variable dépendante Y pour l'observation i
- β_0 , la constante, soit la valeur prédictive pour Y quand toutes les variables indépendantes sont égales à 0
- k le nombre de variables indépendantes
- β_1 à β_k , les coefficients de régression pour les variables indépendantes de 1 à k (X_1 à X_k)
- ϵ_i , le résidu pour l'observation de i , soit la partie de la valeur de y_i qui n'est pas expliquée par le modèle de régression.

Notez qu'il existe plusieurs écritures simplifiées de cette équation. D'une part, il est possible de ne pas indiquer l'observation i et de remplacer les lettres grecques *bêta* et *epsilon* (β et ϵ) par les lettres b et e :

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_k X_k + e \quad (7.2)$$

D'autre part, cette équation peut être présentée sous forme matricielle. Rappelez-vous que, pour chacune des n observations de l'échantillon, une équation est formulée :

$$\begin{cases} y_1 = \beta_0 + \beta_1 x_{1,1} + \dots + \beta_p x_{1,k} + \epsilon_1 \\ y_2 = \beta_0 + \beta_1 x_{2,1} + \dots + \beta_p x_{2,k} + \epsilon_2 \\ \dots \\ y_n = \beta_0 + \beta_1 x_{n,1} + \dots + \beta_p x_{n,k} + \epsilon_n \end{cases} \quad (7.3)$$

Par conséquent, sous forme matricielle, l'équation s'écrit :

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,k} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix} \quad (7.4)$$

ou tout simplement :

$$Y = X\beta + \epsilon \quad (7.5)$$

avec :

- Y , un vecteur de dimension $n \times 1$ pour la variable dépendante, soit une colonne avec n observations
- X , une matrice de dimension $n \times (k + 1)$ pour les k variables indépendantes, incluant une autre colonne (avec la valeur de 1 pour les n observations) pour la constante d'où $k + 1$
- β , un vecteur de dimension $k + 1$, soit les coefficients de régression pour les k variables et la constante
- ϵ , un vecteur de dimension $n \times 1$ pour les résidus.



Vous aurez compris que, comme pour la régression linéaire simple (section 4.4), l'équation de la régression linéaire multiple comprend aussi une partie expliquée et une autre non expliquée (stochastique) par le modèle :

$$Y = \underbrace{\beta_0 + \beta_1 X_i + \beta_2 X_i + \dots + \beta_k X_k}_{\text{partie expliquée par le modèle}} + \underbrace{\epsilon}_{\text{partie non expliquée (stochastique)}} \quad (7.6)$$

$$Y = \underbrace{X\beta}_{\text{partie expliquée par le modèle}} + \underbrace{\epsilon}_{\text{partie non expliquée (stochastique)}} \quad (7.7)$$

7.2.2 Hypothèses de la régression linéaire multiple

Un modèle est bien construit s'il respecte plusieurs hypothèses liées à la régression, dont les principales étant :

- **Hypothèse 1.** *La variable dépendante doit être continue et non-bornée.* Quant aux variables indépendantes (VI), elles peuvent être quantitatives (discrètes ou continues) et qualitatives (nominale ou ordinaire).
- **Hypothèse 2.** *La variance de chaque VI doit être supérieure à 0.* Autrement dit, toutes les observations ne peuvent avoir la même valeur.
- **Hypothèse 3.** *Indépendance des termes d'erreur.* Les résidus des observations $(\epsilon_1, \epsilon_2, \dots, \epsilon_n)$ ne doivent pas être corrélés entre eux. Autrement dit, les observations doivent être indépendantes les unes des autres, ce qui n'est souvent pas le cas pour des mesures temporelles. Par exemple, l'application du cadre conceptuel sur la modélisation de l'exposition des cyclistes au bruit et à la pollution atmosphérique (figure 7.1) est basée sur des données primaires collectées lors de trajets réalisés à vélo dans une ville donnée. Par conséquent, deux observations qui se suivent ont bien plus de chances de se ressembler – du point de vue des mesures de pollution et des caractéristiques de l'environnement urbain – que deux observations tirées au hasard dans le jeu de données. Ce problème d'autocorrélation temporelle doit être contrôlé, sinon, les coefficients de régression seront biaisés.
- **Hypothèse 4.** *Normalité des résidus* avec une moyenne centrée sur zéro.
- **Hypothèse 5.** *Absence de colinéarité parfaite entre les variables explicatives.* Par exemple, dans un modèle, nous ne pouvons pas introduire à la fois les pourcentages de locataires et de propriétaires, car pour chaque observation, la somme des deux donne 100%. Nous avons donc une corrélation parfaite entre ces deux variables : le coefficient de corrélation de Pearson entre ces deux variables est égal à 1. Par conséquent, le modèle ne peut pas être estimé avec ces deux variables et l'une des deux est automatiquement ôtée.
- **Hypothèse 6.** *Homoscédasticité des erreurs (ou absence d'hétéroscédasticité).* Les résidus doivent avoir une variance constante, c'est-à-dire qu'elle doit être la même pour chaque observation. Il y a homoscédasticité lorsqu'il y a une absence de corrélation entre les résidus et les valeurs prédictives. Si cette condition n'est pas respectée, nous parlons alors d'hétéroscédasticité.
- **Hypothèse 7.** *Le modèle est bien spécifié.* Un modèle est mal spécifié (construit) quand « une ou plu-

sieurs variables non pertinentes sont incluses dans le modèle » ou « qu'une ou plusieurs variables pertinentes sont exclues du modèle » (Bressoux 2010, 138-139). Concrètement, l'inclusion d'une variable non pertinente ou l'omission d'une variable peut entraîner une mauvaise estimation des effets des variables explicatives du modèle.

Pour connaître les conséquences de la violation de chacune de ces hypothèses, vous pourrez notamment consulter l'excellent ouvrage de Bressoux (2010, 103-110). Retenez ici que le non-respect de ces hypothèses produit des coefficients de régression biaisés.

7.3 Évaluation de la qualité d'ajustement du modèle

Pour illustrer la régression linéaire multiple, nous utilisons un jeu de données tiré d'un article portant sur la distribution spatiale de la végétation sur l'île de Montréal abordée sous l'angle de l'équité environnementale (Apparicio, Pham et al. 2016). Dans cette étude, les auteurs veulent vérifier si certains groupes de population (personnes à faible revenu, minorités visibles, personnes âgées et enfants de moins de 15 ans) ont ou non une accessibilité plus limitée à la végétation urbaine. En d'autres termes, cet article tente de répondre à la question suivante : une fois contrôlées les caractéristiques de la forme urbaine (densité de population et âge du bâti), est-ce que les quatre groupes de population résident dans des îlots urbains avec proportionnellement moins ou plus de végétation ?

Dans le tableau 7.1, sont reportées les variables utilisées (calculées au niveau des îlots de l'île de Montréal) introduites dans le modèle de régression :

- le pourcentage de la superficie de l'îlot couverte par de la végétation, soit la variable indépendante (VI) ;
- deux variables indépendantes de contrôle (VC) relatives à la forme urbaine ;
- les pourcentages des quatre groupes de population comme variables indépendantes explicatives (VE).

Notez que ce jeu de données est utilisé tout au long du chapitre. L'équation de départ du premier modèle de régression est donc :

VegPct ~ HABHA + AgeMedian + Pct_014 + Pct_65P + Pct_MV + Pct_FR

7.3.1 Mesures de la qualité d'un modèle

Comme pour la régression linéaire simple (section 4.4), les trois mesures les plus couramment utilisées pour évaluer la qualité d'un modèle sont :

- Le **coefficient de détermination** (R^2) qui indique la proportion de la variance de la variable dépendante expliquée par les variables indépendantes du modèle (équation (7.9)). Il varie ainsi de 0 à 1.

TAB. 7.1 : Statistiques descriptives pour les variables du modèle

Nom	Intitulé	Type	Moy.	E.-T.	Q1	Q2	Q3
VegPct	Végétation (%)	VD	35,1	18,6	20,3	33,8	49,0
HABHA	Habitants au km ²	VC	87,8	74,0	36,9	68,4	120,5
AgeMedian	Âge médian des bâtiments	VC	52,1	25,2	37,2	49,0	61,0
Pct_014	Moins de 15 ans (%)	VE	15,9	5,3	12,5	15,9	19,3
Pct_65P	65 ans et plus (%)	VE	14,9	8,3	9,6	13,9	18,2
Pct_MV	Minorités visibles (%)	VE	21,0	16,4	8,3	17,2	29,6
Pct_FR	Personnes à faible revenu (%)	VE	23,6	16,0	11,1	21,3	33,7

- La **statistique de Fisher** qui permet d'évaluer la significativité globale du modèle (équation (7.10)). Dans le cas d'une régression linéaire multiple, l'hypothèse nulle du test F est que toutes les valeurs des coefficients de régression des variables indépendantes sont égales à 0; autrement dit, qu'aucune des variables indépendantes n'a d'effet sur la variable dépendante. Tel que décrit à la section 4.4.3, il est possible d'obtenir une valeur de p rattachée à la statistique F avec k degrés de liberté au dénominateur et $n-k-1$ degrés de liberté au numérateur (k et n étant respectivement le nombre de variables indépendantes et le nombre d'observations). Lorsque la valeur de p est inférieure à 0,05, nous pourrons en conclure que le modèle est globalement significatif, c'est-à-dire qu'au moins un coefficient de régression est significativement différent de zéro. Notez qu'il est plutôt rare qu'un modèle de régression, comprenant plusieurs variables indépendantes, soit globalement non significatif ($P > 0,05$), et ce, surtout s'il est basé sur un cadre conceptuel et théorique solide. Le test de la statistique de Fisher est donc facile à passer et ne constitue pas une preuve absolue de la pertinence du modèle.
- L'**erreur quadratique moyenne (RMSE)** qui indique l'erreur absolue moyenne du modèle exprimée dans l'unité de mesure de la variable dépendante, autrement dit l'écart absolu moyen entre les valeurs observées et prédictes du modèle (équation (7.11)). Une valeur élevée indique que le modèle se trompe largement en moyenne et inversement.



Rappel sur la décomposition de la variance et calcul du R^2 , de la statistique F et du RMSE

Rappelez-vous que la variance totale (SCT) est égale à la somme de la variance expliquée (SCE) par le modèle et de la variance non expliquée (SCR) par le modèle.

$$\underbrace{\sum_{i=1}^n (y_i - \bar{y})^2}_{\text{variance de Y}} = \underbrace{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}_{\text{var. expliquée}} + \underbrace{\sum_{i=1}^n (y_i - \hat{y}_i)^2}_{\text{var. non expliquée}} \Rightarrow SCT = SCE + SCR \quad (7.8)$$

avec :

- y_i est la valeur observée de la variable dépendante pour i ;
- \bar{y} est la valeur moyenne de la variable dépendante;
- \hat{y}_i est la valeur prédictive de la variable dépendante pour i .

À partir des trois variances (totale, expliquée et non expliquée), il est alors possible de calculer les trois mesures de la qualité d'ajustement du modèle.

$$R^2 = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = \frac{SCE}{SCT} \text{ avec } R^2 \in [0, 1] \quad (7.9)$$

$$F = \frac{\frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{k}}{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-k-1}} = \frac{\frac{SCE}{k}}{\frac{SCR}{n-k-1}} = \frac{\frac{R^2}{k}}{\frac{1-R^2}{n-k-1}} = \frac{(n-k-1)R^2}{k(1-R^2)} \quad (7.10)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} = \sqrt{\frac{SCR}{n}} \quad (7.11)$$

Globalement, plus un modèle de régression est efficace, plus les valeurs du R^2 et de la statistique F sont élevées et inversement, plus celle de RMSE est faible. En effet, remarquez qu'à l'équation (7.10), la statistique F peut être obtenue à partir du R^2 ; par conséquent, plus la valeur du R^2 est forte (proche de 1), plus celle de F est aussi élevée. Notez aussi que plus un modèle est performant, plus la partie expliquée par

le modèle (SCE) est importante et plus celle non expliquée (SCR) est faible; ce qui signifie que plus le R^2 est proche de 1 (équation (7.9)), plus le RMSE – calculé à partir du SCR – est faible (équation (7.11)).

La syntaxe R ci-dessous illustre comment calculer les différentes variances (SCT, SCE et SCR) à partir des valeurs observées et prédictes par le modèle, puis les valeurs du R^2 , de F et du RMSE. Nous verrons par la suite qu'il est possible d'obtenir directement ces valeurs à partir de la fonction `summary(VotreModèle)`.

```
# Chargement des données
load("data/lm/DataVegetation.RData")

# Construction du modèle de régression
Modele1 <- lm(VegPct ~ HABHA+AgeMedian+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)

# Nombre d'observations
n <- nrow(DataFinal)

# Nombre de variables indépendantes (coefficients moins la constante)
k <- length(Modele1$coefficients)-1

# Vecteur pour les valeurs observées
Yobs <- DataFinal$VegPct

# Vecteur pour les valeurs prédictes
Ypredict <- Modele1$fitted.values

# Variance totale
SCT <- sum((Yobs-mean(Yobs))^2)

# Variance expliquée
SCE <- sum((Ypredict-mean(Yobs))^2)

# Variance résiduelle
SCR <- sum((Yobs-Ypredict)^2)

# Calcul du coefficient de détermination (R2)
R2 <- SCE / SCT

# Calcul de la valeur de F
valeurF <- (R2 / k) /((1-R2)/(n-k-1))

cat("R2 =", round(SCE / SCT,4),
    "\nF de Fisher = ", round(valeurF,0),
    "\nRMSE =", round(sqrt(SCR/n),4)
  )

## R2 = 0.4182
## F de Fisher = 1223
## RMSE = 14.1575
```

7.3.2 Comparaison des modèles incrémentiels

Tel que signalé plus haut, il est fréquent de construire plusieurs modèles de régression imbriqués. Cette démarche est très utile pour évaluer l'apport de l'introduction d'un nouveau bloc de variables dans un modèle. De manière exploratoire, cela permet également de vérifier si l'introduction d'une variable in-

dépendante supplémentaire dans un modèle a ou non un apport significatif et ainsi de décider de la conserver, ou non, dans le modèle final selon le principe de parcimonie.



Le principe de parcimonie

Le principe de parcimonie appliqué aux régressions correspond à l'idée qu'il est préférable de disposer d'un **modèle plus simple** que d'un **modèle compliqué** pour expliquer un phénomène si la qualité de leurs prédictions – qualité d'ajustement des deux modèles – est équivalente.

Une première justification de ce principe trouve son origine dans la philosophie des sciences avec le **rasoir d'Ockham**. Il s'agit d'un principe selon lequel il est préférable de privilégier des théories faisant appel à un plus petit nombre d'hypothèses. L'idée centrale étant d'éviter d'apporter des réponses à une question qui soulèveraient davantage de nouvelles questions. Dans le cas d'une régression, nous pourrions être tenté d'ajouter de nombreuses variables indépendantes pour améliorer la capacité de prédiction du modèle. Cette stratégie conduit généralement à observer des relations contraires à nos connaissances entre les variables du modèle, ce qui soulève de nouvelles questions de recherche (pas toujours judicieuses...). Dans notre quotidien, si une casserole tombe de son support, il est plus raisonnable d'imaginer que nous l'avons mal fixée que d'émettre l'hypothèse qu'un fantôme l'a volontairement fait tomber! Cette seconde hypothèse soulève d'autres questions (pas toujours judicieuses...) sur la nature d'un fantôme, son identité, la raison le poussant à agir, etc.

Une seconde justification de ce principe s'observe dans la pratique statistique : des modèles plus complexes ont souvent une plus faible capacité de généralisation. En effet, un modèle complexe et trop bien ajusté aux données observées est souvent incapable d'effectuer des prédictions justes pour de nouvelles données. Ce phénomène est appelé surajustement ou surinterprétation (*overfitting* en anglais). Le surajustement résultant de modèles trop complexes entre en conflit direct avec l'enjeu principal de l'inférence en statistique : pouvoir généraliser des observations faites sur un échantillon au reste d'une population.

Notez que ce principe de parcimonie ne signifie pas que vous devez systématiquement retirer toutes les variables non significatives de votre analyse. En effet, il peut y avoir un intérêt théorique à démontrer l'absence de relation entre des variables. Il s'agit plutôt d'une ligne de conduite à garder à l'esprit lors de l'élaboration du cadre théorique et de l'interprétation des résultats.

Mathématiquement, plus nous ajoutons de variables supplémentaires dans un modèle, plus le R^2 augmente. On ne peut donc pas utiliser directement le R^2 pour comparer deux modèles de régression ne comprenant pas le même nombre de variables indépendantes. Nous privilégions alors l'utilisation du R^2 ajusté qui, comme illustré dans l'équation (7.12), tient compte à la fois des nombres d'observations et des variables indépendantes utilisées pour construire le modèle.

$$R_{\text{ajusté}}^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1} \quad \text{avec } R_{\text{ajusté}}^2 \in [0, 1] \quad (7.12)$$

Si le R^2 ajusté du second modèle est supérieur au premier modèle, cela signifie qu'il y a un gain de la variance expliquée entre le premier et le second modèle. Ce gain est-il pour autant significatif? Pour y répondre, il convient de comparer les valeurs des statistiques F des deux modèles. Pour ce faire, nous calculons le F incrémentiel et la valeur de p qui lui est associé avec comme degrés de liberté, le nombre de variables indépendantes ajoutées ($k_2 - k_1$) et $n - k_2 - 1$. Si la valeur de $p < 0,05$, nous pouvons conclure que le gain de variance expliquée par le second modèle est significatif comparativement au premier modèle (au seuil de 5nbsp;%).

$$F_{\text{incrémentiel}} = \frac{\frac{R_2^2 - R_1^2}{k_2 - k_1}}{\frac{1 - R_2^2}{n - k_2 - 1}} \quad (7.13)$$

avec R_1^2 et R_2^2 étant les coefficients de détermination des modèles 1 et 2 et k_1 et k_2 étant les nombres de

variables indépendantes qu'ils comprennent ($k_2 > k_1$).

Illustrons le tout avec deux modèles. Dans la syntaxe R ci-dessous, nous avons construit un premier modèle avec uniquement les variables de contrôle (`modele1`), soit deux variables indépendantes (`HABHA` et `AgeMedian`). Puis, dans un second modèle (`modele2`), nous ajoutons comme variables indépendantes les pourcentages des quatre groupes de population (`Pct_014`, `Pct_65P`, `Pct_MV`, `Pct_FR`). Repérez comment sont calculés les R^2 ajustés pour les modèles et le F incrémentiel.

Le R^2 ajusté passe de 0,269 à 0,418 des modèles 1 à 2 signalant que l'ajout des quatre variables indépendantes augmente considérablement la variance expliquée. Autrement dit, le second modèle est bien plus performant. Le F incrémentiel s'élève à 653,8 et est significatif ($p < 0,001$). Notez que la syntaxe ci-dessous illustre comment calculer les valeurs du R^2 ajusté et du F incrémentiel à partir des équations (7.12) et (7.13). Sachez toutefois qu'il est possible d'obtenir directement le R^2 ajusté avec la fonction `summary(VotreModèle)` et le F incrémentiel avec la fonction `anova(modele1, modele2)`.

```

modele1 <- lm(VegPct ~ HABHA+AgeMedian, data = DataFinal)
modele2 <- lm(VegPct ~ HABHA+AgeMedian+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)

# nombre d'observations pour les deux modèles
n1 <- length(modele1$fitted.values)
n2 <- length(modele2$fitted.values)

# nombre de variables indépendantes
k1 <- length(modele1$coefficients)-1
k2 <- length(modele2$coefficients)-1

# coefficient de détermination
R2m1 <- summary(modele1)$r.squared
R2m2 <- summary(modele2)$r.squared

# coefficient de détermination ajusté
R2ajustm1 <- 1-((n1-1)*(1-R2m1)) / (n1-k1-1)
R2ajustm2 <- 1-((n2-1)*(1-R2m2)) / (n2-k2-1)

# Statistique F
Fm1 <- summary(modele1)$fstatistic[1]
Fm2 <- summary(modele2)$fstatistic[1]

# F incrémentiel
Fincrementiel <- ((R2m2-R2m1) / (k2 - k1)) / ((1-R2m2)/(n2-k2-1))
pFinc <- pf(Fincrementiel, k2-k1, n2-k2-1, lower.tail = FALSE)

cat("\nR2 (modèle 1) =", round(R2m1,4),
    "; R2 ajusté = ", round(R2ajustm1,4),
    "; F =", round(Fm1, 1),
    "\nR2 (modèle 2) =", round(R2m2,4),
    "; R2 ajusté = ", round(R2ajustm2,4),
    "; F =", round(Fm2, 1),
    "\nF incrémentiel =", round(Fincrementiel,1),
    "; p = ", round(pFinc,3)
)

## 
## R2 (modèle 1) = 0.2691 ; R2 ajusté = 0.269 ; F = 1879.2
## R2 (modèle 2) = 0.4182 ; R2 ajusté = 0.4179 ; F = 1222.5

```

```

## F incrémentiel = 653.8 ; p = 0

# F incrémentiel avec la fonction anova
anova(modele1, modele2)

## Analysis of Variance Table
##
## Model 1: VegPct ~ HABHA + AgeMedian
## Model 2: VegPct ~ HABHA + AgeMedian + Pct_014 + Pct_65P + Pct_MV + Pct_FR
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1 10207 2570964
## 2 10203 2046427  4      524537 653.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

7.4 Différentes mesures pour les coefficients de régression

La fonction `summary(nom du modèle)` permet d'obtenir les résultats du modèle de régression. D'emblée, signalons que le modèle est globalement significatif ($F(6,10203) = 1123, p = 0,000$) avec un R^2 de 0,4182 indiquant que les variables indépendantes du modèle expliquent 41,82% de la variance du pourcentage de végétation dans les îlots de l'île de Montréal.

```

modelereg <- lm(VegPct ~ HABHA+AgeMedian+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)
summary(modelereg)

```

```

##
## Call:
## lm(formula = VegPct ~ HABHA + AgeMedian + Pct_014 + Pct_65P +
##     Pct_MV + Pct_FR, data = DataFinal)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -48.876 -9.757 -0.232  9.499 103.830
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.355774  0.882235  29.874 <2e-16 ***
## HABHA       -0.070401  0.002202 -31.975 <2e-16 ***
## AgeMedian    0.010790  0.006369   1.694  0.0902 .
## Pct_014      1.084478  0.032179  33.702 <2e-16 ***
## Pct_65P      0.400531  0.018835  21.265 <2e-16 ***
## Pct_MV      -0.031112  0.010406 -2.990  0.0028 **
## Pct_FR      -0.348256  0.011640 -29.918 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.16 on 10203 degrees of freedom
## Multiple R-squared:  0.4182, Adjusted R-squared:  0.4179
## F-statistic: 1223 on 6 and 10203 DF,  p-value: < 2.2e-16

```

7.4.1 Coefficients de régression : évaluer l'effet des variables indépendantes

Les différents résultats pour les coefficients sont reportés au tableau 7.2.

La constante (β_0) est la valeur attendue de la variable dépendante (Y) quand les valeurs de toutes les variables indépendantes sont égales à 0. Pour ce modèle, quand les variables indépendantes sont égales à 0, plus du quart de la superficie des îlots serait en moyenne couverte par de la végétation ($\beta_0 = 26,36$). Notez que la constante n'a pas toujours une interprétation pratique. Il est par exemple très invraisemblable de trouver un îlot avec de la population dans lequel il n'y aurait aucune personne à faible revenu, aucune personne ne déclarant appartenir à une minorité visible, aucun enfant de moins de 15 ans et aucune personne âgée 65 ans et plus. La constante a donc avant tout un rôle mathématique dans le modèle.

Le coefficient de régression (β_1 à β_k) indique le changement de la variable dépendante (Y) lorsque la variable indépendante augmente d'une unité, toutes choses étant égales par ailleurs. Il permet ainsi d'évaluer l'effet d'une augmentation d'une unité dans laquelle est mesurée la VI sur la VD.

Que signifie l'expression *toutes choses étant égales par ailleurs* pour un coefficient de régression ?

Après l'apprentissage du grec, grâce aux nombreuses équations intégrées au livre, passons au latin ! L'expression *toutes choses étant égales par ailleurs* vient du latin *ceteris paribus*, à ne pas confondre avec *c'est terrible Paris en bus* (petite blague formulée par un étudiant ayant suivi le cours *Méthodes quantitatives appliquées en études urbaines* à l'INRS il y a quelques années)! Certains auteurs emploient encore *ceteris paribus* : il est donc possible que vous la retrouviez dans un article scientifique...

Plus sérieusement, l'expression *toutes choses étant égales par ailleurs* signifie que l'on estime l'effet de la variable indépendante sur la variable dépendante, si toutes les autres variables indépendantes restent constantes ou autrement dit, une fois contrôlés tous les autres prédicteurs.

À partir des coefficients du tableau 7.2, l'équation du modèle de régression s'écrit alors comme suit :

$$\text{VegPct} = 26,356 - 0,070 \text{ HABHA} + 0,011 \text{ AgeMedian} + 1,084 \text{ Pct_014} + 0,401 \text{ Pct_65P} - 0,031 \text{ Pct_MV} - 0,348 \text{ Pct_FR} + e$$

Comment interpréter un coefficient de régression pour une variable indépendante ?

Le signe du coefficient de régression indique si la variable indépendante est associée positivement ou négativement avec la variable dépendante. Par exemple, plus la densité de population est importante à travers les îlots de l'île de Montréal, plus la couverture végétale diminue.

Quant à la valeur absolue du coefficient, elle indique la taille de l'effet du prédicteur. Par exemple, 1,084 signifie que si toutes les autres variables indépendantes restent constantes, alors le pourcentage de végétation dans l'îlot augmente de 1,084 points de pourcentage pour chaque différence d'un point de pourcentage d'enfants de moins de 15 ans. Toutes choses étant égales par ailleurs, une augmentation de 10% d'enfants dans un îlot entraîne alors une hausse de 10,8% de la couverture végétale dans l'îlot.

TAB. 7.2 : Différentes mesures pour les coefficients

Variable	Coef.	Erreur type	Valeur de T	P	coef. 2,5 %	coef. 97,5 %	
Constante	26,356	0,882	29,870	0,000	24,626	28,085	***
HABHA	-0,070	0,002	-31,970	0,000	-0,075	-0,066	***
AgeMedian	0,011	0,006	1,690	0,090	-0,002	0,023	.
Pct_014	1,084	0,032	33,700	0,000	1,021	1,148	***
Pct_65P	0,401	0,019	21,260	0,000	0,364	0,437	***
Pct_MV	-0,031	0,010	-2,990	0,003	-0,052	-0,011	**
Pct_FR	-0,348	0,012	-29,920	0,000	-0,371	-0,325	***

L'analyse des coefficients montre ainsi qu'une fois contrôlées les deux caractéristiques relatives à la forme urbaine (densité de population et âge médian des bâtiments), plus les pourcentages d'enfants et de personnes âgées sont élevés, plus la couverture végétale de l'îlot est importante ($B = 1,084$ et $0,401$), toutes choses étant égales par ailleurs. À l'inverse, de plus grands pourcentages de personnes à faible revenu et de minorités sont associés à une plus faible couverture végétale ($B = -0,348$ et $-0,031$).

L'erreur type du coefficient de régression

L'erreur type d'un coefficient permet d'évaluer son niveau de précision, soit le degré d'incertitude vis-à-vis du coefficient. Succinctement, elle correspond à l'écart-type de l'estimation (coefficient) ; elle est ainsi toujours positive. Plus la valeur de l'erreur type est faible, plus l'estimation du coefficient est précise. Notez toutefois qu'il n'est pas judicieux de comparer les erreurs types des coefficients pour des variables exprimées dans des unités de mesure différentes.

Comme nous le verrons plus loin, l'utilité principale de l'erreur type est qu'elle permet de calculer la valeur de t et l'intervalle de confiance du coefficient de régression.

7.4.2 Coefficients de régression standardisés : repérer les variables les plus importantes du modèle

Un coefficient de régression est exprimé dans les unités de mesure des variables indépendante (VI) et dépendante (VD) : une augmentation d'une unité de la VI a un effet de β (valeur de coefficient) unité de mesure sur la VD, toutes choses étant égales par ailleurs. Prenons l'exemple d'un modèle fictif dans lequel une variable indépendante mesurée en mètres obtient un coefficient de régression de $0,000502$. Si cette variable était exprimée en kilomètres et non en mètres, son coefficient serait alors de $0,502$ ($0,000502 \times 1000 = 0,502$). Cela explique que pour certaines variables, il est souvent préférable de modifier l'unité de mesure, particulièrement pour les variables de distance ou de revenu. Par exemple, dans un modèle de régression, nous introduisons habituellement une variable de revenu par tranche de mille dollars ou le loyer mensuel par tranche de cent dollars, puisque les coefficients du revenu ou de loyer exprimé en dollars risquent d'être extrêmement faibles. Concrètement, cela signifie que nous divisons la variable *revenu* par 1000 et celle du *loyer* par 100 avant de l'introduire dans le modèle.

Du fait de leur unités de mesure souvent différentes, vous aurez compris que nous ne pouvons pas comparer directement les coefficients de régression afin de repérer la ou les variables indépendantes (X) qui ont les effets (impacts) les plus importants sur la variable dépendante (Y). Pour remédier à ce problème, nous utilisons les **coefficients de régression standardisés**. Ces coefficients standardisés sont simplement les valeurs de coefficients de régression qui seraient obtenus si toutes les variables du modèle (VD et VI) étaient préalablement centrées réduites (soit avec une moyenne égale à 0 et un écart-type égal à 1; consultez la section 2.5.5.2 pour un rappel). Puisque toutes les variables du modèle sont exprimées en écarts-types, les coefficients standardisés permettent ainsi d'évaluer l'**effet relatif** des VI sur la VD. Cela permet ainsi de repérer la ou les variables les plus « importantes » du modèle.

L'interprétation d'un coefficient de régression standardisé est donc la suivante : il indique le changement en termes d'unités d'écart-type de la variable dépendante (Y) à chaque ajout d'un écart-type de la variable indépendante, toutes choses étant égales par ailleurs.

Le coefficient de régression standardisé peut être aussi facilement calculé en utilisant les écarts-types des deux variables VI et VD :

$$\beta_z = \beta \frac{s_x}{s_y} \quad (7.14)$$

La syntaxe R ci-dessous illustre trois façons d'obtenir les coefficients standardisés :

- en centrant et réduisant préalablement les variables avec la fonction `scale` avant de construire le modèle avec la fonction `lm`;
- en calculant les écarts-types de VD et de VI et en appliquant l'équation (7.14);
- avec la fonction `lm.beta` du package `QuantPsyc`. Cette dernière méthode est moins « verbeuse » (deux lignes de code uniquement), mais nécessite de charger un package supplémentaire.

```
# Modèle de régression
Modele1 <- lm(VegPct ~ HABHA+AgeMedian+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)

# Méthode 1 : lm sur des variables centrées réduites
ModeleZ <- lm(scale(VegPct) ~ scale(HABHA)+scale(AgeMedian)+
               scale(Pct_014)+scale(Pct_65P)+
               scale(Pct_MV)+scale(Pct_FR), data = DataFinal)
coefs <- ModeleZ$coefficients
coefs[1:length(coefs)]

##      (Intercept)    scale(HABHA)  scale(AgeMedian)  scale(Pct_014)
## 3.721649e-16 -2.806891e-01   1.467299e-02   3.093456e-01
## scale(Pct_65P)  scale(Pct_MV)  scale(Pct_FR)
## 1.788453e-01 -2.755087e-02  -3.004544e-01

# Méthode 2 : à partir de l'équation
# Écart-type de la variable dépendante
VDet <- sd(DataFinal$VegPct)
cat("Écart-type de Y =", round(VDet,3))

## Écart-type de Y = 18.562

# Écarts-types des variables indépendantes
VI <- c("HABHA","AgeMedian","Pct_014","Pct_65P","Pct_MV","Pct_FR")
VIet <- sapply(DataFinal[VI], sd)
# Coefficients de régression du modèle sans la constante
coefs <- Modele1$coefficients[1:length(VIet)+1]
# Coefficients de régression du modèle
coefstand <- coefs * (VIet / VDet)
coefstand

##          HABHA     AgeMedian      Pct_014      Pct_65P      Pct_MV      Pct_FR
## -0.28068906  0.01467299  0.30934560  0.17884535 -0.02755087 -0.30045437

# Méthode 3 : avec la fonction lm.beta du package QuantPsyc
library(QuantPsyc)
lm.beta(lm(VegPct ~ HABHA+AgeMedian+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal))

##          HABHA     AgeMedian      Pct_014      Pct_65P      Pct_MV      Pct_FR
## -0.28068906  0.01467299  0.30934560  0.17884535 -0.02755087 -0.30045437
```

Par exemple, pour la variable `Pct_014`, le coefficient de régression standardisé est égal à :

$$\beta_z = 1,084 \times \frac{5,295}{18,562} = 0,309 \quad (7.15)$$

TAB. 7.3 : Calcul des coefficients standardisés

Variable dépendante	Écart-type	Coef.	Coef. standardisé
HABHA	74,008	-0,070	-0,281
AgeMedian	25,241	0,011	0,015
Pct_014	5,295	1,084	0,309
Pct_65P	8,289	0,401	0,179
Pct_MV	16,438	-0,031	-0,028
Pct_FR	16,015	-0,348	-0,300

avec 1,084 étant le coefficient de régression de Pct_014, 5,295 et 18,562 étant respectivement les écarts-types de Pct_014 (variable indépendante) et de VegPct (variable dépendante).

Au tableau 7.3, nous constatons que la valeur absolue du coefficient de régression pour HABHA est inférieure à celle de Pct_65P ($-0,070$ versus $0,401$), ce qui n'est pas le cas pour leur coefficient standardisé ($-0,281$ versus $0,179$). Rappelez-vous aussi que nous ne pouvons pas directement comparer les effets de ces deux variables à partir des coefficients de régression puisqu'elles sont exprimées dans des unités de mesure différentes : HABHA est exprimée en habitants par hectare et Pct_65P en pourcentage. À la lecture des coefficients standardisés, nous pouvons en conclure que la variable HABHA a un effet relatif plus important que Pct_65P ($-0,281$ versus $0,179$).

7.4.3 Significativité des coefficients de régression : valeurs de t et de p

Une fois les coefficients de régression obtenus, il convient de vérifier s'ils sont ou non significativement différents de 0. Si le coefficient de régression d'une variable indépendante est significativement différent de 0, nous concluons que la variable a un effet significatif sur la variable dépendante, toutes choses étant égales par ailleurs. Pour ce faire, il suffit de calculer la valeur de t qui est simplement le coefficient de régression divisé par son erreur type.

$$t = \frac{\beta_k - 0}{s(\beta_k)} \quad (7.16)$$

avec $s(\beta_k)$ étant l'erreur type du coefficient de régression. Notez que dans l'équation (7.16), nous indiquons habituellement -0 , pour signaler que l'on veut vérifier si le coefficient est différent de 0. En guise d'exemple, au tableau 7.2, la valeur de t de la variable HABHA est bien égale à :

$$-0,070401 / 0,002202 = -31,975.$$

Démarche pour vérifier si un coefficient est significativement différent de 0 avec un seuil de confiance

- Poser l'hypothèse nulle (H_0) stipulant que le coefficient est égal à 0, soit $H_0 : \beta_k = 0$. L'hypothèse alternative (H_1) est que le coefficient est différent de 0, soit $H_1 : \beta_k \neq 0$.
- Calculer la valeur de t , soit le coefficient de régression divisé par son erreur type (équation (7.16)).
- Calculer le nombre de degrés de liberté, soit $dl = n - k - 1$, n et k étant respectivement les nombres d'observations et de variables indépendantes.
- Choisir un seuil de signification alpha (5 %, 1 % ou 0,1 %, soit $p = 0,05, 0,01$ ou $0,01$).
- Trouver la valeur critique de t dans la table T de Student (14.3) avec p et le nombre de degrés de liberté (dl).
- Valider ou réfuter l'hypothèse nulle (H_0) :
 - si la valeur de t est inférieure à la valeur critique de t avec dl et le seuil choisi, nous confirmons H_0 : le coefficient n'est pas significativement différent de 0.

- si la valeur de t est supérieure à la valeur critique de t avec dl et le seuil choisi, nous réfutons l'hypothèse nulle, et choisissons l'hypothèse alternative (H_1) stipulant que le coefficient est significativement différent de 0.

Valeurs critiques de la valeur de t à retenir!

Lorsque le nombre de degrés de liberté ($n - k - 1$) est très important (supérieur à 2500), et donc le nombre d'observations de votre jeu de données, nous retenons habituellement les valeurs critiques suivantes : **1,65** ($p = 0,10$), **1,96** ($p = 0,05$), **2,58** ($p = 0,01$) et **3,29** ($p=0,001$). Concrètement, cela signifie que :

- une valeur de t supérieure à 1,96 ou inférieure à -1,96 nous informe que la relation entre la variable indépendante et la variable dépendante est significative positivement ou négativement au seuil de 5 %. Autrement dit, vous avez moins de 5 % de chances de vous tromper en affirmant que le coefficient de régression est bien significativement différent de 0.
- une valeur de t supérieure à 2,58 ou inférieure à -2,58 nous informe que la relation entre la variable indépendante et la variable dépendante est significative positivement ou négativement au seuil de 5 %. Autrement dit, vous avez moins de 1 % de chances de vous tromper en affirmant que le coefficient de régression est bien significativement différent de 0.
- une valeur de t supérieure à 3,29 ou inférieure à -3,29 nous informe que la relation entre la variable indépendante et la variable dépendante est significative positivement ou négativement au seuil de 5 %. Autrement dit, vous avez moins de 0,1 % de chances de vous tromper en affirmant que le coefficient de régression est bien significativement différent de 0.

Concrètement, retenez et utilisez les seuils de $\pm 1,96$, $\pm 2,58$ et $\pm 3,29$ pour repérer les variables significatives positivement ou négativement aux seuils respectifs de 0,5, 0,1 et 0,001.

Que signifient les seuils 0,10, 0,05 et 0,001 ?

L'interprétation exacte des seuils de significativité des coefficients d'une régression est quelque peu alambiquée, mais mérite de s'y attarder. En effet, indiquer qu'un coefficient est significatif est souvent perçu comme un argument fort pour une théorie, il est donc nécessaire d'avoir du recul et de bien comprendre ce que l'on entend par **significatif**.

Si un coefficient est significatif au seuil de 5 % dans notre modèle, cela signifie que si, pour l'ensemble d'une population, la valeur du coefficient est de 0 en réalité, alors nous avons moins de 5 % de chances de collecter un échantillon (pour cette population) ayant produit un coefficient aussi fort que celui que nous observons dans notre propre échantillon. Par conséquent, il serait très invraisemblable que le coefficient soit 0 puisque nous avons effectivement collecté un tel échantillon. Il s'agit d'une forme d'argumentation par l'absurde propre à la statistique fréquentiste.

Notez que si 100 études étaient conduites sur le même sujet et dans les mêmes conditions, nous nous attendrions à ce que 5 d'entre elles trouvent un coefficient significatif, du fait de la variation des échantillons. Ce constat souligne le fait que la recherche est un effort collectif et qu'une seule étude n'est pas suffisante pour trancher sur un sujet. Les revues systématiques de la littérature sont donc des travaux particulièrement importants pour la construction du consensus scientifique.

Ne pas confondre significativité et effet de la variable indépendante

Attention, un coefficient significatif n'est pas toujours intéressant ! Autrement dit, bien qu'il soit significatif à un seuil donné (par exemple, $p = 0,05$), cela ne veut pas dire pour autant qu'il ait un effet important sur la variable dépendante. Il faut donc analyser simultanément les valeurs de p et des coefficients de régression. Afin de mieux saisir l'effet d'un coefficient significatif, il est intéressant de représenter graphiquement l'effet marginal d'une variable indépendante (VI) sur une variable dépendante (VD), une fois contrôlées les autres VI du modèle de régression (section [7.7.4](#)).

Prenons deux variables indépendantes du tableau [7.2](#) – HABHA et AgeMedian – et vérifions si leurs coefficients de régression respectifs (-0,070 et 0,011) sont significatifs. Appliquons la démarche décrite dans l'encadré ci-dessus :

1. Nous posons l'hypothèse nulle stipulant que la valeur de ces deux coefficients est égale à 0, soit $H_0 : \beta_k = 0$.
 2. La valeur de t est égale à $-0,070401 / 0,002202 = -31,97139$ pour HABHA et à $0,010790 / 0,006369 = 1,694144$ pour AgeMedian.
 3. Le nombre de degrés de liberté est égal à $dl = n - k - 1 = 10\,210 - 6 - 1 = 10\,203$.
 4. Nous choisissons respectivement les seuils α de 0,10, 0,05, 0,01 ou 0,001.
 5. Avec 10210 degrés de liberté, les valeurs critiques de la table T de Student (section 14.3) sont de 1,65 ($p = 0,10$), 1,96 ($p = 0,05$), 2,58 ($p = 0,01$), 3,29 ($p = 0,001$).
 6. Il reste à valider ou réfuter l'hypothèse nulle (H_0) :
- pour HABHA, la valeur absolue de t (-31,975) est supérieure à la valeur critique de 3,29. Son coefficient de régression est donc significativement différent de 0. Autrement dit, ce prédicteur a un effet significatif et négatif sur la variable dépendante.
 - pour AgeMedian, la valeur absolue de t (1,694) est supérieure à 1,65 ($p = 0,10$), mais inférieure à 1,96 ($p = 0,05$), à 2,58 ($p = 0,01$), à 3,29 ($p = 0,001$). Par conséquent, ce coefficient est différent de 0 uniquement au seuil de $p = 0,10$ et non au seuil de $p = 0,05$. Cela signifie que nous avons un peu moins de 10 % de chances de se tromper en affirmant que cette variable a un effet significatif sur la variable dépendante.



Calculer et obtenir des valeurs de p dans R

Il est très rare que d'utiliser directement la table T de Student pour obtenir un seuil de significativité.

D'une part, il est possible de calculer directement la valeur de p à partir de la valeur de t et du nombre de degrés de liberté avec la fonction `pt` avec les paramètres suivants :

```
pt(q= abs(valeur de T), df= nombre de degrés de liberté, lower.tail = F) *2
```

```
# Degrés de liberté
dl <- nrow(DataFinal) - (length(Modele1$coefficients) - 1) + 1

# Valeurs de T
ValeurT <- summary(Modele1)$coefficients[,3]

# Calcul des valeurs de P
ValeurP <- pt(q= abs(ValeurT), df= dl, lower.tail = F) *2

df_tp <- data.frame(
  ValeurT = round(ValeurT,3),
  ValeurP = round(ValeurP,3)
)
print(df_tp)
```

```
##           ValeurT ValeurP
## (Intercept) 29.874  0.000
## HABHA      -31.975  0.000
## AgeMedian    1.694  0.090
## Pct_014     33.702  0.000
## Pct_65P     21.265  0.000
## Pct_MV      -2.990  0.003
## Pct_FR      -29.918  0.000
```

D'autre part, la fonction `summary` renvoie d'emblée les valeurs de t et de p . Par convention, R, comme la plupart des logiciels d'analyses statistiques, utilise aussi des symboles pour indiquer le seuil de signification du

coefficient (voir tableau 7.3) :

'***' $p \leq 0,001$
 '**' $p \leq 0,01$
 '*' $p \leq 0,05$
 '.' $p \leq 0,10$

7.4.4 Intervalle de confiance des coefficients

Finalement, il est possible de calculer l'intervalle de confiance d'un coefficient à partir d'un niveau de signification (habituellement 0,95 ou encore 0,99). Pour ce faire, la fonction `confint(nom du modèle, level=.95)` est très utile. L'intérêt de ces intervalles de confiance pour les coefficients de régression est double :

- il permet de vérifier si le coefficient est ou non significatif au seuil retenu. Pour cela, la borne inférieure et la borne supérieure du coefficient doivent être toutes deux négatives ou positives. À l'inverse, un intervalle à cheval sur 0, soit avec une borne inférieure négative et une borne supérieure positive, n'est pas significatif.
- il permet d'estimer la précision de l'estimation ; plus l'intervalle du coefficient est réduit, plus l'estimation de l'effet de la variable indépendante est précise. Inversement, un intervalle large signale que le coefficient est incertain.

Cela explique que de nombreux auteurs reportent les intervalles de confiance dans les articles scientifiques (habituellement à 95 %). Dans le modèle présenté ici, il est alors possible d'écrire : toutes choses étant égales par ailleurs, le pourcentage d'enfants de moins de 15 ans est positivement et significativement associé avec le pourcentage de la couverture végétale dans l'îlot ($B = 1,084$; IC 95 % = [1,021 - 1,148], $p < 0,001$).

En guise d'exemple, à la lecture de la sortie R ci-dessous, l'estimation de l'effet de la variable indépendante `AgeMedian` sur la variable `VegPct` se situe dans l'intervalle -0,002 à 0,023 qui est à cheval sur 0. Contrairement aux autres variables, nous ne pouvons donc pas en conclure que cet effet est significatif avec $p = 0,05$.

```
# Intervalle de confiance à 95 % des coefficients
round(confint(Modele1, level=.95), 3)
```

```
##              2.5 % 97.5 %
## (Intercept) 24.626 28.085
## HABHA       -0.075 -0.066
## AgeMedian   -0.002  0.023
## Pct_014      1.021  1.148
## Pct_65P      0.364  0.437
## Pct_MV      -0.052 -0.011
## Pct_FR      -0.371 -0.325
```

💡 Comment est calculé un intervalle de confiance ?

L'intervalle du coefficient est obtenu à partir de :

1. la valeur du coefficient (β_k),

2. la valeur de son erreur type $s(\beta_k)$ et
3. la valeur critique de T ($t_{\alpha/2}$) obtenue avec $n - k - 1$ degrés de liberté et le niveau de significativité retenu (95 %, 99 % ou 99,9 %).

$$IC_{\beta_k} = [\beta_k - t_{\alpha/2} \times s(\beta_k); \beta_k + t_{\alpha/2} \times s(\beta_k)] \quad (7.17)$$

Autrement dit, lorsque vous disposez d'un nombre très important d'observations, les intervalles de confiance s'écrivent simplement avec les fameuses valeurs critiques de T de 1,96, 2,58, 3,29 :

$$\text{Intervalle à 95 \% } IC_{\beta_k} = [\beta_k - 1,96 \times s(\beta_k); \beta_k + 1,96 \times s(\beta_k)] \quad (7.18)$$

$$\text{Intervalle à 99 \% } IC_{\beta_k} = [\beta_k - 2,58 \times s(\beta_k); \beta_k + 2,58 \times s(\beta_k)] \quad (7.19)$$

$$\text{Intervalle à 99,9 \% } IC_{\beta_k} = [\beta_k - 3,29 \times s(\beta_k); \beta_k + 3,29 \times s(\beta_k)] \quad (7.20)$$

La syntaxe R ci-dessous illustre comment calculer les intervalles de confiance à 95 % à partir de l'équation (7.17). Rappelez-vous toutefois qu'il est bien plus simple d'utiliser la fonction `confint` :

- `round(confint(Model1, level=.95),3)`
- `round(confint(Model1, level=.99),3)`
- `round(confint(Model1, level=.999),3)`

```
# Coefficients de régression
coeffs <- Model1$coefficients

# Erreur type des coef.
coeffs_se <- summary(Model1)$coefficients[,2]

# Nombre de degrés de liberté
n <- length(Model1$fitted.values)
k <- length(Model1$coefficients)-1
dl <- n-k-1

# Valeurs critiques de T
t95 <- qt(p=1 - (0.05/2), df=dl)
t99 <- qt(p=1 - (0.01/2), df=dl)
t99.9 <- qt(p=1 - (0.001/2), df=dl)
cat("Valeurs critiques de T en fonction du niveau de confiance",
    "\n et du nombre de degrés de liberté",
    "\n95 % : ", t95,
    "\n99 % : ", t99,
    "\n99,9 % : ", t99.9
)

## Valeurs critiques de T en fonction du niveau de confiance
## et du nombre de degrés de liberté
## 95 % : 1.960197
## 99 % : 2.576311
## 99,9 % : 3.291481
```

```
# Intervalle de confiance à 95

data.frame(
  IC2.5 = round(coefs-t95*coefs_se,3),
  IC97.5 = round(coefs+t95*coefs_se,3)
)

##           IC2.5 IC97.5
## (Intercept) 24.626 28.085
## HABHA      -0.075 -0.066
## AgeMedian   -0.002  0.023
## Pct_014     1.021  1.148
## Pct_65P     0.364  0.437
## Pct_MV     -0.052 -0.011
## Pct_FR     -0.371 -0.325

# Intervalle de confiance à 99

data.frame(
  IC0.5 = round(coefs-t99*coefs_se,3),
  IC99.5 = round(coefs+t99*coefs_se,3)
)

##           IC0.5 IC99.5
## (Intercept) 24.083 28.629
## HABHA      -0.076 -0.065
## AgeMedian   -0.006  0.027
## Pct_014     1.002  1.167
## Pct_65P     0.352  0.449
## Pct_MV     -0.058 -0.004
## Pct_FR     -0.378 -0.318

# Intervalle de confiance à 99.9

data.frame(
  IC0.05 = round(coefs-t99.9*coefs_se,3),
  IC99.95 = round(coefs+t99.9*coefs_se,3)
)

##           IC0.05 IC99.95
## (Intercept) 23.452 29.260
## HABHA      -0.078 -0.063
## AgeMedian   -0.010  0.032
## Pct_014     0.979  1.190
## Pct_65P     0.339  0.463
## Pct_MV     -0.065  0.003
## Pct_FR     -0.387 -0.310
```

7.5 Introduction de variables explicatives particulières

7.5.1 Exploration des relations non linéaires

7.5.1.1 Variable indépendante avec une fonction polynomiale

Dans la section 4.1, nous avons vu que la relation entre deux variables continues n'est pas toujours linéaire ; elle peut être aussi curvilinearéaire. Pour explorer les relations curvilinearéaires, nous introduisons la variable indépendante sous la forme polynomiale d'ordre 2 (voir le prochain encadré). L'équation de régression s'écrit alors :

$$Y = b_0 + b_1 X_1 + b_{11} X_1^2 + b_2 X_2 + \dots + b_k X_k + e \quad (7.21)$$

Dans l'équation (7.21), la première variable indépendante est introduite dans le modèle de régression à la fois dans sa forme originelle et mise au carré : $b_1 X_1 + b_{11} X_1^2$. Un coefficient différent est ajusté pour chacune de ces deux versions de la variable X_1 .

La démographie est probablement la discipline des sciences sociales qui a le plus recours aux régressions polynomiales. En effet, la variable âge est souvent introduite comme variable explicative dans sa forme originelle et mise au carré. L'objectif est de vérifier si l'âge partage ou non une relation curvilinearéaire avec un phénomène donné : par exemple, il pourrait y être associé positivement jusqu'à un certain seuil (45 ans par exemple), puis négativement à partir de ce seuil.



Régression polynomiale et nombre d'ordres.

Sachez qu'il est aussi possible de construire des régressions polynomiales avec plus de deux ordres. Par exemple, une régression polynomiale d'ordre 3 comprend une variable dans sa forme originelle, puis mise au carré et au cube. Cela a l'inconvénient d'augmenter corolairement le nombre de coefficients. Nous verrons au chapitre 11 qu'il existe une solution plus élégante et efficace : le recours aux modèles de régressions linéaires généralisés additifs avec des *splines*. Dans le cadre de cette section, nous nous limitons à des régressions polynomiales d'ordre 2.

$$\text{Ordre 2 : } Y = b_0 + b_1 X_1 + b_{11} X_1^2 + b_2 X_2 + \dots + b_k X_k + e \quad (7.22)$$

$$\text{Ordre 3 : } Y = b_0 + b_1 X_1 + b_{11} X_1^2 + b_{111} X_1^3 + b_2 X_2 + \dots + b_k X_k + e \quad (7.23)$$

$$\text{Ordre 4 : } Y = b_0 + b_1 X_1 + b_{11} X_1^2 + b_{111} X_1^3 + b_{1111} X_1^4 + b_2 X_2 + \dots + b_k X_k + e \quad (7.24)$$

Pour construire une régression polynomiale dans R, il est possible d'utiliser deux fonctions de R :

- `I(VI^2)` avec `VI` qui est la variable indépendante sur laquelle est appliquée la mise au carré.
- `poly(VI, 2)` qui utilise une forme polynomiale orthogonale pour éviter les problèmes de corrélation entre les deux termes, c'est-à-dire entre `VI` et `VI^2`.

Ces deux méthodes produisent les mêmes résultats pour les autres variables dépendantes et pour la qualité d'ajustement du modèle (R^2 , F, etc.). Nous privilégions la seconde fonction pour éviter de détecter à tort des problèmes de multicolinéarité excessive.

Appliquons cette démarche à la variable `AgeMedian` (âge médian des bâtiments) afin de vérifier si elle partage ou non une relation curvilinearéaire avec la couverture végétale de l'îlot. À la lecture des résultats pour les deux modèles, les constats suivants peuvent être avancés :

- Le R^2 ajusté passe de 0,4179 à 0,4378 du modèle 1 au modèle 2, ce qui signale un gain de variance expliquée.
- Le F incrémentiel entre les deux modèles s'élève à 362,64 et est significatif ($p < 0,001$). Nous pouvons donc en conclure que le second modèle est plus performant que le premier, ce qui signale que la forme curvilinéaire pour AgeMedian (modèle 2) est plus efficace que la forme linéaire (modèle 1).
- Dans le premier modèle, le coefficient de régression pour AgeMedian n'est pas significatif. L'âge médian des bâtiments n'est donc pas associé linéairement avec la variable dépendante.
- Dans le second modèle, la valeur du coefficient de `poly(AgeMedian, 2)1` est positive et celle de `poly(AgeMedian, 2)2` est négative et significative. Cela indique qu'il existe une relation linéaire en forme de U inversé. Si le premier coefficient avait été négatif et le second positif, nous aurions alors conclu que la forme curvilinéaire prend la forme d'un U.

```
# régression linéaire
modele1 <- lm(VegPct ~ HABHA+AgeMedian+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)

# régression polynomiale
modele2 <- lm(VegPct ~ HABHA+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)

# affichage des résultats du modèle 1
summary(modele1)

## 
## Call:
## lm(formula = VegPct ~ HABHA + AgeMedian + Pct_014 + Pct_65P +
##     Pct_MV + Pct_FR, data = DataFinal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -48.876  -9.757  -0.232   9.499 103.830
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.355774  0.882235 29.874 <2e-16 ***
## HABHA       -0.070401  0.002202 -31.975 <2e-16 ***
## AgeMedian    0.010790  0.006369  1.694  0.0902 .
## Pct_014      1.084478  0.032179 33.702 <2e-16 ***
## Pct_65P      0.400531  0.018835 21.265 <2e-16 ***
## Pct_MV      -0.031112  0.010406 -2.990  0.0028 **
## Pct_FR      -0.348256  0.011640 -29.918 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.16 on 10203 degrees of freedom
## Multiple R-squared:  0.4182, Adjusted R-squared:  0.4179
## F-statistic: 1223 on 6 and 10203 DF,  p-value: < 2.2e-16

# affichage des résultats du modèle 1
summary(modele2)

##
```

```

## Call:
## lm(formula = VegPct ~ HABHA + poly(AgeMedian, 2) + Pct_014 +
##     Pct_65P + Pct_MV + Pct_FR, data = DataFinal)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -49.659 -9.361 -0.159  9.034 105.160 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            2.968e+01  7.535e-01 39.383 < 2e-16 ***
## HABHA                 -7.107e-02  2.164e-03 -32.839 < 2e-16 ***
## poly(AgeMedian, 2)1   1.134e+01  1.598e+01   0.710  0.47788  
## poly(AgeMedian, 2)2   -2.721e+02  1.429e+01  -19.043 < 2e-16 *** 
## Pct_014                9.969e-01  3.196e-02  31.198 < 2e-16 *** 
## Pct_65P                3.219e-01  1.896e-02  16.972 < 2e-16 *** 
## Pct_MV                -2.888e-02  1.023e-02  -2.823  0.00476 ** 
## Pct_FR                -3.562e-01  1.145e-02 -31.116 < 2e-16 *** 
## ---                
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 13.92 on 10202 degrees of freedom
## Multiple R-squared:  0.4382, Adjusted R-squared:  0.4378 
## F-statistic: 1137 on 7 and 10202 DF,  p-value: < 2.2e-16

```

```

# test de Fisher pour comparer les modèles
anova(modele1, modele2)

```

```

## Analysis of Variance Table
##
## Model 1: VegPct ~ HABHA + AgeMedian + Pct_014 + Pct_65P + Pct_MV + Pct_FR
## Model 2: VegPct ~ HABHA + poly(AgeMedian, 2) + Pct_014 + Pct_65P + Pct_MV +
##     Pct_FR
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)    
## 1 10203 2046427
## 2 10202 1976182  1     70245 362.64 < 2.2e-16 ***
## ---                
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Construction d'un graphique des effets marginaux

Pour visualiser la relation linéaire et curvilinéaire, nous vous proposons de réaliser un graphique des effets marginaux à partir de la syntaxe ci-dessous.

Les graphiques des effets marginaux permettent de visualiser l'impact d'une variable indépendante sur la variable dépendante d'une régression. Nous nous basons pour cela sur les prédictions effectuées par le modèle. Admettons que nous nous intéressons à l'effet de la variable X_1 sur la variable Y . Il est possible de créer de nouvelles données fictives pour lesquelles l'ensemble des autres variables X sont fixées à leur moyenne respective, et seule X_1 est autorisée à varier. En utilisant l'équation de régression du modèle sur ces données fictives, nous pouvons observer l'évolution de la valeur prédite de Y quand X_1 augmente ou diminue, et ce, toutes choses étant égales par ailleurs (puisque toutes les autres variables ont une valeur

fixe). Cette approche est particulièrement intéressante pour décrire des effets non linéaires obtenus avec des polynomiales, mais aussi des interactions comme nous le verrons plus tard. Elle est également utilisée dans les modèles linéaires généralisés (GLM) et additifs (GAM) (chapitres 8 et 11). Notez qu'il est aussi important de représenter, sur ce type de graphique, l'incertitude de la prédiction. Pour cela, il est possible de construire des intervalles de confiance à 95 % autour de la prédiction en utilisant l'erreur standard de la prédiction (renvoyée par la fonction `predict`).

```

library(ggplot2)
# Statistique sur la variable AgeMedian qui varie de 0 à 226 ans
summary(DataFinal$AgeMedian)

##      Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##      0.00   37.25  49.00  52.11  61.00  226.00

# Création d'un DataFrame temporaire
# remarquez que les autres variables indépendantes sont constantes :
# nous leur avons attribué leur moyenne correspondante
df <- data.frame(
  HABHA = mean(DataFinal$HABHA),
  AgeMedian= seq(0,200, by = 2),
  AgeMedian2 = seq(0,200, by = 2)**2,
  Pct_014= mean(DataFinal$Pct_014),
  Pct_65P= mean(DataFinal$Pct_65P),
  Pct_MV= mean(DataFinal$Pct_MV),
  Pct_FR= mean(DataFinal$Pct_FR)
)

# calcul de la valeur de t pour un intervalle à 95 %
n <- length(modele1$fitted.values)
k <- length(modele1$coefficients)-1
t95 <- qt(p=1 - (0.05/2), df=n-k-1)

# Calcul des valeurs prédictes pour le 1er modèle
# avec l'intervalle de confiance à 95 %
predsM1 <- predict(modele1, se = T, newdata = df)
df$predM1 <- predsM1$fit
df$lowerM1 <- predsM1$fit - t95*predsM1$se.fit
df$upperM1 <- predsM1$fit + t95*predsM1$se.fit

# Calcul des valeurs prédictes pour le 2e modèle
# avec l'intervalle de confiance à 95 %
predsM2 <- predict(modele2, se = T, newdata = df)
df$predM2 <- predsM2$fit
df$lowerM2 <- predsM2$fit - t95*predsM2$se.fit
df$upperM2 <- predsM2$fit + t95*predsM2$se.fit

# Graphique
ggplot(data = df) +
  geom_ribbon(aes(x = AgeMedian, ymin = lowerM1, ymax = upperM1),
              fill = rgb(0.1,0.1,0.1,0.4)) +
  geom_path(aes(x = AgeMedian, y = predM1), color = 'blue', size = 1) +
  geom_ribbon(aes(x = AgeMedian, ymin = lowerM2, ymax = upperM2),
              fill = rgb(0.1,0.1,0.1,0.4)) +

```

```
geom_path(aes(x = AgeMedian, y = predM2), color = 'red', size = 1)+

labs(title="Effet marginal de l'âge médian des bâtiments sur la",
     subtitle = "couverture végétale des îlots de l'île de Montréal",
     caption = "bleu : relation linéaire; rouge : curvilinéaire",
     x = "Âge médian des bâtiments",
     y = "Couverture végétale (%)")
```

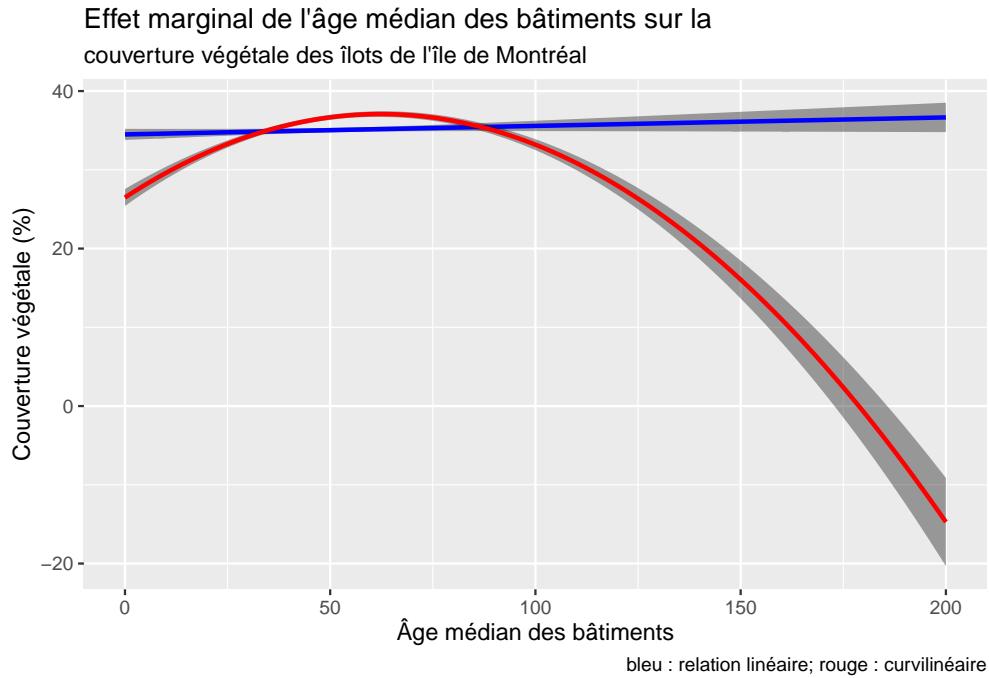


FIG. 7.2 : Relations linéaire et curvilinéaire

La figure 7.2 démontre bien que la relation linéaire n'est pas significative : la pente est extrêmement faible, ce qui signale que l'effet de l'âge médian est presque nul ($B = 0,0108$, $p = 0,0902$). En revanche, la relation curvilinéaire est plus intéressante : la couverture végétale croît quand l'âge médian des bâtiments dans l'îlot augmente de 0 à 60 ans environ, puis elle décroît.

7.5.1.2 Variable indépendante sous forme logarithmique

Une autre manière d'explorer une relation non linéaire est d'intégrer la variable sous forme logarithmique (Hanck et al. 2019, 212-218). L'interprétation du coefficient de régression est alors plus complexe : un 1 % d'augmentation de la variable X_k entraîne un changement de $0,01 \times \beta_k$ de la variable dépendante. Autrement dit, il n'est plus exprimé dans les unités de mesure originales des deux variables.

Au tableau 7.4, le coefficient de -6,855 pour la variable `logHABHA` s'interprète alors comme suit : un changement de 1 % de la variable densité de population entraîne une diminution de $0,01 \times -6,855 = -0,07$ de la couverture végétale dans l'île, toutes choses étant égales par ailleurs.

Puisque l'interprétation du coefficient de régression de $\log(\beta_k)$ est plus complexe, il convient de s'assurer que son apport au modèle est justifié, et ce, de deux façons :

- **Comparez les mesures d'ajustement des deux modèles (surtout les R^2 ajustés).** Si le R^2 ajusté du modèle avec $\log(\beta_k)$ est plus élevé que celui avec β_k , alors la transformation logarithmique fait

TAB. 7.4 : Modèle avec une variable indépendante sous forme logarithmique

Variable	Coef.	Erreur type	Valeur de T	P	coef. 2,5 %	coef. 97,5 %
Constante	52,831	1,001	52,780	0,000	50,868	54,793 ***
logHABHA	-6,855	0,168	-40,730	0,000	-7,185	-6,525 ***
AgeMedian ordre 1	11,985	15,586	0,770	0,442	-18,568	42,537
AgeMedian ordre 2	-286,144	13,942	-20,520	0,000	-313,473	-258,816 ***
Pct_014	0,941	0,031	30,090	0,000	0,879	1,002 ***
Pct_65P	0,306	0,019	16,550	0,000	0,270	0,343 ***
Pct_MV	-0,036	0,010	-3,650	0,000	-0,056	-0,017 ***
Pct_FR	-0,344	0,011	-31,210	0,000	-0,366	-0,323 ***

de votre variable indépendante un meilleur prédicteur, toutes choses étant égales par ailleurs.

- **Construisez les graphiques des effets marginaux** de votre variable afin de vérifier si la relation qu'elle partage avec votre VD est plutôt logarithmique que linéaire (figure 7.3). Notez que cette approche graphique peut aussi ne donner aucune indication lorsque vos données sont très dispersées ou que la relation est faible entre votre variable dépendante et indépendante.

```

library(ggpubr)
library(ggplot2)
library(ggeffects)

# Modèles
modele1a <- lm(VegPct ~ HABHA+poly(AgeMedian,2) +
  Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)

modele1b <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2) +
  Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)

# Valeurs prédites
fit1a <- ggpredict(modele1a, terms = "HABHA")
fit1b <- ggpredict(modele1b, terms = "HABHA")

# Graphiques
G1a <- ggplot(fit1a, aes(x, predicted)) +
  geom_point(data = DataFinal, mapping = aes(x=HABHA, y = VegPct),
  size = 0.2, color = rgb(0.1,0.1,0.1,0.4)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3, fill ="red")+
  geom_line(color = "red") +
  labs(title="Variable non transformée",
  y="VD: valeur prédite",
  x = "Habitants km2") +
  ylim(0,100) + xlim(0,600)

G1b <- ggplot(fit1b, aes(x, predicted)) +
  geom_point(data = DataFinal, mapping = aes(x=HABHA, y = VegPct),
  size = 0.2, color = rgb(0.1,0.1,0.1,0.4)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3, fill ="red")+
  geom_line(color = "red") +
  labs(title="Variable transformée (log)",
  y="VD: valeur prédite",
  x = "Habitants km2")

```

```
G1aG1b <- ggarrange(G1a, G1b, nrow = 1)
G1aG1b
```

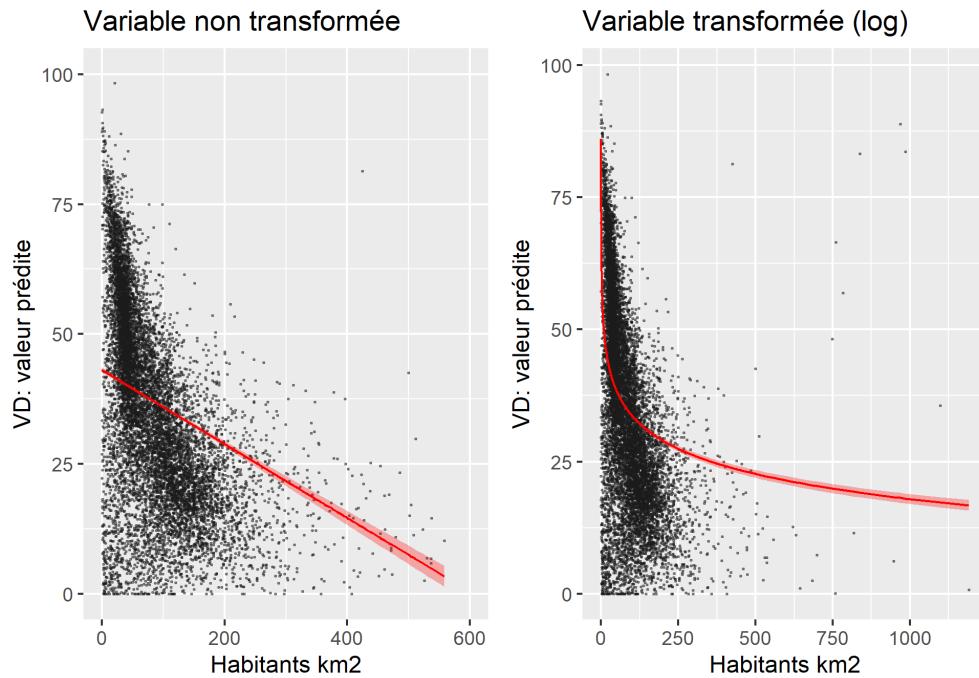


FIG. 7.3 : Effet marginal de la densité de population

7.5.2 Variable indépendante qualitative dichotomique

Il est très fréquent d'introduire une variable qualitative dichotomique comme variable explicative ou de contrôle dans un modèle. À titre de rappel, une variable dichotomique comprend deux modalités (section 2.1.2).

Dans le modèle ci-dessous, nous voulons vérifier si un îlot situé sur le territoire de la ville de Montréal a proportionnellement moins de végétation qu'un îlot situé dans une autre municipalité de l'île de Montréal, toutes choses étant égales par ailleurs. Pour ce faire, nous créons une variable binaire dénommée `VilleMtl` qui prend la valeur de 1 pour les îlots de la ville de Montréal et 0 pour ceux d'une autre municipalité.

Nous obtenons ainsi un coefficient de régression pour `VilleMtl` de -7,699 (tableau 7.5). Cela signifie que si toutes les autres variables indépendantes du modèle étaient constantes, alors un îlot de la ville de Montréal aurait en moyenne une valeur de -7,7 % de moins de végétation comparativement à un îlot situé dans une autre municipalité.

```
# Création d'une variable muette pour Montréal (0 ou 1)
DataFinal$VilleMtl <- ifelse(DataFinal$SDRNOM == "Montréal", 1, 0)

# Modèle avec la variable dichotomique
modele3 <- lm(VegPct ~ VilleMtl+log(HABHA)+poly(AgeMedian,2) +
  Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)
```

TAB. 7.5 : Modèle avec une variable dichotomique

Variable	Coef.	Erreur type	Valeur de T	P	coef. 2,5 %	coef. 97,5 %
Constante	57,676	1,009	57,140	0,000	55,697	59,654 ***
VilleMtl	-7,699	0,377	-20,430	0,000	-8,438	-6,960 ***
log(HABHA)	-6,174	0,168	-36,680	0,000	-6,504	-5,844 ***
AgeMedian ordre 1	-14,871	15,334	-0,970	0,332	-44,929	15,186
AgeMedian ordre 2	-280,251	13,668	-20,500	0,000	-307,044	-253,459 ***
Pct_014	0,794	0,031	25,230	0,000	0,732	0,856 ***
Pct_65P	0,270	0,018	14,810	0,000	0,234	0,306 ***
Pct_MV	-0,028	0,010	-2,890	0,004	-0,047	-0,009 **
Pct_FR	-0,294	0,011	-26,550	0,000	-0,316	-0,273 ***



Bien interpréter un coefficient d'une variable dichotomique

Nous avons vu que le coefficient de régression (β_k) indique le changement de la variable dépendante (Y), lorsque la variable indépendante augmente d'une unité, toutes choses étant égales par ailleurs.

Pour une variable dichotomique, le coefficient indique le changement de Y quand les observations appartiennent à la modalité qui a la valeur de 1 (ici la ville de Montréal), comparativement à celle qui a la valeur de 0 (autres municipalités de l'île de Montréal), toutes choses étant égales par ailleurs.

La modalité qui a la valeur de 0 est alors appelée **modalité ou catégorie de référence**.

Autrement dit, si la variable avait été codée : 0 pour la ville de Montréal et 1 pour les autres municipalités, alors le coefficient aurait été de 7,699.

Pour éviter d'oublier quelle est la modalité de référence (valeur de 0), nous verrons plus tard (dans la section mise en œuvre des modèles de régression dans R (section 7.7) qu'il peut être préférable de définir un facteur avec la fonction `as.factor` et d'indiquer la catégorie de référence avec la fonction `relevel(x, ref)`.

Comme pour une variable indépendante introduite avec une fonction polynomiale, il peut être très intéressant d'illustrer l'effet marginal de la variable dichotomique avec un graphique qui montre l'écart entre les moyennes des deux modalités, une fois contrôlées les autres variables indépendantes (figure 7.4). Notez que dans ce graphique, les barres d'erreurs situées au sommet des rectangles représentent les intervalles à 95 % des prédictions du modèle.

7.5.3 Variable indépendante qualitative polytomique

Il est possible d'introduire une variable qualitative polytomique comme variable explicative ou de contrôle dans un modèle. À titre de rappel, une variable polytomique comprend plus de deux modalités, qu'elle soit nominale ou ordinale (section 2.1.2).

En guise d'exemple, une variable qualitative pourrait être : différents groupes de population (groupes d'âge, minorités visibles, catégories socioprofessionnelles, etc.), différents territoires ou régions (ville centrale, première couronne, deuxième couronne, etc.), une variable continue transformée en quatre ou cinq catégories ordinaires selon les quartiles ou les quintiles.

7.5.3.1 Comment construire un modèle de régression avec une variable explicative qualitative polytomique ?

Prenons l'exemple d'un modèle de régression comprenant deux variables indépendantes : l'une continue (x_1), l'autre qualitative (x_2) avec quatre modalités (A, B, C et D). L'introduction de la variable qualitative dans le modèle revient à :

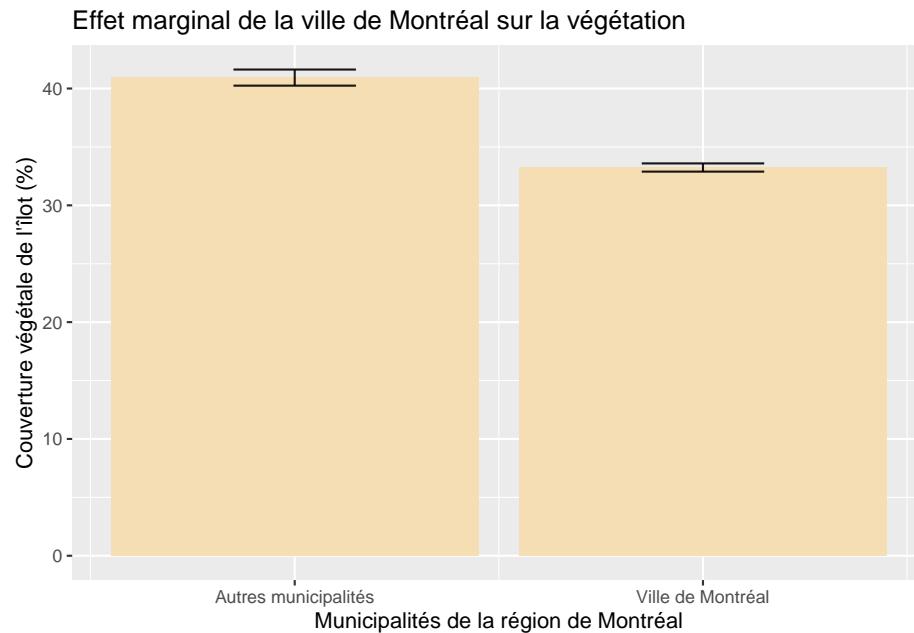


FIG. 7.4 : Effet marginal d'une variable dichotomique

- Transformer chaque modalité en variable muette (binaire). Nous avons ainsi quatre nouvelles variables binaires : x_{2A} , x_{2B} , x_{2C} et x_{2D} . Par exemple, pour x_{2A} , les observations de la modalité A se verront affecter la valeur de 1 versus 0 pour les autres observations. La même démarche s'applique à x_{2B} , x_{2C} et x_{2D} (voir tableau 7.6).
- Toutes les modalités transformées en variables muettes sont introduites dans le modèle comme variables indépendantes **sauf celle servant de catégorie de référence**. Pourquoi sauf une ? Si nous mettions toutes les modalités en variable muette, alors chaque observation serait repérée par une valeur de 1, « il y aurait alors une parfaite multicollinearité et aucune solution unique pour les coefficients de régression ne pourrait être trouvée » (Bressoux 2010, 128).
- Par exemple, si nous choisissons la modalité A comme catégorie de référence, l'équation de régression s'écrit alors :

$$Y = b_0 + b_1 X_1 + b_{2B} X_{2B} + b_{2C} X_{2C} + b_{2D} X_{2D} + e \quad (7.25)$$

- Vous aurez compris que choisir la modalité D comme catégorie de référence revient à écrire l'équa-

TAB. 7.6 : Transformation d'une variable qualitative en variables muettes pour chaque modalité

obs	Y	X1	X2	X2A	X2B	X2C	X2D
1	33,31	21,67	A	1	0	0	0
2	44,35	19,44	A	1	0	0	0
3	41,22	19,05	A	1	0	0	0
4	31,36	18,06	B	0	1	0	0
5	47,36	19,62	B	0	1	0	0
6	46,31	9,66	B	0	1	0	0
7	44,75	26,55	C	0	0	1	0
8	57,96	29,72	C	0	0	1	0
9	39,21	26,40	D	0	0	0	1
10	31,53	27,55	D	0	0	0	1

tion suivante :

$$Y = b_0 + b_1 X_1 + b_{2A} X_{2A} + b_{2B} X_{2B} + b_{2C} X_{2C} + e \quad (7.26)$$

7.5.3.2 Comment interpréter les coefficients des modalités d'une variable explicative qualitative polytomique

Les coefficients des différentes modalités s'interprètent en fonction de la catégorie de référence. Dans l'exemple ci-dessous, nous avons inclus la ville de Montréal comme catégorie de référence (tableau 7.7). Toutes choses étant égales par ailleurs, nous pouvons alors constater que :

- en moyenne, les îlots résidentiels de Senneville et de Baie-D'Urfé ont respectivement 23,235 % et 21,400 % plus de végétation que ceux de la ville de Montréal.
- la seule municipalité comprenant en moyenne moins de végétation dans ses îlots résidentiels est Montréal-Est (-13,334 %)
- nous remarquons aussi que les îlots des municipalités de Sainte-Anne-de-Bellevue, de Montréal-Ouest et de Côte-Saint-Luc ne présentent pas significativement moins ou plus de végétation que ceux de la ville de Montréal (leurs valeurs de p sont supérieures à 0,05).

Par conséquent, les valeurs de t et de p pour une modalité permettent de vérifier si elle est ou non significativement différente de la catégorie de référence.

Utilisons maintenant comme référence la municipalité qui avait le coefficient le plus fort dans le modèle précédent, soit Senneville (tableau 7.8). Bien entendu, les coefficients des variables continues et de la constante ne changent pas. Par contre, les coefficients de toutes les municipalités sont négatifs puisque la municipalité de Senneville est celle qui a proportionnellement le plus de végétation dans ses îlots, toutes choses étant égales par ailleurs.

TAB. 7.7 : Modèle avec une variable polytomique (ville de Montréal en catégorie de référence)

Variable	Coef.	Erreur type	Valeur de T	P	
Constante	48,193	0,992	48,580	0,000	***
log(HABHA)	-5,836	0,168	-34,840	0,000	***
AgeMedian ordre 1	-11,807	15,648	-0,750	0,451	
AgeMedian ordre 2	-266,469	13,613	-19,570	0,000	***
Pct_014	0,794	0,032	25,190	0,000	***
Pct_65P	0,277	0,018	15,130	0,000	***
Pct_MV	-0,036	0,010	-3,740	0,000	***
Pct_FR	-0,279	0,011	-25,340	0,000	***
<i>Municipalité</i>					
ref : Montréal	-	-	-	-	-
Baie-D'Urfé	21,400	1,635	13,090	0,000	***
Beaconsfield	14,112	0,893	15,810	0,000	***
Côte-Saint-Luc	0,172	1,035	0,170	0,868	
Dollard-Des Ormeaux	7,960	0,748	10,640	0,000	***
Dorval	11,157	0,971	11,490	0,000	***
Hampstead	3,080	1,599	1,930	0,054	.
Kirkland	6,937	1,014	6,840	0,000	***
Mont-Royal	12,699	0,894	14,210	0,000	***
Montréal-Est	-13,334	1,920	-6,940	0,000	***
Montréal-Ouest	3,306	1,819	1,820	0,069	.
Pointe-Claire	9,896	0,866	11,430	0,000	***
Sainte-Anne-de-Bellevue	0,342	1,904	0,180	0,858	
Senneville	23,235	3,793	6,130	0,000	***
Westmount	2,255	1,088	2,070	0,038	*

TAB. 7.8 : Modèle avec une variable polytomique (Senneville en catégorie de référence)

Variable	Coef.	Erreur type	Valeur de T	P	
Constante	71,429	3,846	18,570	0,000	***
log(HABHA)	-5,836	0,168	-34,840	0,000	***
AgeMedian ordre 1	-11,807	15,648	-0,750	0,451	
AgeMedian ordre 2	-266,469	13,613	-19,570	0,000	***
Pct_014	0,794	0,032	25,190	0,000	***
Pct_65P	0,277	0,018	15,130	0,000	***
Pct_MV	-0,036	0,010	-3,740	0,000	***
Pct_FR	-0,279	0,011	-25,340	0,000	***
<i>Municipalité</i>					
ref : Senneville	-	-	-	-	-
Baie-D'Urfé	-1,835	4,093	-0,450	0,654	
Beaconsfield	-9,123	3,866	-2,360	0,018	*
Côte-Saint-Luc	-23,064	3,918	-5,890	0,000	***
Dollard-Des Ormeaux	-15,275	3,852	-3,970	0,000	***
Dorval	-12,078	3,891	-3,100	0,002	**
Hampstead	-20,156	4,094	-4,920	0,000	***
Kirkland	-16,298	3,911	-4,170	0,000	***
Mont-Royal	-10,537	3,875	-2,720	0,007	**
Montréal	-23,235	3,793	-6,130	0,000	***
Montréal-Est	-36,570	4,231	-8,640	0,000	***
Montréal-Ouest	-19,930	4,187	-4,760	0,000	***
Pointe-Claire	-13,339	3,865	-3,450	0,001	***
Sainte-Anne-de-Bellevue	-22,893	4,225	-5,420	0,000	***
Westmount	-20,980	3,927	-5,340	0,000	***

À l'inverse, si nous utilisons Montréal-Est comme modalité de référence, soit la municipalité avec le coefficient le plus faible dans le premier modèle, tous les coefficients deviendront positifs (tableau 7.9).



Comment choisir la catégorie de référence ?

Plusieurs options sont possibles. Vous pouvez retenir :

- la modalité comprenant le plus d'observations;
- la modalité avec la plus forte valeur pour la variable dépendante;
- la modalité avec la plus faible valeur pour la variable dépendante;
- la modalité qui fait le plus de sens avec votre cadre théorique. Prenons l'exemple d'une variable qualitative comprenant plusieurs groupes d'âge (15-29 ans, 30-39 ans, 40-49 ans, 50-54 ans, 65 ans et plus). Si votre étude porte sur les jeunes et que vous souhaitez comparer leur situation comparativement aux autres groupes d'âge, toutes choses étant égales par ailleurs, sélectionnez bien évidemment la modalité des 15 à 29 ans comme catégorie de référence.

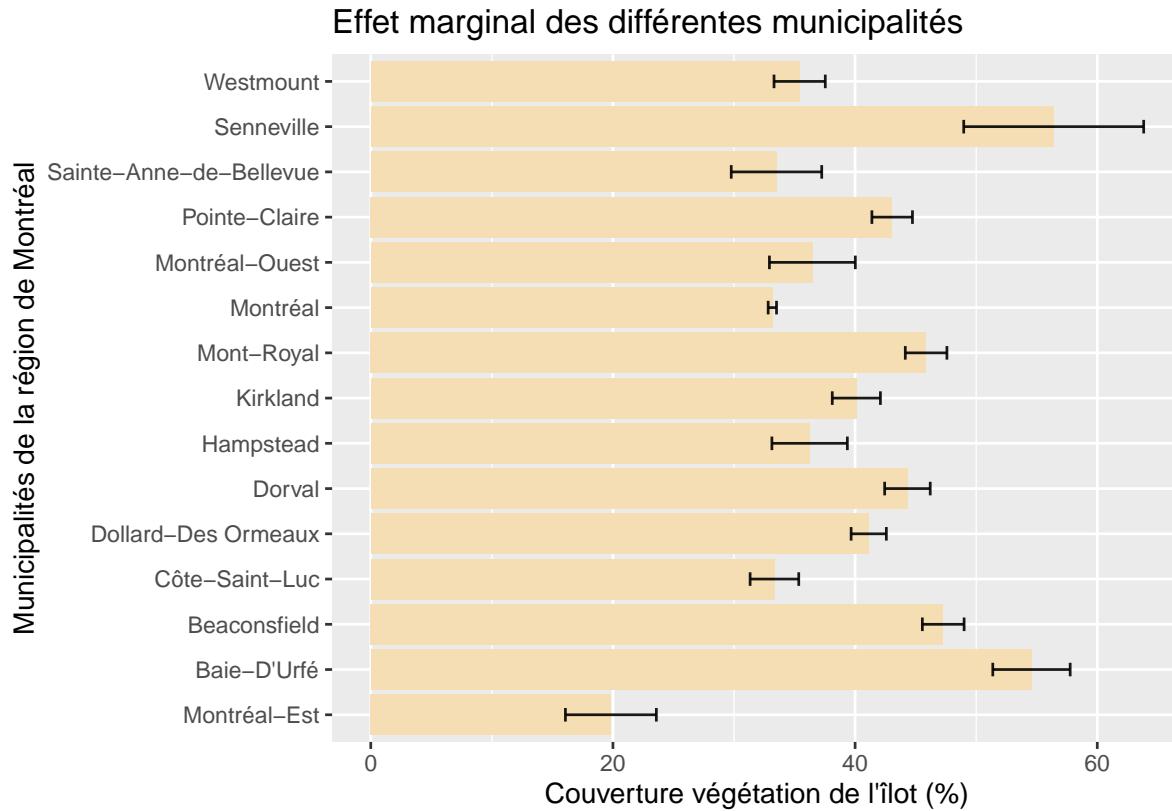
Mais surtout, évitez de choisir une catégorie comprenant très peu d'observations.

7.5.3.3 Effet marginal d'une variable explicative qualitative polytomique

Comme pour une variable dichotomique, il est possible d'illustrer l'effet marginal de la variable qualitative dichotomique avec un graphique. Quelle que soit la catégorie de référence choisie, le graphique est le même. La figure 7.5 illustre ainsi la valeur moyenne, avec son intervalle de confiance à 95 %, de la végétation dans les îlots résidentiels de chacune des municipalités de la région de Montréal, *ceteris paribus*.

Tab. 7.9 : Modèle avec une variable polytomique (Montréal-Est en catégorie de référence)

Variable	Coef.	Erreurs type	Valeur de T	P	
Constante	34,859	2,109	16,530	0,000	***
log(HABHA)	-5,836	0,168	-34,840	0,000	***
AgeMedian ordre 1	-11,807	15,648	-0,750	0,451	
AgeMedian ordre 2	-266,469	13,613	-19,570	0,000	***
Pct_014	0,794	0,032	25,190	0,000	***
Pct_65P	0,277	0,018	15,130	0,000	***
Pct_MV	-0,036	0,010	-3,740	0,000	***
Pct_FR	-0,279	0,011	-25,340	0,000	***
<i>Municipalité</i>					
ref : Montréal-Est	-	-	-	-	-
Baie-D'Urfé	34,735	2,495	13,920	0,000	***
Beaconsfield	27,446	2,091	13,130	0,000	***
Côte-Saint-Luc	13,506	2,167	6,230	0,000	***
Dollard-Des Ormeaux	21,294	2,053	10,370	0,000	***
Dorval	24,491	2,134	11,480	0,000	***
Hampstead	16,414	2,478	6,620	0,000	***
Kirkland	20,272	2,159	9,390	0,000	***
Mont-Royal	26,033	2,101	12,390	0,000	***
Montréal	13,334	1,920	6,940	0,000	***
Montréal-Ouest	16,640	2,628	6,330	0,000	***
Pointe-Claire	23,230	2,087	11,130	0,000	***
Sainte-Anne-de-Bellevue	13,676	2,687	5,090	0,000	***
Senneville	36,570	4,231	8,640	0,000	***
Westmount	15,590	2,196	7,100	0,000	***

**Fig. 7.5 :** Effet marginal d'une variable polytomique

7.5.4 Variables d'interaction

7.5.4.1 Variable d'interaction entre deux variables continues

Une interaction entre deux variables indépendantes continues consiste à simplement les multiplier ($X_1 \times X_2$). Le modèle s'écrit alors :

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 (X_1 \times X_2) + \dots + \beta_k X_k + e \quad (7.27)$$

Un nouveau coefficient (β_3) s'ajoute pour l'interaction (la multiplication) entre les deux variables continues. **Pourquoi ajouter une interaction entre deux variables?** L'objectif est d'évaluer l'effet d'une augmentation de β_1 en fonction d'un niveau donné de β_2 et inversement. Cela permet ainsi de répondre à la question suivante : l'effet de la variable β_1 est-il influencé par la variable β_2 et inversement ?

Prenons un exemple concret pour illustrer le tout. Premièrement, nous ajoutons `DistCBDkm` comme VI, soit la distance au centre-ville exprimée en kilomètres. Notez que pour ne pas surspécifier le modèle, les variables dichotomique `VilleMtl` ou polytomique `Municipalité` ont été préalablement ôtées. Le coefficient ($B = 0,659, p < 0,001$) signale que plus nous nous éloignons du centre-ville, plus la couverture végétale des îlots augmente significativement. En guise d'exemple, toutes choses étant égales par ailleurs, un îlot situé à dix kilomètres du centre-ville aura en moyenne 6,59 % plus de végétation (tableau 7.10).

Dans ce modèle (tableau 7.10), les pourcentages d'enfants de moins de 15 ans et de 65 ans et plus (`Pct_014` et `Pct_65P`) sont associés positivement à la variable dépendante tandis que le pourcentage de personnes à faible revenu (`Pct_FR`) est associé négativement.

Que se passe-t-il si nous introduisons une variable d'interaction entre `DistCBDkm` et `Pct_FR` (tableau 7.11)? L'effet du pourcentage de personnes à faible revenu (%) est significatif et négatif lorsqu'il est mis en interaction avec la distance au centre-ville. Cela indique que plus l'îlot est éloigné du centre-ville, plus `Pct_FR` a un effet négatif sur la couverture végétale ($B = -0,011, p < 0,001$).

À nouveau, il est possible de représenter l'effet de cette interaction à l'aide d'un graphique des effets marginaux. Notez cependant que nous devons représenter l'effet simultané de deux variables indépendantes sur notre variable dépendante, ce qu'il est possible de faire avec une carte de chaleur. La figure 7.6 représente donc l'effet moyen de l'interaction sur la prédiction dans le premier panneau, ainsi que l'intervalle de confiance à 95 % de la prédiction dans les deuxième et troisième panneaux.

Nous constatons ainsi que le modèle prédit des valeurs de végétation les plus faibles lorsque le pourcentage de personnes à faible revenu est élevé et que la distance au centre-ville est élevée (en haut à droite à la figure 7.6). En revanche, les valeurs les plus élevées de végétation sont atteintes lorsque la distance au centre-ville est élevée et que le pourcentage de personnes à faible revenu est faible (en bas à droite).

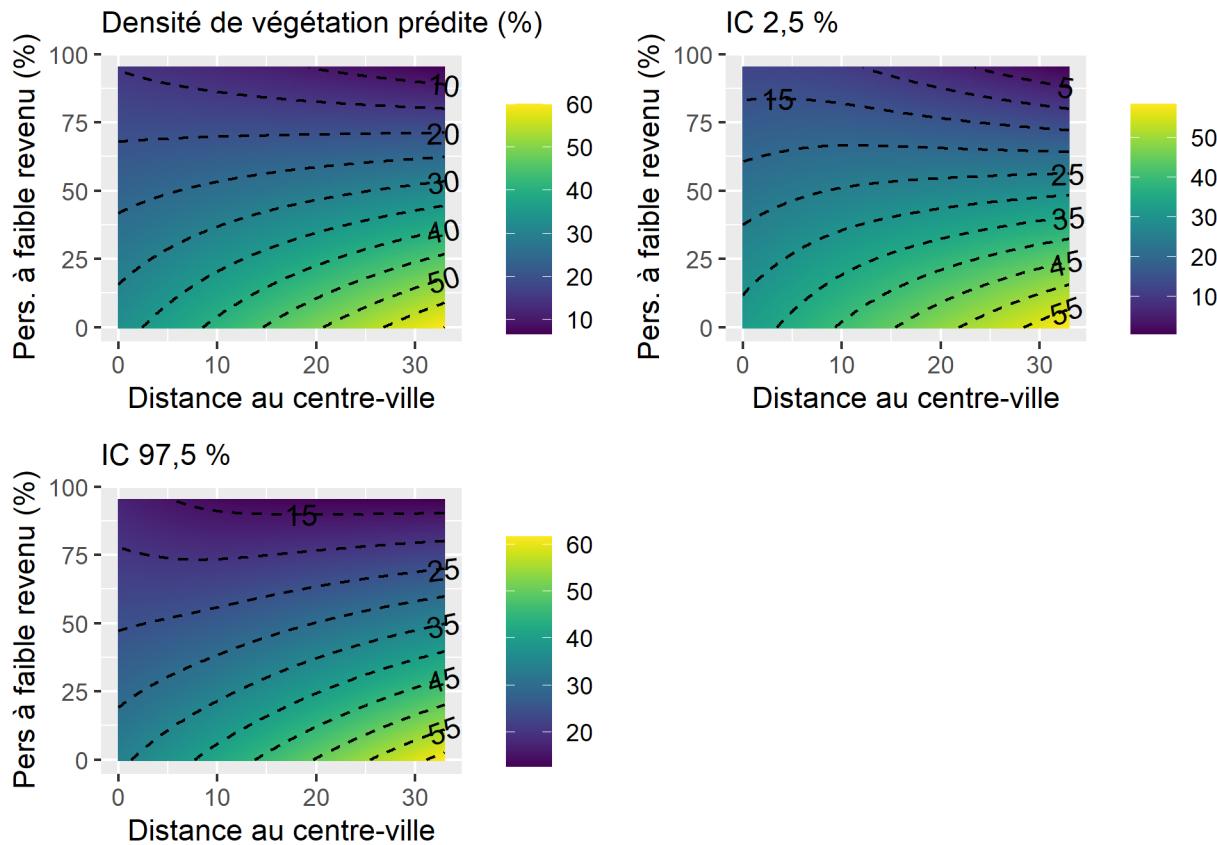
TAB. 7.10 : Modèle avec la distance au centre-ville (km)

Variable	Coef.	Erreur type	Valeur de T	P	
Constante	41,061	1,085	37,830	0,000	***
log(HABHA)	-5,555	0,172	-32,300	0,000	***
AgeMedian ordre 1	176,921	16,582	10,670	0,000	***
AgeMedian ordre 2	-298,735	13,560	-22,030	0,000	***
Pct_014	0,763	0,031	24,440	0,000	***
Pct_65P	0,321	0,018	17,860	0,000	***
Pct_MV	-0,018	0,010	-1,880	0,060	.
Pct_FR	-0,288	0,011	-26,260	0,000	***
DistCBDkm	0,659	0,027	24,460	0,000	***

TAB. 7.11 : Modèle avec une variable d'interaction entre deux VI continues

Variable	Coef.	Erreur type	Valeur de T	P	
Constante	38,382	1,137	33,760	0,000	***
log(HABHA)	-5,505	0,172	-32,080	0,000	***
AgeMedian ordre 1	160,523	16,672	9,630	0,000	***
AgeMedian ordre 2	-310,666	13,610	-22,830	0,000	***
Pct_014	0,786	0,031	25,130	0,000	***
Pct_65P	0,345	0,018	18,960	0,000	***
Pct_MV	-0,018	0,010	-1,820	0,069	.
Pct_FR	-0,191	0,017	-11,500	0,000	***
DistCBDkm	0,821	0,034	24,060	0,000	***
DistCBDkmX_Pct_FR	-0,011	0,001	-7,700	0,000	***

Il semble donc que l'éloignement au centre-ville soit associé avec une augmentation de la densité végétale, mais que cette augmentation puisse être mitigée par l'augmentation parallèle du pourcentage de personnes à faible revenu.

**FIG. 7.6 :** Effet marginal de l'interaction entre deux variables continues

Notez que dans la figure 7.6, la relation entre les deux variables indépendantes et la variable dépendante apparaît non linéaire du fait de l'interaction. À titre de comparaison, si nous utilisons les prédictions du modèle 5 (sans interaction), nous obtenons les prédictions présentées à la figure 7.7. Vous pouvez constater sur cette figure sans interaction que les deux effets des variables indépendantes sont linéaires puisque toutes les lignes sont parallèles.

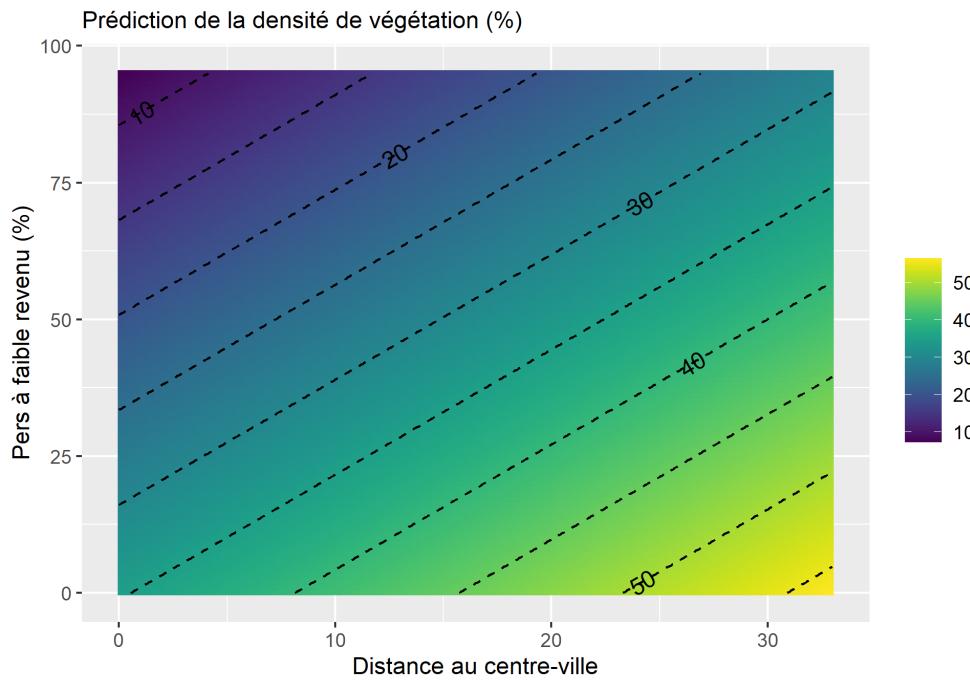


FIG. 7.7 : Effets marginaux de deux variables continues en cas d'absence d'interaction

7.5.4.2 Variable d'interaction entre une variable continue et une variable dichotomique

Une interaction entre une VI continue et une VI dichotomique consiste aussi à les multiplier ($X_1 \times D_2$); le modèle s'écrit alors :

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 D_2 + \beta_3 (X_1 \times D_2) + \dots + \beta_k X_k + e \quad (7.28)$$

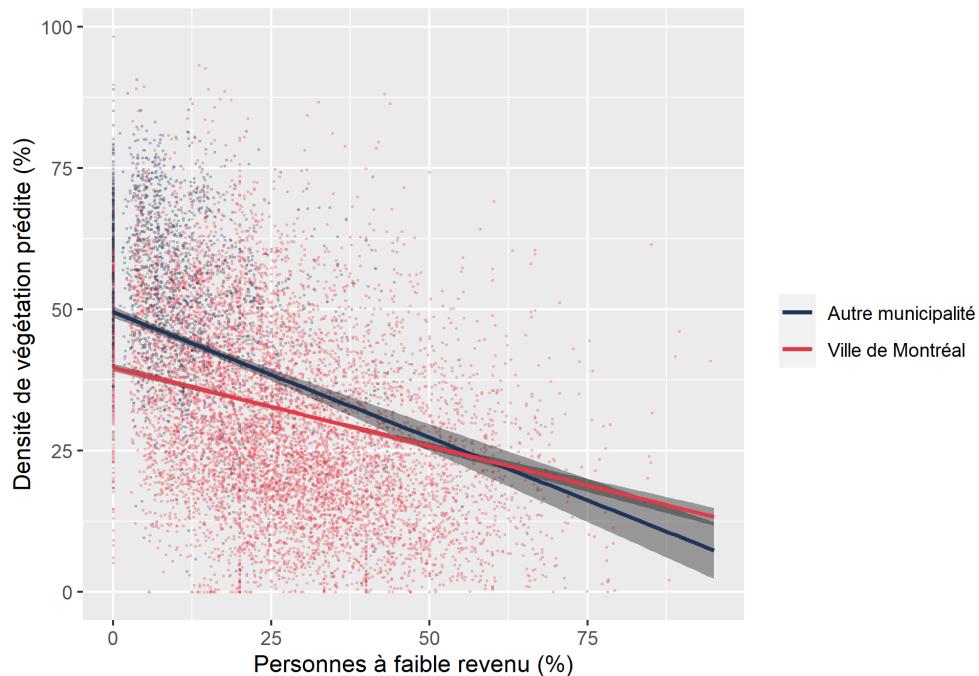
Pour interpréter le coefficient β_3 , il convient alors de bien connaître le nom de la modalité ayant la valeur de 1 (0 étant la modalité de référence). Dans le modèle présenté au tableau 7.12, nous avons multiplié la variable dichotomique ville de Montréal (`VilleMt1`) avec le pourcentage de personnes à faible revenu (`Pct_FR`). Les résultats de ce modèle démontrent que, toutes choses étant égales par ailleurs :

- à chaque augmentation d'une unité du pourcentage à faible revenu (`Pct_FR`), le pourcentage de la couverture végétale diminue significativement de -0,444;
- comparativement à un îlot situé dans une autre municipalité de l'île de Montréal, un îlot de la ville de Montréal a en moyenne -9,804 de couverture végétale;
- à chaque augmentation d'une unité de `Pct_FR` pour un îlot de la Ville Montréal, la couverture végétale augmente de 0,166 comparativement à une autre municipalité de l'île. En d'autres termes, le `Pct_FR` sur le territoire de la ville de Montréal est associé à une diminution de la couverture végétale moins forte que les autres municipalités, comme illustré à la figure 7.8 (pentes en rouge et en bleu).

L'interaction entre une variable qualitative et une variable quantitative peut être représentée par un graphique des effets marginaux. La pente (coefficient) de la variable quantitative varie en fonction des deux catégories de la variable qualitative dichotomique.

TAB. 7.12 : Modèle avec les variables d'interaction entre une VI continue et une VI dichotomique

Variable	Coef.	Erreur type	Valeur de T	P	
Constante	59,275	1,053	56,300	0,000	***
log(HABHA)	-6,160	0,168	-36,640	0,000	***
AgeMedian ordre 1	-20,719	15,354	-1,350	0,177	
AgeMedian ordre 2	-278,141	13,656	-20,370	0,000	***
Pct_014	0,789	0,031	25,100	0,000	***
Pct_65P	0,278	0,018	15,200	0,000	***
Pct_MV	-0,030	0,010	-3,030	0,002	**
Pct_FR	-0,444	0,030	-14,550	0,000	***
VilleMtl	-9,804	0,549	-17,850	0,000	***
VilleMtlX_Pct_FR	0,166	0,032	5,260	0,000	***

**FIG. 7.8 :** Graphique de l'effet marginal de l'interaction entre une variable quantitative et qualitative

7.5.4.3 Variable d'interaction entre deux variables dichotomiques



Variable d'interaction entre deux variables dichotomiques

Nous avons vu qu'il est possible d'introduire une variable d'interaction entre deux variables continues ou entre une variable continue et une autre dichotomique. Sachez qu'il est aussi possible d'introduire une interaction entre deux variables dichotomiques. Sur le sujet, vous pouvez consulter la section 8.3 de l'excellent ouvrage de Hanck et al. (2019).

7.6 Diagnostics de la régression

Pour illustrer comment vérifier si le modèle respecte ou non les hypothèses de la régression, nous utilisons le modèle suivant :

```
modele3 <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = Data-
```

Final)

7.6.1 Nombre d'observations

Tous les auteurs ne s'entendent pas sur le nombre d'observations minimal que devrait comprendre une régression linéaire multiple, tant s'en faut ! Parallèlement, d'autres auteurs proposent aussi des méthodes de simulation pour estimer les coefficients de régression sur un jeu de données comprenant peu d'observations. Bien qu'aucune règle ne soit bien établie, la question du nombre d'observations mérite d'être posée puisqu'un modèle basé sur trop peu d'observations risque de produire des coefficients de régression peu fiables. Par faible fiabilité des coefficients, nous entendons que la suppression d'une ou de plusieurs observations pourrait drastiquement changer l'effet et/ou la significativité d'une ou de plusieurs variables explicatives.

Dans un ouvrage classique intitulé *Using Multivariate Statistics*, Barbara Tabachnick et Linda Fidell (2007, 123-124) proposent deux règles de pouce (à la louche) :

1. $n \geq 50 + 8k$ avec n et k étant respectivement les nombres d'observations et de variables indépendantes, pour tester le coefficient de corrélation multiple (R^2).
2. $n \geq 104 + k$ pour tester individuellement chaque variable indépendante.

Dans le modèle, nous avons 10 210 observations et variables indépendantes. Les deux conditions sont donc largement respectées.

7.6.2 Normalité des résidus

Pour vérifier si les résidus sont normalement distribués, trois démarches largement décrites dans la section 2.5.4 peuvent être utilisées :

- le calcul des coefficients d'asymétrie et d'aplatissement;
- les tests de normalité, particulièrement celui de Jarque-Bera basé sur un test multiplicateur de Lagrange;
- les graphiques (histogramme avec courbe normale et diagramme quantile-quantile) (figure 7.9).

Les deux premières démarches étant parfois très restrictives, nous accordons habituellement une attention particulière aux graphiques.

Pour notre modèle, les coefficients d'asymétrie (-0,263) et d'aplatissement (1,149) signalent que la distribution est plutôt symétrique, mais leptokurtique, c'est-à-dire que les valeurs des résidus sont bien réparties autour de 0, mais avec une faible dispersion. Puisque la valeur de p associée au test de Jarque-Bera est inférieure à 0,05, nous pouvons en conclure que la distribution des résidus est anormale. La forme pointue de la distribution est d'ailleurs confirmée à la lecture de l'histogramme avec la courbe normale et du diagramme quantile-quantile.

```
## Skewness Kurtosis
## -0.161 1.193

##
## Robust Jarque Bera Test
##
## data: residus
## X-squared = 513.15, df = 2, p-value < 2.2e-16
```

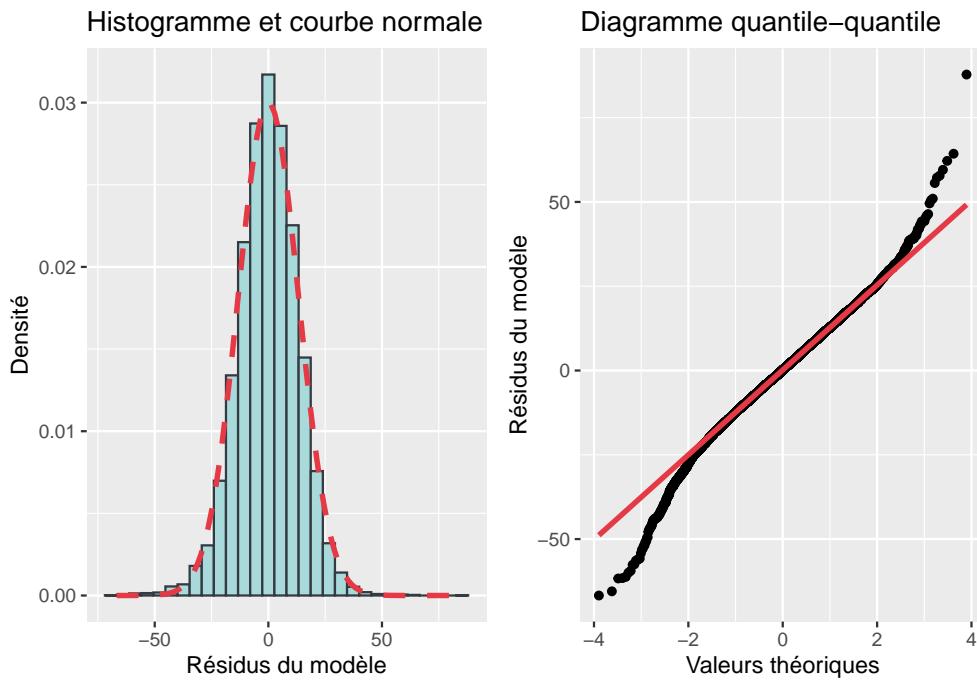


FIG. 7.9 : Vérification de la normalité des résidus

7.6.3 Linéarité et homoscédasticité des résidus

Un modèle est efficace si la dispersion des résidus est homogène sur tout le spectre des valeurs prédictes de la variable dépendante. Dans le cas d'une absence d'homoscédasticité – appelée problème d'hétéroscédasticité –, le nuage de points construit à partir des résidus et des valeurs prédictes (figure 7.10) prend la forme d'une trompette ou d'un entonnoir : les résidus sont alors faibles quand les valeurs prédictes sont faibles et sont de plus en plus élevés au fur et à mesure que les valeurs prédictes augmentent.

Le test de Breusch-Pagan est souvent utilisé pour vérifier l'homoscédasticité des résidus. Il est construit avec les hypothèses suivantes :

- H_0 : homoscédasticité, c'est-à-dire que les termes d'erreur ont une variance constante à travers les valeurs prédictes.
- H_1 : hétéroscédasticité.

Si la valeur de p associée à ce test est inférieure à 0,05, nous réfutons l'hypothèse nulle et nous concluons qu'il y a un problème d'hétéroscédasticité, ce qui est le cas pour notre modèle.

```
## 
## studentized Breusch-Pagan test
## 
## data: modele3
## BP = 1722, df = 8, p-value < 2.2e-16
```

7.6.4 Absence de multicolinéarité excessive

Un modèle présente un problème de multicolinéarité excessive lorsque deux variables indépendantes ou plus sont très fortement corrélées entre elles. Rappelez-vous qu'un coefficient de régression estime l'effet d'une variable dépendante (X_k) si toutes les autres VI restent constantes (c'est-à-dire une fois les autres VI contrôlées, toutes choses étant égales par ailleurs...).



FIG. 7.10 : Distribution des résidus en fonction des valeurs prédictes

Prenons deux variables indépendantes (X_1 et X_2) fortement corrélées avec un coefficient de Pearson très élevé (0,90 par exemple). Admettons que chacune des deux VI a un effet important et significatif sur votre VD lorsqu'une seule est introduite dans le modèle. Si les deux variables sont introduites dans le même modèle, vous évaluez donc l'effet de X_1 une fois contrôlé X_2 et l'effet de X_2 une fois contrôlé X_1 . Par conséquent, l'effet de l'une des deux devient très faible, voire probablement non significatif.

7.6.4.1 Comment évaluer la multicolinéarité ?

Pour ce faire, nous utilisons habituellement le facteur d'inflation de la variance (*Variance Inflation Factor – VIF* en anglais). Le calcul de ce facteur pour chaque VI est basé sur trois étapes.

1. Pour chaque VI, nous construisons un modèle de régression multiple où elle est expliquée par toutes les autres variables indépendantes du modèle. Par exemple, pour la première VI (\hat{X}_1), l'équation du modèle s'écrit :

$$X_1 = b_0 + b_2 X_2 + \dots + b_k X_k + e \quad (7.29)$$

2. À partir de cette équation, nous obtenons ainsi un R^2 qui nous indique la proportion de la variance de X_1 expliquée par les autres VI. Par convention, nous calculons la tolérance (équation (7.30)) qui indique la proportion de la variance de X_k n'étant pas expliquée par les autres VI. En guise d'exemple, une valeur de tolérance égale à 0,1 signale que 90nbsp;% de la variance de X_k est expliqué par les autres variables, ce qui est un problème de multicolinéarité en soit. Concrètement, plus la valeur de la tolérance est proche de zéro, plus c'est problématique.

$$\text{Tolérance}_k = 1 - R_k^2 = \frac{1}{VIF_k} \quad (7.30)$$

3. Puis, nous calculons le facteur d'inflation de la variance (équation (7.31)). Là encore, des règles de pouce (à la louche) sont utilisées. Certains considéreront une valeur de VIF supérieur à 10 (soit

une tolérance à 0,1 ou inférieure) comme problématique, d'autres retiendront le seuil de 5 plus conservateur (soit une tolérance à 0,2 ou inférieure).

$$VIF_k = \frac{1}{1 - R_k^2} \quad (7.31)$$

Pour notre modèle, toutes les valeurs de VIF sont inférieures à 2, indiquant, sans l'ombre d'un doute, l'absence de multicolinéarité excessive.

```
##          GVIF Df GVIF^(1/(2*Df))
## VilleMtl      1.319  1      1.149
## log(HABHA)    1.342  1      1.159
## poly(AgeMedian, 2) 1.399  2      1.087
## Pct_014       1.601  1      1.265
## Pct_65P        1.317  1      1.147
## Pct_MV         1.483  1      1.218
## Pct_FR         1.818  1      1.348
```

7.6.4.2 Comment régler un problème de multicolinéarité?

- La prudence est de mise ! Si une ou plusieurs variables présentent une valeur de VIF supérieure à 5, construisez une matrice de corrélation de Pearson (section 4.3.7) et repérez les valeurs de corrélation supérieures à 0,8 ou inférieures à -0,8. Vous repérerez ainsi les corrélations problématiques entre deux variables indépendantes du modèle.
- Refaites ensuite un modèle en ôtant la variable indépendante avec la plus forte valeur de VIF (7 ou 12 par exemple), et revérifiez les valeurs de VIF. Refaites cette étape si le problème de multicolinéarité excessive persiste.



Une multicolinéarité excessive n'est pas forcément inquiétante

Nous avons vu plus haut comment introduire des variables indépendantes particulières comme des variables d'interaction ($X_1 \times X_2$) ou des variables sous une forme polynomiale (ordre 2 : $X_1 + X_1^2$; ordre 3 : $X_1 + X_1^2 + X_1^3$, etc.). Bien entendu, ces termes composant les variables d'interaction ou d'une forme polynomiale sont habituellement fortement corrélés entre eux. Cela n'est toutefois pas problématique !

Dans l'exemple ci-dessous, nous obtenons deux valeurs de VIF très élevées pour la variable d'interaction `Pct_014:DistCBDkm` (16,713) et l'un des paramètres à partir duquel elle est calculée, soit `DistCBDkm` (12,526).

```
##          GVIF Df GVIF^(1/(2*Df))
## log(HABHA)    1.426  1      1.194
## poly(AgeMedian, 2) 1.768  2      1.153
## Pct_014       3.326  1      1.824
## Pct_65P        1.359  1      1.166
## Pct_MV         1.495  1      1.223
## Pct_FR         1.810  1      1.345
## DistCBDkm     12.526  1      3.539
## Pct_014:DistCBDkm 16.713  1      4.088
```

7.6.5 Absence d'observations aberrantes

7.6.5.1 Détection des observations très influentes du modèle

Lors de l'analyse des corrélations (section 4.3), nous avons vu que des valeurs extrêmes peuvent avoir un impact important sur le coefficient de corrélation de Pearson. Le même principe s'applique à la régression multiple, pour laquelle nous nous s'attendrions à ce que chaque observation joue un rôle équivalent dans la détermination de l'équation du modèle.

Autrement dit, il est possible que certaines observations avec des valeurs extrêmes – fortement dissemblables des autres – aient une influence importante, voire démesurée, dans l'estimation du modèle. Concrètement, cela signifie que si elles étaient ôtées, les coefficients de régression et la qualité d'ajustement du modèle pourraient changer drastiquement. Deux mesures sont habituellement utilisées pour évaluer l'influence de chaque observation sur le modèle :

- **La statistique de la distance de Cook** qui mesure l'influence de chaque observation sur les résultats du modèle. Brièvement, la distance de Cook évalue l'influence de l'observation i en la supprimant du modèle (équation (7.32)). Plus sa valeur est élevée, plus l'observation joue un rôle important dans la détermination de l'équation de régression.

$$D_i = \frac{\sum_{j=1}^n (\hat{y}_i - \hat{y}_{i(j)})^2}{ks^2} \quad (7.32)$$

avec $\hat{y}_{i(j)}$ la valeur prédite quand l'observation i est ôté du modèle, k le nombre de variables indépendantes et s^2 l'erreur quadratique moyenne du modèle.

- **La statistique de l'effet levier** (*leverage value* en anglais) qui varie de 0 (aucune influence) à 1 (explique tout le modèle). La somme de toutes les valeurs de cette statistique est égale au nombre de VI dans le modèle.

Quel critère retenir pour détecter les observations avec potentiellement une trop grande influence sur le modèle ?

Pour les repérer, voire les supprimer, plusieurs auteur(e)s proposent les seuils suivants : $4/n$ ou $8/n$ ou $16/n$. Avec 10210 observations dans le modèle, les seuils seraient les suivants :

```
## Nombre d'observations = 10210 (100 %)
## 4/n = 0.00039
## 8/n = 0.00078
## 16/n = 0.00157
## Observations avec une valeur supérieure ou égale aux différents seuils
## 4/n = 605 soit 5.93 %
## 8/n = 275 soit 2.69 %
## 16/n = 133 soit 1.3 %
```

Le critère de $4/n$ étant plutôt sévère, nous privilégions généralement celui de $8/n$, voire $16/n$. Il est aussi possible de construire un nuage de points pour les repérer (figure 7.11).

7.6.5.2 Quoi faire avec les observations très influentes du modèle

Trois approches sont possibles :

- **Recourir à des régressions bootstrap**, ce qui permet généralement de supprimer l'effet de ces observations. Brièvement, le principe général est de créer un nombre élevé d'échantillons du jeu de

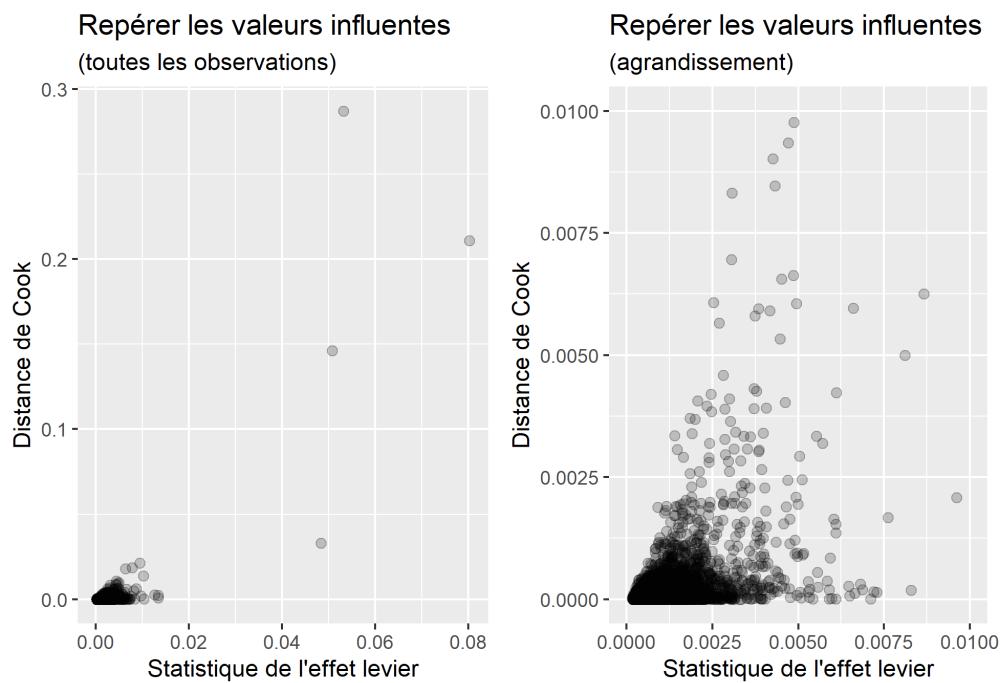


FIG. 7.11 : Repérage graphique les valeurs influentes du modèle

données initial (1000 à 2000 itérations par exemple) et de construire un modèle de régression pour chacun d'eux. On obtiendra ainsi des intervalles de confiance pour les coefficients de régression et les mesures d'ajustement du modèle.

- **Supprimer les observations trop influentes** (avec l'un des critères de $4/n$, $8/n$ et $16/n$ vus plus haut). Une fois supprimées, il convient 1) de recalculer le modèle, 2) de refaire le diagnostic de la régression au complet et finalement, 3) de comparer les modèles avant et après suppression des valeurs trop influentes, notamment la qualité d'ajustement du modèle (R^2 ajusté) et les coefficients de régression. Des changements importants indiqueront que le premier modèle est potentiellement biaisé.
- **Utiliser un modèle linéaire généralisé (GLM)** permettant d'utiliser une distribution différente correspondant plus à votre jeu de données (chapitre 8).

7.7 Mise en œuvre dans R

7.7.1 Fonctions `lm`, `summary()` et `confint()`

Les fonctions de base `lm`, `summary()` et `confint()` permettent respectivement 1) de construire un modèle, 2) d'afficher ces résultats et 3) d'obtenir les intervalles de confiance des coefficients de régression :

- `monModele <- lm(Y ~X1+X2+...+Xk)` avec Y étant la variable dépendante et les variables indépendantes (X_1 à X_k) étant séparées par le signe $+$.
- `summary(monModele)`
- `confint(monModele, level=.95)`.

Dans la syntaxe ci-dessous, vous retrouverez les différents modèles abordés dans les sections précédentes; remarquez que toutes que les lignes `summary` sont mises en commentaires afin de ne pas afficher les résultats des modèles.

```

# Chargement des données
load("data/lm/DataVegetation.RData")

# 1er modèle de régression
modele1 <- lm(VegPct ~ HABHA+AgeMedian+Pct_014+Pct_65P+Pct_MV+Pct_FR,
               data = DataFinal)
# summary(modele1)

# 2e modèle de régression : fonction polynomiale d'ordre 2 (poly(AgeMedian,2))
modele2 <- lm(VegPct ~ HABHA+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR,
               data = DataFinal)
# summary(modele2)

# 3e modèle de régression : forme logarithmique (log(HABHA))
modele3 <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR,
               data = DataFinal)
# summary(modele3)

# 4e modèle de régression : VI dichotomique
# création de la variable dichotomique (VilleMtl)
DataFinal$VilleMtl <- ifelse(DataFinal$SDRNOM == "Montréal", 1, 0)
modele4 <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR+
               VilleMtl, # variable dichotomique
               data = DataFinal)
# summary(modele4)

# 5e modèle de régression : VI polytomique
# création de la variable polytomique (Munic)
DataFinal$Munic <- relevel(DataFinal$SDRNOM, ref="Montréal")
modele5 <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR+
               Munic, data = DataFinal)
# summary(modele5)

# 6e modèle de régression : interaction entre deux VI continues,
# soit DistCBDkm*Pct_014
modele6 <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR+
               DistCBDkm+DistCBDkm*Pct_014,
               data = DataFinal)
# summary(modele6)

# 7e modèle de régression : interaction entre une VI continue et une VI dichotomique,
# soit VilleMtl*Pct_FR
modele7 <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR+
               VilleMtl*Pct_FR,
               data = DataFinal)
# summary(modele7)

```

À la figure 7.12, les résultats de la régression linéaire multiple, obtenus avec la `summary(monModele)`, sont présentés en quatre sections distinctes :

- a. Le rappel de l'équation du modèle.
- b. Quelques statistiques descriptives sur les résidus du modèle, soit la différence entre les valeurs observées et prédictes.
- c. Un tableau pour les coefficients de régression comprenant plusieurs colonnes, à savoir les coeffi-

- cients de régression (*Estimate*), l'erreur type du coefficient (*Std. Error*), la valeur de t (*t value*) et la probabilité associée à la valeur de t (*Pr(>|t|)*). La première ligne de ce tableau (*Estimate*) est pour la constante (*Intercept* en anglais) et celles qui suivent sont pour les variables indépendantes.
- d. Les mesures d'ajustement du modèle, dont le RMSE (*Residual standard error*), les R^2 classique (*Multiple R-squared*) et ajusté (*Adjusted R-squared*), la statistique F avec le nombre de degrés de liberté en lignes (nombre d'observations) et en colonnes ($n-k-1$) ainsi que la valeur de p qui est lui associée (*F-statistic: 1223 on 6 and 10203 DF, p-value: < 2.2e-16*).

```

## a. Rappel de l'équation du modèle
## Call:
## lm(formula = VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 +
##     Pct_65P + Pct_MV + Pct_FR, data = DataFinal)

## b. Statistiques sur les résidus
## Residuals:
##   Min     1Q Median     3Q    Max 
## -66.848 -8.660  0.381  8.961 83.269 

## c. Tableau pour les coefficients de régression
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             5.283e+01  1.001e+00  52.781 < 2e-16 ***
## log(HABHA)              -6.855e+00  1.683e-01 -40.730 < 2e-16 ***
## poly(AgeMedian, 2)1    1.198e+01  1.559e+01   0.769 0.441958  
## poly(AgeMedian, 2)2   -2.861e+02  1.394e+01 -20.525 < 2e-16 *** 
## Pct_014                 9.406e-01  3.126e-02  30.093 < 2e-16 *** 
## Pct_65P                 3.062e-01  1.851e-02  16.546 < 2e-16 *** 
## Pct_MV                 -3.630e-02  9.943e-03 -3.651 0.000262 *** 
## Pct_FR                 -3.443e-01  1.103e-02 -31.212 < 2e-16 *** 
## ---                     
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## d. Mesures pour la qualité d'ajustement
## Residual standard error: 13.57 on 10202 degrees of freedom
## Multiple R-squared:  0.4657,      Adjusted R-squared:  0.4653 
## F-statistic: 1270 on 7 and 10202 DF,  p-value: < 2.2e-16

```

FIG. 7.12 : Différentes parties obtenues avec la fonction summary(Modèle)

```
# Intervalle de confiance des coefficients à 95 %
confint(modele3)
```

```

##                               2.5 %         97.5 %
## (Intercept)      50.8684505  54.79255157
```

```

## log(HABHA)           -7.1847527   -6.52495353
## poly(AgeMedian, 2)1 -18.5676034   42.53686203
## poly(AgeMedian, 2)2 -313.4726002 -258.81630119
## Pct_014              0.8793672   1.00190861
## Pct_65P               0.2699504   0.34250907
## Pct_MV                -0.0557951  -0.01681481
## Pct_FR                -0.3659445  -0.32269562

```

7.7.2 Comparaison des modèles

Tel que détaillé à la section 7.3.2, pour comparer des modèles imbriqués, il convient d'analyser les valeurs du R^2 ajusté et du F incrémentiel, ce qui peut être fait en trois étapes.

Première étape. Il peut être judicieux d'afficher l'équation des différents modèles afin de se remémorer les VI introduites dans chacun d'eux, et ce, avec la fonction `MonModèle$call$formula`.

```

# Rappel des équations des huit modèles
print(modele1$call$formula)

## VegPct ~ HABHA + AgeMedian + Pct_014 + Pct_65P + Pct_MV + Pct_FR

print(modele2$call$formula)

## VegPct ~ HABHA + poly(AgeMedian, 2) + Pct_014 + Pct_65P + Pct_MV +
##          Pct_FR

print(modele3$call$formula)

## VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 + Pct_65P +
##          Pct_MV + Pct_FR

print(modele4$call$formula)

## VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 + Pct_65P +
##          Pct_MV + Pct_FR + VilleMtl

print(modele5$call$formula)

## VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 + Pct_65P +
##          Pct_MV + Pct_FR + Munic

print(modele6$call$formula)

## VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 + Pct_65P +
##          Pct_MV + Pct_FR + DistCBDkm + DistCBDkm * Pct_014

print(modele7$call$formula)

```

```
## VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 + Pct_65P +
##      Pct_MV + Pct_FR + VilleMtl + VilleMtlX_Pct_FR
```

Deuxième étape. La syntaxe ci-dessous vous permet de comparer les R^2 ajustés des différents modèles. Nous constatons ainsi que :

- La valeur du R^2 ajusté du modèle 2 est supérieure à celle du modèle 1 (0,4378 versus 0,4179), signalant que la forme polynomiale d'ordre 2 pour l'âge médian des bâtiments (`poly(AgeMedian, 2)`) améliore la prédiction comparativement à la forme originelle de (`AgeMedian`).
- La valeur du R^2 ajusté du modèle 3 est supérieure à celle du modèle 2 (0,4653 versus 0,4378), signalant que la forme logarithmique pour la densité de population (`log(HABHA)`) améliore la prédiction comparativement à la forme originelle (`HABHA`).
- La valeur du R^2 ajusté du modèle 4 est supérieure à celle du modèle 3 (0,4863 versus 0,4653), signalant que l'introduction de la variable dichotomique (`VilleMtl`) pour la municipalité apporte un gain de variance expliquée non négligeable.
- La valeur du R^2 ajusté du modèle 5 est supérieure à celle du modèle 4 (0,5064 versus 0,4863), signalant que l'introduction de la variable polytomique pour les municipalités de l'île de Montréal (`Muni`) améliore la prédiction du modèle comparativement à la variable dichotomique (`VilleMtl`).
- La valeur du R^2 ajusté du modèle 6 est supérieure à celle du modèle 2 (0,4953 versus 0,4378), signalant que l'introduction d'une variable d'interaction entre deux variables continues (`DistCBDkm + DistCBDkm * Pct_014`) apporte également un gain substantiel comparativement au modèle 2, ne comprenant pas cette variable d'interaction.
- La valeur du R^2 ajusté du modèle 7 est supérieure à celle du modèle 2 (0,4877 versus 0,4378), signalant que l'introduction d'une variable d'interaction entre une variable continue et la variable dichotomique (`DistCBDkm + DistCBDkm * Pct_014`) apporte également un gain substantiel comparativement au modèle 2, ne comprenant pas cette variable d'interaction.

```
cat("\nComparaison des R2 ajustés :",
  "\nModèle 1.", round(summary(modele1)$adj.r.squared,4),
  "\nModèle 2.", round(summary(modele2)$adj.r.squared,4),
  "\nModèle 3.", round(summary(modele3)$adj.r.squared,4),
  "\nModèle 4.", round(summary(modele4)$adj.r.squared,4),
  "\nModèle 5.", round(summary(modele5)$adj.r.squared,4),
  "\nModèle 6.", round(summary(modele6)$adj.r.squared,4),
  "\nModèle 7.", round(summary(modele7)$adj.r.squared,4)
)

## 
## Comparaison des R2 ajustés :
## Modèle 1. 0.4179
## Modèle 2. 0.4378
## Modèle 3. 0.4653
## Modèle 4. 0.4863
## Modèle 5. 0.5064
## Modèle 6. 0.4953
## Modèle 7. 0.4877
```

Troisième étape. La syntaxe ci-dessous permet d'obtenir le F incrémentiel pour des modèles ne comprenant pas le même nombre de variables dépendantes, et ce, en utilisant la fonction `anova(modele1, modele2, ..., modelen)`.

Par exemple, la syntaxe `anova(modele1, modele2)` permet de comparer les deux modèles et signale que le

gain de variance expliquée entre les deux modèles (R^2 de 0,4179 et de 0,4378) est significatif (F incrémentiel = 362,64; $p < 0,001$).

```
# Comparaison des deux modèles uniquement (modèles 1 et 2)
anova(modele1, modele2)

## Analysis of Variance Table
##
## Model 1: VegPct ~ HABHA + AgeMedian + Pct_014 + Pct_65P + Pct_MV + Pct_FR
## Model 2: VegPct ~ HABHA + poly(AgeMedian, 2) + Pct_014 + Pct_65P + Pct_MV +
##             Pct_FR
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1 10203 2046427
## 2 10202 1976182  1      70245 362.64 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Il est aussi possible de comparer plusieurs modèles simultanément. Notez que dans la syntaxe ci-dessous, le troisième modèle n'est pas inclus, car il comprend le même nombre de variables indépendantes que le second modèle; il en va de même pour le sixième modèle comparativement au cinquième. Ici aussi, l'analyse des valeurs de F et de p vous permettent de vérifier si les modèles, et donc leurs R^2 ajustés, sont significativement différents (quand $p < 0,05$).

```
# Comparaison de plusieurs modèles
anova(modele1, modele2, modele4, modele5, modele7)

## Analysis of Variance Table
##
## Model 1: VegPct ~ HABHA + AgeMedian + Pct_014 + Pct_65P + Pct_MV + Pct_FR
## Model 2: VegPct ~ HABHA + poly(AgeMedian, 2) + Pct_014 + Pct_65P + Pct_MV +
##             Pct_FR
## Model 3: VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 + Pct_65P +
##             Pct_MV + Pct_FR + VilleMtl
## Model 4: VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 + Pct_65P +
##             Pct_MV + Pct_FR + Munic
## Model 5: VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 + Pct_65P +
##             Pct_MV + Pct_FR + VilleMtl + VilleMtlX_Pct_FR
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1 10203 2046427
## 2 10202 1976182  1      70245 412.995 < 2.2e-16 ***
## 3 10201 1805547  1     170636 1003.224 < 2.2e-16 ***
## 4 10188 1732849 13     72698  32.878 < 2.2e-16 ***
## 5 10200 1800664 -12    -67815  33.226 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Quel modèle choisir?

Nous avons déjà évoqué le principe de parcimonie. À titre de rappel, l'ajout de variables indépendantes qui s'avèrent significatives fait inévitablement augmenter la variance expliquée et ainsi la valeur R^2 ajusté. Par

contre, elle peut rendre le modèle plus complexe à analyser, voire entraîner un surajustement du modèle. Nous avons vu que l'introduction des variables dichotomique, polytomique et d'interaction avait pour effet d'augmenter la capacité de prédiction du modèle. Quoi qu'il en soit, le gain de variance expliquée s'élève à environ 4nbsp;% entre le troisième modèle versus le cinquième et le sixième :

- Modèle 3 ($R^2=0,465$). $\text{VegPct} \sim \log(\text{HABHA}) + \text{poly}(\text{AgeMedian}, 2) + \text{Pct_014} + \text{Pct_65P} + \text{Pct_MV} + \text{Pct_FR}$
- Modèle 5 ($R^2=0,506$). $\text{VegPct} \sim \log(\text{HABHA}) + \text{poly}(\text{AgeMedian}, 2) + \text{Pct_014} + \text{Pct_65P} + \text{Muni}$
- Modèle 6 ($R^2=0,495$). $\text{VegPct} \sim \log(\text{HABHA}) + \text{poly}(\text{AgeMedian}, 2) + \text{Pct_014} + \text{Pct_65P} + \text{Pct_MV} + \text{Pct_FR} + \text{DistCBDkm} + \text{DistCBDkm} * \text{Pct_014}$

Par conséquent, il est légitime de se questionner sur le bien-fondé de conserver ces variables indépendantes additionnelles : `Muni` pour le modèle 5 et `DistCBDkm + DistCBDkm * Pct_014` pour le modèle 6. Trois options sont alors envisageables :

- Bien entendu, conservez l'une ou l'autre de ces variables additionnelles si elles sont initialement reliées à votre cadre théorique.
- Conservez l'une ou l'autre de ces variables additionnelles si elles permettent de répondre à une question spécifique (non prévue initialement) et si les associations ainsi révélées méritent, selon vous, discussion.
- Supprimez-les si leur apport est limité et ne fait que complexifier le modèle.

7.7.3 Diagnostic sur un modèle

7.7.3.1 Vérification le nombre d'observations

La syntaxe suivante permet de vérifier si le nombre d'observations est suffisant pour tester le R^2 et chacune des variables indépendantes.

```
modele3 <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2)+
                 Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal)

# Nombre d'observation
nobs <- length(modele3$fitted.values)

# Nombre de variables indépendantes (coefficients moins la constante)
k <- length(modele3$coefficients)-1

# Première règle de pouce
if(nobs >= 50+(8*k)){
  cat("\nNombre d'observations suffisant pour tester le R2")
} else{
  cat("\nAttention! Nombre d'observations insuffisant pour tester le R2")
}

## 
## Nombre d'observations suffisant pour tester le R2

# Deuxième règle de pouce
if(nobs >= 104+k){
  cat("\nNombre d'observations suffisant pour tester individuellement chaque VI")
} else{
  cat("\nAttention! Nombre d'observations insuffisant",
      "\npour tester individuellement chaque VI")
}
```

```
##  
## Nombre d'observations suffisant pour tester individuellement chaque VI
```

7.7.3.2 Vérification la normalité des résidus

La syntaxe suivante permet de vérifier la normalité des résidus selon les trois démarches classiques : 1) coefficients d'asymétrie et d'aplatissement, 2) test de normalité de Jarque-Bera (fonction `JarqueBeraTest` du package `DescTools`) et 3) les graphiques (histogramme avec courbe normale et diagramme quantile-quantile).

```
library(DescTools)  
library(stats)  
library(ggplot2)  
library(ggpubr)  
  
# Vecteur pour les résidus du modèle  
residus <- modele3$residuals  
  
# 1. coefficients d'asymétrie et d'aplatissement  
c(Skewness= round(DescTools:::Skew(residus),3),  
  Kurtosis = round(DescTools:::Kurt(residus),3))  
  
## Skewness Kurtosis  
## -0.263 1.149  
  
# 2. Test de normalité de Jarque-Bera  
JarqueBeraTest(residus)  
  
##  
## Robust Jarque Bera Test  
##  
## data: residus  
## X-squared = 528.51, df = 2, p-value < 2.2e-16  
  
# 3. Graphiques  
Ghisto <- ggplot() +  
  geom_histogram(aes(x = residus, y = ..density..),  
    bins = 30, color = "#343a40", fill = "#a8dadc") +  
  stat_function(fun = dnorm,  
    args = list(mean = mean(residus),  
               sd = sd(residus)),  
    color = "#e63946", size = 1.2, linetype = "dashed") +  
  labs(title="Histogramme et courbe normale",  
    y = "densité", "Résidus du modèle")  
  
Gqqplot <- qplot(sample = residus)+  
  geom_qq_line(line.p = c(0.25, 0.75),  
    color = "#e63946", size=1.2)+  
  labs(title="Diagramme quantile-quantile",
```

```
x="Valeurs théoriques",
y = "Résidus")

ggarrange(Ghisto, Gqqplot, ncol=2, nrow=1)
```

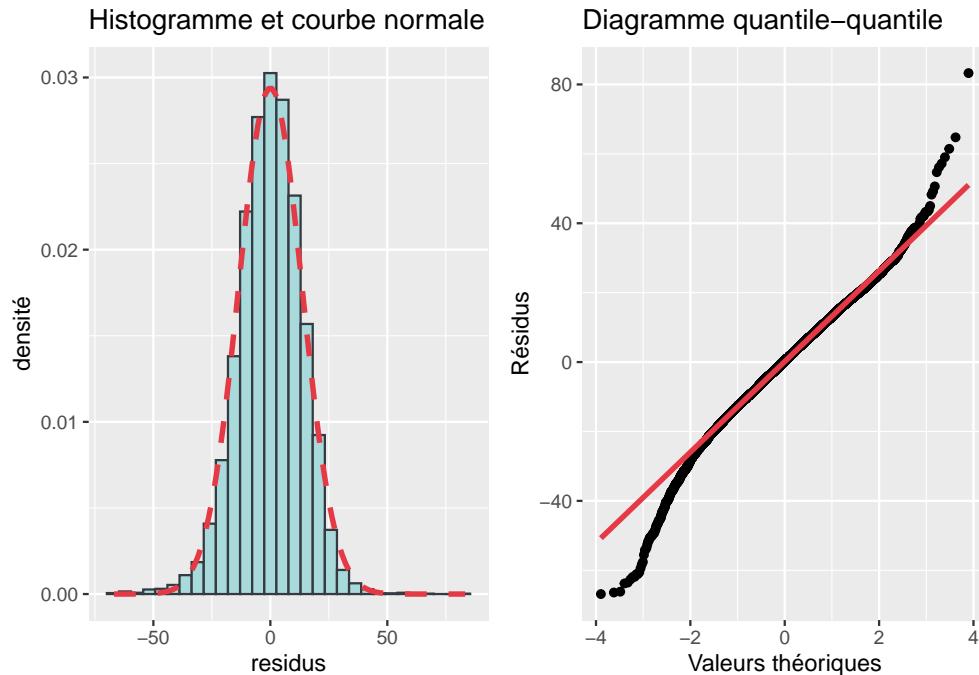


FIG. 7.13 : Diagnostic : la normalité des résidus

7.7.3.3 Évaluation de la linéarité et l'homoscédasticité des résidus

La syntaxe suivante permet de vérifier si l'hypothèse d'homoscédasticité des résidus est respectée avec : 1) un nuage de points entre les valeurs prédictes et des résidus, 3) les graphiques (histogramme avec courbe normale et diagramme quantile-quantile) et 2) le test de Breusch-Pagan (fonction `bptest` du package `lmtest`).

```
# 1. Test de Breusch-Pagan pour vérifier l'homoscédasticité
library(lmtest)
bptest(modele3)
```

```
## 
## studentized Breusch-Pagan test
## 
## data: modele3
## BP = 1651.5, df = 7, p-value < 2.2e-16

if(bptest(modele3)$p.value < 0.05){
  cat("\nAttention : problème d'hétérosécédasticité des résidus")
} else{
  cat("\nParfait : homoscédasticité des résidus")
}
```

```
##  
## Attention : problème d'hétérosécédasticité des résidus
```

```
# 2. Graphique entre les valeurs prédictes et les résidus  
residus <- modele3$residuals  
ypredicts <- modele3$fitted.values  
  
ggplot() +  
  geom_point(aes(x = ypredicts, y = residus),  
             color = "#343a40", fill = "#a8dadc",  
             alpha = 0.2, size = 0.8) +  
  geom_smooth(aes(x = ypredicts, y = residus),  
              method = lm, color = "red") +  
  labs(x = "Valeurs prédictes", y = "Résidus")
```

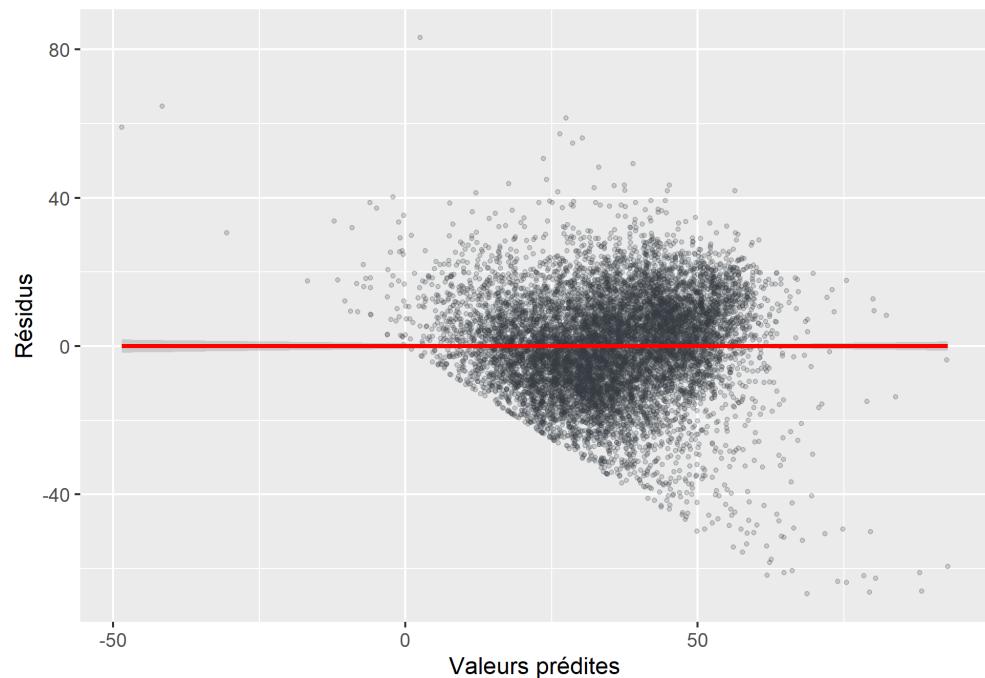


FIG. 7.14 : Distribution des résidus en fonction des valeurs prédictes

7.7.3.4 Vérification la multicolinéarité excessive

Pour vérifier la présence ou l'absence de multicolinéarité excessive, nous utilisons habituellement la fonction `vif` du package `car`.

```
library(car)  
  
# facteur d'inflation de la variance  
round(car::vif(modele3), 3)
```

	GVIF	Df	GVIF^(1/(2*Df))
## log(HABHA)	1.289	1	1.136
## poly(AgeMedian, 2)	1.387	2	1.085

```

## Pct_014          1.518  1        1.232
## Pct_65P          1.304  1        1.142
## Pct_MV           1.480  1        1.217
## Pct_FR           1.730  1        1.315

```

```

# problème de multicolinéarité (VIF > 10)?
car::vif(modele3) > 10

```

```

##                  GVIF     Df GVIF^(1/(2*Df))
## log(HABHA)      FALSE FALSE      FALSE
## poly(AgeMedian, 2) FALSE FALSE      FALSE
## Pct_014         FALSE FALSE      FALSE
## Pct_65P          FALSE FALSE      FALSE
## Pct_MV           FALSE FALSE      FALSE
## Pct_FR           FALSE FALSE      FALSE

```

```

# problème de multicolinéarité (VIF > 5)?
car::vif(modele3) > 5

```

```

##                  GVIF     Df GVIF^(1/(2*Df))
## log(HABHA)      FALSE FALSE      FALSE
## poly(AgeMedian, 2) FALSE FALSE      FALSE
## Pct_014         FALSE FALSE      FALSE
## Pct_65P          FALSE FALSE      FALSE
## Pct_MV           FALSE FALSE      FALSE
## Pct_FR           FALSE FALSE      FALSE

```

7.7.3.5 Répérage des valeurs très influentes du modèle

La syntaxe suivante permet d'évaluer le nombre de valeurs très influentes dans le modèle avec les critères de $4/n$, $8/n$ et $16/n$ pour la distance de Cook.

```

nobs <- length(modele3$fitted.values)
DistanceCook <- cooks.distance(modele3)
n4 <- length(DistanceCook[DistanceCook > 4/nobs])
n8 <- length(DistanceCook[DistanceCook > 8/nobs])
n16 <- length(DistanceCook[DistanceCook > 16/nobs])
cat("Nombre d'observations =", nobs, "(100 %)",
    "\n 4/n =", round(4/nobs,5),
    "\n 8/n =", round(8/nobs,5),
    "\n 16/n =", round(16/nobs,5),
    "\nObservations avec une valeur supérieure ou égale aux différents seuils",
    "\n 4/n =", n4, "soit", round(n4/nobs*100,2), "%",
    "\n 8/n =", n8, "soit", round(n8/nobs*100,2), "%",
    "\n 16/n =", n16, "soit", round(n16/nobs*100,2), "%"
)

```

```

## Nombre d'observations = 10210 (100 %)
## 4/n = 0.00039
## 8/n = 0.00078
## 16/n = 0.00157

```

```
## Observations avec une valeur supérieure ou égale aux différents seuils
## 4/n = 604 soit 5.92 %
## 8/n = 285 soit 2.79 %
## 16/n = 132 soit 1.29 %
```

Vous pouvez également construire un nuage de points avec la distance de Cook et l'effet de levier (*leverage value*) pour repérer visuellement les observations très influentes.

```
library(car)
library(ggpubr)
DistanceCook <- cooks.distance(modele3)
LeverageValue <- hatvalues(modele3)

G1 <- ggplot()+
  geom_point(aes(x = LeverageValue, y = DistanceCook),
             alpha = 0.2, size = 2, col="black", fill="red")+
  labs(x = "Effet levier",
       y = 'Distance de Cook',
       title = 'Repérer les valeurs influentes',
       subtitle = '(toutes les observations)')

G2 <- ggplot()+
  geom_point(aes(x = LeverageValue, y = DistanceCook),
             alpha = 0.2, size = 2, col="black", fill="red")+
  ylim(0,0.01)+
  xlim(0,0.01)+
  labs(x = "Effet levier",
       y = 'Distance de Cook',
       title = 'Repérer les valeurs influentes',
       subtitle = '(agrandissement)')

ggarrange(G1,G2, nrow=1, ncol=2)
```

7.7.3.6 Construction d'un nouveau modèle en supprimant les observations très influentes du modèle

Dans un premier temps, il convient de construire un nouveau modèle sans les valeurs influentes du modèle de départ.

```
# Nombre d'observation dans le modèle 3
nobs <- length(modele3$fitted.values)

# Distance de Cook
cook <- cooks.distance(modele3)

# Les observations très influentes avec le critère de 16/n
DataSansOutliers <- cbind(DataFinal, cook)
DataSansOutliers <- DataSansOutliers[DataSansOutliers$cook < 8/nobs, ]
modele3b <- lm(VegPct ~ log(HABHA)+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR,
                data = DataSansOutliers)
nobsb <- length(modele3b$fitted.values)
```

Comparez les valeurs du R² ajusté des deux modèles. Habituellement, la suppression des valeurs très influentes s'accompagne d'une augmentation du R² ajusté. C'est notamment le cas ici puisque sa valeur

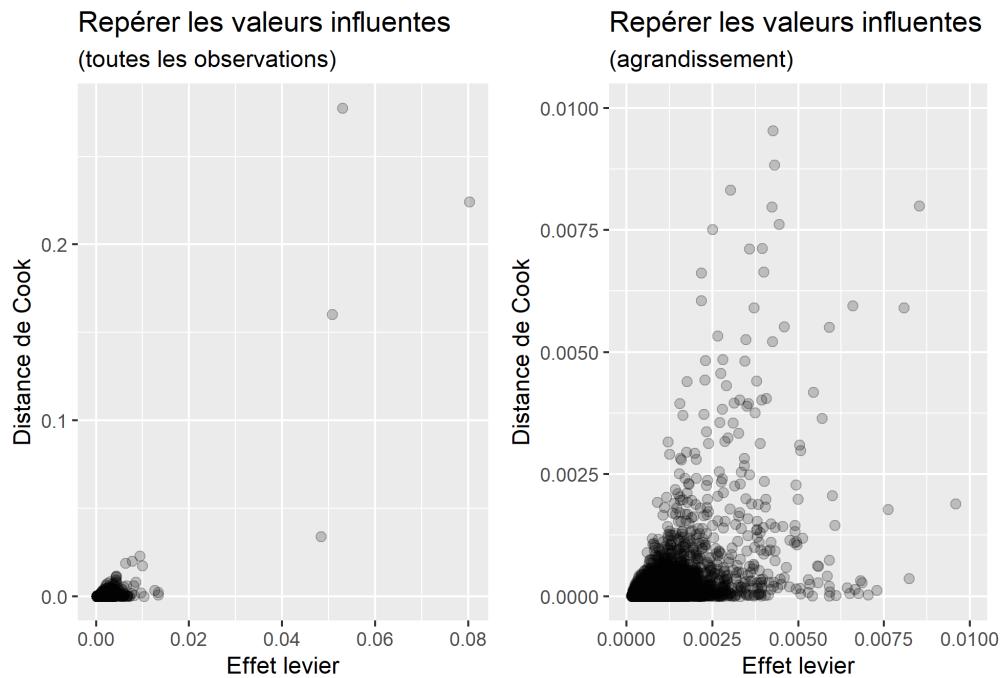


FIG. 7.15 : Repérage graphique des valeurs influentes du modèle

grimpe de 0,4653 à 0,5684, signalant ainsi un gain important pour la variance expliquée.

```
# Comparaison des mesures d'ajustement
cat("\nComparaison des R2 ajustés :",
  "\nModèle de départ (n=", nobs, ")", ",
  round(summary(modele3)$adj.r.squared,4),

  "\nModèle sans les observations très influentes (n=", nobsb, ")", ",
  round(summary(modele3b)$adj.r.squared,4),
  sep=""
)

## 
## Comparaison des R2 ajustés :
## Modèle de départ (n=10210), 0.4653
## Modèle sans les observations très influentes (n=9925), 0.5684
```

Pour le modèle, il convient alors de refaire le diagnostic de la régression et de vérifier si la suppression des observations très influentes a amélioré : 1) la normalité, la linéarité et l'homoscédasticité des résidus, 2) la multicolinéarité excessive et 3) l'absence de valeurs trop influentes.

La normalité des résidus s'est-elle ou non améliorée ?

Pour ce faire, comparez les valeurs d'asymétrie, d'aplatissement et du test de Jarque-Bera et les graphiques de normalité. À la lecture des valeurs :

- l'asymétrie est très similaire (-0,260 à -0,265);
- l'aplatissement s'est amélioré (1,183 à 0,164);
- le test de Jarque-Bera signale toujours un problème de normalité ($p < 0,001$), mais sa valeur a nettement diminué (548,7 à 131,24);

- les graphiques démontrent une nette amélioration de la normalité des résidus.

```
# 1. coefficients d'asymétrie et d'aplatissement
resmodele3 <- rstudent(modele3)
resmodele3b <- rstudent(modele3b)

c(Skewness= round(Skew(resmodele3),3),
  Kurtosis = round(Kurt(resmodele3),3))
```

```
## Skewness1 Skewness2 Skewness3 Skewness4 Kurtosis1 Kurtosis2 Kurtosis3 Kurtosis4
##      -0.260      0.024     -10.739      0.000      1.185      0.048     24.448      0.000
```

```
c(Skewness= round(Skew(resmodele3b),3),
  Kurtosis = round(Kurt(resmodele3b),3))
```

```
## Skewness1 Skewness2 Skewness3 Skewness4 Kurtosis1 Kurtosis2 Kurtosis3 Kurtosis4
##      -0.265      0.025     -10.790      0.000      0.165      0.049      3.360      0.000
```

```
# 2. Test de normalité de Jarque-Bera
JarqueBeraTest(resmodele3)
```

```
##
## Robust Jarque Bera Test
##
## data: resmodele3
## X-squared = 548.7, df = 2, p-value < 2.2e-16
```

```
JarqueBeraTest(resmodele3b)
```

```
##
## Robust Jarque Bera Test
##
## data: resmodele3b
## X-squared = 131.24, df = 2, p-value < 2.2e-16
```

```
# 3. Graphiques
Ghisto1 <- ggplot() +
  geom_histogram(aes(x = resmodele3, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#a8dadc") +
  stat_function(fun = dnorm, args = list(mean = mean(resmodele3),
                                         sd = sd(resmodele3)),
                color = "#e63946", size = 1.2, linetype = "dashed") +
  labs(title="Modèle de départ", y = "densité", x="Résidus studentisés")
```

```
Gqqplot1 <- qplot(sample = residus) +
  geom_qq_line(line.p = c(0.25, 0.75), color = "#e63946", size=1.2) +
  labs(title="Modèle de départ", x="Valeurs théoriques", y = "Résidus studentisés")
```

```
Ghisto2 <- ggplot() +
  geom_histogram(aes(x = resmodele3b, y = ..density..),
                 bins = 30, color = "#343a40", fill = "#a8dadc") +
```

```

stat_function(fun = dnorm, args = list(mean = mean(resmodele3b),
                                         sd = sd(resmodele3b)),
               color = "#e63946", size = 1.2, linetype = "dashed")+
  labs(title="Modèle après suppression", x="Valeurs théoriques", y="Résidus studentisés")

Gqqplot2 <- qplot(sample = resmodele3b)+
  geom_qq_line(line.p = c(0.25, 0.75), color = "#e63946", size=1.2)+ 
  labs(title="Modèle après suppression",x="Valeurs théoriques", y = "Résidus studentisés")

library(ggpubr)
ggarrange(Ghisto1, Ghisto2, Gqqplot1, Gqqplot2, ncol=2, nrow=2)

```

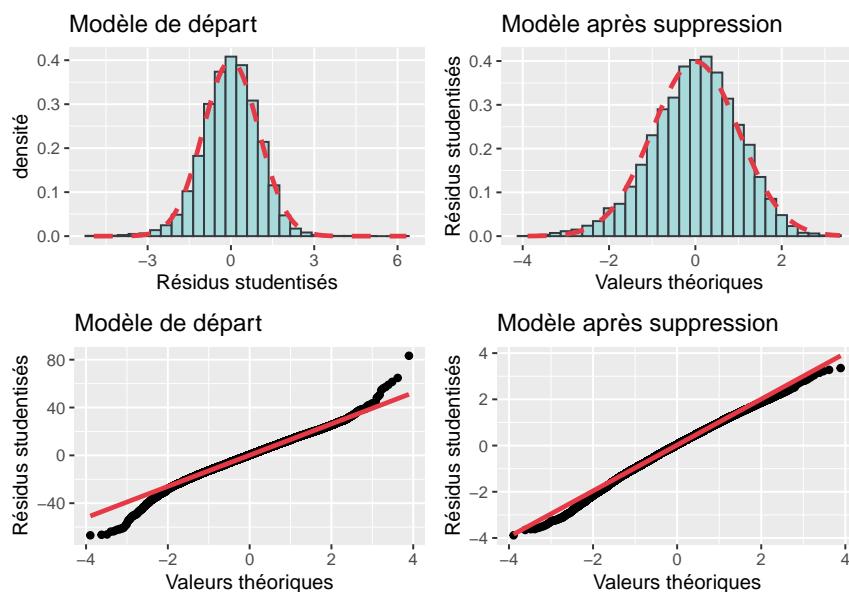


FIG. 7.16 : Normalité des résidus avant et après la suppression des valeurs influentes

Le problème d'hétérosécédasticité est-il corrigé ?

- la valeur du test de Breusch-Pagan est beaucoup plus faible, mais il semble persister un problème d'hétérosécédasticité.

```

# homoscédasticité des résidus améliorée ou non?
library(lmtest)
library(ggpubr)

bptest(modele3)

```

```

##
## studentized Breusch-Pagan test
##
## data: modele3
## BP = 1651.5, df = 7, p-value < 2.2e-16

```

```
bptest(modele3b)
```

```

## 
## studentized Breusch-Pagan test
##
## data: modele3b
## BP = 640.53, df = 7, p-value < 2.2e-16

resmodele3 <- residuals(modele3)
resmodele3b <- residuals(modele3b)
ypredicts3 <- modele3$fitted.values
ypredicts3b <- modele3b$fitted.values

G1 <- ggplot() +
  geom_point(aes(x = ypredicts3, y = resmodele3),
             color = "#343a40", fill = "#a8dadc", alpha = 0.2, size = 0.8) +
  geom_smooth(aes(x = ypredicts3, y = resmodele3), method = lm, color = "red") +
  labs(title="Modèle de départ", x="Valeurs prédictes", y = "Résidus studentisés")

G2 <- ggplot() +
  geom_point(aes(x = ypredicts3b, y = resmodele3b),
             color = "#343a40", fill = "#a8dadc", alpha = 0.2, size = 0.8) +
  geom_smooth(aes(x = ypredicts3b, y = resmodele3b), method = lm, color = "red") +
  labs(title="Modèle après suppression", x="Valeurs prédictes", y = "Résidus studentisés")

ggarrange(G1, G2, ncol=2, nrow=1)

```

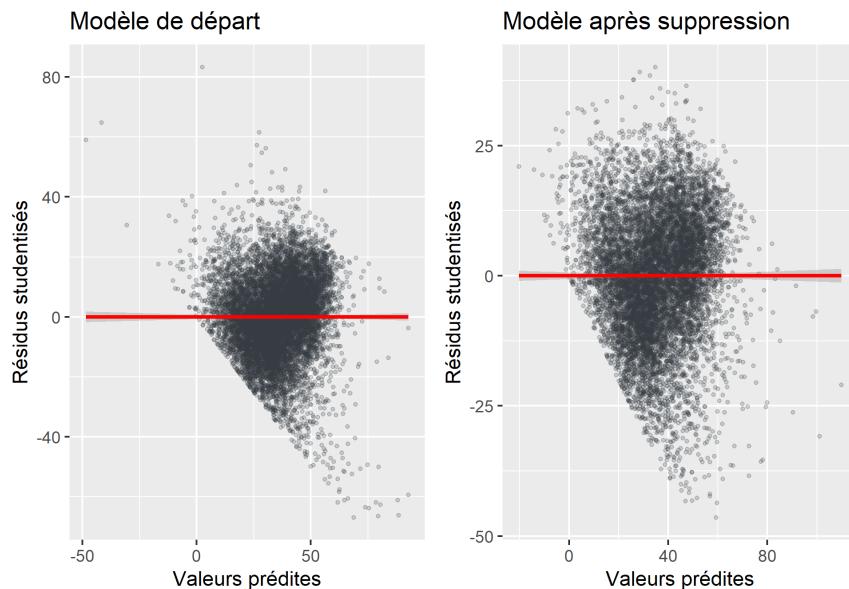


FIG. 7.17 : Amélioration de l'homoscédasticité des résidus

Finalement, il convient de comparer les coefficients de régression.

```

# Comparaison des coefficients
summary(modele3)

```

```

## 
## Call:

```

```

## lm(formula = VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 +
##     Pct_65P + Pct_MV + Pct_FR, data = DataFinal)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -66.848 -8.660  0.381  8.961 83.269
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.283e+01  1.001e+00  52.781 < 2e-16 ***
## log(HABHA)                -6.855e+00  1.683e-01 -40.730 < 2e-16 ***
## poly(AgeMedian, 2)1      1.198e+01  1.559e+01   0.769 0.441958
## poly(AgeMedian, 2)2     -2.861e+02  1.394e+01 -20.525 < 2e-16 ***
## Pct_014                  9.406e-01  3.126e-02  30.093 < 2e-16 ***
## Pct_65P                  3.062e-01  1.851e-02  16.546 < 2e-16 ***
## Pct_MV                  -3.630e-02  9.943e-03 -3.651 0.000262 ***
## Pct_FR                  -3.443e-01  1.103e-02 -31.212 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.57 on 10202 degrees of freedom
## Multiple R-squared:  0.4657, Adjusted R-squared:  0.4653
## F-statistic:  1270 on 7 and 10202 DF,  p-value: < 2.2e-16

```

```
summary(modele3b)
```

```

## 
## Call:
## lm(formula = VegPct ~ log(HABHA) + poly(AgeMedian, 2) + Pct_014 +
##     Pct_65P + Pct_MV + Pct_FR, data = DataSansOutliers)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -46.417 -7.734  0.456  8.290 40.085
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)              6.748e+01  9.869e-01  68.370 < 2e-16 ***
## log(HABHA)                -1.000e+01  1.720e-01 -58.167 < 2e-16 ***
## poly(AgeMedian, 2)1      4.357e+01  1.387e+01   3.142  0.00168 **
## poly(AgeMedian, 2)2     -3.564e+02  1.250e+01 -28.510 < 2e-16 ***
## Pct_014                  8.351e-01  2.870e-02  29.101 < 2e-16 ***
## Pct_65P                  2.271e-01  1.807e-02  12.566 < 2e-16 ***
## Pct_MV                  -8.517e-03  9.109e-03 -0.935  0.34976
## Pct_FR                  -2.924e-01  1.028e-02 -28.440 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.96 on 9917 degrees of freedom
## Multiple R-squared:  0.5687, Adjusted R-squared:  0.5684

```

```
## F-statistic: 1868 on 7 and 9917 DF, p-value: < 2.2e-16
```

7.7.4 Graphiques pour les effets marginaux

Tel que signalé ultérieurement, il est courant de représenter l'effet marginal d'une VI sur une VD, une fois contrôlées les autres VI. Pour ce faire, il est possible d'utiliser les packages `ggplot2` et `ggeffects`.

7.7.4.1 Effet marginal pour une variable continue

La syntaxe ci-dessous illustre comment obtenir un graphique pour nos quatre variables explicatives. Bien entendu, si le coefficient de régression est positif (comme pour les pourcentages de jeunes de moins de 15 ans et les personnes âgées), la pente est alors montante, et inversement descendante pour des coefficients négatifs (comme pour les personnes ayant déclaré appartenir à une minorité visible et les personnes à faible revenu). En outre, plus la valeur absolue du coefficient est forte, plus la pente est prononcée.

```
library(ggplot2)
library(ggeffects)
library(ggpubr)

# Création d'un DataFrame pour les valeurs prédites pour chaque VI continue
fitV1 <- ggpredict(modele3, terms = "Pct_014")
fitV2 <- ggpredict(modele3, terms = "Pct_65P")
fitV3 <- ggpredict(modele3, terms = "Pct_MV")
fitV4 <- ggpredict(modele3, terms = "Pct_FR")

# Construction des graphiques
G1 <- ggplot(fitV1, aes(x, predicted)) +
  # ligne de régression
  geom_line(color = 'red', size = 1) +
  # intervalle de confiance à 95 %
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3) +
  # Titres
  labs(y="valeur prédite Y", x = "Moins de 15 ans (%)")

G2 <- ggplot(fitV2, aes(x, predicted)) +
  geom_line(color = 'red', size = 1) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3) +
  labs(y="valeur prédite Y", x = "65 ans et plus (%)")

G3 <- ggplot(fitV3, aes(x, predicted)) +
  geom_line(color = 'blue', size = 1) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3) +
  labs(y="valeur prédite Y", x = "Minorités visibles (%)")

G4 <- ggplot(fitV4, aes(x, predicted)) +
  geom_line(color = 'blue', size = 1) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3) +
  labs(y="valeur prédite Y", x = "Personne à faible revenu (%)")

# Assemblage des graphiques
ggarrange(G1, G2, G3, G4, ncol =2, nrow =2)
```

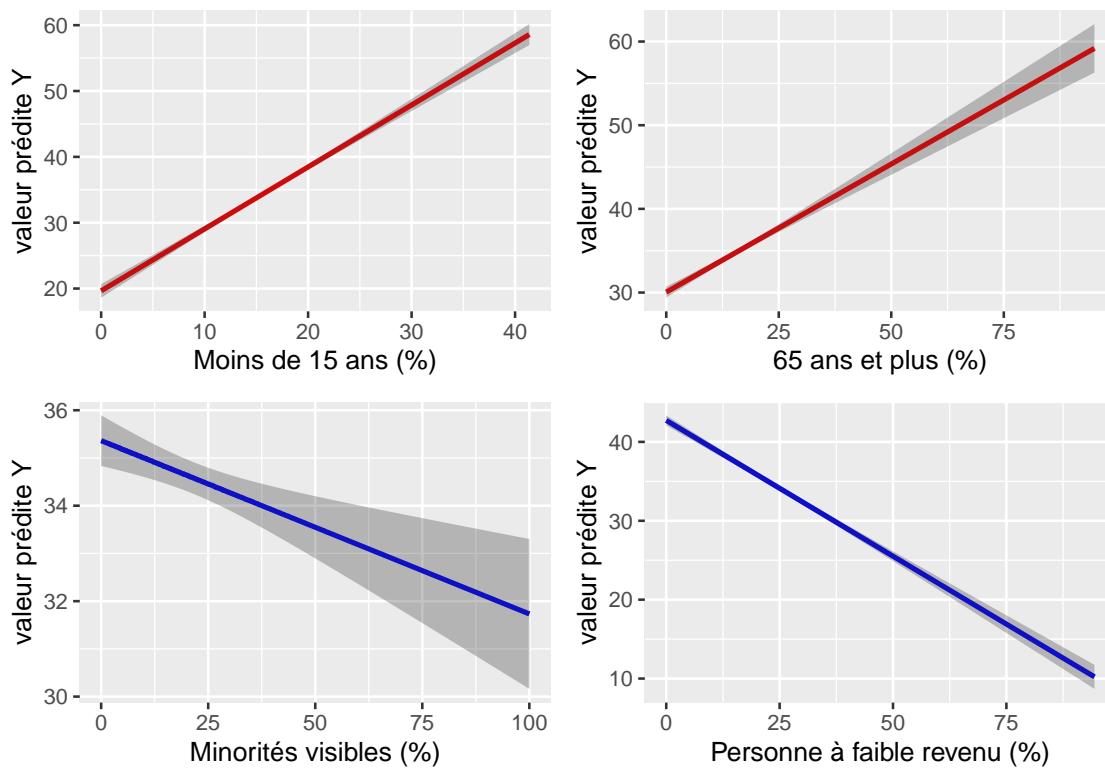


FIG. 7.18 : Effets marginaux pour des variables continues

7.7.4.2 Effet marginal pour une variable avec une fonction polynomiale d'ordre 2

```
library(ggplot2)
library(ggeffects)
library(ggpubr)

fitAgeMedian <- ggpredict(modele3, terms = "AgeMedian")

ggplot(fitAgeMedian, aes(x, predicted)) +
  geom_line(color = 'green', size = 1) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3) +
  labs(title="Variable sous forme polynomiale (ordre 2)",
       y="VD: valeur prédictive", x = "Âge médian des bâtiments")
```

7.7.4.3 Effet marginal pour une variable transformée en logarithme

```
fitHabHa <- ggpredict(modele3, terms = "HABHA")

ggplot(fitHabHa, aes(x, predicted)) +
  geom_line(color = 'blue', size = 1) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .3) +
  labs(y="VD: valeur prédictive", x = "Habitants km2")
```

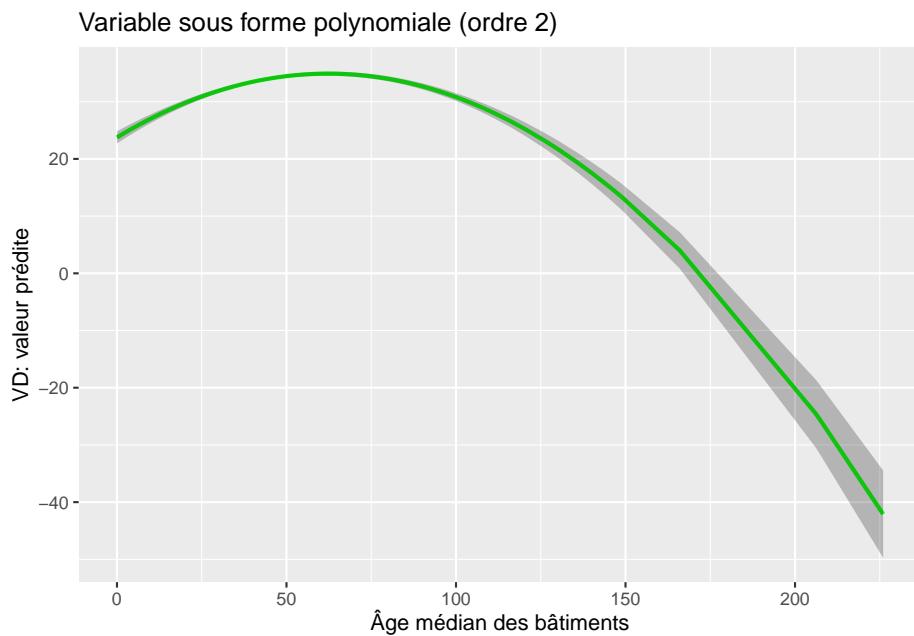


FIG. 7.19 : Effet marginal d'une variable avec un fonction polynomiale d'ordre 2

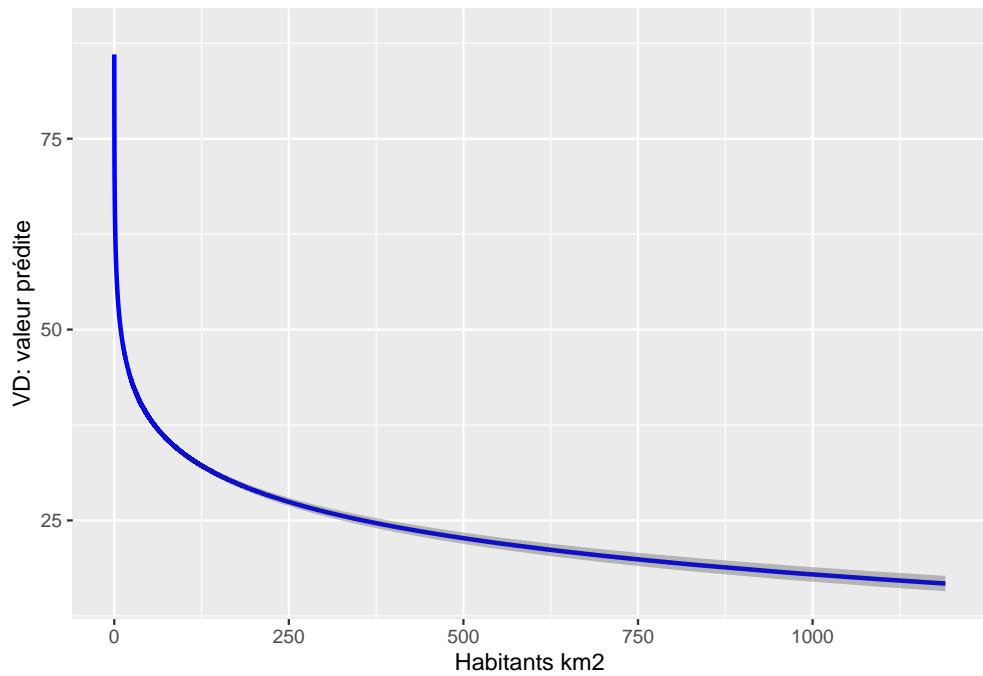


FIG. 7.20 : Effet du logarithme de la densité

7.7.4.4 Effet marginal pour une variable dichotomique

```
# Valeurs prédites selon le modèle avec la variable dichotomique
fitVilleMtl <- ggpredict(modele4, terms = "VilleMtl")

# Graphique
ggplot(fitVilleMtl, aes(x=x, y=predicted)) +
  geom_bar(stat = "identity", position = position_dodge(), fill="wheat") +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high), alpha = .9, position = position_dodge())+
  labs(title="Effet marginal de la ville de Montréal sur la végétation",
       x="Municipalités de la région de Montréal",
       y="Couverture végétation de l'îlot (%)")+
  scale_x_continuous(breaks=c(0,1),
                     labels = c("Autre municipalité", "Ville de Montréal"))
```

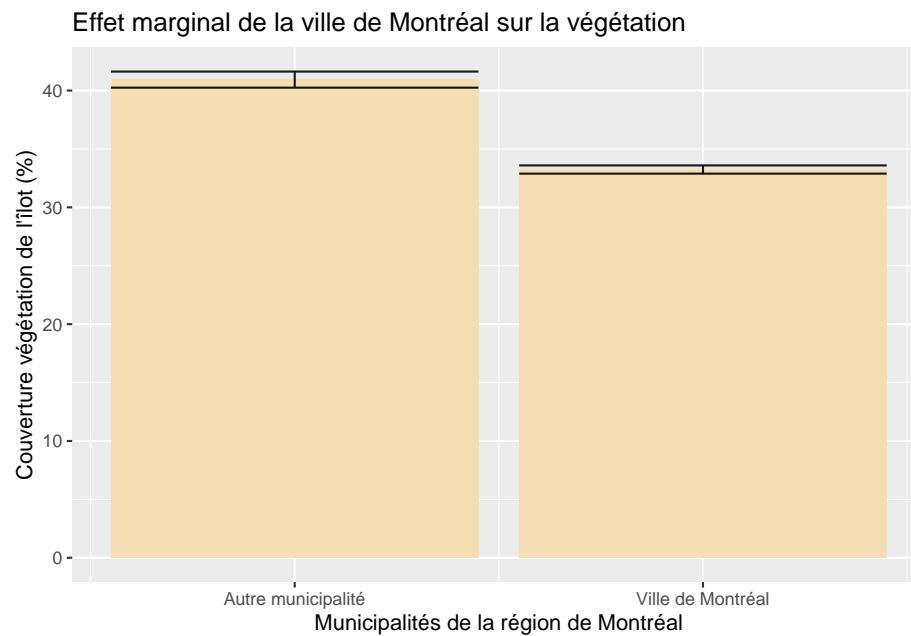


FIG. 7.21 : Effet marginal d'une variable dichotomique

7.7.4.5 Effet marginal pour une variable polytomique

```
# Valeurs prédites selon le modèle avec la variable polytomique
fitVilles <- ggpredict(modele5, terms = "Munic")

# Graphique
Graphique <- ggplot(fitVilles, aes(x=x, y=predicted)) +
  geom_bar(stat = "identity", position = position_dodge(), fill="wheat") +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high), alpha = .9, position = position_dodge())+
  labs(title="Effet marginal de la ville de Montréal sur la végétation",
       x="Municipalités de la région de Montréal",
       y="Couverture végétation de l'îlot (%)")
```

```
# Rotation du graphique
Graphique + coord_flip()
```

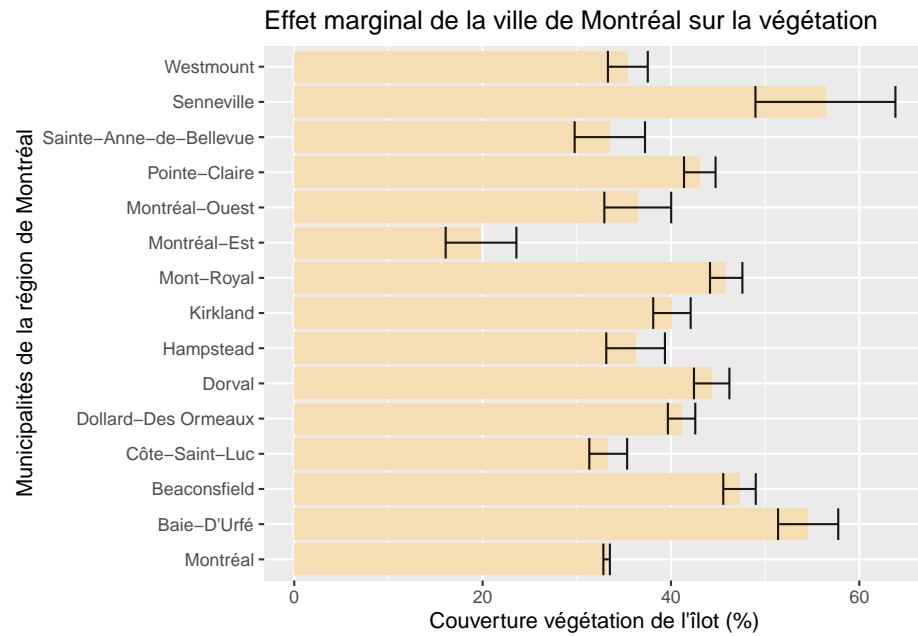


FIG. 7.22 : Effet marginal d'une variable polytomique

7.7.4.6 Effet marginal pour une variable d'interaction (deux VI continues)

```
library(metR) # pour ajouter des labels aux contours

df <- expand.grid(
  DistCBDkm = seq(0,33,0.1),
  Pct_FR = seq(0,95,1),
  HABHA = mean(DataFinal$HABHA),
  AgeMedian = mean(DataFinal$AgeMedian),
  Pct_014 = mean(DataFinal$Pct_014),
  Pct_65P = mean(DataFinal$Pct_65P),
  Pct_MV = mean(DataFinal$Pct_MV)
)

df$DistCBDkmX_Pct_FR <- df$DistCBDkm * df$Pct_FR
pred <- predict(modele6, newdata = df, se = T)
df$pred <- pred$fit
df$pred_se <- pred$se.fit
df$lower <- df$pred - 1.96 * df$pred_se
df$upper <- df$pred + 1.96 * df$pred_se

P1 <- ggplot(data = df) +
  geom_tile(aes(x = DistCBDkm, y = Pct_FR, fill = pred)) +
  stat_contour(aes(x = DistCBDkm, y = Pct_FR, z = pred),
               color = "black", linetype = "dashed") +
  geom_text_contour(aes(x = DistCBDkm, y = Pct_FR, z = pred), )+
```

```

scale_fill_viridis(discrete=FALSE) +
  labs(x = "Distance au centre-ville",
       y = "Pers à faible revenu (%)",
       fill = "",
       subtitle = "Prédiction")

P2 <- ggplot(data = df) +
  geom_tile(aes(x = DistCBDkm, y = Pct_FR, fill = lower)) +
  stat_contour(aes(x = DistCBDkm, y = Pct_FR, z = lower),
               color = "black", linetype = "dashed") +
  geom_text_contour(aes(x = DistCBDkm, y = Pct_FR, z = lower), )+
  scale_fill_viridis(discrete=FALSE) +
  labs(x = "Distance au centre-ville",
       y = "Pers à faible revenu (%)",
       fill = "",
       subtitle = "IC 2,5 %")

P3 <- ggplot(data = df) +
  geom_tile(aes(x = DistCBDkm, y = Pct_FR, fill = upper)) +
  stat_contour(aes(x = DistCBDkm, y = Pct_FR, z = upper),
               color = "black", linetype = "dashed") +
  geom_text_contour(aes(x = DistCBDkm, y = Pct_FR, z = upper), )+
  scale_fill_viridis(discrete=FALSE) +
  labs(x = "Distance au centre-ville",
       y = "Pers à faible revenu (%)",
       fill = "",
       subtitle = "IC 97,5 %")

ggarrange(P1,P2,P3,common.legend = F, ncol = 2, nrow = 2)

```

7.7.4.7 Effet marginal pour une variable d'interaction (une VI continue et une VI dichotomique)

```

df <- expand.grid(
  VilleMtl = c(0,1),
  Pct_FR = seq(0,95,1),
  HABHA = mean(DataFinal$HABHA),
  AgeMedian = mean(DataFinal$AgeMedian),
  Pct_014 = mean(DataFinal$Pct_014),
  Pct_65P = mean(DataFinal$Pct_65P),
  Pct_MV = mean(DataFinal$Pct_MV)
)
df$VilleMtlX_Pct_FR <- df$VilleMtl * df$Pct_FR

head(df, n=5)

##   VilleMtl Pct_FR   HABHA AgeMedian  Pct_014  Pct_65P  Pct_MV VilleMtlX_Pct_FR
## 1       0     0 87.7694  52.11494 15.89268 14.86761 20.96675      0
## 2       1     0 87.7694  52.11494 15.89268 14.86761 20.96675      0
## 3       0     1 87.7694  52.11494 15.89268 14.86761 20.96675      0
## 4       1     1 87.7694  52.11494 15.89268 14.86761 20.96675      1
## 5       0     2 87.7694  52.11494 15.89268 14.86761 20.96675      0

```

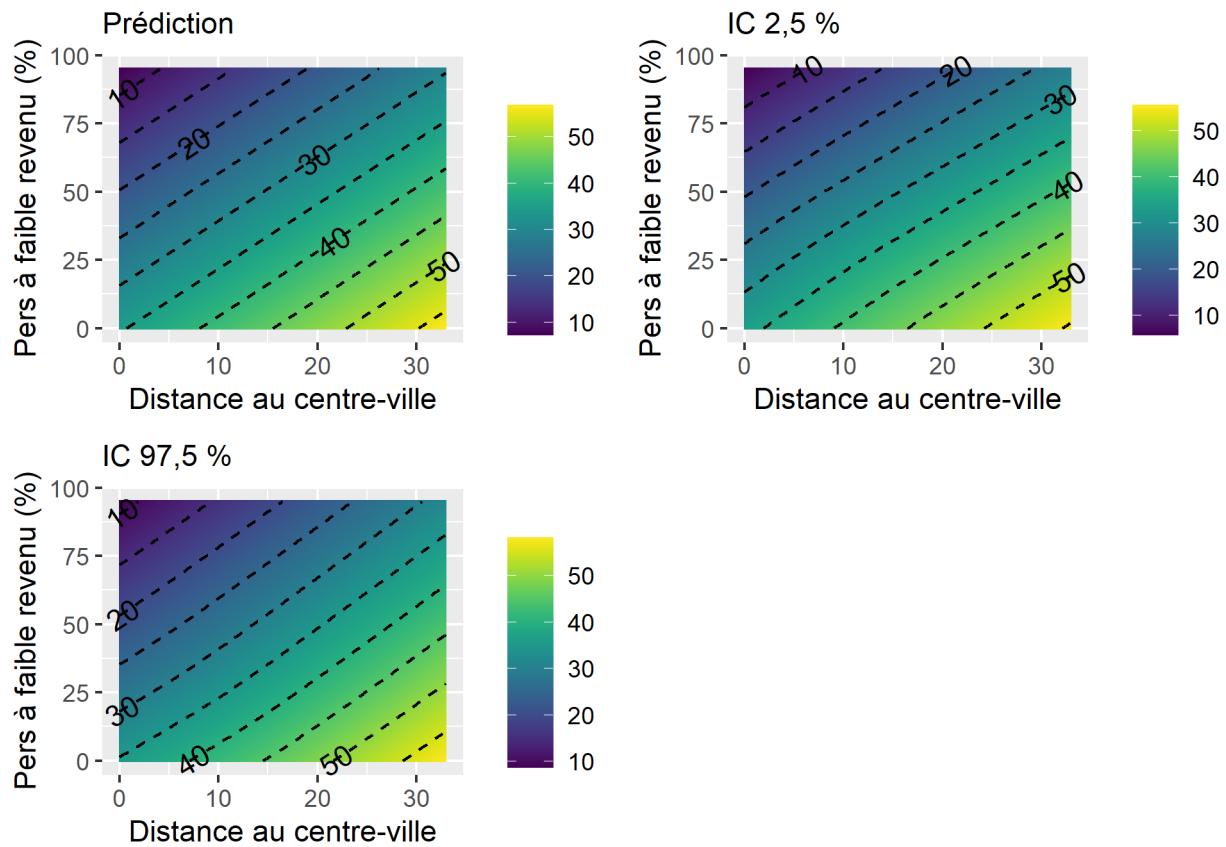


FIG. 7.23 : Effet marginal de l'interaction entre deux variables continues

```

pred <- predict(modele7, se = T, newdata = df)
df$pred <- pred$fit
df$upper <- df$pred + 1.96*pred$se.fit
df$lower <- df$pred - 1.96*pred$se.fit

df$VilleMtl_str <- ifelse(df$VilleMtl==0,"Autre municipalité","Ville de Montréal")
DataFinal$VilleMtl_str <- ifelse(DataFinal$VilleMtl==0,"Autre municipalité","Ville de Montréal")

cols <- c("Autre municipalité" ="#1d3557" , "Ville de Montréal"="#e63946")

ggplot(data = df) +
  geom_point(data = DataFinal, mapping = aes(x = Pct_FR, y = VegPct, color = VilleMtl_str),
             size = 0.2, alpha = 0.2) +
  geom_ribbon(aes(x = Pct_FR, ymin = lower, ymax = upper, group = VilleMtl_str),
              fill = rgb(0.1,0.1,0.1,0.4)) +
  geom_path(aes(x = Pct_FR, y = pred, color = VilleMtl_str), size = 1) +
  scale_colour_manual(values = cols) +
  labs(x = "Personnes à faible revenu (%)",
       y = "Densité de végétation prédictive (%)",
       color = "")

```

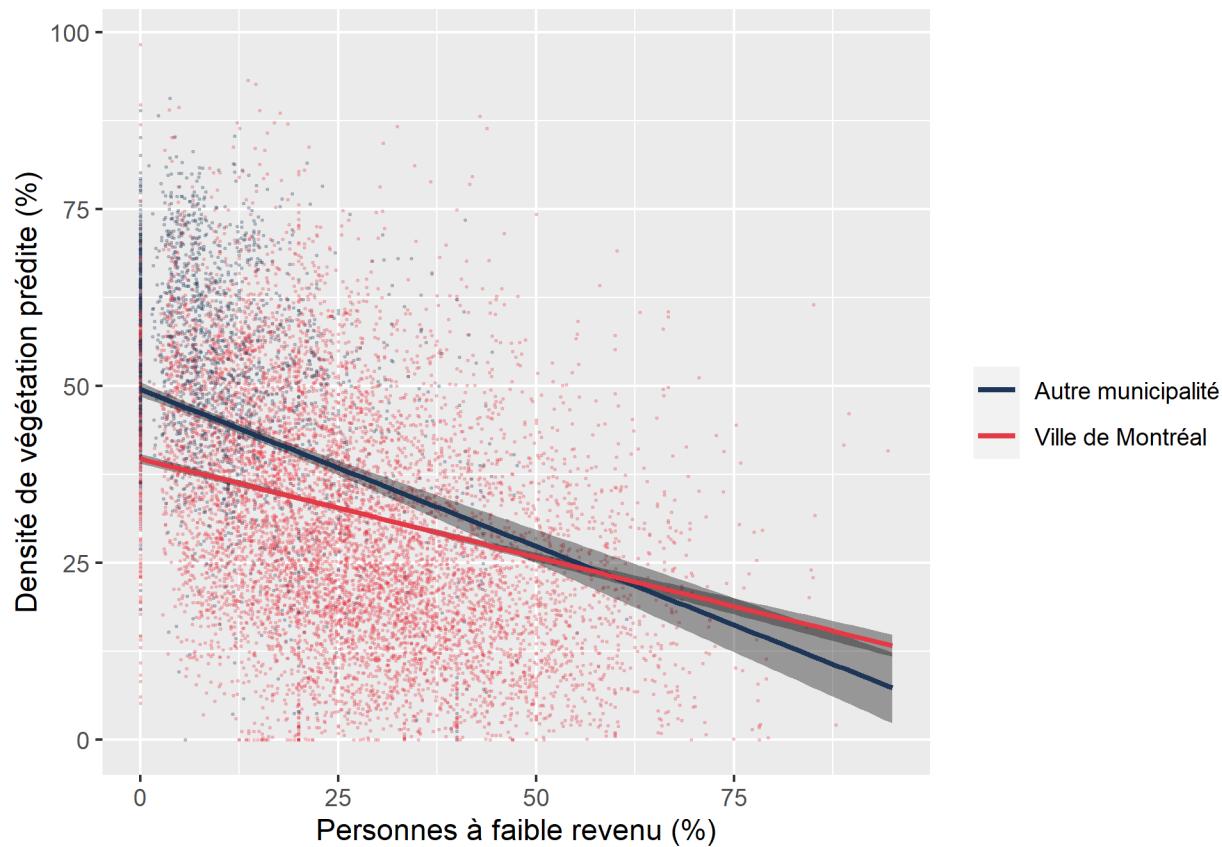


FIG. 7.24 : Graphique de l'effet marginal de l'interaction entre une variable quantitative et qualitative

7.8 Quiz de révision du chapitre

Questions

- **Quels modèles sont imbriqués ?**
 - Modèle A : $Y = X_1 + X_2 + X_3$ / Modèle B : $Y = X_1 + X_2 + X_3 + X_4 + X_5$
 - Modèle A : $Y = X_1 + X_4 + X_5$ / Modèle B : $Y = X_1 + X_2 + X_3 + X_7 + X_8$

Relisez au besoin la section [7.3.2](#).
- **Quelle mesure relative à la qualité d'ajustement du modèle indique la proportion de la variance de la variable dépendante expliquée par les variables indépendantes du modèle ?**
 - Statistique de Fisher
 - Coefficient de détermination (R^2)
 - L'erreur quadratique moyenne (RMSE)
 - Le coefficient de régression standardisé

Relisez au besoin la section [7.3.1](#).
- **Le R^2 ajusté permet de comparer des modèles avec des nombres de variables indépendantes et/ou d'observations différents**
 - Vrai
 - Faux

Relisez au besoin le début de la section 7.3.2.

- **Comment repérer les variables les plus importantes du modèle ?**

- Coefficients de régression
- Coefficients de régression standardisés
- Erreurs types

Relisez au besoin la section 7.4.2.

- **Comment évaluer la significativité des coefficients ?**

- Valeur de t
- Valeur de p
- R²
- F de Fisher

Relisez au besoin la section 7.4.3.

- **Pour un nombre très élevé d'observations, quelle affirmation est vraie pour les valeurs de t et de p ?**

- 1,96 (p <= 0,05); 2,58 (p <= 0,01); 3,29 (p <= 0,001)
- 1,96 (p <= 0,001); 2,58 (p <= 0,01); 3,29 (p <= 0,05)
- 2,96 (p <= 0,05); 3,58 (p <= 0,01); 4,29 (p <= 0,001)

Relisez au besoin la section 7.4.3.

- **Une variable indépendante dont l'intervalle de confiance à 95 % du coefficient de régression est de [-15,06; 28,17] est-elle significatif au seuil de p = 0,05 ?**

- Vrai
- Faux

Relisez au besoin la section 7.4.4.

- **Comment poser un diagnostic sur un modèle de régression linéaire ?**

- Nombre d'observations
- Normalité des résidus
- Linéarité et homoscédasticité des résidus
- Absence de multicolinéarité excessive
- Absence d'observations trop influentes
- Nombre de variables indépendantes

Relisez le deuxième encadré à la section 7.6.

Réponses

- Quels modèles sont imbriqués ?
 - Modèle A : Y = X₁ + X₂ + X₃ / Modèle B : Y = X₁ + X₂ + X₃ + X₄ + X₅
- Quelle mesure relative à la qualité d'ajustement du modèle indique la proportion de la variance de la variable dépendante expliquée par les variables indépendantes du modèle ?
 - Coefficient de détermination (R²)
- Le R² ajusté permet de comparer des modèles avec des nombres de variables indépendantes et/ou d'observations différents
 - Vrai
- Comment repérer les variables les plus importantes du modèle ?

- Coefficients de régression standardisés
- Comment évaluer la significativité des coefficients ?
 - Valeur de t
 - Valeur de p
- Pour un nombre très élevé d'observations, quelle affirmation est vraie pour les valeurs de t et de p ?
 - 1,96 ($p \leq 0,05$) ; 2,58 ($p \leq 0,01$) ; 3,29 ($p \leq 0,001$)
- Une variable indépendante dont l'intervalle de confiance à 95 % du coefficient de régression est de [-15,06 ; 28,17] est-elle significatif au seuil de $p = 0,05$?
 - Faux
- Comment poser un diagnostic sur un modèle de régression linéaire ?
 - Nombre d'observations
 - Normalité des résidus
 - Linéarité et homoscédasticité des résidus
 - Absence de multicolinéarité excessive
 - Absence d'observations trop influentes

Chapitre 8

Régressions linéaires généralisées (GLM)

Dans ce chapitre, nous présentons les modèles linéaires généralisés plus communément appelés GLM (*generalized linear models* en anglais). Il s'agit d'une extension directe du modèle de régression linéaire multiple (LM) basé sur la méthode des moindres carrés ordinaires, décrite dans le chapitre précédent. Pour aborder cette section sereinement, il est important d'avoir bien compris le concept de distribution présenté dans la section 2.4. À la fin de cette section, vous serez en mesure de :

- comprendre la distinction entre un modèle LM classique et un GLM;
- identifier les composantes d'un GLM;
- interpréter les résultats d'un GLM;
- effectuer les diagnostics d'un GLM.



Dans ce chapitre, nous utilisons principalement les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2`, le seul, l'unique!
 - * `ggpubr` pour combiner des graphiques et réaliser des diagrammes.
- Pour ajuster des modèles GLM :
 - * `VGAM` et `gamlss` offrent tous les deux un très large choix de distributions et de fonctions de diagnostic, mais nécessitent souvent un peu plus de code.
 - * `mgcv` offre moins de distributions que les deux précédents, mais est plus simple d'utilisation.
- Pour analyser des modèles GLM :
 - `car` essentiellement pour la fonction `vif`.
 - `DHARMA` pour le diagnostic des résidus simulés.
 - `ROCR` et `caret` pour l'analyse de la qualité d'ajustement de modèles pour des variables qualitatives.
 - `AER` pour des tests de surdispersion.
 - `fitdistrplus` pour ajuster des distributions à des données.
 - `LaplaceDemon` pour manipuler certaines distributions.
 - `sandwich` pour générer des erreurs standards robustes pour le modèle GLM logistique binomial.

8.1 Qu'est qu'un modèle GLM ?

Nous avons vu qu'une régression linéaire multiple (LM) ne peut être appliquée que si la variable dépendante analysée est continue et si elle est normalement distribuée, une fois les variables indépendantes contrôlées. Il s'agit d'une limite très importante puisqu'elle ne peut être utilisée pour modéliser et prédire des variables binaires, multinomiales, de comptage, ordinaires ou plus simplement des données anormalement distribuées. Une seconde limite importante des LM est que l'influence des variables indépendantes

sur la variable dépendante ne peut être que linéaire. L'augmentation d'une unité de X conduit à une augmentation (ou diminution) de β (coefficients de régression) unités de Y , ce qui n'est pas toujours représentatif des phénomènes étudiés. Afin de dépasser ces contraintes, Nelder et Wedderburn (1972) ont proposé une extension des modèles LM, soit les modèles linéaires généralisés (GLM).

8.1.1 Formulation d'un GLM

Puisqu'un modèle GLM est une extension des modèles LM, il est possible de traduire un modèle LM sous forme d'un GLM. Nous utilisons ce point de départ pour détailler la morphologie d'un GLM. Nous avons vu dans la section précédente qu'un modèle LM est formulé de la façon suivante (notation matricielle) :

$$Y = \beta_0 + X\beta + \epsilon \quad (8.1)$$

Avec β_0 la constante (*intercept* en anglais) et β un vecteur de coefficients de régression pour les k variables indépendantes (X).

D'après cette formule, nous modélisons la variable Y avec une équation de régression linéaire et un terme d'erreur que nous estimons être normalement distribué. Nous pouvons reformuler ce simple LM sous forme d'un GLM avec l'écriture suivante :

$$\begin{aligned} Y &\sim Normal(\mu, \sigma) \\ g(\mu) &= \beta_0 + \beta X \\ g(x) &= x \end{aligned} \quad (8.2)$$

Pas de panique ! Cette écriture se lit comme suit : la variable Y est issue d'une distribution normale ($Y \sim Normal$) avec deux paramètres : μ (sa moyenne) et σ (son écart-type). μ varie en fonction d'une équation de régression linéaire ($\beta_0 + \beta X$) transformée par une fonction de lien g (détailée plus loin). Dans ce cas précis, la fonction de lien est appelée fonction identitaire puisqu'elle n'applique aucune transformation ($g(x) = x$). Notez ici que le second paramètre de la distribution normale σ (paramètre de dispersion) a une valeur fixe et ne dépend donc pas des variables indépendantes à la différence de μ . Dans ce modèle spécifiquement, les paramètres à estimer sont σ , β_0 et β . Notez que dans la notation traditionnelle, la fonction de lien est appliquée au paramètre modélisé. Il est possible de renverser cette notation en utilisant la réciproque (g') de la fonction de lien (g) :

$$g(\mu) = \beta_0 + \beta X \iff \mu = g'(\beta_0 + \beta X) \text{ si } g'(g(x)) = x \quad (8.3)$$

Dans un modèle GLM, la distribution attendue de la variable Y est déclarée de façon explicite ainsi que la façon dont nos variables indépendantes conditionnent cette distribution. Ici, c'est la moyenne (μ) de la distribution qui est modélisée, nous nous intéressons ainsi au changement moyen de Y provoqué par les variables X .

Avec cet exemple, nous voyons les deux composantes supplémentaires d'un modèle GLM :

- La distribution supposée de la variable Y conditionnée par les variables X (ici, la distribution normale).
- Une fonction de lien associant l'équation de régression formée par les variables indépendantes et un paramètre de la distribution retenue (ici, la fonction identitaire et le paramètre μ).

Notez également que l'estimation des paramètres d'un modèle GLM (ici, β_0 , βX et σ) ne se fait plus avec la méthode des moindres carrés ordinaires utilisée pour les modèles LM. À la place, la méthode par maximum de vraisemblance (*maximum likelihood*) est la plus souvent utilisée, mais certains *packages* utilisent également la méthode des moments (*method of moments*). Ces deux méthodes nécessitent des échantillons

plus grands que la méthode des moindres carrés. Dans le cas spécifique d'un modèle GLM utilisant une distribution normale, la méthode des moindres carrés et la méthode par maximum de vraisemblance produisent les mêmes résultats.

8.1.2 Autres distributions et rôle de la fonction de lien

À première vue, il est possible de se demander pourquoi ajouter ces deux éléments puisqu'ils ne font que complexifier le modèle. Pour mieux saisir la pertinence des GLM, prenons un exemple appliqué au cas d'une variable binaire. Admettons que nous souhaitons modéliser / prédire la probabilité qu'une personne à vélo décède lors d'une collision avec un véhicule motorisé. Notre variable dépendante est donc binaire (0 = survie, 1 = décès) et nous souhaitons la prédire avec trois variables continues que sont : la vitesse de déplacement du ou de la cycliste (x_1), la vitesse de déplacement du véhicule (x_2) et la masse du véhicule (x_3). Puisque la variable Y n'est pas continue, il serait absurde de supposer qu'elle est issue d'une distribution normale. Il serait plus logique de penser qu'elle provient d'une distribution de Bernoulli (pour rappel, une distribution de Bernoulli permet de modéliser un phénomène ayant deux issues possibles comme un lancer de pièce de monnaie, section 2.4). Plus spécifiquement, nous pourrions formuler l'hypothèse que nos trois variables x_1 , x_2 et x_3 influencent le paramètre p (la probabilité d'occurrence de l'événement) d'une distribution de Bernoulli. À partir de ces premières hypothèses, nous pouvons écrire le modèle suivant :

$$\begin{aligned} Y &\sim \text{Bernoulli}(p) \\ g(p) &= \beta_0 + \beta X \\ g(x) &= x \end{aligned} \tag{8.4}$$

Toutefois, le résultat n'est pas entièrement satisfaisant. En effet, p est une probabilité et, par nature, ce paramètre doit être compris entre 0 et 1 (entre 0 et 100 % de « chances de décès », ni plus ni moins). L'équation de régression que nous utilisons actuellement peut produire des résultats compris entre $-\infty$ et $+\infty$ pour p puisque rien ne contraint la somme $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ à être comprise entre 0 et 1. Il est possible de visualiser le problème soulevé par cette situation avec les figures suivantes. Admettons que nous avons observé une variable Y binaire et que nous savons qu'elle est influencée par une variable X qui, plus elle augmente, plus les chances que Y soit 1 augmentent (figure 8.1).

Si nous utilisons l'équation de régression actuelle, cela revient à trouver la droite la mieux ajustée passant dans ce nuage de points (figure 8.2).

Ce modèle semble bien cerner l'influence positive de X sur Y , mais la droite est au final très éloignée de chaque point, indiquant un faible ajustement du modèle. De plus, la droite prédit des probabilités négatives lorsque X est inférieur à -2,5 et des probabilités supérieures à 1 quand X est supérieur à 1. Elle est donc loin de bien représenter les données.

C'est ici qu'intervient la fonction de lien. La fonction identitaire que nous avons utilisée jusqu'ici n'est pas satisfaisante, nous devons la remplacer par une fonction qui conditionnera la somme $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ pour donner un résultat entre 0 et 1. Une candidate toute désignée est la fonction *sigmoidale*, plus souvent appelée la fonction *logistique* !

$$\begin{aligned} Y &\sim \text{Bernoulli}(p) \\ S(p) &= \beta_0 + \beta X \\ S(x) &= \frac{e^x}{e^x + 1} \end{aligned} \tag{8.5}$$

La fonction logistique prend la forme d'un S . Plus la valeur entrée dans la fonction est grande, plus le résultat produit par la fonction est proche de 1 et inversement. Si nous reprenons l'exemple précédent,



FIG. 8.1 : Exemple de données issues d'une distribution de Bernoulli

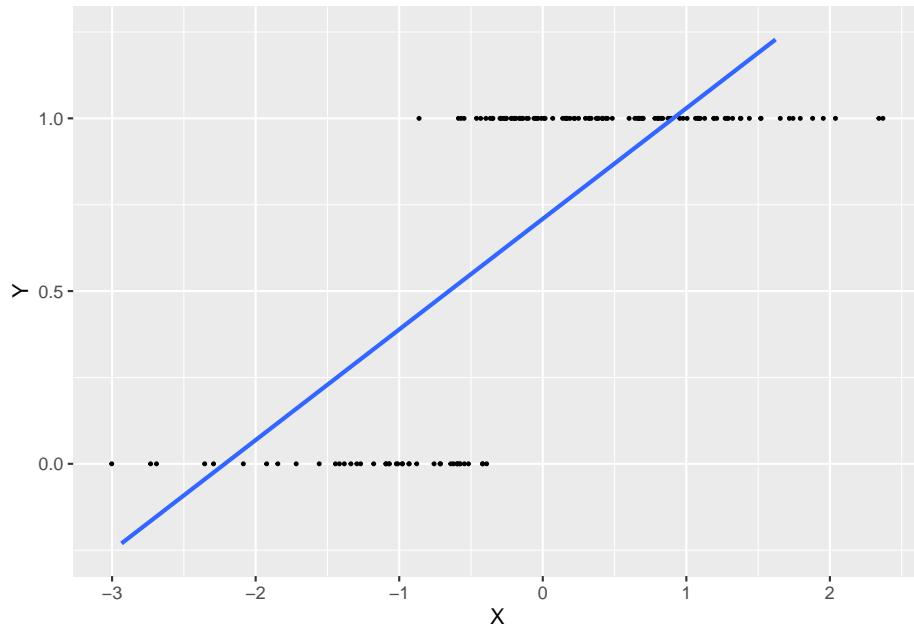


FIG. 8.2 : Ajustement d'une droite de régression aux données issues d'une distribution de Bernoulli

nous obtenons le modèle illustré à la figure 8.3.

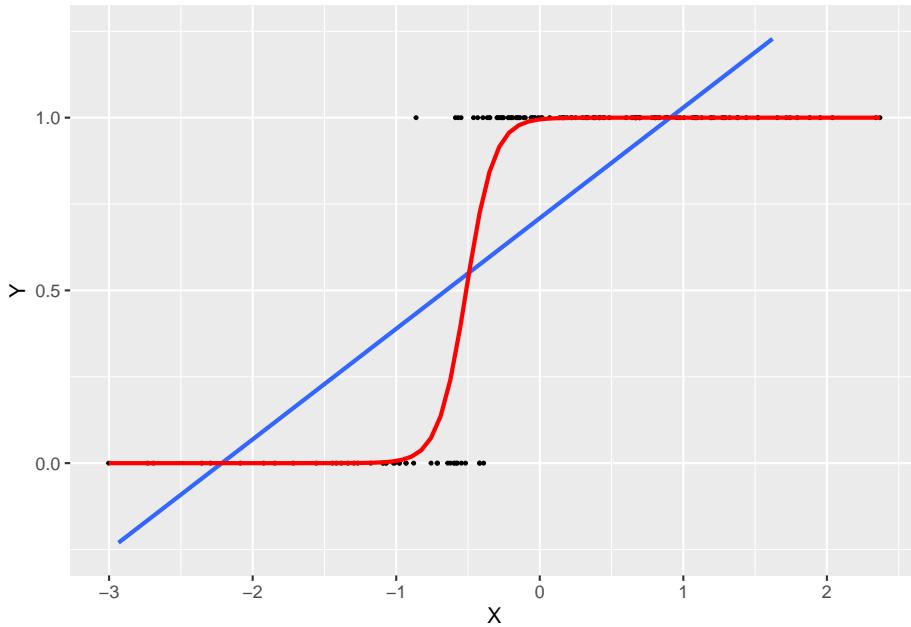


FIG. 8.3 : Utilisation de la fonction de lien logistique

Une fois cette fonction insérée dans le modèle, nous constatons qu'une augmentation de la somme $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$ conduit à une augmentation de la probabilité p et inversement, et que cet effet est non linéaire. Nous avons donc maintenant un GLM permettant de prédire la probabilité d'un décès lors d'un accident en combinant une distribution et une fonction de lien adéquates.

8.1.3 Conditions d'application

La famille des GLM englobe de (très) nombreux modèles du fait de la diversité de distributions existantes et des fonctions de liens utilisables. Cependant, certaines combinaisons sont plus souvent utilisées que d'autres. Nous présentons donc dans les prochaines sections les modèles GLM les plus communs. Les conditions d'application varient d'un modèle à l'autre, il existe cependant quelques conditions d'application communes à tous ces modèles :

- l'indépendance des observations (et donc des erreurs);
- l'absence de valeurs aberrantes / fortement influentes;
- l'absence de multicolinéarité excessive entre les variables indépendantes.

Ces trois conditions sont également valables pour les modèles LM tel qu'abordés dans le chapitre 7. La distance de *Cook* peut ainsi être utilisée pour détecter les potentielles valeurs aberrantes et le facteur d'inflation de la variance (*VIF*) pour détecter la multicolinéarité. Les conditions d'application particulières sont détaillées dans les sections dédiées à chaque modèle.

8.1.4 Résidus et déviance

Dans la section sur la régression linéaire simple, nous avons présenté la notion de résidu, soit l'écart entre la valeur observée (réelle) de Y et la valeur prédictive par le modèle. Pour un modèle GLM, ces résidus traditionnels (aussi appelés résidus naturels) ne sont pas très informatifs si la variable à modéliser est

binaire, multinomiale ou même de comptage. Lorsque l'on travaille avec des GLM, nous préférerons utiliser trois autres formes de résidus, soit les résidus de Pearson, les résidus de déviance et les résidus simulés.

Les résidus de Pearson sont une forme ajustée des résidus classiques, obtenus par la division des résidus naturels par la racine carrée de la variance modélisée. Leur formule varie donc d'un modèle à l'autre puisque l'expression de la variance change en fonction de la distribution du modèle. Pour un modèle GLM gaussien, elle s'écrit :

$$r_i = \frac{y_i - \mu_i}{\sigma} \quad (8.6)$$

Pour un modèle GLM de Bernoulli, elle s'écrit :

$$r_i = \frac{y_i - p_i}{\sqrt{p_i(1-p_i)}} \quad (8.7)$$

avec μ_i et p_i les prédictions du modèle pour l'observation i .

Les résidus de déviance sont basés sur le concept de *likelihood* présenté dans la section 2.5.4.2. Pour rappel, le *likelihood*, ou la vraisemblance d'un modèle, correspond à la probabilité conjointe d'avoir observé les données Y selon le modèle étudié. Pour des raisons mathématiques (voir section 2.5.4.2), le *log likelihood* est plus souvent calculé. Plus cette valeur est forte, moins le modèle se trompe. Cette interprétation est donc inverse à celle des résidus classiques, c'est pourquoi le *log likelihood* est généralement multiplié par -2 pour retrouver une interprétation intuitive. Ainsi, pour chaque observation i , nous pouvons calculer :

$$d_i = -2 \times \log(P(y_i|M_e)) \quad (8.8)$$

avec d_i le résidu de déviance et $P(y_i|M_e)$ la probabilité d'avoir observé la valeur y_i selon le modèle étudié (M_e).

La somme de tous ces résidus est appelée la déviance totale du modèle.

$$D(M_e) = \sum_{i=1}^n -2 \times \log(P(y_i|M_e)) \quad (8.9)$$

Il s'agit donc d'une quantité représentant à quel point le modèle est erroné vis-à-vis des données. Notez qu'en tant que telle, la déviance n'a pas d'interprétation directe en revanche, elle est utilisée pour calculer des mesures d'ajustement des modèles GLM.

Les résidus simulés sont une avancée récente dans le monde des GLM, ils fournissent une définition et une interprétation harmonisée des résidus pour l'ensemble des modèles GLM. Dans la section sur les LM (voir section 7.2.2), nous avons vu comment interpréter les graphiques des résidus pour détecter d'éventuels problèmes dans le modèle. Cependant, cette technique est bien plus compliquée à mettre en œuvre pour les GLM puisque la forme attendue des résidus varie en fonction de la distribution choisie pour modéliser Y . La façon la plus efficace de procéder est d'interpréter les graphiques des résidus simulés qui ont la particularité d'être **identiquement distribués, quel que soit le modèle GLM construit**. Ces résidus simulés sont compris entre 0 et 1 et sont calculés de la manière suivante :

- À partir du modèle GLM construit, simuler S fois (généralement 1 000) une variable Y' avec autant d'observation (n) que Y . Cette variable simulée est une combinaison de la prédiction du modèle (coefficients et variables indépendantes) et de sa dispersion (variance). Ces simulations représentent des variations vraisemblables de la variable Y si le modèle est correctement spécifié. En d'autres termes, si le modèle représente bien le phénomène à l'origine de la variable Y , alors les simulations Y' issues du modèle devraient être proches de la variable Y originale. Pour une explication plus

détaillée de ce que signifie simuler des données à partir d'un modèle, référez-vous au *bloc attention* intitulé *Distinction entre simulation et prédition* dans la section 8.1.5.2.

- Pour chaque observation, nous obtenons ainsi S valeurs formant une distribution Ds_i , soit les valeurs simulées par le modèle pour cette observation.
- Pour chacune de ces distributions, nous calculons la probabilité cumulative d'observer la vraie valeur Y_i d'après la distribution Ds_i . Cette valeur est comprise entre 0 (toutes les valeurs simulées sont plus grandes que Y_i) et 1 (toutes les valeurs simulées sont inférieures à Y_i).

Si le modèle est correctement spécifié, le résultat attendu est que la distribution de ces résidus est uniforme. En effet, il y a autant de chances que les simulations produisent des résultats supérieurs ou inférieurs à Y_i si le modèle représente bien le phénomène (Dunn et Smyth 1996; Gelman et Hill 2006). Si la distribution des résidus ne suit pas une loi uniforme, cela signifie que le modèle échoue à reproduire le phénomène à l'origine de Y , ce qui doit nous alerter sur sa pertinence.

8.1.5 Vérification l'ajustement

Il existe trois façons de vérifier l'ajustement d'un modèle GLM :

- utiliser des mesures d'ajustement (AIC, pseudo-R², déviance expliquée, etc.);
- comparer les distributions de la variable originale et celle des prédictions;
- comparer les prédictions du modèle avec les valeurs originales.

Notez d'emblée que vérifier la qualité d'ajustement d'un modèle (ajustement aux données originales) ne revient pas à vérifier la validité d'un modèle (respect des conditions d'application). Cependant, ces deux éléments sont généralement liés, car un modèle mal ajusté a peu de chances d'être valide et inversement.

8.1.5.1 Mesures d'ajustement

Les mesures d'ajustement sont des indicateurs plus ou moins arbitraires dont le principal intérêt est de faciliter la comparaison entre plusieurs modèles similaires. Il est nécessaire de les reporter, car dans certains cas, ils peuvent indiquer que des modèles sont très mal ajustés.

8.1.5.1.1 Déviance expliquée

Rappelons que la déviance d'un modèle est une quantité représentant à quel point le modèle est erroné. L'objectif de l'indicateur de la déviance expliquée est d'estimer le pourcentage de la déviance maximale observable dans les données que le modèle est parvenu à expliquer. La déviance maximale observable dans les données est obtenue en utilisant la déviance totale du modèle nul (notée M_n , soit un modèle dans lequel aucune variable indépendante n'est ajoutée et ne comportant qu'une constante). Cette déviance est maximale puisqu'aucune variable indépendante n'est présente dans le modèle. Nous calculons ensuite le pourcentage de cette déviance totale qui a été contrôlée par le modèle étudié (M_e).

$$\text{déviance expliquée} = \frac{D(M_n) - D(M_e)}{D(M_n)} = 1 - \frac{D(M_e)}{D(M_n)} \quad (8.10)$$

Il s'agit donc d'un simple calcul de pourcentage entre la déviance maximale ($D(M_n)$) et la déviance expliquée par le modèle étudié ($D(M_n) - D(M_e)$). Cet indicateur est compris entre 0 et 1 : plus il est petit, plus la capacité de prédiction du modèle est faible. Attention, cet indicateur ne tient pas compte de la complexité du modèle. Ajouter une variable indépendante supplémentaire ne fait qu'augmenter la déviance expliquée, ce qui ne signifie pas que la complexification du modèle soit justifiée (*voir l'encadré sur le principe de parcimonie*, section 7.3.2).

8.1.5.1.2 Pseudo-R²

Le R² est une mesure d'ajustement représentant la part de la variance expliquée dans un modèle linéaire classique. Cette mesure n'est pas directement transposable au cas des GLM puisqu'ils peuvent être appliqués à des variables non continues et anormalement distribuées. Toutefois, il existe des mesures semblables appelées pseudo-R², remplissant un rôle similaire. Notez cependant qu'ils ne peuvent pas être interprétés comme le R² classique (d'une régression linéaire multiple) : **ils ne représentent pas la part de la variance expliquée**. Ils sont compris dans l'intervalle 0 et 1 ; plus leurs valeurs s'approchent de 1, plus le modèle est ajusté.

TAB. 8.1 : Principaux pseudo- R^2

Nom	Formule	Commentaire
McFadden	$1 - \frac{\text{loglike}(M_e)}{\text{loglike}(M_n)}$	Le rapport des <i>loglikelihood</i> , très proche de la déviance expliquée.
McFadden ajusté	$1 - \frac{\text{loglike}(M_e) - K}{\text{loglike}(M_n)}$	Version ajustée du R ² de McFadden tenant compte du nombre de paramètres (k) dans le modèle.
Efron	$1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$	Rapport entre la somme des résidus classiques au carré (numérateur) et de la somme des écarts au carré à la moyenne (dénominateur). Notez que pour un GLM gaussien, ce pseudo-R ² est identique au R ² classique.
Cox & Snell	$1 - e^{-\frac{2}{n}(\text{loglike}(M_e) - \text{loglike}(M_n))}$	Transformation de la déviance afin de la mettre sur une échelle de 0 à 1 (mais ne pouvant atteindre exactement 1).
Nagelkerke	$\frac{1 - e^{-\frac{2}{n}(\text{loglike}(M_e) - \text{loglike}(M_n))}}{1 - e^{-\frac{2 * \text{loglike}(M_n)}{n}}}$	Ajustement du R ² de Cox et Snell pour que l'échelle de valeurs possibles puisse comporter 1 (attention, car les valeurs de ce R ² tendent à être toujours plus fortes que les autres).

En dehors du pseudo-R² de McFadden ajusté, aucune de ces mesures ne tient compte de la complexité du modèle. Il est cependant important de les reporter, car des valeurs très faibles indiquent vraisemblablement un modèle avec une moindre capacité informative. À l'inverse, des valeurs trop fortes pourraient indiquer un problème de surajustement (**voir encadré sur le principe de parcimonie**, section 7.3.2).

8.1.5.1.3 Critère d'information d'Akaike (AIC)

Probablement l'indicateur le plus répandu, sa formule est relativement simple, car il s'agit seulement d'un ajustement de la déviance :

$$AIC = D(M_e) + 2K \quad (8.11)$$

avec K le nombre de paramètres à estimer dans le modèle (coefficients, paramètres de distribution, etc.).

L'AIC n'a pas d'interprétation directe, mais permet de comparer deux modèles imbriqués (voir section 7.3.2). Plus l'AIC est petit, mieux le modèle est ajusté. L'idée derrière cet indicateur est relativement simple. Si la déviance D est grande, alors le modèle est mal ajusté. Ajouter des paramètres (des coefficients pour de nouvelles variables X , par exemple) ne peut que réduire D , mais cette réduction n'est pas forcément suffisamment grande pour justifier la complexification du modèle. L'AIC pondère donc D en lui ajoutant 2 fois le nombre de paramètres du modèle. Un modèle plus simple (avec moins de paramètres) parvenant à une même déviance est préférable à un modèle complexe (principe de parcimonie ou du rasoir d'Ockham), ce que permet de « quantifier » l'AIC. Attention, l'AIC **ne peut pas être utilisé pour comparer des modèles non imbriqués**. Notez que d'autres indicateurs similaires comme le WAIC, le BIC et le DIC sont utilisés dans un contexte d'inférence bayésienne. Retenez simplement que ces indicateurs sont conceptuellement proches du AIC et s'interprètent (à peu de choses près) de la même façon.

8.1.5.2 Comparaison des distributions originales et prédictes

Une façon rapide de vérifier si un modèle est mal ajusté est de comparer la forme de la distribution originale et celle capturée par le modèle. L'idée est la suivante : si le modèle est bien ajusté aux données, il est possible de se servir de celui-ci pour générer de nouvelles données dont la distribution ressemble à celle des données originales. Si une différence importante est observable, alors les résultats du modèle ne sont pas fiables, car le modèle échoue à reproduire le phénomène étudié. Cette lecture graphique ne permet pas de s'assurer que le modèle est valide ou bien ajusté, mais simplement d'écarte rapidement les mauvais candidats. Notez que cette méthode ne s'applique pas lorsque la variable modélisée est binaire, multinomiale ou ordinale. Le graphique à réaliser comprend donc la distribution de la variable dépendante Y (représentée avec un histogramme ou un graphique de densité) et plusieurs distributions simulées à partir du modèle. Cette approche est plus répandue dans la statistique bayésienne, mais elle reste pertinente dans l'approche fréquentiste. Il est rare de reporter ces figures, mais elles doivent faire partie de votre diagnostic.



Distinction entre simulation et prédition

Notez ici que **simuler des données** à partir d'un modèle et **effectuer des prédictions** à partir d'un modèle sont deux opérations différentes. Prédire une valeur à partir d'un modèle revient simplement à appliquer son équation de régression à des données. Si nous réutilisons les mêmes données, la prédition renvoie toujours le même résultat, il s'agit de la partie systématique (ou déterministe) du modèle. Pour illustrer cela, admettons que nous avons ajusté un modèle GLM de type gaussien (fonction de lien identitaire) avec trois variables continues X_1 , X_2 et X_3 et des coefficients respectifs de 0,5, 1,2 et 1,8 ainsi qu'une constante de 7. Nous pouvons utiliser ces valeurs pour prédire la valeur attendue de Y quand $X_1 = 3$, $X_2 = 5$ et $X_3 = 5$:

$$\text{Prédiction} = 7 + 3 \times 0,5 + 5 \times 1,2 + 5 \times 1,8 = 23,5$$

En revanche, simuler des données à partir d'un modèle revient à ajouter la dimension stochastique (aléatoire) du modèle. Puisque notre modèle GLM est gaussien, il comporte un paramètre σ (son écart-type); admettons, pour cet exemple, qu'il est de 1,2. Ainsi, avec les données précédentes, il est possible de simuler un ensemble infini de valeurs dont la distribution est la suivante : $Normal(\mu = 23,5, \sigma = 1,2)$. 95 % du temps, ces valeurs simulées se trouveront dans l'intervalle $[21,1-25,9]$ ($\mu - 2\sigma ; \mu + 2\sigma$), puisque cette distribution est normale. Les valeurs simulées dépendent donc de la distribution choisie pour le modèle et de l'ensemble des paramètres du modèle, pas seulement de l'équation de régression.

Si vous aviez à ne retenir qu'une seule phrase de ce bloc, retenez que la prédition ne se réfère qu'à la partie systématique du modèle (équation de régression), alors que la simulation incorpore la partie stochastique (aléatoire) de la distribution du modèle. Deux prédictions effectuées sur des données identiques donnent nécessairement des résultats identiques, ce qui n'est pas le cas pour la simulation.

8.1.5.3 Comparaison des prédictions du modèle avec les valeurs originales

Les prédictions d'un modèle devraient être proches des valeurs réelles observées. Si ce n'est pas le cas, alors le modèle n'est pas fiable et ses paramètres ne sont pas informatifs. Dépendamment de la nature de la variable modélisée (quantitative ou qualitative), plusieurs approches peuvent être utilisées pour quantifier l'écart entre valeurs réelles et valeurs prédictes.

8.1.5.3.1 Pour une variable quantitative

La mesure la plus couramment utilisée pour une variable quantitative est l'erreur moyenne quadriatique (*Root Mean Square Error – RMSE* en anglais).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (8.12)$$

Il s'agit de la racine carrée de la moyenne des écarts au carré entre valeurs réelles et prédictes. Le RMSE est exprimé dans la même unité que la donnée originale et nous donne une indication sur l'erreur moyenne de la prédiction du modèle. Admettons, par exemple, que nous modélisons les niveaux de bruit environnemental en ville en décibels et que notre modèle de régression ait un RMSE de 3,5. Cela signifierait qu'en moyenne notre modèle se trompe de 3,5 décibels (erreur pouvant être négative ou positive), ce qui serait énorme (3 décibels correspondent à une multiplication par deux de l'intensité sonore) et nous amènerait à reconsidérer la fiabilité du modèle. Notez que l'usage d'une moyenne quadratique plutôt qu'une moyenne arithmétique permet de donner plus d'influence aux larges erreurs et donc de pénaliser davantage des modèles faisant parfois de grosses erreurs de prédiction. Le RMSE est donc très sensible à la présence de valeurs aberrantes. À la place de la moyenne quadratique, il est possible d'utiliser la simple moyenne arithmétique des valeurs absolues des erreurs (MAE). Cette mesure est cependant moins souvent utilisée :

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (8.13)$$

Ces deux mesures peuvent être utilisées pour comparer la capacité de prédiction de deux modèles appliqués aux mêmes données, même s'ils ne sont pas imbriqués. Elles ne permettent cependant pas de prendre en compte la complexité du modèle. Un modèle plus complexe aura toujours des valeurs de RMSE et de MAE plus faibles.

8.1.5.3.2 Pour une variable qualitative

Lorsque l'on modélise une variable qualitative, une erreur revient à prédire la mauvaise catégorie pour une observation. Il est ainsi possible de compter, pour un modèle, le nombre de bonnes et de mauvaises prédictions et d'organiser cette information dans une **matrice de confusion**. Cette dernière prend la forme suivante pour un modèle binaire :

TAB. 8.2 : Exemple de matrice de confusion

Valeur prédictive / Valeur réelle	A	B	Total (%)
A	15	3	18 (41,9)
B	5	20	25 (51,1)
Total (%)	20 (46,6)	23 (53,5)	43 (81,4)

En colonne du tableau 8.2, nous avons les catégories observées et en ligne, les catégories prédictives. La diagonale représente les prédictions correctes. Dans le cas présent, le modèle a bien catégorisé 35 (15 + 20) observations sur 43, soit une précision totale de 81,4 % ; huit sont mal classifiées (18,6 %) ; cinq avec la modalité A ont été catégorisées comme des B, soit 20 % des A, et seuls trois B ont été catégorisées comme des A (13 %).

La matrice ci-dessus (tableau 8.2) ne comporte que deux catégories possibles puisque la variable Y modélisée est binaire. Il est facile d'étendre le concept de matrice de confusion au cas des variables avec plus de deux modalités (multinomiale). Le tableau 8.3 est un exemple de matrice de confusion multinomiale.

Tab. 8.3 : Exemple de matrice de confusion multinomiale

Valeur prédictée / Valeur réelle	A	B	C	D	Total (%)
A	15	3	1	5	24 (18,7)
B	5	20	2	12	39 (30,4)
C	2	10	25	8	45 (35,2)
D	1	0	5	14	20 (15,6)
Total (%)	23 (18,1)	33 (25,7)	33 (25,7)	39 (30,5)	128

Trois mesures pour chaque catégorie peuvent être utilisées pour déterminer la capacité de prédiction du modèle :

- La précision (*precision* en anglais), soit le nombre de fois où une catégorie a été correctement prédite, divisée par le nombre de fois où la catégorie a été prédite.
- Le rappel (*recall* en anglais), soit le nombre de fois où une catégorie a été correctement prédite divisée par le nombre de fois où elle se trouve dans les données originales.
- Le score *F1* est la moyenne harmonique entre la précision et le rappel, soit :

$$F1 = 2 \times \frac{\text{précision} \times \text{rappel}}{\text{précision} + \text{rappel}} \quad (8.14)$$

Il est possible de calculer les moyennes pondérées des différents indicateurs (macro-indicateurs) afin de disposer d'une valeur d'ensemble pour le modèle. La pondération est faite en fonction du nombre de cas observé de chaque catégorie ; l'idée étant qu'il est moins grave d'avoir des indicateurs plus faibles pour des catégories moins fréquentes. Cependant, il est tout à fait possible que cette pondération ne soit pas souhaitable. C'est par exemple le cas dans de nombreuses études en santé portant sur des maladies rares où l'attention est concentrée sur ces catégories peu fréquentes.

Le **coefficient de Kappa** (variant de 0 à 1) peut aussi être utilisé pour quantifier la fidélité générale de la prédiction du modèle. Il est calculé avec l'équation (8.15) :

$$k = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \quad (8.15)$$

avec $Pr(a)$ la proportion d'accord entre les catégories observées et les catégories prédites, et $Pr(e)$ la probabilité d'un accord aléatoire entre les catégories observées et les catégories prédites (équation (8.16))

$$Pr(e) = \sum_{j=1}^J \frac{Cnt_{predit}(j)}{n \times 2} \times \frac{Cnt_{rel}(j)}{n \times 2} \quad (8.16)$$

avec n le nombre d'observations, $Cnt_{predit}(j)$ le nombre de fois où le modèle prédit la catégorie j et $Cnt_{rel}(j)$ le nombre de fois où la catégorie j a été observée.

Pour l'interprétation du coefficient de Kappa, référez-vous au tableau 8.4.

TAB. 8.4 : Interprétation des valeurs du coefficient de Kappa

K	Interprétation
< 0	Désaccord
0 - 0,20	Accord très faible
0,21 - 0,40	Accord faible
0,41 - 0,60	Accord modéré
0,61 - 0,80	Accord fort
0,81 - 1	Accord presque parfait

Enfin, un test statistique basé sur la distribution binomiale peut être utilisé pour vérifier que le modèle atteint un niveau de précision supérieur au seuil de non-information. Ce seuil correspond à la proportion de la modalité la plus présente dans le jeu de données. Dans la matrice de confusion utilisée dans le tableau 8.4, ce seuil est de 30,5 % (catégorie D), ce qui signifie qu'un modèle prédisant tout le temps la catégorie D aurait une précision de 30,5 % pour cette catégorie. Il est donc nécessaire que notre modèle fasse mieux que ce seuil.

Dans le cas de la matrice de confusion du tableau 8.3, nous obtenons donc les valeurs affichées dans le tableau 8.5.

TAB. 8.5 : Indicateurs de qualité de prédiction

	précision	rappel	F1
A	65,2	31,3	42,3
B	60,6	25,6	36,0
C	75,8	27,8	40,7
D	35,9	35,0	35,4
macro	57,8	30,0	38,2
Kappa	0,44		
Valeur de p (précision > NIR)	< 0,0001		

À la lecture du tableau 8.5, nous remarquons que :

- la catégorie D est la moins bien prédite des quatre catégories (faible précision et faible rappel) ;
- la catégorie C a une forte précision, mais un faible rappel, ce qui signifie que de nombreuses observations étant originellement des A, B ou D ont été prédites comme des C. Ce constat est également vrai pour la catégorie B ;
- le coefficient de Kappa indique un accord modéré entre les valeurs originales et la prédiction ;
- la probabilité que la précision du modèle ne dépasse pas le seuil de non-information est inférieure à 0,001, indiquant que le modèle à une précision supérieure à ce seuil.

8.1.6 Comparaison de deux modèles GLM

Tel qu'abordé dans le chapitre sur les régressions linéaires classiques, il est courant de comparer plusieurs modèles imbriqués (section 7.3.2). Cette procédure permet de déterminer si l'ajout d'une ou de plusieurs variables contribue à significativement améliorer le modèle. Il est possible d'appliquer la même démarche aux GLM à l'aide du test de rapport de vraisemblance (*likelihood ratio test*). Le principe de base de ce test est de comparer le *likelihood* de deux modèles GLM imbriqués ; la valeur de ce test se calcule avec l'équation suivante :

$$LR = 2(\loglik(M_2) - \loglik(M_1)) \quad (8.17)$$

avec M_2 un modèle reprenant toutes les variables du modèle M_1 , impliquant donc que $\text{loglik}(M_2) \geq \text{loglik}(M_1)$.

Avec ce test, nous supposons que le modèle M_2 , qui comporte plus de paramètres que le modèle M_1 , devrait être mieux ajusté aux données. Si c'est bien le cas, la différence entre les *loglikelihood* de deux modèles devrait être supérieure à zéro. La valeur calculée LR suit une distribution du khi-deux avec un nombre de degrés de liberté égal au nombre de paramètres supplémentaires dans le modèle M_2 comparativement à M_1 . Avec ces deux informations, il est possible de déterminer la valeur de p associée à ce test et de déterminer si M_2 est significativement mieux ajusté que M_1 aux données. Notez qu'il existe aussi deux autres tests (test de Wald et test de Lagrange) ayant la même fonction. Il s'agit, dans les deux cas, d'approximation du test de rapport des vraisemblances dont la puissance statistique est inférieure au test de rapport de vraisemblance (Neyman, Pearson et Pearson 1933).

Dans les prochaines sections, nous décrivons les modèles GLM les plus couramment utilisés. Il en existe de nombreuses variantes que nous ne pouvons pas toutes décrire ici. L'objectif est de comprendre les rouages de ces modèles afin de pouvoir, en cas de besoin, transposer ces connaissances sur des modèles plus spécifiques. Pour faciliter la lecture de ces sections, nous vous proposons une carte d'identité de chacun des modèles présentés. Elles contiennent l'ensemble des informations pertinentes à retenir pour chaque modèle.

8.2 Modèles GLM pour des variables qualitatives

Nous abordons en premier les principaux GLM utilisés pour modéliser des variables binaires, multinoomiales et ordinaires. Prenez bien le temps de saisir le fonctionnement du modèle logistique binomial, car il sert de base pour les trois autres modèles présentés.

8.2.1 Modèle logistique binomial

Le modèle logistique binomial est une généralisation du modèle de Bernoulli que nous avons présenté dans l'introduction de cette section. Le modèle logistique binomiale couvre donc deux cas de figure :

1. La variable observée est binaire (0 ou 1). Dans ce cas, le modèle logistique binomiale devient un simple modèle de Bernoulli.
2. La variable observée est un comptage (nombre de réussites) et nous disposons d'une autre variable avec le nombre de réplications de l'expérience. Par exemple, pour chaque intersection d'un réseau routier, nous pourrions avoir le nombre de décès à vélo (variable Y de comptage) et le nombre de collisions vélo / automobile (variable quantifiant le nombre d'expériences, chaque collision étant une expérience). Spécifiquement, nous tentons de prédire le paramètre p de la distribution binomiale à l'aide de notre équation de régression et de la fonction logistique comme fonction de lien. Notez ici que cette fonction de lien influence directement l'interprétation des paramètres du modèle. Pour rappel, cette fonction est définie comme :

$$g(x) = \ln\left(\frac{x}{1-x}\right) \quad (8.18)$$

avec \ln le logarithme naturel.

Au-delà de sa propriété mathématique assurant que $g(x) \in [0, 1]$, cette fonction offre une interprétation intéressante. La partie $\frac{x}{1-x}$ est une cote et s'interprète en termes de chances d'observer un évènement. Par exemple, dans le cas des accidents de cyclistes, si la probabilité d'observer un décès suite à une collision est de 0,1, alors la cote de cet évènement est $\frac{1}{9} = \frac{1}{10}$ soit un contre neuf. Dans un modèle GLM logis-

tique, les coefficients ajustés pour les variables indépendantes représentent des **logarithmes de rapport de cote**, car ils comparent les chances d'observer l'évènement ($y = 1$) en fonction des valeurs des variables indépendantes.

TAB. 8.6 : Carte d'identité du modèle logistique binomial

Type de variable dépendante	Variable binaire (0 ou 1) ou comptage de réussite à une expérience (ex : 3 réussites sur 5 expériences)
Distribution utilisée	Binomiale
Formulation	$Y \sim Binomial(p)$ $g(p) = \beta_0 + \beta X$ $g(x) = \log(\frac{x}{1-x})$
Fonction de lien	Logistique
Paramètre modélisé	p
Paramètres à estimer	β_0, β
Conditions d'application	Non-séparation complète, absence de sur-dispersion ou de sous-dispersion

8.2.1.1 Interprétation des paramètres

Les seuls paramètres à estimer du modèle sont les coefficients β et la constante β_0 . La fonction de lien logistique transforme la valeur de ces coefficients, en conséquence, **ils ne peuvent plus être interprétés directement**. β_0 et β sont exprimés dans une unité particulière : des logarithmes de rapports de cote (*log odd ratio*). Le rapport de cote est relativement facile à interpréter contrairement à son logarithme. Pour l'obtenir, il suffit d'utiliser la fonction exponentielle (l'inverse de la fonction logarithme) pour passer des log rapport de cote à de simples rapports de cote. Donc si $\exp(\beta)$ est inférieur à 1, il réduit les chances d'observer l'évènement et inversement si $\exp(\beta)$ est supérieur à 1.

Par exemple, admettons que nous ayons un coefficient β_1 de 1,2 pour une variable X_1 dans une régression logistique. Il est nécessaire d'utiliser son exponentiel pour l'interpréter de façon intuitive. $\exp(1,2) = 3,32$, ce qui signifie que lorsque X_1 augmente d'une unité, les chances d'observer 1 plutôt que 0 comme valeur de Y sont multipliées par 3,32. Admettons maintenant que β_1 vaille -1,2, nous calculons donc $\exp(-1,2) = 0,30$, ce qui signifie qu'à chaque augmentation d'une unité de X_1 , les chances d'observer 1 plutôt que 0 comme valeur de Y sont multipliées par 0,30. En d'autres termes, les chances d'observer 1 plutôt que 0 sont divisées par 3,33 ($1/0,30 = 3,33$), soit une diminution de 70 % ($1 - 0,3 = 0,7$) des chances d'observer 1 plutôt que 0.



Les rapports de cotes

Le rapport de cote ou rapport des chances est une mesure utilisée pour exprimer l'effet d'un facteur sur une probabilité. Il est très utilisé dans le domaine de la santé, mais aussi des paris. Prenons un exemple concret avec le port du casque à vélo. Si sur 100 accidents impliquant des cyclistes portant un casque, nous observons seulement 3 cas de blessures graves à la tête, contre 15 dans un second groupe de 100 cyclistes ne portant pas de casque, nous pouvons calculer le rapport de cote suivant :

$$\frac{p(1-q)}{q(1-p)} = \frac{0,15 \times (1-0,03)}{0,03 \times (1-0,15)} = 5,71 \quad (8.19)$$

avec p la probabilité d'observer le phénomène (ici la blessure grave à la tête) dans le groupe 1 (ici les cyclistes sans casque) et q la probabilité d'observer le phénomène dans le groupe 2 (ici les cyclistes avec un casque). Ce rapport de cote indique que les cyclistes sans casques ont 5,71 fois plus de risques de se blesser gravement à la tête lors d'un accident comparativement aux cyclistes portant un casque.

8.2.1.2 Conditions d'application

La non-séparation complète signifie qu'aucune des variables X n'est, à elle seule, capable de parfaitement distinguer les deux catégories 0 et 1 de la variable Y . Dans un tel cas de figure, les algorithmes d'ajustement utilisés pour estimer les paramètres des modèles sont incapables de converger. Notez aussi l'absurdité de créer un modèle pour prédire une variable Y si une variable X est capable à elle seule de la prédire à coup sûr. Ce problème est appelé un effet de Hauck-Donner. Il est assez facile de le repérer, car la plupart du temps les fonctions de R signalent ce problème (message d'erreur sur la convergence). Sinon, des valeurs extrêmement élevées ou faibles pour certains rapports de cote peuvent aussi indiquer un effet de Hauck-Donner.

La sur-dispersion est un problème spécifique aux distributions n'ayant pas de paramètre de dispersion (binomiale, de Poisson, exponentielle, etc.), pour lesquelles la variance dépend directement de l'espérance. La sur-dispersion désigne une situation dans laquelle les résidus (ou erreurs) d'un modèle sont plus dispersés que ce que suppose la distribution utilisée. À l'inverse, il est aussi possible (mais rare) d'observer des cas de sous-dispersion (lorsque la dispersion des résidus est plus petite que ce que suppose la distribution choisie). Ce cas de figure se produit généralement lorsque le modèle parvient à réaliser une prédiction trop précise pour être crédible. Si vous rencontrez une forte sous-dispersion, cela signifie souvent que l'une de vos variables indépendantes provoque une séparation complète. La meilleure option, dans ce cas, est de supprimer la variable en question du modèle. La variance attendue d'une distribution binomiale est $nb \times p \times (1 - p)$, soit le produit entre le nombre de tirages, la probabilité de réussite et la probabilité d'échec. À titre d'exemple, si nous considérons une distribution binomiale avec un seul tirage et 50 % de chances de réussite, sa variance serait : $1 \times 0,5 \times (1 - 0,5) = 0,25$.

Plusieurs raisons peuvent expliquer la présence de sur-dispersion dans une modèle :

- il manque des variables importantes dans le modèle, conduisant à un mauvais ajustement et donc une sur-dispersion des erreurs ;
- les observations ne sont pas indépendantes, impliquant qu'une partie de la variance n'est pas contrôlée et augmente les erreurs ;
- la probabilité de succès de chaque expérience varie d'une répétition à l'autre (différentes distributions).

La conséquence directe de la sur-dispersion est la sous-estimation de la variance des coefficients de régression. En d'autres termes, la sur-dispersion conduit à sous-estimer notre incertitude quant aux coefficients obtenus et réduit les valeurs de p calculées pour ces coefficients. Les risques de trouver des résultats significatifs à cause des fluctuations d'échantillonnage augmentent.

Pour détecter une sur-dispersion ou une sous-dispersion dans un modèle logistique binomial, il est possible d'observer les résidus de déviance du modèle. Ces derniers sont supposés suivre une distribution du khi-deux avec $n-k$ degrés de liberté (avec n le nombre d'observations et k le nombre de coefficients dans le modèle). Par conséquent, la somme des résidus de déviance d'un modèle logistique binomiale divisée par le nombre de degrés de liberté devrait être proche de 1. Une légère déviation (jusqu'à 0,15 au-dessus ou au-dessous de 1) n'est pas alarmante ; au-delà, il est nécessaire d'ajuster le modèle.

Notez que si la variable Y modélisée est exactement binaire (chaque expérience est indépendante et n'est composée que d'un seul tirage) et que le modèle utilise donc une distribution de Bernoulli, le test précédent pour détecter une éventuelle sur-dispersion n'est pas valide. Hilbe (2009) parle de sur-dispersion implicite pour le modèle de Bernoulli et recommande notamment de toujours ajuster les erreurs standards des modèles utilisant des distributions de Bernoulli, binomiale et de Poisson. L'idée ici est d'éviter d'être trop optimiste face à l'incertitude du modèle sur les coefficients et de l'ajuster en conséquence. Pour cela, il est possible d'utiliser des quasi-distributions ou des estimateurs robustes (Zeileis 2004). Notez que si le modèle ne souffre pas de sur ou sous-dispersion, ces ajustements produisent des résultats

équivalents aux résultats non ajustés.

8.2.1.3 Exemple appliqué dans R

Présentation des données

Pour illustrer le modèle logistique binomial, nous utilisons ici un jeu de données proposé par l'Union européenne : l'enquête de déplacement sur la demande pour des systèmes de transports innovants¹. Pour cette enquête, un échantillon de 1 000 individus représentatifs de la population a été constitué dans chacun des 26 États membres de l'UE, soit un total de 26 000 observations. Pour chaque individu, plusieurs informations ont été collectées relatives à la catégorie socioprofessionnelle, le mode de transport le plus fréquent, le temps du trajet de son déplacement le plus fréquent et son niveau de sensibilité à la cause environnementale. Nous modélisons ici la probabilité qu'un individu déclare utiliser le plus fréquemment le vélo comme moyen de transport. Les variables explicatives sont résumées au tableau 8.7. Il existe bien évidemment un grand nombre de facteurs individuels qui influencent la prise de décision sur le mode de transport. Les résultats de ce modèle ne doivent donc pas être pris avec un grand sérieux ; il est uniquement construit à des fins pédagogiques, sans cadre conceptuel solide.

TAB. 8.7 : Variables indépendantes utilisées pour prédire le mode de transport le plus utilisé

Nom de la variable	Signification	Type de variable	Mesure
Pays	Pays de résidence	Variable multinomiale	Le nom d'un des 26 pays membres de l'UE
Sexe	Sexe biologique	Variable binaire	Homme ou femme
Age	Âge biologique	Variable continue	L'âge en nombre d'années variant de 16 à 84 ans dans le jeu de données
Education	Niveau d'éducation maximum atteint	Variable multinomiale	Premier cycle, secondaire inférieur (classes supérieures de l'école élémentaire), secondaire, troisième cycle
StatutEmploi	Employé ou non	Variable binaire	Employé ou non
Revenu	Niveau de revenu autodéclaré	Variable multinomiale	Très faible revenu, faible revenu, revenu moyen, revenu élevé, revenu très élevé, sans réponse
Residence	Lieu de résidence	Variable multinomiale	Zone rurale, petite ou moyenne ville (moins de 250 000 habitants), grande ville (entre 250 000 et 1 million d'habitants), aire métropolitaine (plus d'un million d'habitants)
Duree	Durée du voyage le plus fréquent autodéclarée (en minutes)	Variable continue	Nombre de minutes
ConsEnv	Préoccupation environnementale	Variable ordinaire	Échelle de Likert de 1 à 10

Vérification des conditions d'application

La première étape de la vérification des conditions d'application est de calculer les valeurs du facteur d'inflation de variance (VIF) pour s'assurer de l'absence de multicolinéarité trop forte entre les variables indépendantes. L'ensemble des valeurs de VIF sont inférieures à 5, indiquant l'absence de multicolinéarité excessive dans le modèle.

```
library(car)
# Chargement des données
dfenquete <- read.csv("data/glm/enquete_transport_UE.csv", encoding = 'UTF-8')
dfenquete$Pays <- relevel(as.factor(dfenquete$Pays), ref = "Allemagne")
```

¹<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/P82V9X>

```
# Vérification du VIF
model1 <- glm(y ~
  Pays + Sexe + Age + Education + StatutEmploi + Revenu +
  Residence + Duree + ConsEnv,
  family = binomial(link="logit"),
  data = dfenquete
)
vif(model1)

##          GVIF Df GVIF^(1/(2*Df))
## Pays      1.794797 27   1.010890
## Sexe     1.028618  1   1.014208
## Age       1.060256  1   1.029687
## Education 1.428872  3   1.061285
## StatutEmploi 1.151879  1   1.073256
## Revenu    1.220934  5   1.020162
## Residence 1.130526  3   1.020658
## Duree     1.042638  1   1.021096
## ConsEnv   1.090987  1   1.044503
```

La seconde étape de vérification est le calcul des distances de Cook et l'identification d'éventuelles valeurs aberrantes (figure 8.4).

```
# Calcul et représentation des distances de Cook
cookd <- data.frame(
  dist = cooks.distance(model1),
  oid = 1:nrow(dfenquete)
)
ggplot(cookd) +
  geom_point(aes(x = oid, y = dist), color = rgb(0.1,0.1,0.1,0.4), size = 1) +
  geom_hline(yintercept = 0.002, color = "red") +
  labs(x = "observations", y = "distance de Cook") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
```

Le calcul de la distance de Cook révèle un ensemble d'observations se démarquant nettement des autres (délimitées dans la figure 8.4 par la ligne rouge). Nous les isolons dans un premier temps pour les analyser.

```
# Isoler les observations avec de très fortes valeurs de Cook
# valeur seuil choisie : 0,002
cas_étranges <- subset(dfenquete, cookd$dist >= 0.002)
cat(nrow(cas_étranges), 'observations se démarquant dans le modèle')

## 19 observations se démarquant dans le modèle
```

```
print(cas_étranges)

##           X y      Pays Sexe Age      Education Statut_emploi
## 7660    7660 1 Slovaquie homme 50 universite      Employed
## 25150  25150 1         Malte homme 16 secondaire Not Employed
```

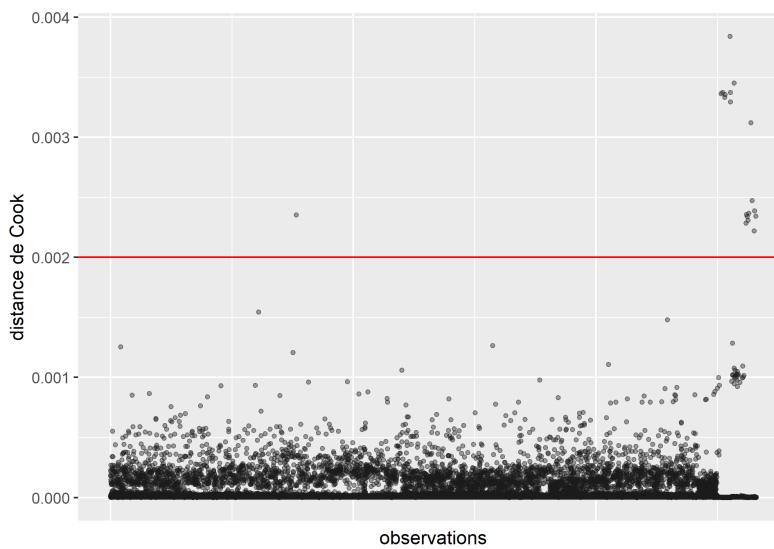


FIG. 8.4 : Distances de Cook pour le modèle binomial avec toutes les observations

```

## 25227 25227 1      Malte femme 53 secondaire inferieur Not Employed
## 25309 25309 1      Malte femme 32            secondaire Employed
## 25322 25322 1      Malte homme 38            universite Employed
## 25536 25536 1      Malte homme 27            universite Employed
## 25541 25541 1      Malte homme 38 secondaire inferieur Employed
## 25549 25549 1      Malte homme 31            universite Employed
## 25690 25690 1 Luxembourg homme 32            universite Employed
## 26190 26190 1      Chypre homme 24            secondaire Not Employed
## 26201 26201 1      Chypre homme 25            secondaire Employed
## 26244 26244 1      Chypre homme 32            secondaire Employed
## 26269 26269 1      Chypre homme 60            secondaire Not Employed
## 26303 26303 1      Chypre homme 59            secondaire Not Employed
## 26393 26393 1      Chypre homme 30            premier cycle Employed
## 26444 26444 1      Chypre femme 52            universite Employed
## 26516 26516 1      Chypre homme 21            universite Not Employed
## 26549 26549 1      Chypre homme 28            universite Employed
## 26600 26600 1      Chypre homme 36            secondaire Employed
##
##           Revenu             Residence Duree mode_pref StatutEmploi ConsEnv
## 7660      moyen       zone rurale   775     velo    employe      7
## 25150     moyen       zone rurale    15      velo  sans emploi     3
## 25227     moyen       zone rurale   45      marche sans emploi     5
## 25309     moyen petite-moyenne ville   25      marche employe      4
## 25322     eleve       zone rurale   30      marche employe     10
## 25536     tres eleve petite-moyenne ville   14      velo    employe     10
## 25541     moyen       zone rurale    5      marche employe      8
## 25549     sans reponse petite-moyenne ville   60      velo    employe     10
## 25690     tres eleve petite-moyenne ville  720      velo    employe      6
## 26190     moyen       grande ville   20      velo  sans emploi     5
## 26201     faible       zone rurale   20      velo    employe      5
## 26244     tres faible petite-moyenne ville   18      velo    employe      4
## 26269     moyen petite-moyenne ville      5      velo  sans emploi     7

```

```

## 26303      moyen      zone rurale    7     velo  sans emploi    8
## 26393  tres eleve petite-moyenne ville   61     velo      employe    5
## 26444      eleve petite-moyenne ville  120     velo      employe    3
## 26516      moyen petite-moyenne ville   25     velo  sans emploi    8
## 26549  tres faible petite-moyenne ville   15     velo      employe    2
## 26600      moyen petite-moyenne ville    8     velo      employe    1

```

À la lecture des valeurs pour ces 19 cas étranges, nous remarquons que la plupart des observations proviennent de Malte et de Chypre. Ces deux petites îles constituent des cas particuliers en Europe et devraient vraisemblablement faire l'objet d'une analyse séparée. Nous décidons donc de les retirer du jeu de données. Deux autres observations étranges sont observables en Slovaquie et au Luxembourg. Dans les deux cas, les répondants ont renseigné des temps de trajet fantaisistes de respectivement 775 et 720 minutes. Nous les retirons donc également de l'analyse.

```

# Retirer les observations aberrantes
dfenquete2 <- subset(dfenquete, (dfenquete$Pays %in% c("Malte", "Chypre")) == F &
                      dfenquete$Duree < 400)

# Réajuster le modèle
model2 <- glm(y ~
                 Pays + Sexe + Age + Education + StatutEmploi + Revenu +
                 Residence + Duree + ConsEnv,
                 family = binomial(link="logit"),
                 data = dfenquete2)

# Recalculer la distance de Cook
cookd <- data.frame(
  dist = cooks.distance(model2),
  oid = 1:nrow(dfenquete2)
)
ggplot(cookd) +
  geom_point(aes(x = oid, y = dist), color = rgb(0.1,0.1,0.1,0.4), size = 1) +
  labs(x = "observations", y = "distance de Cook") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank())

```

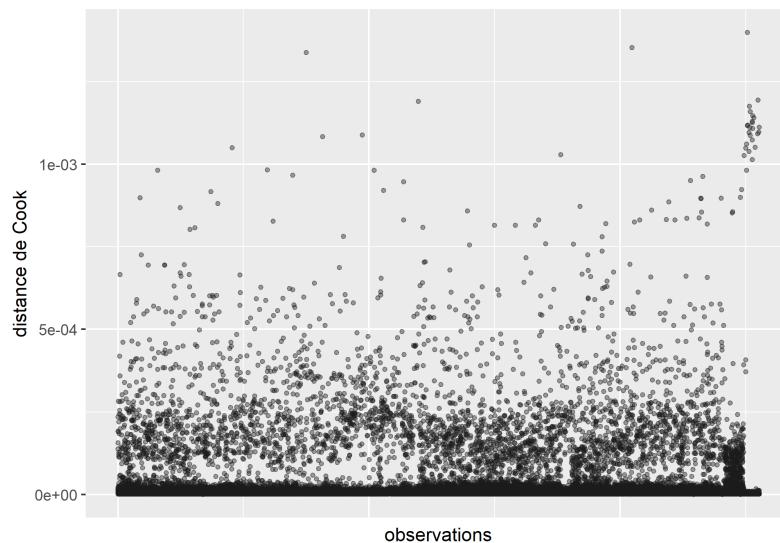


FIG. 8.5 : Distances de Cook pour le modèle binomial sans les valeurs aberrantes

Après avoir retiré ces valeurs aberrantes, nous n'observons plus de nouveaux cas singuliers avec les distances de Cook (figure 8.5).

La prochaine étape de vérification des conditions d'application est l'analyse des résidus simulés. Nous commençons donc par calculer ces résidus et afficher leur histogramme (figure 8.6).

```
library(DHARMA)
# Extraire les probabilités prédites par le modèle
probs <- predict(model2, type = "response")
# Calculer 1000 simulations à partir du modèle ajusté
sims <- lapply(1:length(probs), function(i){
  p <- probs[[i]]
  vals <- rbinom(n = 1000, size = 1, prob = p)
})
matsim <- do.call(rbind, sims)
# Utiliser le package DHARMA pour calculer les résidus simulés
sim_res <- createDHARMA(simulatedResponse = matsim,
                           observedResponse = dfenquete2$y,
                           fittedPredictedResponse = probs,
                           integerResponse = T)

ggplot() +
  geom_histogram(aes(x = residuals(sim_res)),
                 bins = 30, fill = "white", color = rgb(0.3,0.3,0.3)) +
  labs(x = "résidus simulés", y = "fréquence")
```

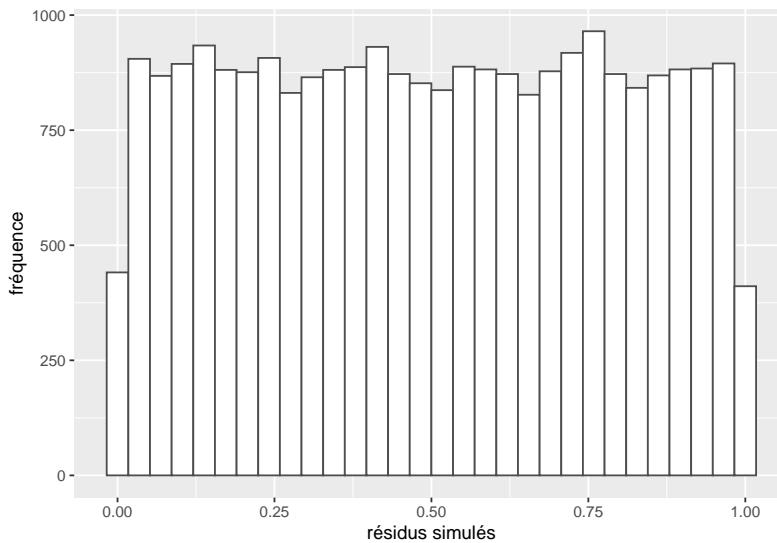


FIG. 8.6 : Distribution des résidus simulés pour le modèle binomial

L'histogramme indique clairement que les résidus simulés suivent une distribution uniforme (figure 8.6). Il est possible d'aller plus loin dans le diagnostic en utilisant la fonction `plot` sur l'objet `sim_res`. La partie de droite de la figure ainsi obtenue (figure 8.7) est un diagramme de quantiles-quantiles (ou Q-Q plot). Les points du graphique sont supposés suivre une ligne droite matérialisée par la ligne rouge. Une déviation de cette ligne indique un éloignement des résidus de leur distribution attendue. Trois tests sont également réalisés par la fonction :

- Le premier (Test de Kolmogorov-Smirnov, *KS test*) permet de tester si les points dévient significativement de la ligne droite. Dans notre cas, la valeur de p n'est pas significative, indiquant que les

résidus ne dévient pas de la distribution uniforme.

- Le second test permet de vérifier la présence de sur ou sous-dispersion. Dans notre cas, ce test n'est pas significatif, n'indiquant aucun problème de sur-dispersion ou de sous-dispersion.
- Le dernier test permet de vérifier si des valeurs aberrantes sont présentes dans les résidus. Une valeur non significative indique une absence de valeurs aberrantes.

Le second graphique permet de comparer les résidus et les valeurs prédites. L'idéal est donc d'observer une ligne droite horizontale au milieu du graphique qui indiquerait une absence de relation entre les valeurs prédites et les résidus (ce que nous observons bien ici).

```
plot(sim_res)
```

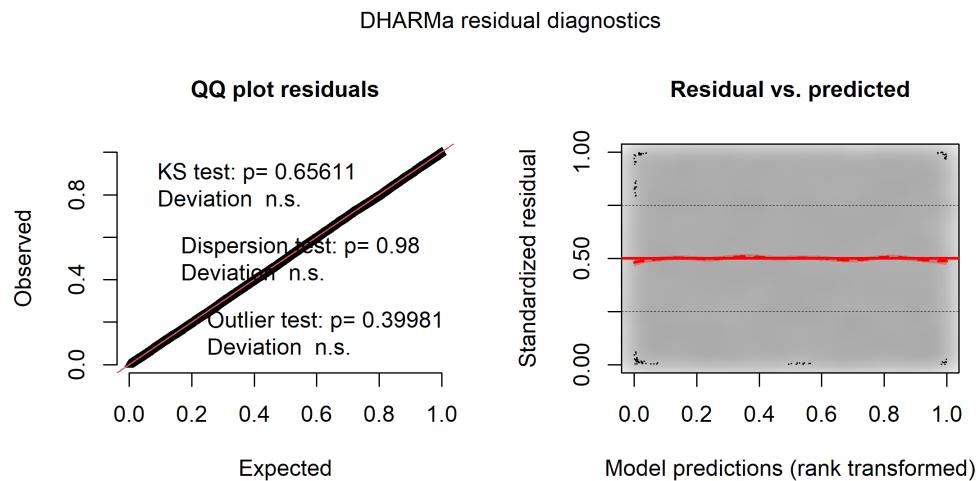


FIG. 8.7 : Diagnostic des résidus simulés par le package DHARMA

L'analyse approfondie des résidus nous permet donc de conclure que le modèle respecte les conditions d'application et que nous pouvons passer à la vérification de la qualité d'ajustement du modèle.

Vérification de la qualité d'ajustement

Pour calculer les différents R^2 d'un modèle GLM, nous proposons la fonction suivante :

```
rsqs <- function(loglike.full, loglike.null, full.deviance, null.deviance, nb.params, n){
  # Calcul de la déviance expliquée
  explained_dev <- 1 - (full.deviance / null.deviance)
  K <- nb.params
  # R2 de McFadden ajusté
  r2_faddenadj <- 1 - (loglike.full - K) / loglike.null
  Lm <- loglike.full
  Ln <- loglike.null
  # R2 de Cox and Snell
  Rcs <- 1 - exp((-2/n) * (Lm - Ln))
  # R2 de Nagelkerke
  Rn <- Rcs / (1 - exp(2 * Ln / n))
  return(
    list("deviance expliquée" = explained_dev,
        "McFadden ajusté" = r2_faddenadj,
```

```

    "Cox and Snell" = Rcs,
    "Nagelkerke" = Rn
)
}
}
}
```

Nous l'utilisons pour l'ensemble des modèles GLM de ce chapitre. Dans le cas du modèle binomial, nous obtenons :

```

# Ajuster un modèle null avec seulement une constante
model2.null <- glm(y ~1,
                     family = binomial(link="logit"),
                     data = dfenquete2)

# Calculer les R2
rsqs(loglike.full = as.numeric(logLik(model2)), # loglikelihood du modèle complet
      loglike.null = as.numeric(logLik(model2.null)), # loglikelihood du modèle nul
      full.deviance = deviance(model2), # déviance du modèle complet
      null.deviance = deviance(model2.null), # déviance du modèle nul
      nb.params = model2$rank, # nombre de paramètres dans le modèle
      n = nrow(dfenquete2) # nombre d'observations
)

## `$`deviance expliquée`
## [1] 0.0876057
##
## `$`McFadden ajusté`
## [1] 0.08357379
##
## `$`Cox and Snell`
## [1] 0.0689509
##
## $Nagelkerke
## [1] 0.1236597
```

La déviance expliquée par le modèle est de 8,8 %, les pseudos R² de McFadden (ajusté), d'Efron et de Nagelkerke sont respectivement 0,084, 0,069 et 0,124. Toutes ces valeurs sont relativement faibles et indiquent qu'une large partie de la variabilité de Y reste inexpliquée.

Pour vérifier la qualité de prédiction du modèle, nous devons comparer les catégories prédictes et les catégories réelles de notre variable dépendante et construire une matrice de confusion. Cependant, un modèle GLM binomial prédit **la probabilité d'appartenance au groupe 1** (ici les personnes utilisant le vélo pour effectuer leur déplacement le plus fréquent). Pour convertir ces probabilités prédictes en catégories prédictes, il faut choisir un seuil de probabilité au-delà duquel nous considérons que la valeur attendue est 1 (cycliste) plutôt que 0 (autre). Un exemple naïf serait de prendre le seuil 0,5, ce qui signifierait que si le modèle prédit qu'une observation a au moins 50 % de chance d'être une personne à vélo, alors nous l'attribuons à cette catégorie. Cependant, cette méthode est rarement optimale ; il est donc plus judicieux de fixer le seuil de probabilité en trouvant le point d'équilibre entre la sensibilité (proportion de 1 correctement identifiés) et la spécificité (proportion de 0 correctement identifiés). Ce point d'équilibre est identifiable graphiquement en calculant la spécificité et la sensibilité de la prédiction selon toutes les valeurs possibles du seuil.

```

library(ROCR)
# Obtention des prédictions du modèle
prob <- predict(model2, type = "response")
# Calcul de la sensibilité et de la spécificité (package ROCR)
predictions <- prediction(prob, dfenquete2$y)
sens <- data.frame(x=unlist(performance(predictions, "sens")@x.values),
                     y=unlist(performance(predictions, "sens")@y.values))
spec <- data.frame(x=unlist(performance(predictions, "spec")@x.values),
                     y=unlist(performance(predictions, "spec")@y.values))
# Trouver numériquement la valeur seuil (minimiser la différence absolue
# entre sensibilité et spécificité)
real <- dfenquete2$y
find_cutoff <- function(seuil){
  pred <- ifelse(prob>seuil,1,0)
  sensi <- sum(real==1 & pred==1) / sum(real==1)
  spec <- sum(real==0 & pred==0) / sum(real==0)
  return(abs(sensi-spec))
}
prob_seuil <- optimize(find_cutoff,interval = c(0,1), maximum = F)$minimum
cat("Le seuil de probabilité à retenir équilibrant",
    "la sensibilité et la spécificité est de",prob_seuil)

```

Le seuil de probabilité à retenir équilibrant la sensibilité et la spécificité est de 0.14785

```

# Affichage du graphique
ggplot() +
  geom_line(data = sens, mapping = aes(x = x, y = y)) +
  geom_line(data = spec, mapping = aes(x = x,y = y,col="red")) +
  scale_y_continuous(sec.axis = sec_axis(~., name = "Spécificité")) +
  labs(x='Seuil de probabilité', y="Sensibilité") +
  geom_vline(xintercept = prob_seuil, color = "black", linetype = "dashed") +
  annotate(geom = "text", x = prob_seuil, y = 0.01, label = round(prob_seuil,3))+
  theme(axis.title.y.right = element_text(colour = "red"), legend.position="none")

```

Nous constatons à la figure 8.8 que si la valeur du seuil est 0 %, alors la prédiction a une sensibilité parfaite (le modèle prédit toujours 1, donc tous les 1 sont détectés); à l'inverse, si le seuil choisi est 100 %, alors la prédiction à une spécificité parfaite (le modèle prédit toujours 0, donc tous les 0 sont détectés). Dans notre cas, la valeur d'équilibre est d'environ 0,148, donc si le modèle prédit une probabilité au moins égale à 14,8 % qu'un individu utilise le vélo pour son déplacement le plus fréquent, nous devons l'attribuer à la catégorie *cycliste*. Avec ce seuil, nous pouvons convertir les probabilités prédites en classes prédites et construire notre matrice de confusion.

```

library(caret) # pour la matrice de confusion
# Calcul des catégories prédites
ypred <- ifelse(predict(model2,type="response")>0.148,1,0)
info <- confusionMatrix(as.factor(dfenquete2$y), as.factor(ypred))
# Affichage des valeurs brutes de la matrice de confusion
print(info)

```

```

## Confusion Matrix and Statistics
##
##          Reference

```

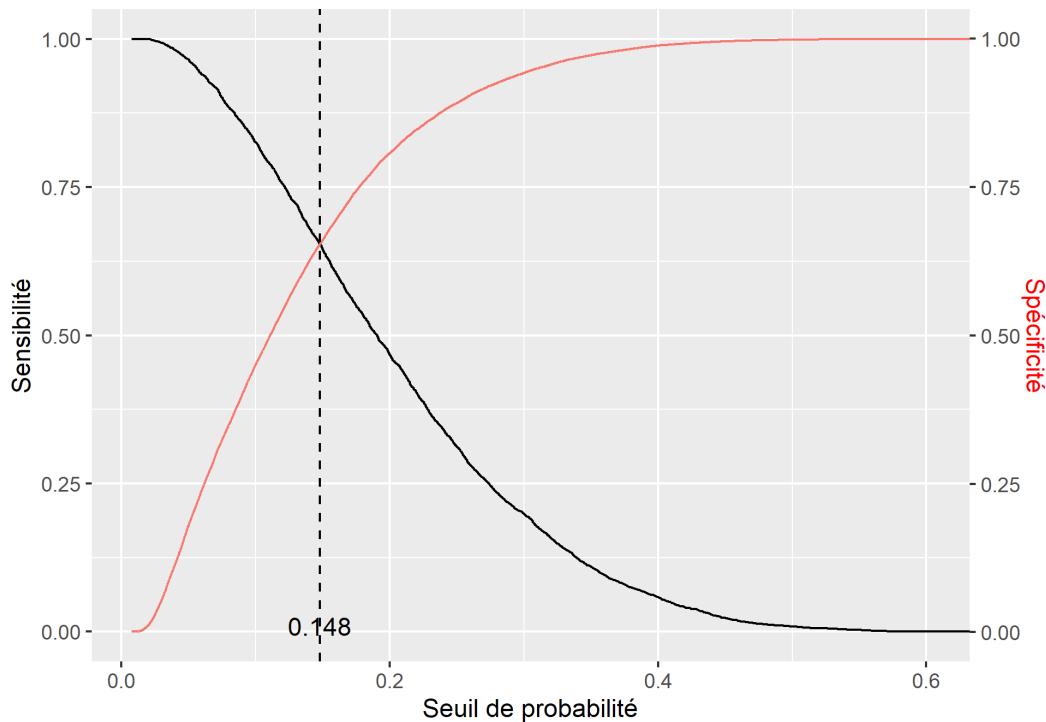


FIG. 8.8 : Point d'équilibre entre sensibilité et spécificité

```

## Prediction     0      1
##             0 14355  7576
##             1 1251   2365
##
##                   Accuracy : 0.6545
##                   95% CI  : (0.6486, 0.6603)
##       No Information Rate : 0.6109
##       P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.1783
##
## Mcnemar's Test P-Value : < 2.2e-16
##
##                   Sensitivity : 0.9198
##                   Specificity  : 0.2379
##       Pos Pred Value : 0.6546
##       Neg Pred Value : 0.6540
##           Prevalence  : 0.6109
##       Detection Rate : 0.5619
## Detection Prevalence : 0.8585
##       Balanced Accuracy : 0.5789
##
## 'Positive' Class : 0
##

```

Les résultats proposés par le package *caret* sont exhaustifs; nous vous proposons ici une façon de les présenter dans deux tableaux : l'un présente la matrice de confusion (tableau 8.8) et l'autre, les indicateurs

de qualité de prédiction (tableau 8.9).

TAB. 8.8 : Matrice de confusion pour le modèle binomial

	0 (réel)	1 (réel)	Total	%
0 (prédit)	14355	7576	21931	85.8
1 (prédit)	1251	2365	3616	14.2
Total	15606	9941	25547	
%	61.1	38.9		

D'après ces indicateurs, nous constatons que le modèle a une capacité de prédiction relativement faible, mais tout de même significativement supérieure au seuil de non-information. La valeur de rappel pour la catégorie 1 (cycliste) est faible, indiquant que le modèle a manqué un nombre important de cyclistes lors de sa prédiction.

TAB. 8.9 : Matrice de confusion pour le modèle binomial

	Précision	Rappel	F1
0	0.65	0.92	0.76
1	0.65	0.24	0.35
macro	0.65	0.65	0.6
Kappa	0.18		
Valeur de p (précision > NIR)	< 0,0001		

Interprétation des résultats du modèle

L'interprétation des résultats d'un modèle binomial passe par la lecture des rapports de cotes (exponentiel des coefficients) et de leurs intervalles de confiance. Nous commençons donc par calculer la version robuste des erreurs standards des coefficients :

```
library(sandwich) # pour calculer les erreurs standards robustes
covModel2 <- vcovHC(model2, type = "HC0") # méthode HC0, basée sur les résidus
stdErrRobuste <- sqrt(diag(covModel2)) # extraire la diagonale
# Extraction des coefficients
coeffs <- model2$coefficients
# Recalcul des scores Z
zvalRobuste <- coeffs / stdErrRobuste
# Recalcul des valeurs de P
pvalRobuste <- 2 * pnorm(abs(zvalRobuste), lower.tail = FALSE)
# Calcul des rapports de cote
oddRatio <- exp(coeffs)
# Calcul des intervalles de confiance à 95 % des rapports de cote
lowerBound <- exp(coeffs - 1.96 * stdErrRobuste)
upperBound <- exp(coeffs + 1.96 * stdErrRobuste)
# Étoiles pour les valeurs de p
starsp <- case_when(pvalRobuste <= 0.001 ~ "***",
                      pvalRobuste > 0.001 & pvalRobuste <= 0.01 ~ "**",
                      pvalRobuste > 0.01 & pvalRobuste <= 0.05 ~ "*",
                      pvalRobuste > 0.05 & pvalRobuste <= 0.1 ~ ".",
                      TRUE ~ "")
```

```
# Compilation des résultats dans un tableau
tableau_binom <- data.frame(
  coefficients = coeffs,
  rap.cote = oddRatio,
  err.std = stdErrRobuste,
  score.z = zvalRobuste,
  p.val = pvalRobuste,
  rap.cote.2.5 = lowerBound,
  rap.cote.97.5 = upperBound,
  sign = starsp
)
```

Considérant que la variable *Pays* a 24 modalités, il est plus judicieux de présenter ses 23 rapports de cotes sous forme d'un graphique. Nous avons choisi l'Allemagne comme catégorie de référence puisqu'elle fait partie des pays avec une importante part modale pour le vélo sans pour autant constituer un cas extrême comme le Danemark.

```
# Isoler les lignes du tableau récapitulatif pour les pays
paysdf <- subset(tableau_binom, grepl("Pays", row.names(tableau_binom), fixed = T))
#paysdf$Pays <- gsub("Pays", "", row.names(paysdf), fixed=T)
paysdf$Pays <- substr(row.names(paysdf), 5, nchar(row.names(paysdf)))
ggplot(data = paysdf) +
  geom_vline(xintercept = 1, color = "red") + #afficher la valeur de référence
  geom_errorbarh(aes(xmin = rap.cote.2.5, xmax = rap.cote.97.5,
                     y = reorder(Pays, rap.cote)), height = 0) +
  geom_point(aes(x = rap.cote, y = reorder(Pays, rap.cote))) +
  geom_text(aes(x = rap.cote.97.5, y = reorder(Pays, rap.cote),
                label = paste("RC : ", round(rap.cote, 2), sep = "")),
            size = 3, nudge_x = 0.25) +
  labs(x = "Rapports de cote", y = "Pays (référence : Allemagne)")
```

Dans la figure 8.9, la barre horizontale pour chaque pays représente l'intervalle de confiance de son rapport de cotes (le point); plus cette ligne est longue, plus grande est l'incertitude autour de ce paramètre. Lorsque les lignes de deux pays se chevauchent, cela signifie qu'il n'y a pas de différence significative au seuil 0,05 entre les rapports de cotes des deux pays. La ligne rouge tracée à $x = 1$, représente le rapport de cotes du pays de référence (ici l'Allemagne). Nous constatons ainsi que comparativement à un individu vivant en Allemagne, ceux vivant au Danemark et aux Pays-Bas ont 2,4 fois plus de chances d'utiliser le vélo pour leur déplacement le plus fréquent. Les Pays de l'Ouest (France, Luxembourg, Royaume-Uni, Irlande) et du Sud (Grèce, Italie, Espagne, Portugal) ont en revanche des rapports de cotes plus faibles. En France, les chances qu'un individu utilise le vélo pour son trajet le plus fréquent sont 3,22 (1/0,31) fois plus faibles que si l'individu vivait en Allemagne.

Pour le reste des coefficients et des rapports de cotes, nous les rapportons dans le tableau 8.10.

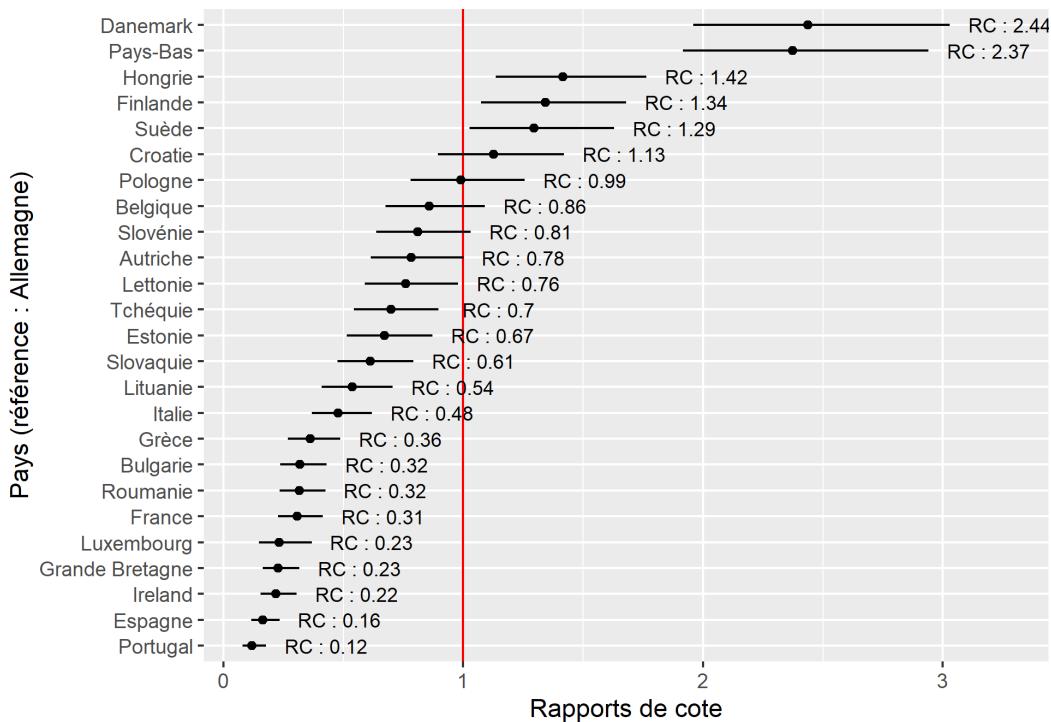


FIG. 8.9 : Rapports de cote pour les différents pays de l'UE

TAB. 8.10 : Résultats du modèle binomial

Variable	Coefficient	Rapport de cote	Err.std	Val.z	P	RC 2,5 %	RC 97,5 %
Constante	-2,497	0,082	0,183	-13,674	0,000	0,058	0,118
<i>Sexe</i>							
ref : femme	—	—	—	—	—	—	—
homme	0,372	1,451	0,038	9,803	0,000	1,347	1,562
Age	-0,009	0,991	0,002	-5,361	0,000	0,988	0,994
<i>Education</i>							
ref : premier cycle	—	—	—	—	—	—	—
secondaire	0,193	1,213	0,105	1,836	0,066	0,987	1,490
secondaire inferieur	0,301	1,351	0,114	2,649	0,008	1,081	1,687
universite	0,146	1,157	0,108	1,349	0,177	0,936	1,432
<i>StatutEmploi</i>							
ref : employé	—	—	—	—	—	—	—
sans emploi	0,257	1,293	0,043	6,045	0,000	1,190	1,405
<i>Revenu</i>							
ref : elevé	—	—	—	—	—	—	—
faible	0,077	1,080	0,072	1,067	0,286	0,938	1,244
moyen	0,042	1,043	0,065	0,639	0,523	0,917	1,185
sans réponse	0,217	1,242	0,102	2,120	0,034	1,016	1,517
très élevé	-0,120	0,887	0,188	-0,637	0,524	0,613	1,283
très faible	0,240	1,271	0,086	2,776	0,006	1,073	1,505
<i>Residence</i>							
ref : aire métropolitaine	—	—	—	—	—	—	—
grande ville	0,273	1,314	0,070	3,911	0,000	1,146	1,507
petite-moyenne ville	0,277	1,319	0,061	4,503	0,000	1,169	1,487
zone rurale	-0,119	0,888	0,069	-1,713	0,087	0,775	1,017
Duree	-0,001	0,999	0,001	-0,981	0,326	0,998	1,001
ConsEnv	0,102	1,108	0,010	10,502	0,000	1,087	1,130

Les chances pour un individu d'utiliser le vélo pour son trajet le plus fréquent sont augmentées de 45 % s'il s'agit d'un homme plutôt qu'une femme. Pour l'âge, nous constatons un effet relativement faible puisque chaque année supplémentaire réduit les chances qu'un individu utilise le vélo comme mode de transport pour son trajet le plus fréquent de 0,9 % ($(0,991 - 1) \times 100$). Le fait d'être sans emploi augmente les chances d'utiliser le vélo de 29 % comparativement au fait d'avoir un emploi. Concernant le niveau d'éducation, seul le coefficient pour le groupe des personnes de la catégorie « secondaire inférieure » est significatif, indiquant que les personnes de ce groupe ont 35 % de chances en plus d'utiliser le vélo comme mode de transport pour leur déplacement le plus fréquent comparativement aux personnes de la catégorie « premier cycle ». Pour le revenu, seul le groupe avec de très faibles revenus se distingue significativement du groupe avec un revenu élevé avec un rapport de cotes de 1,27, soit 27 % de chances en plus d'utiliser le vélo.

Comparativement à ceux vivant dans une aire métropolitaine, les personnes vivant dans de petites, moyennes et grandes villes ont des chances accrues d'utiliser le vélo comme mode de déplacement pour leur trajet le plus fréquent. En revanche, nous n'observons aucune différence entre la probabilité d'utiliser le vélo dans une métropole et en zone rurale. La figure 8.10 permet de clairement visualiser cette situation. Rappelons que la référence est la situation : vivre dans une région métropolitaine, représentée par la ligne verticale rouge. Plusieurs pistes d'interprétation peuvent être envisagées pour ce résultat :

- En métropole et dans les zones rurales, les distances domicile-travail tendent à être plus grandes que dans les petites, moyennes et grandes villes.
- En métropole, le système de transport en commun est davantage développé et entre donc en concurrence avec les modes de transport actifs.

```
# Isoler les lignes du tableau récapitulatif pour les lieux de résidence
residdf <- subset(tableau_binom, grepl("Residence", row.names(tableau_binom), fixed = T))
residdf$resid <- gsub("Residence","", row.names(residdf), fixed=T)
ggplot(data = residdf) +
  geom_vline(xintercept = 1, color = "red") # afficher la valeur de référence
  geom_errorbarh(aes(xmin = rap.cote.2.5, xmax = rap.cote.97.5, y = resid), height = 0) +
  geom_point(aes(x = rap.cote, y = resid)) +
  geom_text(aes(x = rap.cote.97.5, y = resid,
                label = paste("RC : ", round(rap.cote,2), sep="")),
            size = 3, nudge_x = 0.1) +
  labs(x = "Rapports de cotes",
       y = "Lieu de résidence (référence : aire métropolitaine)")
```

Il est aussi intéressant de noter que la durée des trajets ne semble pas influencer la probabilité d'utiliser le vélo. Enfin, une conscience environnementale plus affirmée semble être associée avec une probabilité supérieure d'utiliser le vélo pour son déplacement le plus fréquent, avec une augmentation des chances de 11 % pour chaque point supplémentaire sur l'échelle de Likert.

Afin de simplifier la présentation de certains résultats, il est possible de calculer exactement les prédictions réalisées par le modèle. Un bon exemple ici est le cas de la variable *âge*. À quelle différence pouvons-nous nous attendre entre deux individus identiques ayant seulement une différence d'âge de 15 ans ?

Prenons comme individu un homme de 30 ans, vivant dans une grande ville allemande, ayant un niveau d'éducation de niveau secondaire, employé, dans la tranche de revenu moyen, déclarant effectuer un trajet de 45 minutes et ayant rapporté un niveau de conscience environnementale de 5 (sur 10). Nous pouvons prédire la probabilité qu'il utilise le vélo pour son trajet le plus fréquent en utilisant la formule suivante :

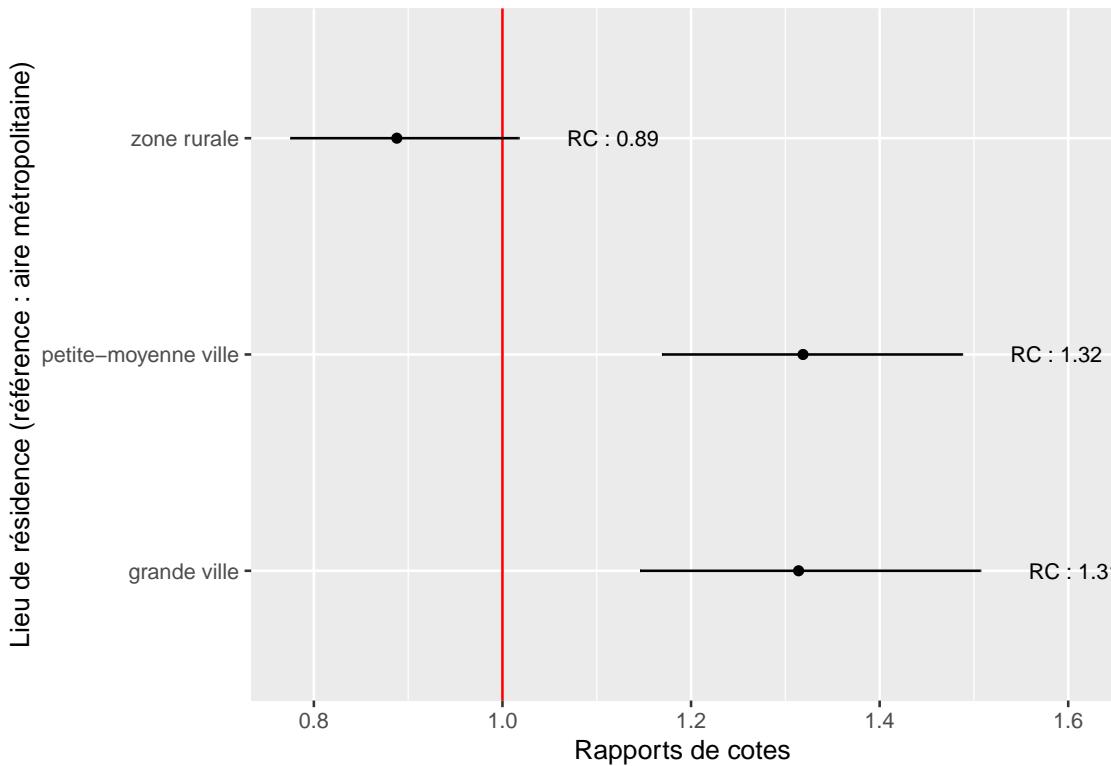


FIG. 8.10 : Rapports de cote pour les différents lieux de résidence

$$\text{logit}(p) = -2,497 + 1 \times 0,372 + 30 \times -0,009 + 1 \times 0,193 + 1 \times 0,042 + 1 \times 0,273 + 45 \times -0,001 + 5 \times 0,102$$

$$p = \exp(-2,497 + 1 \times 0,372 + 30 \times -0,009 + 1 \times 0,193 + 1 \times 0,042 + 1 \times 0,273 + 45 \times -0,001 + 5 \times 0,102) / (1 + \exp(-2,497 + 1 \times 0,372 + 30 \times -0,009 + 1 \times 0,193 + 1 \times 0,042 + 1 \times 0,273 + 45 \times -0,001 + 5 \times 0,102)) = 0,194$$

Il y aurait 19,4 % de chances pour que cette personne soit cycliste. Comme cette probabilité dépasse le seuil que nous avons sélectionné, cette personne serait classée comme cycliste. Si nous augmentons son âge de 15 ans, nous obtenons :

$$p = \exp(-2,497 + 1 \times 0,372 + 45 \times -0,009 + 1 \times 0,193 + 1 \times 0,042 + 1 \times 0,273 + 45 \times -0,001 + 5 \times 0,102) / (1 + \exp(-2,497 + 1 \times 0,372 + 45 \times -0,009 + 1 \times 0,193 + 1 \times 0,042 + 1 \times 0,273 + 45 \times -0,001 + 5 \times 0,102)) = 0,174$$

soit une réduction de 2 points de pourcentages. Il est également possible de représenter cette évolution sur un graphique pour montrer l'effet sur l'étendue des valeurs possibles. Sur ces graphiques des effets marginaux, il est essentiel de représenter l'incertitude quant à la prédiction. En temps normal, la fonction `predict` calcule directement l'erreur standard de la prédiction et cette dernière peut être utilisée pour calculer l'intervalle de confiance de la prédiction. Cependant, nous voulons ici utiliser nos erreurs standards robustes. Nous devons donc procéder par simulation pour déterminer l'intervalle de confiance à 95 % de nos prédictions. Cette opération nécessite de réaliser plusieurs opérations manuellement dans R.

```
# Créer un jeu de données fictif pour la prédiction
mat <- model.matrix(model2$terms, model2$model)
age2seq <- seq(20,80)
mat2 <- matrix(mat[1,, nrow=length(age2seq), ncol=length(mat[1,]), byrow=TRUE]
colnames(mat2) <- colnames(mat)
mat2[, "Age"] <- age2seq
```

```

mat2[,"PaysBelgique"] <- 0
mat2[,"Duree"] <- 45
mat2[,"ConsEnv"] <- 5
mat2[,"StatutEmploisans emploi"] <- 0
mat2[,"Residencegrande ville"] <- 1
mat2[,"Educationsecondaire"] <- 1
mat2[,"Sexehomme"] <- 1
mat2[,"Revenumoyen"] <- 1
mat2[,"Revenufiable"] <- 0
# Calculer la prédition comme un log de rapport de cote (avec les erreurs standards)
# en multipliant les coefficient par les valeurs des données fictives
coeffs <- model2$coefficients
pred <- coeffs %*% t(mat2)
# Simulation de prédictions (toujours en log de rapport de cote)
# Étape 1 : simuler 1000 valeurs pour chaque coefficient
sim_coeffs <- lapply(1:length(coeffs), function(i){
  coef <- coeffs[[i]]
  std.err <- stdErrRobuste[[i]]
  vals <- rnorm(n = 1000, mean = coef, sd = std.err)
  return(vals)
})
mat_sim_coeffs <- do.call(rbind,sim_coeffs)
# Étape 2 : effectuer les prédictions à partir des coefficients simulés
sim_preds <- lapply(1:ncol(mat_sim_coeffs),function(i){
  temp_coefs <- mat_sim_coeffs[,i]
  temp_pred <- as.vector(temp_coefs %*% t(mat2))
  return(temp_pred)
})
mat_sim_preds <- do.call(cbind,sim_preds)
# Étape 3 : extraire les intervalles de confiance pour les simulations
intervals <- apply(mat_sim_preds,MARGIN = 1, FUN = function(vec){
  return(quantile(vec,probs = c(0.025, 0.975)))
})
# Étape 4 : récupérer tous ces éléments dans un DataFrame
df <- data.frame(
  Age = seq(20,80),
  pred = as.vector(pred),
  lower = as.vector(intervals[1,]),
  upper = as.vector(intervals[2,])
)
# Étape 5 : appliquer l'inverse de la fonction de lien pour
# obtenir les prédictions en termes de probabilité
ilink <- family(model2)$linkinv
df$prob_pred <- ilink(df$pred)
df$prob_lower <- ilink(df$lower)
df$prob_upper <- ilink(df$upper)
# Étape 6 : représenter le tout sur un graphique
ggplot(df) +
  geom_ribbon(aes(x = Age, ymax = prob_upper, ymin = prob_lower),
              fill = rgb(0.1,0.1,0.1,0.4)) +
  geom_path(aes(x = Age, y = prob_pred), color = "blue", size = 1) +
  geom_hline(yintercept = 0.15, linetype = "dashed", size = 0.7) +
  labs(x = "Âge", y = "Probabilité prédictive (intervalle de confiance à 95 %)")

```

La figure 8.11 permet de bien constater la diminution de la probabilité d'utiliser le vélo pour son trajet

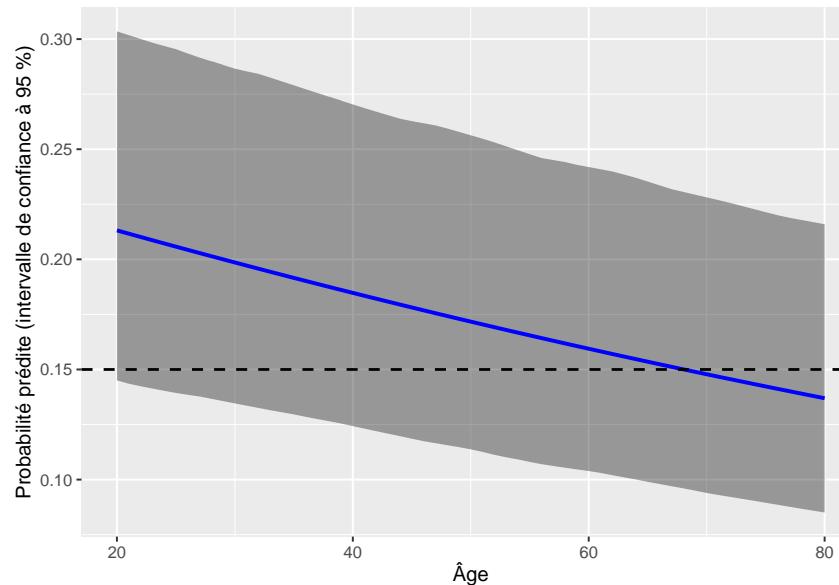


FIG. 8.11 : Effet de l'âge sur la probabilité d'utiliser le vélo comme moyen de déplacement pour son trajet le plus fréquent

le plus fréquent avec l'âge, mais cette réduction est relativement ténue. Dans le cas utilisé en exemple, l'individu ne serait plus classé cycliste qu'après 67 ans.

8.2.2 Modèle probit binomial

Le modèle GLM probit binomial est pour ainsi dire le frère du modèle logistique binomial. La seule différence entre les deux réside dans l'utilisation d'une autre fonction de lien : probit plutôt que logistique. La fonction de lien probit (Φ) correspond à la fonction cumulative de la distribution normale et a également une forme de S . Cette version du modèle est plus souvent utilisée par les économistes. Le principal changement réside dans l'interprétation des coefficients β_0 et β . Du fait de la transformation probit, ces derniers indiquent le changement en termes de scores Z de la probabilité modélisée. Vous conviendrez qu'il ne s'agit pas d'une échelle très intuitive ; la plupart du temps, seuls la significativité et le signe (positif ou négatif) des coefficients sont interprétés.

TAB. 8.11 : Carte d'identité du modèle probit binomial

Type de variable dépendante	Variable binaire (0 ou 1) ou comptage de réussite à une expérience (ex : 3 réussites sur 5 expériences)
Distribution utilisée	Binomiale
Formulation	$Y \sim Binomial(p)$ $g(p) = \beta_0 + \beta X$ $g(x) = \Phi^{-1}(x)$
Fonction de lien	probit
Paramètre modélisé	p
Paramètres à estimer	β_0, β
Conditions d'application	Non-séparation complète, absence de sur-dispersion ou de sous-dispersion

8.2.3 Modèle logistique des cotes proportionnelles

Le modèle logistique des cotes proportionnelles (aussi appelé modèle logistique cumulatif) est utilisé pour modéliser une variable qualitative ordinale. Un exemple classique de ce type de variable est une échelle de satisfaction (très insatisfait, insatisfait, mitigé, satisfait, très satisfait) qui peut être recodée avec des valeurs numériques (0, 1, 2, 3, 4; ces échelons étant notés j). Il n'existe pas à proprement parler de distribution pour représenter ces données, mais avec une petite astuce, il est possible de simplement utiliser la distribution binomiale. Cette astuce consiste à poser l'hypothèse de la proportionnalité des cotes, soit que le passage de la catégorie 0 à la catégorie 1 est proportionnel au passage de la catégorie 1 à la catégorie 2 et ainsi de suite. Si cette hypothèse est respectée, alors les coefficients du modèle pourront autant décrire le passage de la catégorie *satisfait* à celle *très satisfait* que le passage de *insatisfait* à *mitigé*. Si cette hypothèse n'est pas respectée, il faudrait des coefficients différents pour représenter les passages d'une catégorie à l'autre (ce qui est le cas pour le modèle multinomial présenté dans la section 8.2.4).

TAB. 8.12 : Carte d'identité du modèle logistique des cotes proportionnelles

Type de variable dépendante	Variable qualitative ordinale avec j catégories
Distribution utilisée	Binomiale
Formulation	$Y \sim Binomial(p)$ $g(p \leq j) = \beta_0 j + \beta X$ $g(x) = \log(\frac{x}{1-x})$
Fonction de lien	logistique
Paramètre modélisé	p
Paramètres à estimer	β et $j-1$ constantes β_{0j}
Conditions d'application	Non-séparation complète, absence de sur-dispersion ou de sous-dispersion, Proportionnalité des cotes

Ainsi, dans le modèle logistique binomial vu précédemment, nous modélisons la probabilité d'observer un événement $P(Y = 1)$. Dans un modèle logistique ordinal, nous modélisons la probabilité cumulative d'observer l'échelon j de notre variable ordinale $P(Y \leq j)$. L'intérêt de cette reformulation est que nous conservons la facilité d'interprétation du modèle logistique binomial classique avec les rapports de cotes, à ceci près qu'ils représentent maintenant la probabilité de passer à un échelon supérieur de Y . La différence pratique est que notre modèle se retrouve avec autant de constantes qu'il y a de catégories à prédire moins une, chacune de ces constantes contrôlant la probabilité de base de passer de la catégorie j à la catégorie $j+1$.

8.2.3.1 Conditions d'application

Les conditions d'application sont les mêmes que pour un modèle binomial, avec bien sûr l'ajout de l'hypothèse sur la proportionnalité des cotes. Selon cette hypothèse, l'effet de chaque variable indépendante est identique sur la probabilité de passer d'un échelon de la variable Y au suivant. Afin de tester cette condition, deux approches sont envisageables :

1. Utiliser l'approche de Brant (1990). Il s'agit d'un test statistique comparant les résultats du modèle ordinal avec ceux d'une série de modèles logistiques binomiaux (1 pour chaque catégorie possible de Y).
2. Ajuster un modèle ordinal sans l'hypothèse de proportionnalité des cotes et effectuer un test de ratio des *likelihood* pour vérifier si le premier est significativement mieux ajusté.

Si certaines variables ne respectent pas cette condition d'application, trois options sont possibles pour y remédier :

1. Supprimer la variable du modèle (à éviter si cette variable est importante dans votre cadre théorique).
2. Autoriser la variable à avoir un effet différent entre chaque palier (possible avec le *package VGAM*).
3. Changer de modèle et opter pour un modèle des catégories adjacentes. Il s'agit du cas particulier où toutes les variables sont autorisées à changer à chaque niveau. Ne pas confondre ce dernier modèle et le modèle multinomial (section 8.2.4)), puisque le modèle des catégories adjacentes continue à prédire la probabilité de passer à une catégorie supérieure.

8.2.3.2 Exemple appliqué dans R

Pour cet exemple, nous analysons un jeu de données proposé par Inside Airbnb², une organisation sans but lucratif collectant des données des annonces sur le site d'Airbnb pour alimenter le débat sur l'effet de cette société sur les quartiers. Plus spécifiquement, nous utilisons le jeu de données pour Montréal compilé le 30 juin 2020. Nous modélisons ici le prix par nuit des logements, ce type d'exercice est appelé modélisation hédonique. Il est particulièrement utilisé en économie urbaine pour évaluer les déterminants du marché immobilier et prédire son évolution. Le cas d'Airbnb a déjà été étudié dans plusieurs articles (Teubner, Hawlitschek et Dann 2017; Wang et Nicolau 2017; Zhang et al. 2017). Il en ressort notamment que le niveau de confiance inspiré par l'hôte, les caractéristiques intrinsèques du logement et sa localisation sont les principales variables indépendantes de son prix. Nous construisons donc notre modèle sur cette base. Notez que nous avons décidé de retirer les logements avec des prix supérieurs à 250 \$ par nuit qui constituent des cas particuliers et qui devraient faire l'objet d'une analyse à part entière. Nous avons également retiré les observations pour lesquelles certaines données sont manquantes, et obtenons un nombre final de 9 051 observations.

La distribution originale du prix des logements dans notre jeu de données est présentée à la figure 8.12.

²<http://insideairbnb.com/get-the-data.html>

```
# Charger le jeu de données
data_airbnb <- read.csv("data/glm/airbnb_data.csv")
# Afficher la distribution du prix
ggplot(data = data_airbnb) +
  geom_histogram(aes(x = price), bins = 30,
                 color = "white", fill = "#1d3557", size = 0.02) +
  labs(x="Prix (en dollars)", y="Fréquence")
```

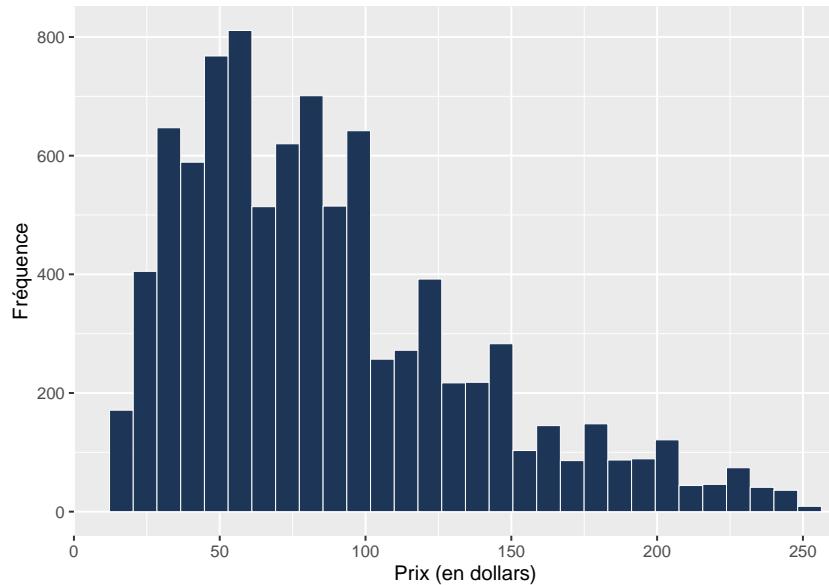


FIG. 8.12 : Distribution des prix des logements Airbnb

Nous avons ensuite découpé le prix des logements en trois catégories : inférieur à 50 \$, de 50 \$ à 99 \$ et de 100 \$ à 249 \$. Ces catégories forment une variable ordinaire de trois échelons que nous modélisons à partir de trois catégories de variables :

- les caractéristiques propres au logement;
- les caractéristiques environnementales autour du logement;
- les notes obtenues par le logement sur le site d'Airbnb.

```
# Afficher le nombre de logement par catégories
table(data_airbnb$fac_price_cat)
```

```
## 
##     1     2     3
## 2212 3911 2928
```

Le tableau 8.13 présente l'ensemble des variables utilisées dans le modèle.

TAB. 8.13 : Variables indépendantes utilisées pour prédire la catégorie de prix de logements Airbnb

Nom de la variable	Description	Type de variable	Mesure
beds	Nombre de lits dans le logement	Variable de comptage	Nombre de lits dans le logement
Garden_or_backyard	Présence d'un jardin ou d'une arrière-cour	Variable binaire	Oui ou non
private	Le logement est entièrement à disposition du locataire ou seulement une pièce	Variable binaire	Privé ou partagé
Free_street_parking	Une place de stationnement gratuite est disponible sur la rue	Variable binaire	Oui ou non
Host_greets_you	L'hôte accueille personnellement les locataires	Variable binaire	Oui ou non
prt_veg_500m	Végétation dans les environs du logement	Variable continue	Pourcentage de surface végétale dans un rayon de 500 mètres autour du logement
has_metro_500m	Présence d'une station de métro à proximité du logement	Variable binaire	Présence d'une station de métro dans un rayon de 500 mètres autour du logement
commercial_1km	Commerce dans les environs du logement	Variable continue	Pourcentage de surface dédiée au commerce (mode d'occupation du sol) dans un rayon d'un kilomètre autour du logement
cat_review	Évaluation de la qualité du logement par les usagers	Variable ordinaire	Note obtenue par le logement sur une échelle allant de 1 (très mauvais) à 5 (parfait)
host_total_listings_count	Nombre total de logements détenus par l'hôte sur Airbnb	Variable de comptage	Nombre total de logements détenus par l'hôte sur Airbnb

Vérification des conditions d'application

Avant d'ajuster le modèle, il convient de vérifier l'absence de multicolinéarité excessive entre les variables indépendantes.

```
# Notez que la fonction vif ne s'intéresse qu'aux variables indépendantes.
# Vous pouvez ainsi utiliser la fonction glm avec la fonction vif
# pour n'importe quel modèle glm
vif(glm(price ~ beds +
         Garden_or_backyard + Host_greets_you + Free_street_parking +
         prt_veg_500m + has_metro_500m + commercial_1km + host_total_listings_count +
         private + cat_review, data = data_airbnb))
```

##	beds	Garden_or_backyard	Host_greets_you
##	1.123595	1.079324	1.046884
##	Free_street_parking	prt_veg_500m	has_metro_500m
##	1.142189	1.532536	1.239516
##	commercial_1km	host_total_listings_count	private
##	1.225301	1.058232	1.143932
##	cat_review		
##	1.015778		

Toutes les valeurs de VIF sont inférieures à 2, indiquant une absence de multicolinéarité excessive. Nous pouvons alors ajuster le modèle et analyser les distances de Cook afin de vérifier la présence ou non d'observations très influentes. Pour ajuster le modèle, nous utilisons le package VGAM et la fonction vglm qui nous donnent accès à la famille cumulative pour ajuster des modèles logistiques ordinaires. Notez que le fonctionnement de base de cette famille est de modéliser $P(Y \leq 1), P(Y \leq 2), \dots, P(Y \leq J)$ avec

J le nombre de catégories. Cependant, nous voulons ici modéliser la probabilité de passer à une catégorie supérieure de prix. Pour cela, il est nécessaire de spécifier le paramètre `reverse = TRUE` pour la famille `cumulative` (voir `help(cumulative)` pour plus de détails).

```
library(VGAM)
modele <- vglm(fac_price_cat ~ beds +
  Garden_or_backyard + Free_street_parking +
  prt_veg_500m + has_metro_500m + commercial_1km +
  private + cat_review + host_total_listings_count ,
  family = cumulative(link="logitlink", # fonction de lien
  parallel = TRUE, # cote proportionnelle
  reverse = TRUE),
  data = data_airbnb, model = T)
```

Notez que, puisque la variable Y a trois catégories différentes et que nous modélisons la probabilité de passer à une catégorie supérieure, chaque observation a alors deux (3-1) valeurs de résidus différentes. Par conséquent, nous calculons deux distances de Cook différentes que nous devons analyser conjointement. Malheureusement, la fonction `cook.distance` ne fonctionne pas avec les objets `vglm`, nous devons donc les calculer manuellement.

```
# Extraction des résidus
res <- residuals(modele, type = "pearson")
# Extraction de la hat matrix (nécessaire pour calculer la distance de Cook)
hat <- hatvaluesvlm(modele)
# Calcul des distances de Cook
cooks <- lapply(1:ncol(res), function(i){
  r <- res[,i]
  h <- hat[,i]
  cook <- (r/(1 - h))^2 * h/(1 * modele@rank)
})
# Structuration dans un DataFrame
matcook <- data.frame(do.call(cbind, cooks))
names(matcook) <- c("dist1","dist2")
matcook$oid <- 1:nrow(matcook)
# Afficher les distances de Cook
plot1 <- ggplot(data = matcook) +
  geom_point(aes(x = oid, y = dist1), size = 0.2, color = rgb(0.1,0.1,0.1,0.4)) +
  labs(x = "", y = "", subtitle = "distance de Cook P(Y>=2)")+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank())
plot2 <- ggplot(data = matcook) +
  geom_point(aes(x = oid, y = dist2), size = 0.2, color = rgb(0.1,0.1,0.1,0.4)) +
  labs(x = "", y = "", subtitle = "distance de Cook P(Y>=3)")+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank())

ggarrange(plot1, plot2, ncol = 2, nrow = 1)
```

Les distances de Cook (figure 8.13) nous permettent d'identifier quelques observations potentiellement trop influentes, mais elles semblent être différentes d'un graphique à l'autre. Nous décidons donc de ne pas retirer d'observations à ce stade et de passer à l'analyse des résidus simulés. Pour effectuer des simulations à partir de ce modèle, nous nous basons sur les probabilités d'appartenance prédites par le modèle.

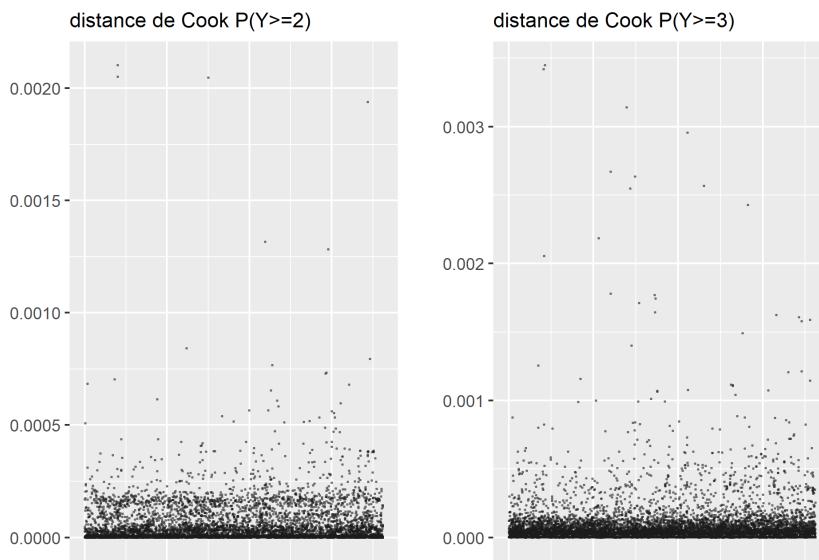


FIG. 8.13 : Distances de Cook pour le modèle logistique des cotes proportionnelles

```
# Extraire les probabilités prédites
predicted <- predict(modele,type = "response")
round(head(predicted,n = 4),3)
```

```
##      1     2     3
## 1 0.706 0.267 0.028
## 2 0.073 0.461 0.466
## 3 0.687 0.283 0.030
## 4 0.049 0.383 0.568
```

Nous constatons ainsi que, pour la première observation, la probabilité prédite d'appartenir au groupe 1 est de 69,4 %, de 27,7 % pour le groupe 2 et de 2,9 % pour le groupe 3. Si nous effectuons 1 000 simulations, nous pouvons nous attendre à ce qu'en moyenne, sur ces 1 000 simulations, 694 indiqueront 1 comme catégorie prédite, 277 indiqueront 2 et seulement 29 indiqueront 3.

```
# Nous effectuerons 1000 simulations
nsim <- 1000
# Lancement des simulations pour chaque observation (lignes dans predicted)
simulations <- lapply(1:nrow(predicted), function(i){
  probs <- predicted[i,]
  sims <- sample(c(1,2,3), size = nsim, replace = T, prob = probs)
  return(sims)
})
# Combiner les prédictions dans un tableau
matsim <- do.call(rbind, simulations)
# Observons si nos simulations sont proches de ce que nous attendions
table(matsim[,1])
```

```
##
##      1     2     3
## 703 271   26
```

À partir de ces simulations de prédiction, nous pouvons réaliser un diagnostic des résidus simulés grâce au package DHARMA.

```
library(DHARMA)
# Extraction de la prédiction moyenne du modèle
pred_cat <- unique(data_airbnb$fac_price_cat)[max.col(predicted)]
# Préparer les données avec le package DHARMA
sim_res <- createDHARMA(simulatedResponse = matsim,
                          observedResponse = as.numeric(data_airbnb$fac_price_cat),
                          fittedPredictedResponse = as.numeric(pred_cat),
                          integerResponse = T)
# Afficher le graphique de diagnostic général
plot(sim_res)
```

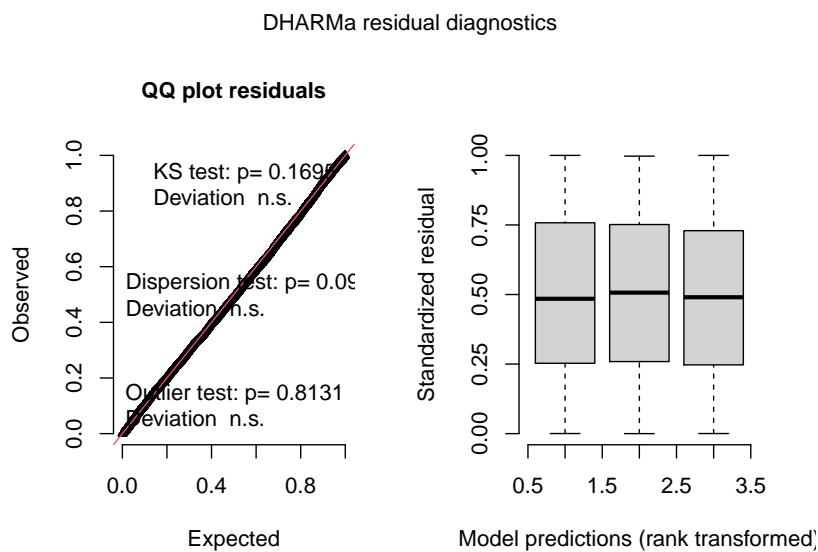


FIG. 8.14 : Diagnostic général des résidus simulés du modèle des cotes proportionnelles

La figure 8.14 nous indique que les résidus simulés suivent bien une distribution uniforme et qu'aucune valeur aberrante n'est observable. Pour affiner notre diagnostic, nous vérifions également si aucune relation ne semble exister entre chaque variable indépendante et les résidus.

```
# Préparons un plot multiple
par(mfrow=c(3,4))
vars <- c("nombre de lits" = "beds",
        "couvert végétal" = "prt_veg_500m",
        "commercial" = "commercial_1km",
        "nb logements hôte" = "host_total_listings_count",
        "jardin" = "Garden_or_backyard",
        "accueil" = "Host_greets_you",
        "stationnement gratuit" = "Free_street_parking",
        "métro" = "has_metro_500m",
        "logement privé" = "private",
        "évaluation" = "cat_review")
```

```
for(name in names(vars)){
```

```

v <- vars[[name]]
plotResiduals(sim_res, data_airbnb[[v]], rank = F, quantreg = F, main = "",
               xlab = name, ylab = "résidus")
}

```

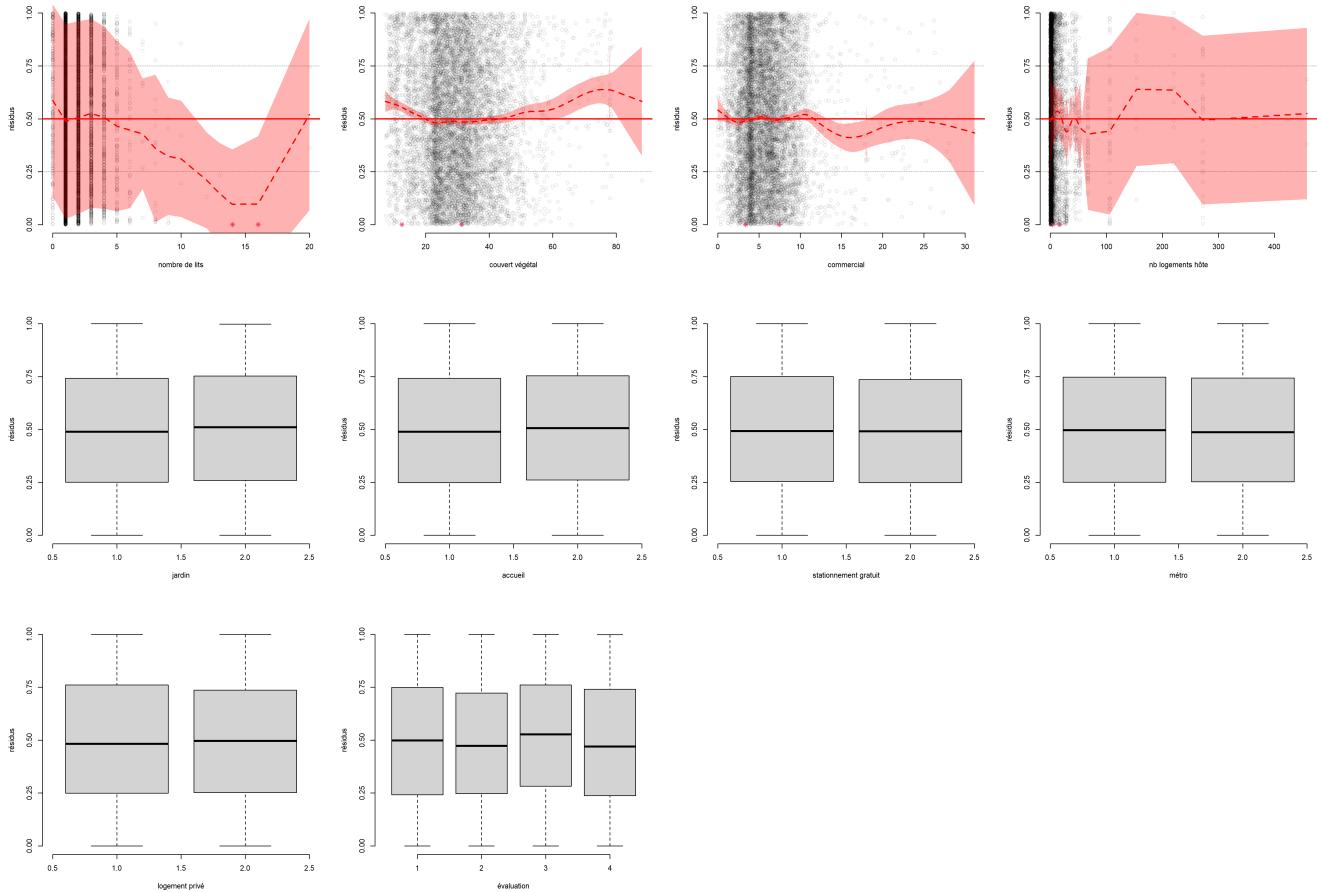


FIG. 8.15 : Diagnostic des variables indépendantes et des résidus simulés du modèle des cotes proportionnelles

La fonction `plotResiduals` du package DHARMA produit des graphiques peu esthétiques, mais pratiques pour effectuer ce type de diagnostic.

La figure 8.15 indique qu'aucune relation marquée n'existe entre nos variables indépendantes et nos résidus simulés, sauf pour la variable *nombre de lits*. En effet, nous pouvons constater que les résidus ont tendance à être toujours plus faibles quand le nombre de lits augmente. Cet effet est sûrement lié au fait qu'au-delà de cinq lits, le logement en question est vraisemblablement un dortoir. Il pourrait être judicieux de retirer ces observations de l'analyse, considérant qu'elles sont peu nombreuses et constituent un type de logement particulier.

```

data_airbnb2 <- subset(data_airbnb, data_airbnb$beds <=5)
modele2 <- vglm(fac_price_cat ~ beds +
                  Garden_or_backyard + Free_street_parking +
                  prt_veg_500m + has_metro_500m + commercial_1km +

```

```

    private + cat_review + host_total_listings_count ,
family = cumulative(link="logitlink", # fonction de lien
                     parallel = TRUE, # cote proportionnelle
                     reverse = TRUE),
data = data_airbnb2, model = T)

```

Nous pouvons ensuite recalculer les résidus simulés pour observer si cette tendance a été corrigée. La figure 8.16 montre qu'une bonne partie du problème a été corrigée; cependant, il semble tout de même que les résidus soient plus forts pour les logements avec un seul lit.

```

# Nous effectuerons 1000 simulations
nsim <- 1000
predicted <- predict(modele2, type = "response")
# Lancement des simulations pour chaque observation (lignes dans predicted)
simulations <- lapply(1:nrow(predicted), function(i){
  probs <- predicted[i,]
  sims <- sample(c(1,2,3), size = nsim, replace = T, prob = probs)
  return(sims)
})
# Combiner les prédictions dans un tableau
matsim <- do.call(rbind, simulations)
# Extraction de la prédiction moyenne du modèle
pred_cat <- unique(data_airbnb2$fac_price_cat)[max.col(predicted)]
# Préparer les données avec le package DHARMA
sim_res <- createDHARMA(simulatedResponse = matsim,
                         observedResponse = as.numeric(data_airbnb2$fac_price_cat),
                         fittedPredictedResponse = as.numeric(pred_cat),
                         integerResponse = T)

```

```

par(mfrow=c(3,4))
vars <- c("nombre de lits" = "beds",
        "couvert végétal" = "prt_veg_500m",
        "commercial" = "commercial_1km",
        "nb logements hôte" = "host_total_listings_count",
        "jardin" = "Garden_or_backyard",
        "accueil" = "Host_greets_you",
        "stationnement gratuit" = "Free_street_parking",
        "métro" = "has_metro_500m",
        "logement privé" = "private",
        "évaluation" = "cat_review")

for(name in names(vars)){
  v <- vars[[name]]
  plotResiduals(sim_res, data_airbnb2[[v]], rank = F, quantreg = F, main = "",
                xlab = name, ylab = "résidus")
}

```

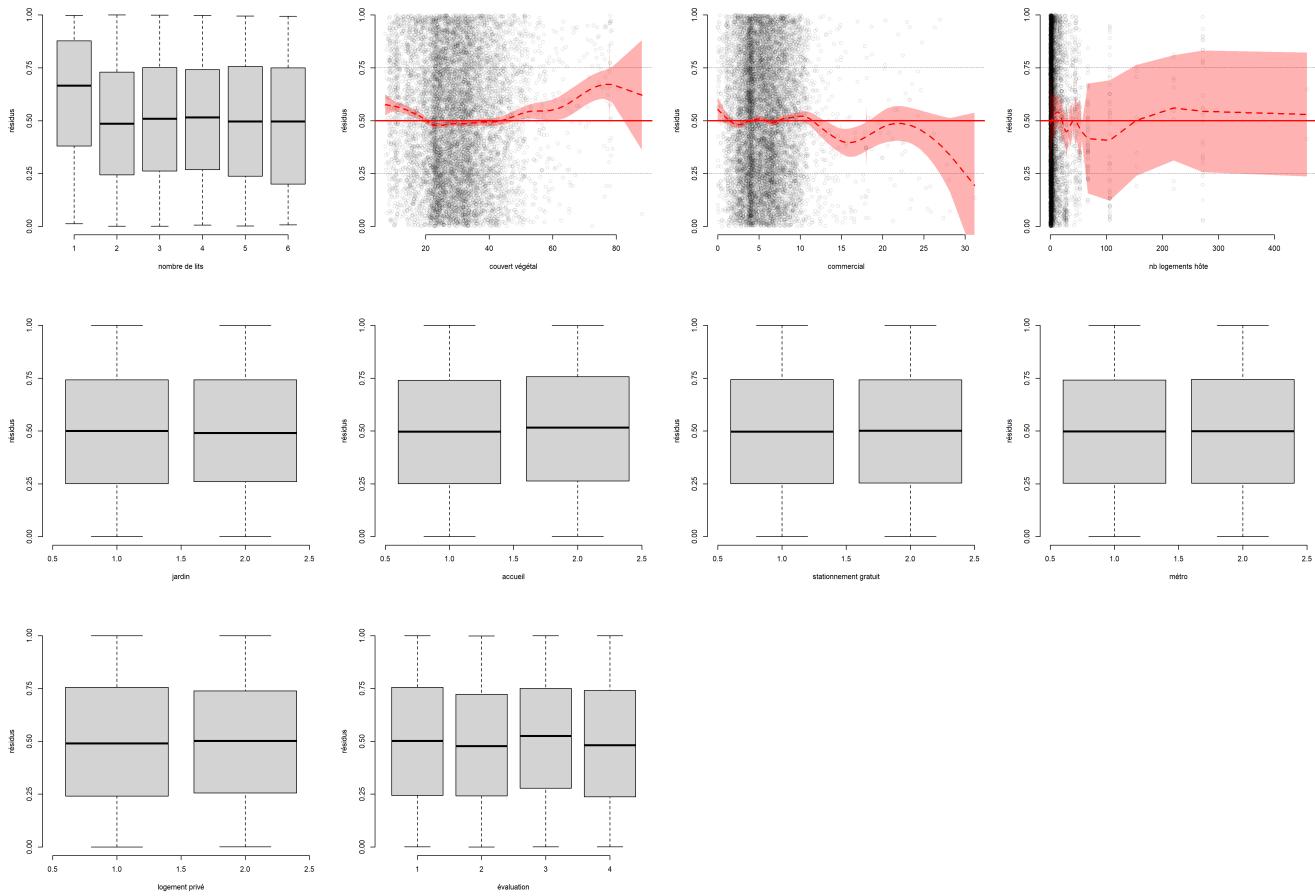


FIG. 8.16 : Diagnostic des variables indépendantes et des résidus simulés du modèle des cotes proportionnelles (après correction)

La prochaine étape du diagnostic est de vérifier l'absence de séparation parfaite provoquée par une de nos variables indépendantes. Le package `VGAM` propose pour cela la fonction `hdeff`.

```
tests <- hdeff(modele2)
problem <- table(tests)
problem
```

```
## tests
## FALSE
##    11
```

La fonction nous informe qu'aucune de nos variables indépendantes ne provoque de séparation parfaite : toutes les valeurs renvoyées par la fonction `hdeff` sont égales à `FALSE`.

Il ne nous reste donc plus qu'à vérifier que l'hypothèse de proportionnalité des cotes est respectée, soit que l'effet de chacune des variables indépendantes est bien le même pour passer de la catégorie 1 à 2 que pour passer de la catégorie 2 à 3. Pour cela, deux approches sont possibles : le test de Brant ou la réalisation d'une séquence de tests de rapport de vraisemblance.

Le package `brant` propose une implémentation du test de Brant, mais celle-ci ne peut être appliquée qu'à des modèles construits avec la fonction `polr` du package `MASS`. Nous avons donc récupéré le code source

de la fonction `brant` du package `brant` et apporté quelques modifications pour qu'elle soit utilisable sur un objet `vglm`. Cette nouvelle fonction, appelée `brant.vglm`, est disponible dans le code source de ce livre.

```
tableau_brant <- round(brant.vglm(modele2), 3)

## -----
## Test for      X2  df  probability
## -----
## Omnibus       113.72  9   0
## beds          0.63   1   0.43
## Garden_or_backyardYES 0.16   1   0.69
## Free_street_parkingYES 0.84   1   0.36
## prt_veg_500m    6.49   1   0.01
## has_metro_500mYES 0.02   1   0.89
## commercial_1km 0.01   1   0.92
## privateEntier  93.56   1   0
## cat_review     0.63   1   0.43
## host_total_listings_count 1.51   1   0.22
## -----
## 
## H0: Parallel Regression Assumption holds
```

Ce premier tableau nous indique que seule la variable indiquant si le logement est disponible en entier ou partagé contrevient à l'hypothèse de proportionnalité des cotes (la seule valeur p significative). Pour confirmer cette observation, nous pouvons réaliser un ensemble de tests de rapport de vraisemblance. Pour chaque variable du modèle, nous créons un second modèle dans lequel cette variable est autorisée à varier pour chaque catégorie et nous comparons les niveaux d'ajustement des modèles. Nous avons implémenté cette procédure dans la fonction `parallel.likelihoodtest.vglm` disponible dans le code source de ce livre.

```
tableau_likelihood <- parallel.likelihoodtest.vglm(modele2, verbose = FALSE)
print(tableau_likelihood)
```

	variable	non_parallel	AIC	loglikelihood	p.val	loglikelihood	ratio	test
## 1	beds	15304	-7640				0.271	
## 2	Garden_or_backyard	15305	-7641				0.417	
## 3	Free_street_parking	15301	-7639				0.079	
## 4	prt_veg_500m	15296	-7636				0.007	
## 5	has_metro_500m	15303	-7640				0.191	
## 6	commercial_1km	15305	-7640				0.271	
## 7	private	15215	-7595				0.000	
## 8	cat_review	15306	-7641				0.430	
## 9	host_total_listings_count	15304	-7640				0.257	

Les résultats de cette seconde série de tests confirment les précédents, la variable concernant le type de logement doit être autorisée à varier en fonction de la catégorie. Ce second tableau nous indique que la variable concernant la densité de végétation pourrait aussi être amenée à varier en fonction du groupe, mais ce changement a un effet très marginal (différence entre les valeurs d'AIC de seulement 8 points). Nous ajustons donc un nouveau modèle autorisant la variable `private` à changer en fonction de la catégorie prédictive.

```
modele3 <- vglm(fac_price_cat ~ beds +
  Garden_or_backyard + Host_greets_you + Free_street_parking +
  prt_veg_500m + has_metro_500m + commercial_1km +
  private + cat_review + host_total_listings_count,
  family = cumulative(link="logitlink",
  parallel = FALSE ~ private ,
  reverse = TRUE),
  data = data_airbnb2, model = T)
```

Vérification l'ajustement du modèle

Maintenant que toutes les conditions d'application ont été passées en revue, nous pouvons passer à la vérification de l'ajustement du modèle.

```
modelenull <- vglm(fac_price_cat ~ 1,
  family = cumulative(link="logitlink",
  parallel = TRUE,
  reverse = TRUE),
  data = data_airbnb2, model = T)
rsqs(loglike.full = logLik(modele3),
  loglike.null = logLik(modelenull),
  full.deviance = deviance(modele3),
  null.deviance = deviance(modelenull),
  nb.params = modele3@rank,
  n = nrow(data_airbnb2)
)
```

```
## $`deviance expliquee`
## [1] 0.2087098
##
## $`McFadden ajuste`
## [1] 0.2073552
##
## $`Cox and Snell`
## [1] 0.3606848
##
## $Nagelkerke
## [1] 0.4085924
```

Le modèle final parvient à expliquer 21 % de la déviance originale. Il obtient un R^2 ajusté de McFadden de 0,21, et des R^2 de Cox et Snell et de Nagelkerke de respectivement 0,36 et 0,41. Construisons à présent la matrice de confusion de la prédiction du modèle (nous utilisons ici la fonction `nice_confusion_matrix` également disponible dans le code source de ce livre).

```
preds_probs <- fitted(modele3)
pred_cat <- c(1,2,3)[max.col(preds_probs)]
library(caret)
matrices <- nice_confusion_matrix(data_airbnb2$fac_price_cat,pred_cat)
# Afficher la matrice de confusion
print(matrices$confusion_matrix)
```

	rowsnames	1	2	3	rs	rp
--	-----------	---	---	---	----	----

```

## colsnames ""           "1 (reel)" "2 (reel)" "3 (reel)" "Total" "%"
## 1      "1 (predit)" "1576"     "736"      "168"      "2480" "27.7"
## 2      "2 (predit)" "579"      "2385"     "1331"     "4295" "48"
## 3      "3 (predit)" "49"       "771"      "1360"     "2180" "24.3"
##      "Total"      "2204"     "3892"     "2859"     "8955" NA
##      "%"         "24.6"     "43.5"     "31.9"    NA      NA

```

```

# Afficher les indicateurs de qualité de prédiction
print(matrices$indicators)

```

	rnames	precision	rappel	F1
## 1	"1"	"0.64"	"0.72"	"0.67"
## 2	"2"	"0.56"	"0.61"	"0.58"
## 3	"3"	"0.62"	"0.48"	"0.54"
## macro_scores	"macro"	"0.6"	"0.59"	"0.59"
##	"Kappa"	"0.37"	NA	NA
##	"Valeur de p (precision > NIR)"	"0"	NA	NA

Le modèle a une précision totale de 61 % (61 % des observations ont été correctement prédictes). La catégorie 1 a de loin la meilleure précision (72 %) et la 3 a la pire (48 %), ce qui indique qu'il manque vraisemblablement des variables indépendantes contribuant à prédire les prix des logements les plus chers. Le coefficient de Kappa () indique un niveau d'accord entre modéré et faible, mais le modèle parvient à une prédiction significativement supérieure au seuil de non-information. Si l'ajustement du modèle est imparfait, il est suffisamment fiable pour nous donner des renseignements pertinents sur le phénomène étudié.

Interprétation des résultats

L'ensemble des coefficients du modèle sont accessibles via la fonction `summary`. À partir des coefficients et de leurs erreurs standards, il est possible de calculer les rapports de cotes ainsi que leurs intervalles de confiances.

```

tableau <- summary(modele3)$coef3
rappCote <- exp(tableau[,1])
rappCote2.5 <- exp(tableau[,1] - 1.96 * tableau[,2])
rappCote97.5 <- exp(tableau[,1] + 1.96 * tableau[,2])
tableau <- cbind(tableau, rappCote, rappCote2.5, rappCote97.5)
print(round(tableau,3))

```

	Estimate	Std. Error	z value	Pr(> z)	rappCote
## (Intercept):1	-1.203	0.137	-8.768	0.000	0.300
## (Intercept):2	-3.240	0.151	-21.516	0.000	0.039
## beds	0.748	0.027	28.143	0.000	2.113
## Garden_or_backyardYES	0.120	0.067	1.795	0.073	1.128
## Host_greets_youYES	0.094	0.060	1.548	0.122	1.098
## Free_street_parkingYES	-0.015	0.046	-0.320	0.749	0.985
## prt_veg_500m	-0.026	0.003	-10.182	0.000	0.975
## has_metro_500mYES	0.063	0.048	1.326	0.185	1.065
## commercial_1km	0.008	0.008	0.955	0.340	1.008
## privateEntier:1	2.445	0.062	39.241	0.000	11.533
## privateEntier:2	1.576	0.081	19.573	0.000	4.834
## cat_review	0.200	0.021	9.563	0.000	1.222

```

## host_total_listings_count -0.002      0.001   -2.094     0.036    0.998
##                               rappCote2.5 rappCote97.5
## (Intercept):1             0.230      0.393
## (Intercept):2             0.029      0.053
## beds                      2.006      2.226
## Garden_or_backyardYES    0.989      1.287
## Host_greets_youYES       0.975      1.236
## Free_street_parkingYES   0.900      1.078
## prt_veg_500m              0.970      0.980
## has_metro_500mYES         0.970      1.170
## commercial_1km             0.992      1.024
## privateEntier:1           10.207     13.031
## privateEntier:2            4.128      5.660
## cat_review                 1.173      1.273
## host_total_listings_count 0.996      1.000

```

Pour faciliter la lecture des résultats, nous proposons le tableau 8.14.

Sans surprise, chaque lit supplémentaire contribue à augmenter les chances que le logement soit dans une catégorie de prix supérieure (multiplication par deux à chaque lit supplémentaire). En revanche, la présence d'un stationnement gratuit, d'un jardin et l'accueil en personne par l'hôte n'ont pas d'effets significatifs. Comme l'indiquent les articles mentionnés en début de section, les revues positives augmentent la probabilité d'appartenir à une catégorie supérieure de prix. Pour chaque point supplémentaire sur l'échelle de 1 à 5, la probabilité d'appartenir à une catégorie de prix supérieure augmente de 22,2 %. Il est intéressant de noter que le fait de disposer du logement entier plutôt que d'une simple chambre augmente davantage les chances de passer du groupe de prix 1 à 2 (multiplication par 2,45) que du groupe 2

TAB. 8.14 : Coefficients du modèle logistique des cotes proportionnelles

Variable	Coefficient	RC	P	RC 2,5 %	RC 97,5 %	sign.
(Intercept):1	-1,200	0,300	0,000	0,230	0,395	***
(Intercept):2	-3,240	0,039	0,000	0,029	0,052	***
beds	0,750	2,113	0,000	2,014	2,226	***
<i>Garden_or_backyard</i>						
ref : NO	—	—	—	—	—	—
YES	0,120	1,128	0,073	0,990	1,284	.
<i>Host_greets_you</i>						
ref : NO	—	—	—	—	—	—
YES	0,090	1,098	0,122	0,980	1,234	
<i>Free_street_parking</i>						
ref : NO	—	—	—	—	—	—
YES	-0,010	0,985	0,749	0,896	1,083	
<i>prt_veg_500m</i>						
has_metro_500m	-0,030	0,975	0,000	0,970	0,980	***
<i>ref : NO</i>	—	—	—	—	—	—
YES	0,060	1,065	0,185	0,970	1,174	
<i>commercial_1km</i>						
ref : Chambre	—	—	—	—	—	—
Entier :1	2,450	11,533	0,000	10,176	13,066	***
Entier :2	1,580	4,834	0,000	4,137	5,641	***
Effets par niveau						
<i>private</i>						
ref : Chambre	—	—	—	—	—	—
Entier :1	2,450	11,533	0,000	10,176	13,066	***
Entier :2	1,580	4,834	0,000	4,137	5,641	***

à 3 (multiplication par 1,58). Il semble également que si l'hôte possède plusieurs logements, la probabilité d'avoir une classe de prix supérieure diminue légèrement. Cependant, l'effet est trop petit pour pouvoir se livrer à des interprétations.

Les variables environnementales ont peu d'effet : le pourcentage de surface commerciale dans un rayon d'un kilomètre et la présence d'une station de métro ne sont pas significatifs. En revanche, une augmentation de la surface végétale dans un rayon de 500 mètres tend à réduire la probabilité d'appartenir à une classe supérieure. Notre hypothèse concernant ce résultat est que cette variable représente un effet associé à la localisation des Airbnb, les plus centraux ayant tendance à être plus dispendieux, mais avec un environnement moins vert et inversement. Pour l'illustrer, prédisons les probabilités d'appartenance aux différents niveaux de prix d'un logement avec les caractéristiques suivantes : entièrement privé, 2 lits, un jardin, une place de stationnement gratuite, l'hôte ne dispose que d'un logement sur Airbnb et accueille les arrivants en personne, 10 % de surface commerciale dans un rayon d'un kilomètre, noté 2 comme catégorie de revue, absence de métro dans un rayon de 500 mètres.

```
# Créer un jeu de données pour effectuer des prédictions
df <- data.frame(
  prt_veg_500m = seq(5,90),
  beds = 2,
  Garden_or_backyard = "YES",
  Host_greets_you = "YES",
  Free_street_parking = "YES",
  has_metro_500m = "NO",
  commercial_1km = 10,
  private = "Entier",
  cat_review = 2,
  host_total_listings_count = 1
)
# Effectuer les prédictions (dans l'échelle log)
preds <- predict(modele3, newdata = df, type = "link", se.fit=T)
# Définir l'inverse de la fonction de lien
ilink <- function(x){exp(x)/(1+exp(x))}
# Calculer les probabilités et leurs intervalles de confiance
df[["P[Y>=2]"]] <- ilink(preds$fitted.values[,1])
df[["P[Y>=2] 2,5%"]] <- ilink(preds$fitted.values[,1] - 1.96 * preds$se.fit[,1])
df[["P[Y>=2] 97,5%"]] <- ilink(preds$fitted.values[,1] + 1.96 * preds$se.fit[,1])
df[["P[Y>=3]"]] <- ilink(preds$fitted.values[,2])
df[["P[Y>=3] 2,5%"]] <- ilink(preds$fitted.values[,2] - 1.96 * preds$se.fit[,2])
df[["P[Y>=3] 97,5%"]] <- ilink(preds$fitted.values[,2] + 1.96 * preds$se.fit[,2])
df[["P[Y=1]"]] = 1-df[["P[Y>=2]"]]
df[["P[Y=1] 2,5%"]] = 1-df[["P[Y>=2] 2,5%"]]
df[["P[Y=1] 97,5%"]] = 1-df[["P[Y>=2] 97,5%"]]
# Afficher les résultats
ggplot(data = df) +
  geom_ribbon(aes(x = prt_veg_500m,
                  ymin = `P[Y>=2] 2,5%`,
                  ymax = `P[Y>=2] 97,5%`), fill ="#f94144", alpha = 0.4) +
  geom_path(aes(x = prt_veg_500m, y = `P[Y>=2]`,color="Y2")) +
  geom_ribbon(aes(x = prt_veg_500m,
                  ymin = `P[Y>=3] 2,5%`,
                  ymax = `P[Y>=3] 97,5%`), fill ="#90be6d", alpha = 0.4) +
  geom_path(aes(x = prt_veg_500m, y = `P[Y>=3]`, color = "Y3" )) +
  geom_ribbon(aes(x = prt_veg_500m,
                  ymin = `P[Y=1] 2,5%`,
```

```

ymax = `P[Y=1] 97,5`),fill ="#277da1" , alpha = 0.4)+
geom_path(aes(x = prt_veg_500m, y = `P[Y=1]` , color = "Y1")) +
scale_color_manual(name = "Probabilités prédites",
                    breaks = c("Y1", "Y2", "Y3"),
                    labels = c("P[Y=1]", "P[Y>=2]", "P[Y>=3]"),
                    values = c("Y2" = "#f94144", "Y3" = "#90be6d",
                               "Y1" = "#277da1")) +
labs(x = "Densité de végétation (%)",
     y = "Probabilité",
     subtitle = "Y1 : moins de 50 $; Y2 : 50 à 99 $; Y3 : 100 à 249 $")

```

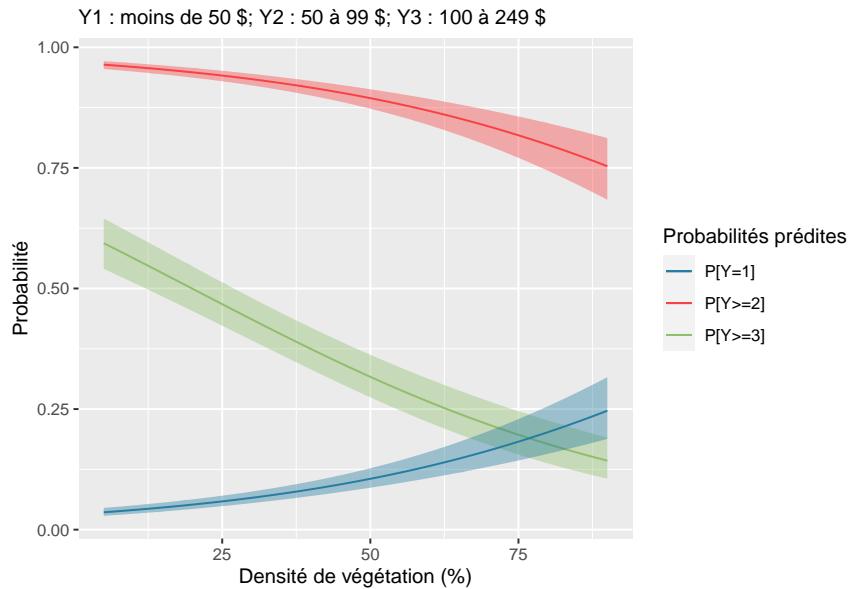


FIG. 8.17 : Prédiction de la probabilité d'appartenance aux trois catégories de prix en fonction de la densité de végétation

Nous constatons, à la figure 8.17, que les probabilités d'appartenir aux niveaux 2 et 3 diminuent à mesure qu'augmente le pourcentage de végétation. La probabilité d'appartenir à la classe 2 et plus (en rouge) passe de plus de 95 % en cas d'absence de végétation et à environ 75 % avec 80 % de végétation dans un rayon de 500 mètres. Comme vous pouvez le constater, la probabilité $P[Y = 1]$ est la symétrie de $P[Y \geq 2]$ puisque $P[Y = 1] + P[Y \geq 2] = 1$.

8.2.4 Modèle logistique multinomial

La régression logistique multinomiale est utilisée pour modéliser une variable Y qualitative multinomiale, c'est-à-dire une variable dont les modalités ne peuvent pas être ordonnées. Dans le modèle précédent, nous avons vu qu'il était possible de modéliser une variable ordinaire avec une distribution binomiale en formulant l'hypothèse de la proportionnalité des cotes. Avec une variable multinomiale, cette hypothèse ne tient plus, car les catégories ne sont plus ordonnées. Il faut donc formuler le modèle différemment.

L'idée derrière un modèle multinomial est de choisir une catégorie de référence, puis de modéliser les probabilités d'appartenir à chaque autre catégorie plutôt qu'à cette catégorie de référence (tableau 8.15). Si nous avons K catégories possibles dans notre variable Y , nous obtenons $K-1$ comparaisons. Chaque comparaison est modélisée avec sa propre équation, ce qui génère de nombreux paramètres. Par exemple, admettons que notre variable Y a cinq catégories et que nous disposons de six variables X prédictives.

Nous avons ainsi 4 (5-1) équations de régression avec 7 paramètres (6 coefficients et une constante), soit 28 coefficients à analyser.

Considérant cette tendance à la multiplication des coefficients, il est fréquent de recourir à une méthode appelée *Analyse de type 3* pour limiter au maximum le nombre de variables indépendantes (VI) dans le modèle. L'idée de cette méthode est de recalculer plusieurs versions du modèle dans lesquelles une variable indépendante est retirée, puis de réaliser un test de rapport de vraisemblance en comparant ce nouveau modèle (complet moins une VI) au modèle complet (toutes les VI) pour vérifier si la variable en question améliore significativement le modèle. Il est alors possible de retirer toutes les variables dont l'apport est négligeable si elles sont également peu intéressantes du point de vue théorique.

TAB. 8.15 : Carte d'identité du modèle logistique multinomial

Type de variable dépendante	Variable qualitative multinomiale avec K catégories
Distribution utilisée	Binomiale
Formulation	$Y \sim Binomial(p)$ $g(p = k \text{ avec } ref = a) = \beta_{0k} + \beta X_k$ $g(x) = \frac{\log(x)}{1-x}$
Fonction de lien	Logistique
Paramètre modélisé	p
Paramètres à estimer	β_{0k}, β_k pour $k \in [2, \dots, K]$
Conditions d'application	Non-séparation complète, absence de sur-dispersion ou de sous-dispersion, Indépendance des alternatives non pertinentes

8.2.4.1 Conditions d'application

Les conditions d'application sont les mêmes que pour un modèle binomial, avec l'ajout de l'hypothèse sur **l'indépendance des alternatives non pertinentes**. Cette dernière suppose que le choix entre deux catégories est indépendant des catégories proposées. Voici un exemple simple pour illustrer cette hypothèse : admettons que nous disposons d'une trentaine de personnes et que nous leur demandons la couleur de leurs yeux. Cette variable ne serait pas affectée par la présence de nouvelles couleurs en dehors de notre échantillon. En revanche, si nous leur demandons de choisir un mode de transport parmi une liste pour se rendre à leur lieu de travail, leur réponse serait nécessairement affectée par la liste des modes de transport disponibles. Les tests développés pour vérifier cette hypothèse sont connus pour leur faible fiabilité³. Il est plus pertinent de décider théoriquement si cette hypothèse est valide ou non. Dans le cas contraire, il est possible d'utiliser une classe de modèle logistique plus rare : le modèle logistique imbriqué.

Notez également que le grand nombre de paramètres dans ce type de modèle implique de disposer d'un plus grand nombre d'observations afin d'avoir suffisamment d'information dans chaque catégorie pour ajuster tous les paramètres.

Pour vérifier la présence de sur-dispersion, il est possible, dans le cas du modèle multinomial, de calculer le rapport entre le khi-deux de Pearson et le nombre de degrés de liberté du modèle. Si ce rapport est supérieur à 1 (des valeurs jusqu'à 1,15 ne sont pas problématiques), alors le modèle souffre de sur-dispersion (SAS Institute Inc 2020a). Le khi-deux de Pearson est simplement la somme des résidus de Pearson au carré dans le cas d'un modèle GLM.

$$\chi^2 = \sum_{i=1}^N \sum_{c=1}^K \frac{(y_{ic} - p_{ic})^2}{p_{ic}} \quad (8.20)$$

Avec y_{ic} 1 si l'observation i appartient à la catégorie c , 0 autrement, p_{ic} la probabilité prédictive pour l'observation i d'appartenir à la catégorie c , N le nombre d'observations et K le nombre de catégories.

³<https://statisticalhorizons.com/iia>

Le ratio est ensuite calculé comme suit : $\frac{\chi^2}{(N-p)(K-1)}$, avec N le nombre d'observations et K le nombre de modalités dans la variable Y . Si ce ratio est égal ou supérieur à 1, alors le modèle souffre de sur-dispersion, si le ratio est inférieur à 1, le modèle souffre de sous-dispersion. Un léger écart ($> 0,15$) n'est pas considéré comme problématique.



Le modèle logistique imbriqué

Du fait de sa proximité avec les modèles à effets mixtes que nous abordons au chapitre 9, nous ne détaillons pas ici le modèle logistique imbriqué, mais présentons plutôt son principe général. Il s'agit d'une généralisation du modèle logistique multinomial basé sur l'idée que certaines catégories pourraient être regroupées dans des « nids » (*nest* en anglais). Dans ces groupes, les erreurs peuvent être corrélées, indiquant ainsi que si une catégorie est manquante, une autre catégorie du même groupe sera préférée. Un paramètre λ contrôle spécifiquement cette corrélation et permet de mesurer sa force une fois le modèle ajusté. Il peut être pertinent de comparer un modèle imbriqué à un modèle multinomial pour déterminer lequel des deux est le mieux ajusté aux données.

8.2.4.2 Exemple appliqué dans R

Pour cet exemple, nous reproduisons une partie de l'analyse effectuée dans l'étude de McFadden (2016). Cet article s'intéresse aux écarts entre les croyances des individus et les connaissances scientifiques sur les sujets des OGM et du réchauffement climatique. Les auteurs utilisent pour cela des données issues d'une enquête auprès de 961 individus formant un échantillon représentatif de la population des États-Unis. Les données issues de cette enquête sont téléchargeables sur le site de l'éditeur⁴, ce qui nous permet ici de reproduire l'analyse effectuée par les auteurs. Deux questions sont centrales dans l'enquête :

- Dans quelle mesure êtes-vous en accord ou en désaccord avec la phrase suivante : les plantations génétiquement modifiées sont sans danger pour la consommation ?
- Dans quelle mesure êtes-vous en accord ou en désaccord avec la phrase suivante : la Terre se réchauffe du fait des activités humaines ?

Pour ces deux questions, les répondants devaient sélectionner leur degré d'accord sur une échelle de Likert allant de 1 (fortement en désaccord) à 5 (fortement en accord). Les réponses à ces deux questions ont été utilisées pour former une variable multinomiale à quatre modalités :

- A. Les individus sont en accord avec les deux propositions.
- B. Les individus sont en désaccord sur les OGM, mais en accord sur le réchauffement climatique.
- C. Les individus sont en accord sur les OGM, mais en désaccord sur le réchauffement climatique.
- D. Les individus sont en désaccord avec les deux propositions.

Un modèle logistique multinomial a été utilisé pour déterminer quels facteurs contribuent à la probabilité d'appartenir à ces différentes catégories. Les variables indépendantes présentes dans le modèle sont détaillées dans le tableau 8.16. Les auteurs avaient notamment conclu que :

- Les effets des connaissances (réelles ou perçues) sur l'appartenance aux différentes catégories n'étaient pas uniformes et pouvaient varier en fonction du sujet.
- L'orientation politique avait une influence significative sur les croyances.
- Les répondants avec de plus hauts résultats au test de cognition CRT avaient plus souvent des opinions divergentes de la communauté scientifique.

⁴ <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0166140>

TAB. 8.16 : Variables indépendantes utilisées dans le modèle logistique multinomial

Nom de la variable	signification	Type de variable	Mesure
PercepOGM	La recherche scientifique supporte ma vision sur la sécurité des plantes OGM	Variable ordinaire	Échelle de Likert de 1 (fortement en désaccord) à 5 (fortement en accord)
PercepRechClim	La recherche scientifique supporte ma vision sur le réchauffement climatique	Variable ordinaire	Échelle de Likert de 1 (fortement en désaccord) à 5 (fortement en accord)
ConnaisOGM	Niveau de connaissance sur les OGM	Variable ordinaire	Nombre de réponses sur trois questions portant sur les OGM
ConnaisRechClim	Niveau de connaissance sur le réchauffement climatique	Variable ordinaire	Nombre de réponses sur trois questions portant sur le réchauffement climatique
CRT	Score obtenu au Cognitive Reflection Test, utilisé pour déterminer la propension à faire preuve d'esprit d'analyse plutôt que choisir des réponses intuitives	Variable ordinaire	Nombre de réponses sur trois questions pièges
Parti	Orientation politique du répondant	Variable multinomiale	Républicain, démocrate et autre
Sexe	Sexe du répondant	Variable binaire	Femme ou homme
Age	Âge du répondant	Variable continue	Âge du répondant
Revenu	Niveau de revenu du répondant	Variable ordinaire	Échelle de 1 (moins de 20 000)(140000 et plus)

Vérification des conditions d'application

Avant d'ajuster le modèle, nous commençons par vérifier l'absence de multicolinéarité excessive entre les variables indépendantes. Toutes les valeurs de VIF sont inférieures à 2, indiquant bien une absence de multicolinéarité.

```
data_quest <- read.csv("data/glm/enquete_PublicOpinion_vs_Science.csv")
# Choix des valeurs de références dans les facteurs
data_quest$Parti <- relevel(as.factor(data_quest$Parti), ref = "Democrate")
data_quest$Sexe <- relevel(as.factor(data_quest$Sexe), ref = "homme")
vif(glm(SCIGM ~ PercepOGM + PercepRechClim + ConnaisOGM + ConnaisRechClim +
       CRT + Parti + AGE + Sexe + Revenu, data = data_quest))
```

```
##                                     GVIF Df GVIF^(1/(2*Df))
## PercepOGM      1.092693  1     1.045320
## PercepRechClim 1.177495  1     1.085125
## ConnaisOGM     1.150662  1     1.072689
## ConnaisRechClim 1.158438  1     1.076307
## CRT            1.155371  1     1.074882
## Parti          1.130817  2     1.031212
## AGE            1.071655  1     1.035208
## Sexe          1.064918  1     1.031949
## Revenu         1.049499  1     1.024451
```

La seconde étape est d'ajuster le modèle et de vérifier l'absence de sur ou sous-dispersion. Pour ajuster le modèle, nous utilisons à nouveau la fonction `vglm` du package VGAM, avec le paramètre `family = multinomial()`. Le ratio entre la statistique de Pearson et le nombre de degrés de liberté du modèle indique une absence de sur ou sous-dispersion (1,04).

```
# Ajustement du modèle
modele <- vglm(Y ~ PercepOGM + PercepRechClim + ConnaisOGM + ConnaisRechClim +
                  CRT + Parti + AGE + Sexe + Revenu,
```

```

        data = data_quest,
        family = multinomial(refLevel="A"), model = T)
# Calcul du Khi2 de Pearson
pred <- predict(modele, type= "response")
cat_predict <- colnames(pred)[max.col(pred)]
freq_real <- table(data_quest$Y)
freq_pred <- table(cat_predict)

khi2 <- sum(residuals(modele, type = "pearson")^2)

N <- nrow(data_quest)
p <- modele@rank
r <- length(freq_real)
ratio <- khi2 / ((N-p)*(r-1))
print(ratio)

## [1] 1.045889

```

La troisième étape de la vérification des conditions d'application est l'analyse des distances de Cook. À nouveau, puisque le modèle évalue la probabilité d'appartenir à $K - 1$ catégorie, nous pouvons calculer $K - 1$ résidus par observation et par extension $K - 1$ distances de Cook. Aucune observation ne semble se détacher nettement dans la figure 8.18. Nous décidons donc pour le moment de conserver toutes les observations.

```

# Extraction des résidus
res <- residuals(modele, type = "pearson")
# Extraction de la hat matrix (nécessaire pour calculer la distance de Cook)
hat <- hatvaluesvlm(modele)
# Calcul des distances de Cook
vals <- c("A","B","C","D")
cooks <- lapply(1:ncol(res),function(i){
  r <- res[,i]
  h <- hat[,i]
  cook <- (r/(1 - h))^2 * h/(1 * modele@rank)
  df <- data.frame(
    oid = 1:length(cook),
    cook = cook
  )
  plot <- ggplot(data = df) +
    geom_point(aes(x = oid, y = cook), size = 0.2, color = rgb(0.1,0.1,0.1,0.4)) +
    labs(x = "", y = "", subtitle = paste("distance de Cook P(",vals[[1]]," VS ",vals[[i+1]],")",sep="")) +
    theme(axis.ticks.x = element_blank(),
          axis.text.x = element_blank())
  return(plot)
})
ggarrange(plotlist = cooks, ncol = 2, nrow = 2)

```

Avant de passer à l'analyse de résidus simulés, il est pertinent de réaliser une analyse de type 3 afin de retirer les variables indépendantes dont l'apport au modèle est négligeable. La fonction `AnalyseType3` (disponible dans le code source de ce livre) permet d'effectuer cette opération automatiquement pour un objet de type `vglm`.

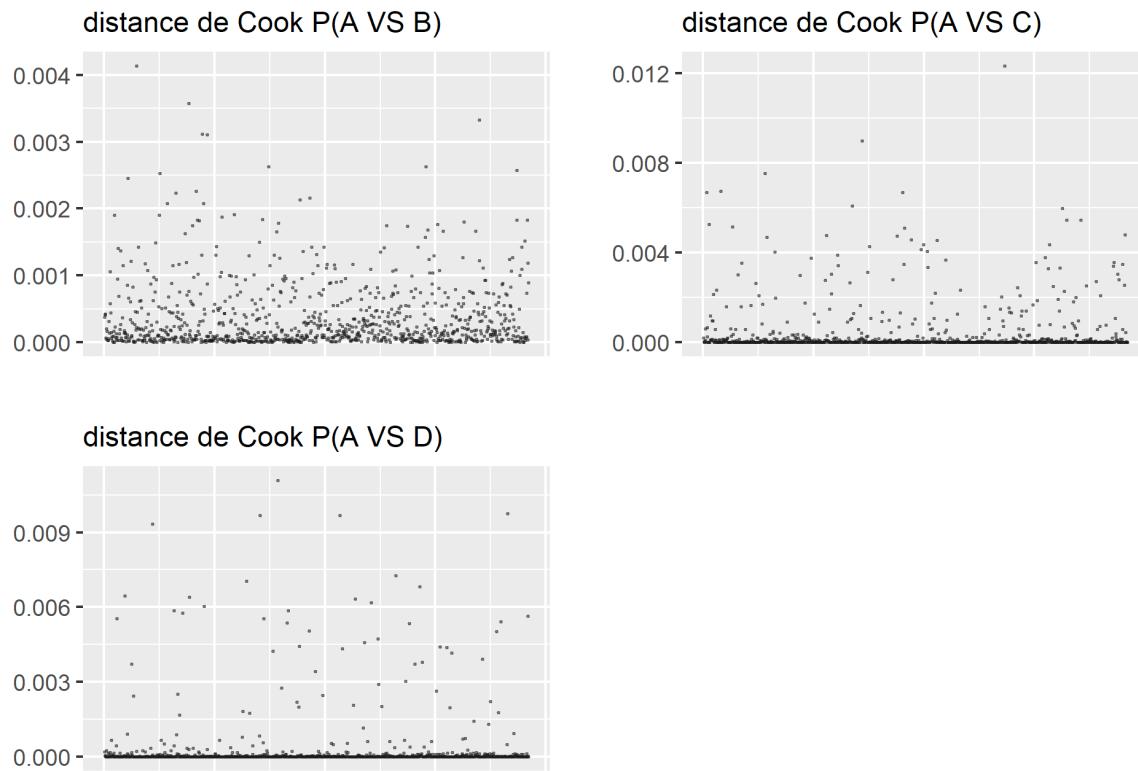


FIG. 8.18 : Distances de Cook pour le modèle logistique multinomial

```
tableau <- AnalyseType3(modele, data_quest, verbose = FALSE)
```

```
## ****
## Type 3 Analysis of Effects
## ****
## AIC model complet : 1855
## loglikelihood model complet : 1789
##   variable retiree  AIC loglikelihood p.val
## 1      PercepOGM 1879      1819      0
## 2      PercepRechClim 1941      1881      0
## 3      ConnaisOGM 1910      1850      0
## 4 ConnaisRechClim 1862      1802 0.0059
## 5          CRT 1860      1800 0.0125
## 6          Parti 1879      1825      0
## 7          AGE 1852      1792 0.4469
## 8          Sexe 1875      1815      0
## 9        Revenu 1850      1790 0.7718
```

L'analyse de type 3 nous permet de déterminer que l'âge et le revenu sont deux variables dont la contribution au modèle est marginale. À titre de rappel, l'analyse de type 3 permet de comparer si le modèle complet (avec l'ensemble des variables indépendantes) est statistiquement mieux ajusté que le modèle avec l'ensemble des variables, sauf une. Or, dans notre cas, le modèle complet n'est pas statistiquement mieux ajusté que le modèle complet sans la variable *Age* ($p=0,44$) et que le modèle complet sans la variable

Revenu ($p = 0,77$). Cela signifie donc que ces deux variables n'ont pas un apport significatif au modèle et peuvent par conséquent être ôtées par parcimonie. Nous décidons donc de les retirer afin d'alléger les tableaux de coefficients que nous présentons plus loin. Nous pouvons également en conclure que ces deux variables ne jouent aucun rôle dans la propension à être en désaccord avec la recherche scientifique. Nous réajustons le modèle en conséquence.

```
modele2 <- vglm(Y ~ PercepOGM + PercepRechClim + ConnaisOGM + ConnaisRechClim +
                  CRT + Parti + Sexe,
                  data = data_quest,
                  family = multinomial(refLevel="A"), model = T)
```

Nous pouvons à présent passer à l'analyse des résidus simulés. Le problème avec ce modèle est que sa variable Y est qualitative alors que la méthode d'analyse des résidus du package DHARMA ne peut traiter que des variables quantitatives, binaires ou ordinaires. Pour rappel, il est possible d'envisager la prédiction d'un modèle logistique multinomial comme la prédiction d'une série de modèles logistiques binomiaux. En représentant nos prédictions de cette façon, nous pouvons à nouveau utiliser le package DHARMA pour analyser nos résidus. Veuillez noter que cette approche n'est pas optimale et que cette section du livre peut être amenée à changer.

La figure 8.19 indique que les résidus suivent bien une distribution uniforme et qu'aucune valeur aberrante n'est observable.

```
# Extraire les prédictions du modèle
categories <- c("B","C","D")
predicted <- predict(modele2, type = "link")
nsim <- 1000
ilink <- function(x){exp(x)/(1+exp(x))}
# Boucler sur chacune des catégories en dehors de la référence
data_sims <- lapply(1:ncol(predicted),function(i){
  categorie <- categories[[i]]
  # Extraire les observations de la catégorie i et de la référence
  test <- data_quest$Y %in% c("A",categorie)
  # Calculer les probabilités d'appartenance à i
  values <- predicted[test,i]
  probs <- ilink(values)
  # Extraire les valeurs réelles et les convertir en 0 / 1
  real <- data_quest[test,]$Y
  real <- ifelse(real=="A",0,1)
  # Enregistrer ces différents éléments
  all_probs <- cbind(1-probs,probs)
  sub_data <- subset(data_quest,test)
  return(list("real" = real, "probs" = all_probs, "data" = sub_data))
})
# Extraire les probabilités prédites
all_probs <- do.call(rbind, lapply(data_sims, function(i){i$probs}))
# Extraire les vrais catégories
all_real <- unlist(lapply(data_sims, function(i){i$real}))
# Effectuer les simulations
simulations <- lapply(1:nrow(all_probs), function(i){
  probs <- all_probs[i,]
  sims <- sample(c(0,1), size = nsim, replace = T, prob = probs)
  return(sims)
})
```

```

matsim <- do.call(rbind, simulations)
# Calculer les résidus simulés
sim_res <- createDHARMA(simulatedResponse = matsim,
                         observedResponse = all_real,
                         fittedPredictedResponse = all_probs[,2],
                         integerResponse = T)
# Afficher les résultats
plot(sim_res)

```

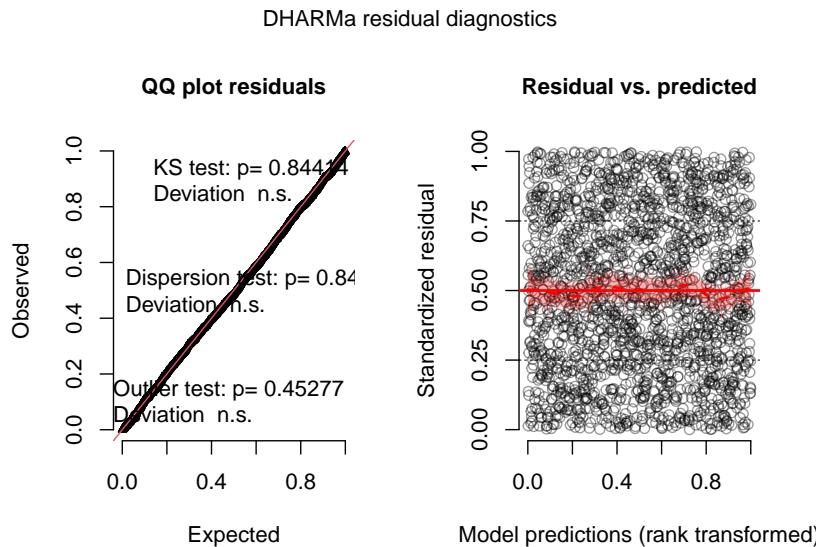


FIG. 8.19 : Diagnostic général des résidus simulés pour le modèle multinomial

La figure 8.20 permet d'affiner le diagnostic en s'assurant de l'absence de relation entre les variables indépendantes et les résidus. Il est possible de remarquer des irrégularités pour les variables de perception (premier et second panneaux). Dans les deux cas, la catégorie 1 (fort désaccord) se démarque nettement des autres catégories. Nous proposons donc de les recoder comme des variables binaires : en désaccord / pas en désaccord pour minimiser cet effet.

```

# Recomposer les données pour coller au format
# étendu de la prediction
etend_data <- do.call(rbind, lapply(data_sims, function(i){i$data}))
par(mfrow=c(3,3))
vars <- c("PercepOGM", "PercepRechClim", "ConnaisOGM",
        "ConnaisRechClim", "CRT", "Parti", "Sexe")
for(v in vars){
  plotResiduals(sim_res, etend_data[[v]], xlab= v, ylab = "résidus")
}

```

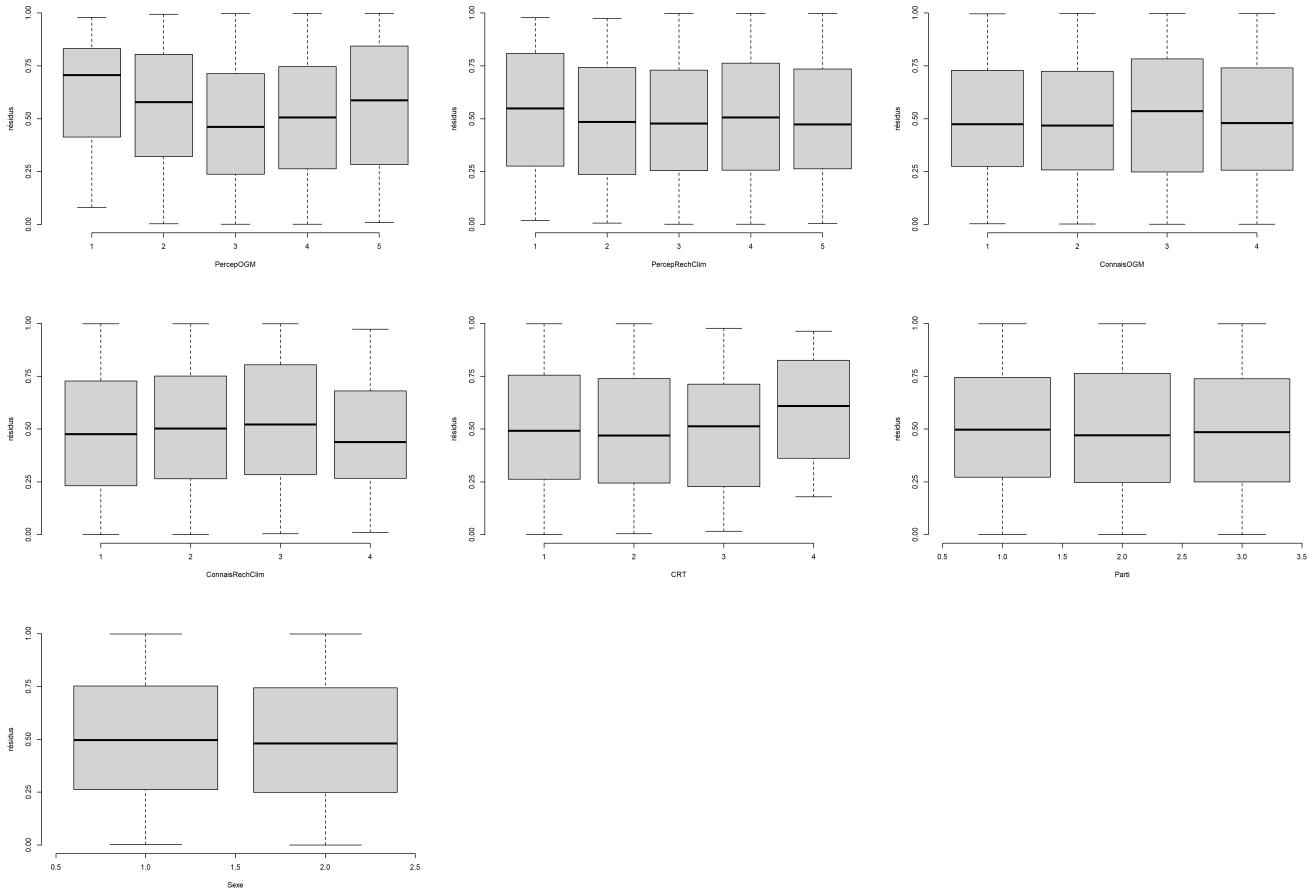


FIG. 8.20 : Diagnostic des variables indépendantes et des résidus simulés pour le modèle multinomial

Nous réajustons donc le modèle et recalculons nos résidus ajustés (masqué ici pour alléger le document). La figure 8.21 nous confirme que le problème a été corrigé.

```
# Convertir les variables ordinaires et variables binaires
data_quest$PercepOGMDes <- ifelse(data_quest$PercepOGM %in% c(1,2), 1,0)
data_quest$PercepRechClimDes <- ifelse(data_quest$PercepRechClim %in% c(1,2), 1,0)
# Réajuster le modèle
modèle3 <- vglm(Y ~ PercepOGMDes + PercepRechClimDes +
  ConnaisOGM + ConnaisRechClim +
  CRT + Parti + Sexe,
  data = data_quest,
  family = multinomial(refLevel="A"), model = T)
```

Profitons du fait que nous utilisons le package VGAM pour vérifier l'absence d'effet de Hauck-Donner qui indiquerait que des variables indépendantes provoquent des séparations parfaites.

```
test <- hdeff(modèle3)
test[test==TRUE]

## (Intercept):2 (Intercept):3
##          TRUE          TRUE
```

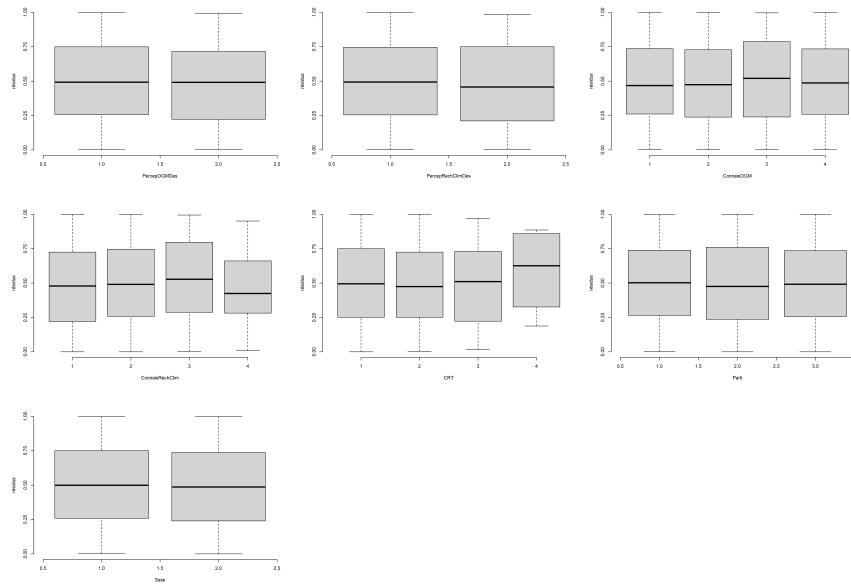


FIG. 8.21 : Diagnostic général des résidus simulés pour le modèle multinomial (version 3)

La fonction nous informe que les constantes permettant de comparer le groupe C au groupe A, et le groupe D au groupe A provoquent des séparations parfaites. Cela s'explique notamment par le faible nombre d'observations tombant dans ces catégories. Considérant que comparativement à la catégorie A, être dans les catégories B, C, ou D signifie remettre en cause au moins un consensus scientifique, il peut être raisonnable de fixer la constante pour qu'elle soit la même pour les trois comparaisons. Ainsi, les chances de passer de A à un autre groupe ne dépendraient pas du groupe en question, mais uniquement des variables indépendantes. Pour cela, nous pouvons forcer le modèle à n'ajuster qu'une seule constante avec la syntaxe suivante :

```

modele4 <- vglm(Y ~ PercepOGMDes + PercepRechClimDes +
  ConnaisOGM + ConnaisRechClim +
  CRT + Parti + Sexe,
  data = data_quest,
  family = multinomial(refLevel="A",
    parallel = TRUE ~1), model = T)
test <- hdeff(modele4)
print(table(test))

## test
## FALSE
##    25

# Calcul du khi2 de Pearson
pred <- predict(modele4, type= "response")
cat_predict <- colnames(pred)[max.col(pred)]
freq_real <- table(data_quest$Y)
freq_pred <- table(cat_predict)
khi2 <- sum(residuals(modele4, type = "pearson")^2)
N <- nrow(data_quest)
p <- modele4@rank
r <- length(freq_real)

```

```
ratio <- kхи2 / ((N)*(r-1)-p)
print(ratio)
```

```
## [1] 1.010319
```

Nous n'avons donc plus de séparation complète ni de sur ou sous-dispersion et les résidus simulés de la quatrième version du modèle sont toujours acceptables (figure 8.22).

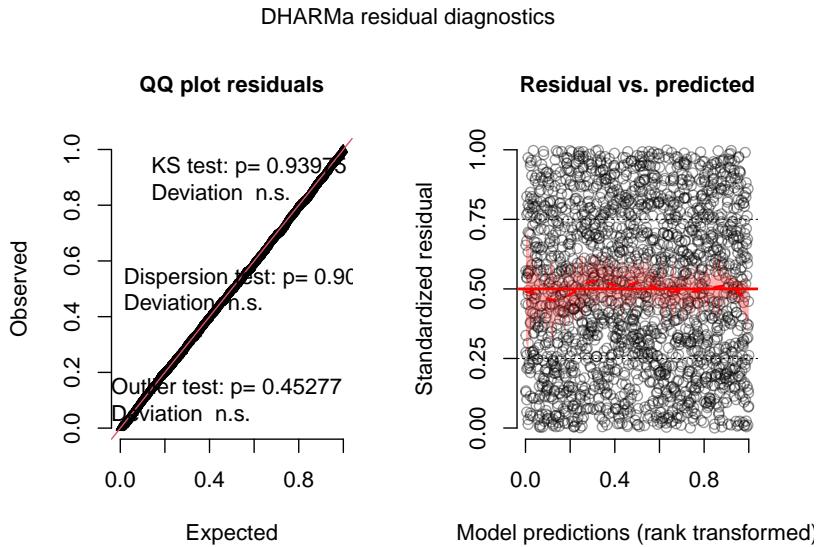


FIG. 8.22 : Diagnostic général des résidus simulés pour le modèle multinomial (version 4)

Vérification l'ajustement du modèle

Puisque les conditions d'application du modèle sont respectées, nous pouvons à présent vérifier sa qualité d'ajustement.

```
modeleNULL <- vglm(Y ~ 1 ,
                     data = data_quest,
                     family = multinomial(refLevel="A",
                                           parallel = TRUE ~ 1 + CRT)
                     , model = T)
rsqs(loglike.full = logLik(modele4),
      loglike.null = logLik(modeleNULL),
      full.deviance = deviance(modele4),
      null.deviance = deviance(modeleNULL),
      nb.params = modele4@rank,
      n = nrow(data_quest))
```

```
## $`deviance expliquee`
## [1] 0.1750071
##
## $`McFadden ajuste`
## [1] 0.1530715
##
```

```
## $`Cox and Snell`  
## [1] 0.3397248  
##  
## $Nagelkerke  
## [1] 0.3746843
```

Le modèle parvient à expliquer 17,5 % de la déviance totale. Il obtient un R^2 ajusté de McFadden de 0,15, et des R^2 de Cox et Snell et de Nagelkerke de respectivement 0,34 et 0,37. Passons à la construction de la matrice de confusion pour analyser la capacité de prédiction du modèle.

```
preds <- predict(modele4, type = "response")  
pred_cats <- colnames(preds)[max.col(preds)]  
real <- data_quest$Y  
matrices <- nice_confusion_matrix(real, pred_cats)  
  
# Afficher la matrice de confusion  
print(matrices$confusion_matrix)
```

	rowsnames	A	B	C	D	rs
## colsnames	""	"A (reel)"	"B (reel)"	"C (reel)"	"D (reel)"	"Total"
## A	"A (predit)"	"482"	"168"	"75"	"29"	"754"
## B	"B (predit)"	"31"	"88"	"9"	"20"	"148"
## C	"C (predit)"	"10"	"6"	"18"	"6"	"40"
## D	"D (predit)"	"3"	"4"	"3"	"9"	"19"
##	"Total"	"526"	"266"	"105"	"64"	"961"
##	"%"	"54.7"	"27.7"	"10.9"	"6.7"	NA
##	rp					
## colsnames	"%"					
## A	"78.5"					
## B	"15.4"					
## C	"4.2"					
## D	"2"					
##	NA					
##	NA					

```
# Afficher les indicateurs de qualité de prédiction  
print(matrices$indicators)
```

	rnames	precision	rappel	F1
## A	"A"	"0.64"	"0.92"	"0.75"
## B	"B"	"0.59"	"0.33"	"0.43"
## C	"C"	"0.45"	"0.17"	"0.25"
## D	"D"	"0.47"	"0.14"	"0.22"
## macro_scores	"macro"	"0.6"	"0.62"	"0.57"
##	"Kappa"	"0.27"	NA	NA
##	"Valeur de p (precision > NIR)"	"0"	NA	NA

La précision globale (macro) du modèle est de 60 % et dépasse significativement le seuil de non-information. L'indicateur de Kappa indique un accord modéré entre la prédiction et les valeurs réelles. Les catégories C et D sont les catégories avec la plus faible précision, indiquant ainsi que le modèle a tendance à manquer les prédictions pour les individus en désaccord avec le consensus scientifique sur

le réchauffement climatique. Les indices de rappel sont également très faibles pour les catégories B, C et D, indiquant que très peu d'observations appartenant originellement à ces groupes ont bien été classées dans ces groupes. La capacité de prédiction du modèle est donc relativement faible.

Interprétation des résultats

Puisque nous disposons de quatre catégories dans notre variable Y , nous obtenons au final trois tableaux de coefficients. Il est possible de visualiser l'ensemble des coefficients du modèle avec la fonction `summary`, nous proposons les tableaux 8.17, 8.18 et 8.19 pour présenter l'ensemble des résultats.

TAB. 8.17 : Coefficients du modèle multinomial A versus B

Variable	Coefficient	RC	val.p	RC 2,5 %	RC 97,5 %	sign.
PercepOGMDes	1,940	6,989	0,000	4,221	11,588	***
PercepRechClimDes	0,430	1,532	0,305	0,677	3,456	
ConnaisOGM	-0,330	0,718	0,000	0,607	0,852	***
ConnaisRechClim	0,180	1,197	0,084	0,980	1,462	.
CRT	0,350	1,417	0,016	1,073	1,878	*
<i>Parti</i>						
ref : Democrat	–	–	–	–	–	–
autre	0,660	1,934	0,001	1,310	2,858	***
Republicain	0,650	1,910	0,003	1,259	2,915	**
<i>Sexe</i>						
ref : homme	–	–	–	–	–	–
femme	1,280	3,581	0,000	2,535	5,053	***

Le tableau 8.17 compare donc le groupe A (en accord avec la recherche scientifique sur les deux sujets) et le groupe B (en désaccord sur la question des OGM). Les résultats indiquent que le fait de se percevoir en désaccord avec le consensus scientifique sur la question des OGM multiplie par sept les chances d'appartenir au groupe B comparativement au groupe A. Cependant, pour chaque bonne réponse supplémentaire sur les questions testant les connaissances sur les OGM, les chances d'appartenir au groupe B comparativement au groupe A diminue de 28 %. Ainsi, un individu ayant répondu correctement aux trois questions verrait ses chances réduites de 63 % d'appartenir au groupe B ($\exp(-0,33 \times 3)$). Il est intéressant de noter que les variables concernant le réchauffement climatique n'ont pas d'effet significatif ici. La variable CRT indique qu'à chaque bonne réponse supplémentaire au test de cognition, les chances d'appartenir au groupe B augmentent de 42 %. Un individu qui aurait répondu aux trois questions du test aurait donc 2,9 fois plus de chances d'appartenir au groupe B qu'au groupe A. Concernant le parti politique, comparativement à une personne se déclarant plus proche du parti démocrate, les personnes proches du parti républicain ou d'un autre parti ont près de deux fois plus de chances d'appartenir au groupe B. Enfin, une femme, comparativement à un homme, a 3,6 fois plus de chance d'appartenir au groupe B.

TAB. 8.18 : Coefficients du modèle multinomial A versus C

variable	Coefficient	RC	val.p	RC 2,5 %	RC 97,5 %	sign.
PercepOGMDes	-0,180	0,834	0,705	0,326	2,138	
PercepRechClimDes	2,060	7,821	0,000	3,819	16,119	***
ConnaisOGM	-0,110	0,896	0,287	0,733	1,094	
ConnaisRechClim	0,280	1,323	0,024	1,041	1,682	*
CRT	0,240	1,275	0,149	0,914	1,768	
<i>Parti</i>						
ref : Democrat	—	—	—	—	—	—
autre	-0,190	0,828	0,496	0,482	1,433	
Republicain	0,920	2,512	0,000	1,584	4,015	***
<i>Sexe</i>						
ref : homme	—	—	—	—	—	—
femme :1	-0,450	0,640	0,040	0,419	0,980	*

Le tableau 8.18 compare les groupes A et C (en désaccord sur le réchauffement climatique). Il est intéressant de noter ici que se percevoir en désaccord avec la recherche scientifique est associé avec une forte augmentation des chances d'appartenir au groupe C. Cependant, un plus grand nombre de bonnes réponses aux questions sur le réchauffement climatique est également associé avec une augmentation des chances (30 % à chaque bonne réponse supplémentaire) d'appartenir au groupe C. Le CRT n'a cette fois-ci pas d'effet. Se déclarer proche du parti républicain, comparativement au parti démocrate, multiplie les chances par 2,5 d'appartenir au groupe C. Comparativement au tableau précédent, le fait d'être une femme diminue les chances de 36 % d'appartenir au groupe C.

Le dernier tableau 8.19 compare le groupe A au groupe D (en désaccord sur les deux sujets). Les variables les plus importantes sont une fois encore le fait de se sentir en désaccord avec la recherche scientifique et le degré de connaissance sur les OGM. La variable concernant le parti politique est significative au seuil 0,05 et exprime toujours une tendance accrue pour les individus du parti républicain à appartenir au groupe D.

Nos propres conclusions corroborent celles de l'article original. Une des conclusions intéressantes est que le rejet du consensus scientifique ne semble pas nécessairement être associé à un déficit d'information ni à une plus faible capacité analytique, mais relèverait davantage d'une polarisation politique. Notez que cette littérature sur les croyances et la confiance dans la recherche est complexe, si le sujet vous intéresse, la discussion de l'article de McFadden (2016) est un bon point de départ.

TAB. 8.19 : Coefficients du modèle multinomial A versus D

variable	Coefficient	RC	val.p	RC 2,5 %	RC 97,5 %	sign.
PercepOGMDes	1,500	4,488	0,000	2,270	8,935	***
PercepRechClimDes	2,440	11,501	0,000	5,312	25,028	***
ConnaisOGM	-0,480	0,621	0,000	0,492	0,787	***
ConnaisRechClim	0,130	1,140	0,399	0,844	1,553	
CRT	0,340	1,409	0,119	0,914	2,160	
<i>Parti</i>						
ref : Democrat	—	—	—	—	—	—
autre	0,190	1,211	0,531	0,664	2,203	
Republicain	0,630	1,872	0,038	1,041	3,387	*
<i>Sexe</i>						
ref : homme	—	—	—	—	—	—
femme :1	-0,110	0,896	0,664	0,543	1,477	

8.2.5 Conclusion sur les modèles pour des variables qualitatives

Nous avons vu dans cette section, les trois principales formes de modèles GLM pour modéliser une variable binaire (modèle binomial), une variable ordinaire (modèle de cotes proportionnel) et une variable multinomiale (modèle multinomial). Pour ces trois modèles, nous avons vu que la distribution utilisée est toujours la distribution binomiale et la fonction de lien logistique. Les coefficients obtenus s'interprètent comme des rapports de cotes, une fois qu'ils sont transformés avec la fonction exponentielle.

8.3 Modèles GLM pour des variables de comptage

Dans cette section, nous présentons les principaux modèles utilisés pour modéliser des variables de comptage. Il peut s'agir de variables comme le nombre d'accidents à une intersection, le nombre de cafés par quartier, le nombre de cas d'une maladie donnée par entité géographique, etc.

8.3.1 Modèle de Poisson

Le modèle GLM de base pour modéliser une variable de comptage est le modèle de Poisson. Pour rappel, la distribution de Poisson a un seul paramètre : λ . Il représente le nombre moyen d'événements observés sur l'intervalle de temps d'une observation, ainsi que la dispersion de la distribution. En conséquence, λ doit être un nombre strictement positif; autrement dit, nous ne pouvons pas observer un nombre négatif d'événements. Il est donc nécessaire d'utiliser une fonction de lien pour contraindre l'équation de régression sur l'intervalle $[0, +\infty]$. La fonction la plus utilisée est le logarithme naturel (\log) dont la réciproque est la fonction exponentielle (\exp).

8.3.1.1 Interprétation des paramètres

Les coefficients du modèle expriment l'effet du changement d'une unité des variables X sur λ (le nombre de cas) dans l'échelle logarithmique (*log-scale*). Pour rappel, l'échelle logarithmique est multiplicative : si nous convertissons les coefficients dans leur échelle originale avec la fonction exponentielle, leur effet n'est plus additif, mais multiplicatif. Prenons un exemple concret, admettons que nous avons ajusté un modèle de Poisson à une variable de comptage Y avec deux variables X_1 et X_2 et que nous avons obtenus les coefficients suivants :

$$\beta_0 = 1,8; \beta_1 = 0,5; \beta_2 = -1,5$$

L'interprétation basique (sur l'échelle logarithmique) est la suivante : une augmentation d'une unité de la variable X_1 est associée avec une augmentation de 0,5 du logarithme du nombre de cas attendus. Une augmentation d'une unité de la variable X_2 est associée avec une réduction de 1,5 unité du logarithme du nombre de cas attendus. Avec la conversion avec la fonction exponentielle, nous obtenons alors :

TAB. 8.20 : Carte d'identité du modèle de Poisson

Type de variable dépendante	Variable de comptage
Distribution utilisée	Poisson
Formulation	$Y \sim Poisson(\lambda)$ $g(\lambda) = \beta_0 + \beta X$ $g(x) = \log(x)$
Fonction de lien	\log
Paramètre modélisé	λ
Paramètres à estimer	β_0, β
Conditions d'application	Absence d'excès de zéros, absence de sur-dispersion ou de sous-dispersion

- $\exp(0,5) = 1,649$, soit une multiplication par 1,649 du nombre de cas attendu (aussi appelé taux d'incident) à chaque augmentation d'une unité de X_1 ;
- $\exp(-1,5) = 0,223$, soit une division par 4,54 (1/0,223) du nombre de cas attendu (aussi appelé taux d'incident) à chaque augmentation d'une unité de X_2 .

Utilisons maintenant notre équation pour effectuer une prédition si $X_1 = 1$ et $X_2 = 1$.

$$\lambda = \exp(1,8 + (0,5 \times 1) + (-1,5 \times 1)) = 2,225$$

Si nous augmentons X_1 d'une unité, nous obtenons alors :

$$\lambda = \exp(1,8 + (0,5 \times 2) + (-1,5 \times 1)) = 3,670$$

En ayant augmenté d'une unité X_1 , nous avons multiplié notre résultat par 1,649 ($2,225 \times 1,649 = 3,670$).

Notez que ces effets se multiplient entre eux. Si nous augmentons à la fois X_1 et X_2 d'une unité chacune, nous obtenons : $\lambda = \exp(1,8 + (0,5 \times 2) + (-1,5 \times 2)) = 0,818$, ce qui correspond bien à $2,225 \times 1,649$ (effet de X_1) $\times 0,223$ (effet de X_2) = 0,818.

Il existe des fonctions dans R qui calculent ces prédictions à partir des équations des modèles. Il est cependant essentiel de bien saisir ce comportement multiplicatif induit par la fonction de lien *log*.

8.3.1.1 Conditions d'application

Puisque la distribution de Poisson n'a qu'un seul paramètre, le modèle GLM de Poisson est exposé au même problème potentiel de sur-dispersion que les modèles binomiaux de la section précédente. Référez-vous à la section 8.2.1.2 pour davantage de détails sur le problème posé par la sur-dispersion. Pour détecter une potentielle sur-dispersion dans un modèle de Poisson, il est possible, dans un premier temps, de calculer le ratio entre la déviance du modèle et son nombre de degrés de liberté (SAS Institute Inc 2020b). Ce ratio doit être proche de 1, s'il est plus grand, le modèle souffre de sur-dispersion.

$$\hat{\phi} = \frac{D(\text{modele})}{N - p} \tag{8.21}$$

avec N et p étant respectivement les nombres d'observations et de paramètres. Il est également possible de tester formellement si la sur-dispersion est significative avec un test de dispersion.

La question de l'excès de zéros a été abordée dans la section 2.4.3.7 du chapitre 2. Il s'agit d'une situation où un plus grand nombre de zéros sont présents dans les données que ce que suppose la distribution de Poisson. Dans ce cas, il convient d'utiliser la distribution de Poisson avec excès de zéros.

8.3.1.2 Exemple appliqué dans R

Pour cet exemple, nous reproduisons l'analyse effectuée dans l'article de Cloutier et al. (2014). L'enjeu de cette étude était de modéliser le nombre de piétons blessés autour de plus de 500 carrefours dans les quartiers centraux de Montréal. Pour cela, trois types de variables étaient utilisées : des variables décrivant l'intersection, des variables décrivant les activités humaines dans un rayon d'un kilomètre autour de l'intersection et des variables représentant le trafic routier autour de l'intersection. Un effet direct de ce type d'étude est bien évidemment l'établissement de meilleures pratiques d'aménagement réduisant les risques encourus par les piétons lors de leurs déplacements en ville. Le tableau 8.21 présente l'ensemble des variables utilisées dans l'analyse.

La distribution originale de la variable est décrite à la figure 8.23. Les barres grises représentent la distribution du nombre d'accidents et les barres rouges une distribution de Poisson ajustée sans variable indépendante (modèle nul). Ce premier graphique peut laisser penser qu'un modèle de Poisson n'est

TAB. 8.21 : Variables indépendantes utilisées dans le modèle de Poisson

Nom de la variable	Signification	Type de variable	Mesure
Feux_auto	Présence de feux de circulation	Variable binaire	0 = absence ; 1 = présence
Feux_piet	Présence de feux de traversée pour les piétons	Variable binaire	0 = absence ; 1 = présence
Pass_piet	Présence d'un passage piéton	Variable binaire	0 = absence ; 1 = présence
Terreplein	Présence d'un terre-plein central	Variable binaire	0 = absence ; 1 = présence
Apaisement	Présence de mesure d'apaisement de la circulation	Variable binaire	0 = absence ; 1 = présence
LogEmploi	Logarithme du nombre d'emplois dans un rayon d'un kilomètre	Variable continue	Logarithme du nombre d'emploi. Utilisation du logarithme, car la variable est fortement asymétrique
Densite_pop	Densité de population dans un rayon d'un kilomètre	Variable continue	Habitants par hectare
Entropie	Diversité des occupations du sol dans un rayon d'un kilomètre (indice d'entropie)	Variable continue	Mesure de 0 à 1 ; 0 = spécialisation parfaite ; 1 = diversité parfaite
DensiteInter	Densité d'intersections dans un rayon d'un kilomètre (connexité)	Variable continue	Nombre d'intersection par km ²
Artere	Présence d'une artère à l'intersection	Variable binaire	0 = absence ; 1 = présence
Long_arterePS	Longueur d'artère dans un rayon d'un kilomètre	Variable continue	Exprimée en mètres
NB_voies5	Présence d'une cinq voies à l'intersection	Variable binaire	0 = absence ; 1 = présence

pas nécessairement le plus adapté considérant le grand nombre d'intersections pour lesquelles nous n'avons aucun accident. Cependant, rappelons que la variable Y n'a pas besoin de suivre une distribution de Poisson. Dans un modèle GLM, l'hypothèse que nous formulons est que la variable dépendante (Y) **conditionnée par les variables indépendantes (X)** suit une certaine distribution (ici Poisson).

```
# Chargement des données
data_accidents <- read.csv("data/glm/accident_pietons.csv", sep = ";")
# Ajustement d'une distribution de Poisson sans variable indépendante
library(fitdistrplus)
model_poisson <- fitdist(data_accidents$Nbr_acci,distr = "pois")
# Création d'un graphique pour comparer les deux distributions
dfpoisson <- data.frame(x=c(0:19),
                         y=dpois(0:19, model_poisson$estimate))
counts <- data.frame(table(data_accidents$Nbr_acci))
names(counts) <- c("nb_accident",'frequence')
counts$nb_accident <- as.numeric(as.character(counts$nb_accident))
counts$prop <- counts$frequence / sum(counts$frequence)
ggplot() +
  geom_bar(aes(x=nb_accident, weight = prop, fill = "real"), width = 0.6, data = counts) +
  geom_bar(aes(x=x, weight = y, fill = "adj"), width = 0.15, data = dfpoisson) +
  scale_x_continuous(limits = c(-0.5,7), breaks = c(0:7)) +
  scale_fill_manual(name = "",
                    breaks = c("real","adj"),
                    labels = c("distribution originale", "distribution de Poisson"),
                    values = c("real" = rgb(0.4,0.4,0.4), "adj" = "red")) +
  labs(subtitle = "",
       x = "nombre d'accidents",
       y = "")
```

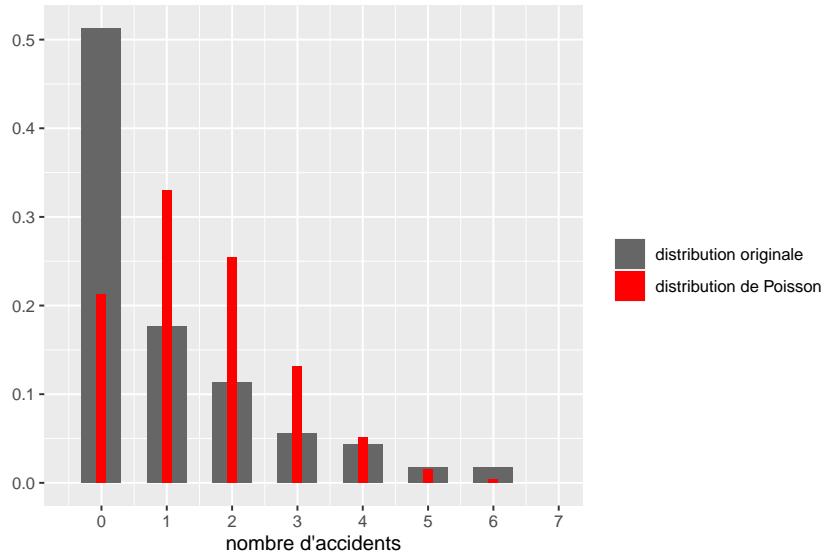


FIG. 8.23 : Distribution originale du nombre d'accidents par intersection

Vérification des conditions d'application

Comme précédemment, notre première étape est de vérifier l'absence de multicolinéarité excessive avec la fonction `vif` du package `car`.

```
vif(glm(Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet + Terreplein + Apaisement +
         LogEmploi + Densite_pop + Entropie + DensiteInter +
         Long_arterePS + Artere + NB_voies5,
         family = poisson(link="log"),
         data = data_accidents))
```

	Feux_auto	Feux_piet	Pass_piet	Terreplein	Apaisement
##	2.861708	1.518668	2.321213	1.221683	1.059722
##	LogEmploi	Densite_pop	Entropie	DensiteInter	Long_arterePS
##	4.763692	1.153096	1.770904	2.040457	4.467841
##	Artere	NB_voies5			
##	1.887728	1.520514			

Toutes les valeurs de VIF sont inférieures à 5. Notons tout de même que le logarithme de l'emploi et la longueur d'artère dans un rayon d'un kilomètre ont des valeurs de VIF proches de 5. La seconde étape du diagnostic consiste à calculer et à visualiser les distances de Cook.

```
# Ajustement d'une première version du modèle
modele <- glm(Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet + Terreplein + Apaisement +
               LogEmploi + Densite_pop + Entropie + DensiteInter +
               Long_arterePS + Artere + NB_voies5,
               family = poisson(link="log"),
               data = data_accidents)

# Calcul des distances de Cook
cooksdf <- cooks.distance(modele)
df <- data.frame(
```

```

cook = cooksd,
oid = 1:length(cooksd)
)
ggplot(data = df) +
  geom_point(aes(x = oid, y = cook), size = 0.5, alpha = 0.5) +
  labs(x = "",
       y = "Distance de Cook")

```

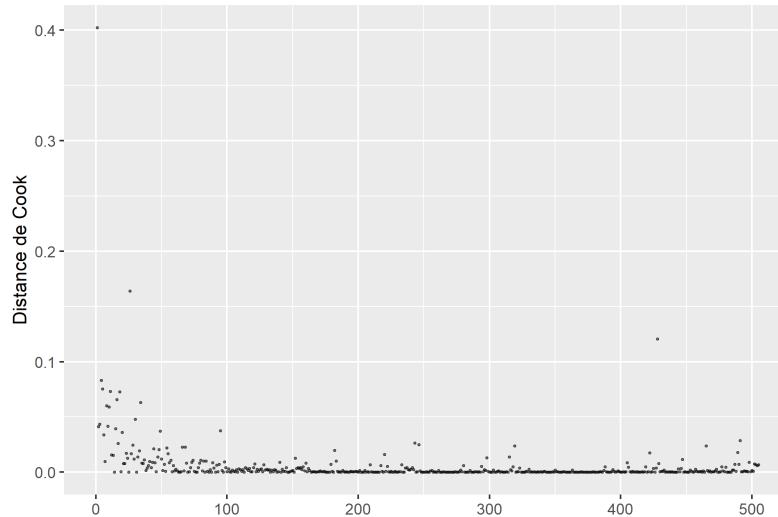


FIG. 8.24 : Distances de Cook pour le modèle de Poisson

La figure 8.24 signale la présence de trois observations avec des valeurs extrêmement fortes de distance de Cook. Nous les isolons dans un premier temps pour les analyser.

```

cas_étrange <- subset(data_accidents, cooksd>0.1)
print(cas_étrange)

##      Nbr_acci Feux_auto Feux_piet Pass_piet Terreplein Apaisement EmpTotBuffer
## 1          19         1         1         1         1         0     7208.538
## 26          7         0         0         1         0         0     1342.625
## 428          0         1         1         1         0         1    13122.560
##      Densite_pop Entropie DensiteInter Long_arterePS Artere NB_voies5 log_acci
## 1      5980.923  0.8073926   42.41597   6955.00     1     1 2.995732
## 26     2751.012  0.0000000   73.35344   2849.66     0     0 2.079442
## 428    14148.827  0.6643891  109.25066   4634.20     0     0 0.000000
##      catego_acci catego_acci2 Arret VAG sum_app LogEmploi AccOrdinal PopHa
## 1            1             1     0     1      4  8.883021      2  5.980923
## 26            1             1     1     1      3  7.202382      2  2.751012
## 428            0             0     0     0      3  9.482088      0 14.148827

```

Les deux premiers cas sont des intersections avec de nombreux accidents (respectivement 19 et 7) qui risquent de perturber les estimations du modèle. Le troisième cas ne comprend en revanche aucun accident. Puisqu'il ne s'agit que de trois observations et que leurs distances de Cook sont très nettement supérieures aux autres, nous les retirons du modèle.

```

data2 <- subset(data_accidents, cooksd<0.1)
# Ajustement d'une première version du modèle
modele <- glm(Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet + Terreplein + Apaisement +
                 LogEmploi + Densite_pop + Entropie + DensiteInter +
                 Long_arterePS + Artere + NB_voies5,
                 family = poisson(link="log"),
                 data = data2)

# Calcul des distances de Cook
cooksd <- cooks.distance(modele)
df <- data.frame(
  cook = cooksd,
  oid = 1:length(cooksd)
)
ggplot(data = df)+
  geom_point(aes(x = oid, y = cook), size = 0.5, alpha = 0.5)+
  labs(x = "",
       y = "distance de Cook")

```

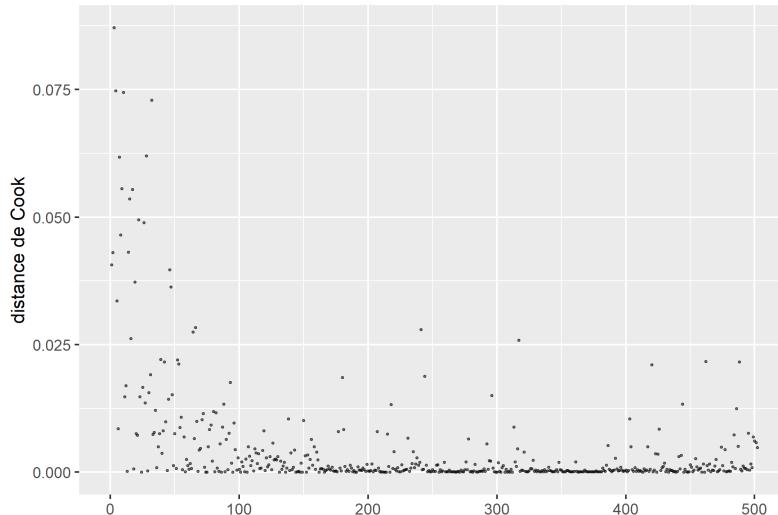


FIG. 8.25 : Distances de Cook pour le modèle de Poisson après avoir retiré les valeurs aberrantes

La figure 8.25 montre que nous n'avons plus d'observations fortement influentes dans le modèle après avoir retiré les trois observations identifiées précédemment. Nous devons à présent vérifier l'absence de sur-dispersion.

```

# Calcul du rapport entre déviance et nombre de degrés de liberté du modèle
deviance(modele)/(nrow(data2) - modele$rank)

```

```
## [1] 1.674691
```

Le rapport entre la déviance et le nombre de degrés de liberté du modèle est nettement supérieur à 1, indiquant une sur-dispersion excessive. Nous pouvons confirmer ce résultat avec la fonction `dispersiontest` du package AER.

```

library(AER)
# Test de sur-dispersion
dispersiontest(modele)

##
## Overdispersion test
##
## data: modele
## z = 5.2737, p-value = 6.686e-08
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
## 1.891565

```

Contrairement à la forme classique d'un modèle de Poisson pour laquelle la dispersion attendue est de 1, le test nous indique qu'une dispersion de 1,89 serait mieux ajustée aux données.

Il est également possible d'illuster cet écart à l'aide d'un graphique représentant les valeurs réelles, les valeurs prédictes, ainsi que la variance (sous forme de barres d'erreurs) attendue par le modèle (figure 8.26). Nous constatons ainsi que les valeurs réelles (en rouge) ont largement tendance à dépasser la variance attendue par le modèle, surtout pour les valeurs les plus faibles de la distribution.

```

# Extraction des prédictions du modèle
lambdas <- predict(modele, type = "response")
# Création d'un DataFrame pour contenir la prédition et les vraies valeurs
df1 <- data.frame(
  lambdas = lambdas,
  reals = data2$Nbr_acci
)
# Calcul de l'intervalle de confiance à 95 % selon la distribution de Poisson
# et stockage dans un second DataFrame
seqa <- seq(0,round(max(lambdas)),1)
df2 <- data.frame(
  lambdas = seqa,
  lower = qpois(p = 0.025, lambda = seqa),
  upper = qpois(p = 0.975, lambda = seqa)
)
# Affichage des valeurs réelles et prédictes (en rouge)
# et de leur variance selon le modèle (en noir)
ggplot() +
  geom_errorbar(data = df2,
    mapping = aes(x = lambdas, ymin = lower, ymax = upper),
    width = 0.2, color = rgb(0.4,0.4,0.4)) +
  geom_point(data = df1,
    mapping = aes(x = lambdas, y = reals),
    color ="red", size = 0.5) +
  labs(x = 'valeurs prédictes',
    y = "valeurs réelles")

```

Pour tenir compte de cette particularité des données, nous modifions légèrement le modèle pour utiliser une distribution de quasi-Poisson, intégrant spécifiquement un paramètre de dispersion. Cet ajustement ne modifie pas l'estimation des coefficients du modèle, mais modifie le calcul des erreurs standards et par extension les valeurs de p pour les rendre moins sensibles au problème de sur-dispersion. Une autre

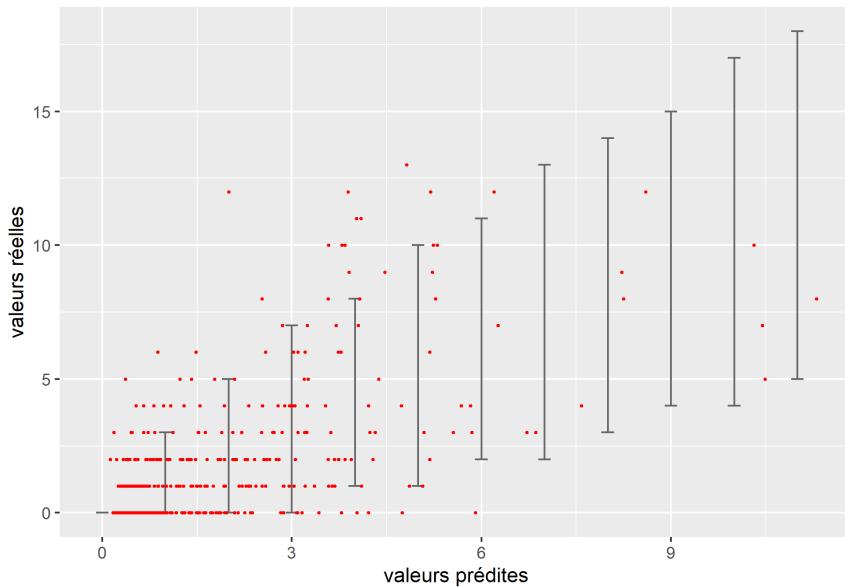


FIG. 8.26 : Représentation de la sur-dispersion des données dans le modèle de Poisson

approche aurait été de calculer une version robuste des erreurs standards avec le *package sandwich* comme nous l'avons fait dans la section 8.2.1 sur le modèle binomial. Après réajustement du modèle, le nouveau paramètre de dispersion estimé est de 1,92.

Les quasi-distributions

Les distributions binomiale et de Poisson ne disposent chacune que d'un paramètre décrivant à la fois leur dispersion et leur espérance. Elles manquent donc de flexibilité et échouent parfois à représenter fidèlement des données avec une forte variance. Il existe donc d'autres distributions, respectivement les distributions quasi-binomiale et quasi-Poisson comprenant chacune un paramètre supplémentaire pour contrôler la dispersion. Bien que cette solution soit attrayante, il ne faut pas perdre de vue que la sur ou la sous dispersion peuvent être causées par l'absence de certaines variables explicatives, la sur-représentation de zéros, ou encore une séparation parfaite de la variable dépendante causée par une variable indépendante.

```
modele2 <- glm(Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet + Terreplein + Apaisement +
  LogEmploi + Densite_pop + Entropie + DensiteInter +
  Long_arterePS + Artere + NB_voies5,
  family = quasipoisson(link="log"),
  data = data2)
```

Nous pouvons à présent comparer la distribution originale des données et les simulations issues du modèle. Notez que contrairement à la distribution de Poisson simple, il n'existe pas dans R de fonction pour simuler des valeurs issues d'une distribution de quasi-Poisson. Il est cependant possible d'exploiter sa proximité théorique avec la distribution binomiale négative pour définir notre propre fonction de simulation. La figure 8.27 permet de comparer la distribution originale (en gris) et l'intervalle de confiance à 95 % des simulations (en rouge). Nous remarquons que le modèle semble capturer efficacement la forme générale de la distribution originale. À titre de comparaison, nous pouvons effectuer le même exercice avec la distribution de Poisson classique (le code n'est pas montré pour éviter les répétitions). La figure 8.28 montre qu'un simple modèle de Poisson est très éloigné de la distribution originale de Y .

```

# Définition d'une fonction pour simuler des données quasi-Poisson
rqpois <- function(n, lambda, disp) {
  rnbinom(n = n, mu = lambda, size = lambda/(disp-1))
}

# Extraction des valeurs prédites par le modèle
preds <- predict(modele2, type="response")
# Génération de 1000 simulations pour chaque prédiction
disp <- summary(modele2)$dispersion
nsim <- 1000
cols <- lapply(1:length(preds), function(i){
  lambda <- preds[[i]]
  sims <- round(rqpois(n = nsim, lambda = lambda, disp = disp))
  return(sims)
})
mat_sims <- do.call(rbind, cols)
# Préparation des données pour le graphique (valeurs réelles)
counts <- data.frame(table(data2$Nbr_acc))
names(counts) <- c("nb_accident", "frequence")
counts$nb_accident <- as.numeric(as.character(counts$nb_accident))
counts$prop <- counts$frequence / sum(counts$frequence)
# Préparation des données pour le graphique (valeurs simulées)
df1 <- data.frame(count = 0:25)
count_sims <- lapply(1:nsim, function(i){
  sim <- mat_sims[,i]
  cnt <- data.frame(table(sim))
  df2 <- merge(df1, cnt, by.x="count", by.y = "sim", all.x = T, all.y=F)
  df2$Freq <- ifelse(is.na(df2$Freq), 0, df2$Freq)
  return(df2$Freq)
})
count_sims <- do.call(cbind, count_sims)
df_sims <- data.frame(
  val = 0:25,
  med = apply(count_sims, MARGIN = 1, median),
  lower = apply(count_sims, MARGIN = 1, quantile, probs = 0.025),
  upper = apply(count_sims, MARGIN = 1, quantile, probs = 0.975)
)

ggplot() +
  geom_bar(aes(x=nb_accident, weight = frequence), width = 0.6, data = counts) +
  geom_errorbar(aes(x = val, ymin = lower, ymax = upper),
                data = df_sims, color = "red", width = 0.6) +
  geom_point(aes(x = val, y = med), color = "red", size = 1.3, data = df_sims) +
  scale_x_continuous(limits = c(-0.5,7), breaks = c(0:7)) +
  xlim(-1,12) +
  labs(subtitle = "",
       x = "nombre d'accidents",
       y = "nombre d'occurrences")

```

La prochaine étape du diagnostic est l'analyse des résidus simulés. La figure 8.29 indique que les résidus du modèle suivent bien une distribution uniforme et qu'aucune valeur aberrante n'est observable.

```

# Génération de 1000 simulations pour chaque prédiction
disp <- 1.918757 # trouvable dans le summary(modele2)
nsim <- 1000

```

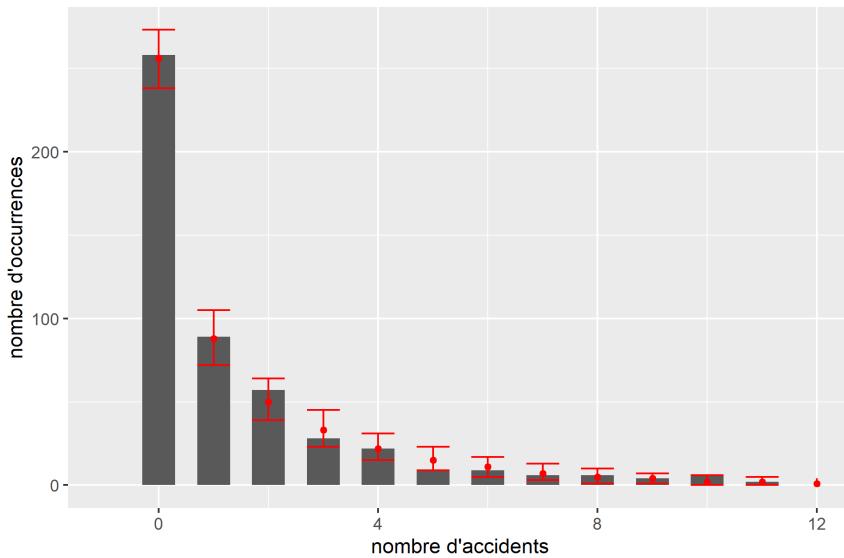


FIG. 8.27 : Comparaison de la distribution originale et des simulations pour le modèle de quasi-Poisson

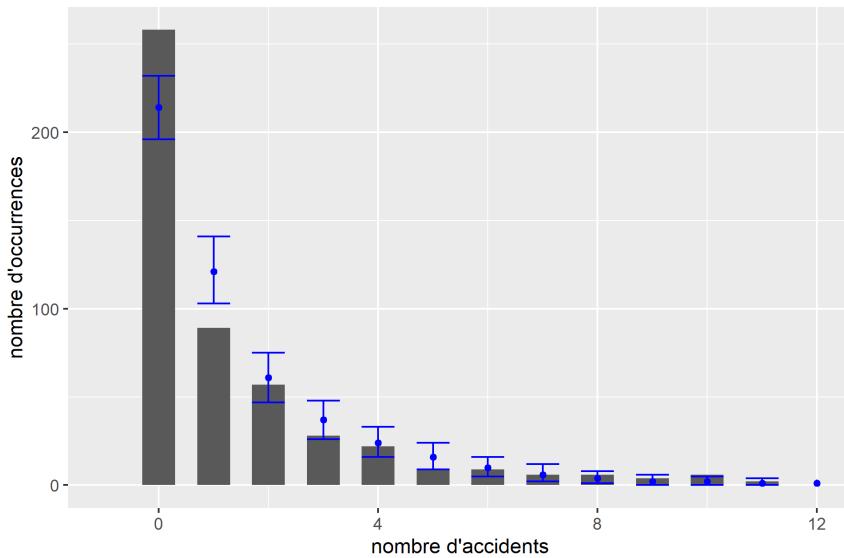


FIG. 8.28 : Comparaison de la distribution originale et des simulations pour le modèle de Poisson

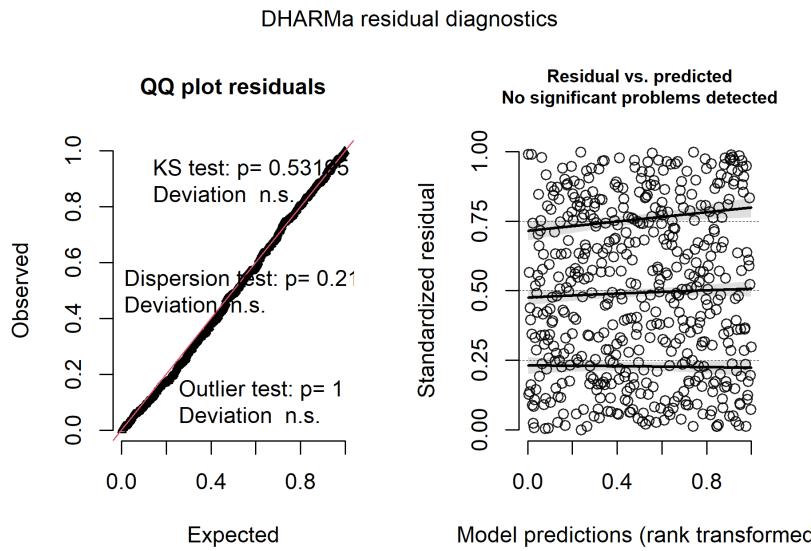


FIG. 8.29 : Analyse globale des résidus simulés pour le modèle de quasi-Poisson

Pour affiner notre diagnostic, nous pouvons également comparer les résidus simulés et chaque variable indépendante. La figure 8.30 n'indique aucune relation problématique entre nos variables indépendantes et les résidus.

```
par(mfrow=c(3,4))
vars <- c("Feux_auto", "Feux_piet", "Pass_piet", "Terreplein", "Apaisement",
        "LogEmploi", "Densite_pop", "Entropie", "DensiteInter",
        "Long_arterePS", "Artere", "NB_voies5")
for(v in vars){
  plotResiduals(sim_res, data2[[v]], xlab = v, main = "", ylab = "résidus")
}
```

Maintenant que l'ensemble des diagnostics a été effectué, nous pouvons passer à la vérification de la qualité d'ajustement.

Vérification de la qualité d'ajustement

Pour le calcul des pseudo-R², notez qu'il n'existe pas à proprement parler de *loglikelihood* pour les quasi-distributions. Pour contourner ce problème, il est possible d'utiliser le *loglikelihood* d'un simple modèle de Poisson (puisque les coefficients ne changent pas), mais il est important de garder à l'esprit que ces pseudo-R² seront d'autant plus faibles que la sur-dispersion originale était forte.

```
modelnull <- glm(Nbr_acci ~ 1,
                  family = poisson(link="log"),
                  data = data2)
rsqs(loglike.full = logLik(modele),
      loglike.null = logLik(modelnull),
      full.deviance = deviance(modele),
      null.deviance = deviance(modelnull),
      nb.params = modele$rank,
      n = nrow(data2))
```

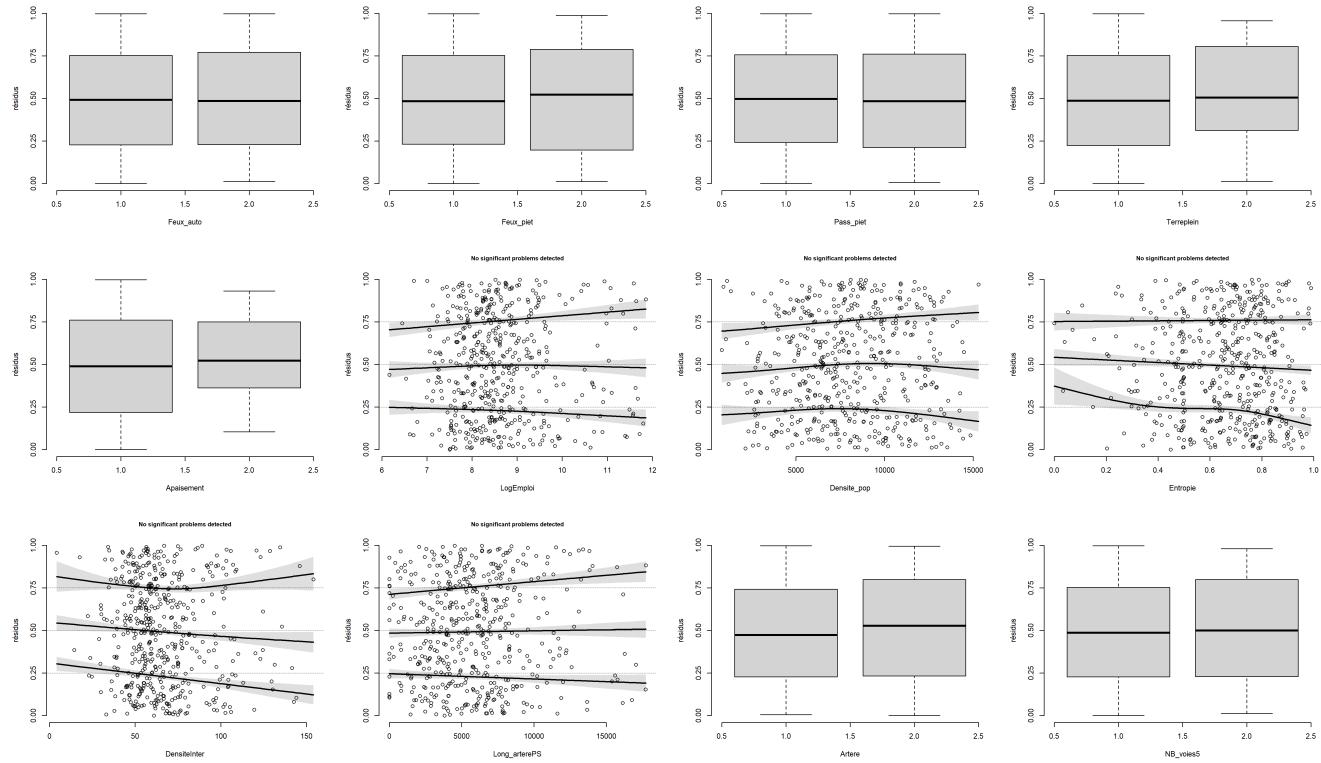


FIG. 8.30 : Comparaison des résidus simulés et de chaque variable indépendante

```
## `$deviance expliquee`  
## [1] 0.4805998  
##  
## `$McFadden ajuste`  
## 'log Lik.' 0.3258375 (df=13)  
##  
## `$Cox and Snell`  
## 'log Lik.' 0.7789704 (df=13)  
##  
## $Nagelkerke  
## 'log Lik.' 0.787958 (df=13)
```

Le modèle parvient ainsi à expliquer 48 % de la déviance totale. Il obtient un R^2 ajusté de McFadden de 0,33 et un R^2 de Cox et Snell de 0,78.

```
# Calcul du RMSE  
sqrt(mean((predict(modele2, type = "response") - data2$Nbr_acci) ** 2))
```

```
## [1] 1.838026
```

```
# Nombre moyen d'accidents  
mean(data2$Nbr_acci)
```

```
## [1] 1.503984
```

L'erreur quadratique moyenne du modèle est de 1,84, ce que signifie qu'en moyenne le modèle se trompe d'environ deux accidents pour chaque intersection. Cette valeur est relativement élevée si nous la comparons avec le nombre moyen d'accidents, soit 1,5. Cela s'explique certainement par le grand nombre de zéros dans la variable Y qui tendent à tirer les prédictions vers le bas.

Interprétation des résultats

L'ensemble des coefficients du modèle sont accessibles via la fonction `summary`. Puisque la fonction de lien du modèle est la fonction `log`, il est pertinent de convertir les coefficients avec la fonction `exp` afin de pouvoir les interpréter sur l'échelle originale (`nombre d'accidents`) plutôt que l'échelle logarithmique (`log(nombre d'accidents)`). N'oubliez pas que ces effets sont multiplicatifs une fois transformés avec la fonction `exp`. Nous pouvons également utiliser les erreurs standards pour calculer des intervalles de confiance à 95 % des exponentiels des coefficients. Le tableau 8.22 présente l'ensemble des informations pertinentes pour l'interprétation des résultats.

```
# Calcul des coefficients en exponentiel et des intervalles de confiance
tableau <- summary(modele2)$coefficients
coeffs <- tableau[,1]
err.std <- tableau[,2]
expcoeff <- exp(coeffs)
exp2.5 <- exp(coeffs - 1.96*err.std)
exp975 <- exp(coeffs - 1.96*err.std)
pvals <- tableau[,4]
tableauComplet <- cbind(coeffs,err.std,expcoeff,exp2.5,exp975,pvals)
# print(tableauComplet)
```

TAB. 8.22 : Résultats du modèle de quasi-Poisson

Variable	Coeff.	exp(Coeff.)	Val.p	IC 2,5 % exp(Coeff.)	IC 97,5 % exp(Coeff.)	Sign.
Constante	-3,680	0,030	0,000	0,010	0,090	***
Feux_auto	1,100	3,000	0,000	1,970	4,660	***
Feux_piet	0,330	1,390	0,009	1,090	1,790	**
Pass_piet	0,340	1,400	0,149	0,880	2,200	
Terreplein	-0,360	0,700	0,099	0,440	1,050	.
Apaisement	0,290	1,330	0,157	0,880	1,950	
LogEmploi	0,230	1,260	0,017	1,040	1,520	*
Densite_pop	0,000	1,000	0,000	1,000	1,000	***
Entropie	-0,420	0,660	0,271	0,320	1,390	
DensiteInter	0,000	1,000	0,410	1,000	1,010	
Long_arterePS	0,000	1,000	0,684	1,000	1,000	
Artere	0,030	1,030	0,842	0,780	1,360	
NB_voies5	0,640	1,890	0,000	1,460	2,440	***

Parmi les variables décrivant les aménagements de l'intersection, nous constatons que les présences d'un feu de circulation et d'un feu de traversée pour les piétons multiplient le nombre attendu d'accidents à une intersection par 3,0 et 1,39. Par contre, les présences d'un passage piéton, d'un terre-plein ou de mesures d'apaisement n'ont pas d'effets significatifs (valeurs de $p > 0,05$). Concernant les variables décrivant l'environnement à proximité des intersections, nous observons que la concentration d'emplois et la densité de population contribuent toutes les deux à augmenter le nombre d'accidents à une intersection, bien que leurs effets soient limités. Enfin, la présence d'une rue à cinq voies à l'intersection augmente le nombre d'accidents attendu à l'intersection de 89 %. Nous ne détaillons pas plus les résultats, car nous utilisons le même jeu de données dans les prochaines sections.

8.3.2 Modèle binomial négatif

Dans le cas où une variable de comptage est marquée par une sur ou sous-dispersion, la distribution de Poisson n'est pas en mesure de capturer efficacement sa variance. Pour contourner ce problème, il est possible d'utiliser la distribution binomiale négative plutôt que la distribution de Poisson (ou quasi-Poisson). Cette distribution peut être décrite comme une généralisation de la distribution de Poisson : elle inclut un second paramètre θ contrôlant la dispersion. L'intérêt premier de ce changement de distribution est que l'interprétation des paramètres est la même pour les deux modèles, tout en contrôlant directement l'effet d'une potentielle sur ou sous-dispersion.

8.3.2.1 Conditions d'application

Les conditions d'application d'un modèle binomial négatif sont presque les mêmes que celles d'un modèle de Poisson. La seule différence est que la condition d'absence de sur ou sous-dispersion est remplacée par une condition de respect du lien espérance-variance. En effet, dans un modèle binomial négatif, le paramètre de dispersion θ est combiné avec μ (l'espérance) pour exprimer la dispersion de la distribution. Dans le package `mgcv` que nous utilisons dans l'exemple, le lien entre μ , θ et la variance est le suivant :

$$\text{variance} = \mu + \mu^{\frac{2}{\theta}} \quad (8.22)$$

Il s'agit donc d'un **modèle hétéroscléastique** : sa variance n'est pas fixe, mais varie en fonction de sa propre espérance. Si celle-ci augmente, la variance augmente (comme pour un modèle de Poisson), et l'intensité de cette augmentation est contrôlée par le paramètre θ . Si cette condition n'est pas respectée, l'analyse des résidus simulés révélera un problème de dispersion.

8.3.2.2 Exemple appliqué dans R

Dans l'exemple précédent avec le modèle de Poisson, nous avons observé une certaine sur-dispersion que nous avons contournée en utilisant un modèle de quasi-Poisson. Dans l'article original, les auteurs ont opté pour un modèle binomial négatif, ce que nous reproduisons ici. Les variables utilisées sont les mêmes que pour le modèle de Poisson. Nous utilisons le package `mgcv` et sa fonction `gam` pour ajuster le modèle.

Vérification des conditions d'application

Nous avons vu précédemment que nos variables indépendantes ne sont pas marquées par une multicolinéarité forte. Il n'est pas nécessaire de recalculer les valeurs de VIF puisque nous utilisons les mêmes données. La première étape du diagnostic est donc de calculer les distances de Cook.

TAB. 8.23 : Carte d'identité du modèle binomial négatif

Type de variable dépendante	Variable de comptage
Distribution utilisée	Négative binomiale
Formulation	$Y \sim NB(\mu, \theta)$ $g(\mu) = \beta_0 + \beta X$ $g(x) = \log(x)$
Fonction de lien	\log
Paramètre modélisé	μ
Paramètres à estimer	β_0, β et θ
Conditions d'application	Absence d'excès de zéros, respect du lien variance-moyenne

```

library(mgcv)
# Chargement des données
data_accidents <- read.csv("data/glm/accident_pietons.csv", sep = ";")
# Ajustement d'une première version du modèle
modelnb <- gam(Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet +
                 Terreplein + Apaisement +
                 LogEmploi + Densite_pop + Entropie + DensiteInter +
                 Long_arterePS + Artere + NB_voies5,
                 family = nb(link="log"),
                 data = data_accidents)
# Calcul et affichage des distances de Cook
cooksd <- cooks.distance(modelnb)
df <- data.frame(
  cook = cooksd,
  oid = 1:length(cooksd)
)
ggplot(data = df) +
  geom_point(aes(x = oid, y = cook), size = 0.5, color = rgb(0.4,0.4,0.4)) +
  labs(x = "", y = "distance de Cook")+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank())

```

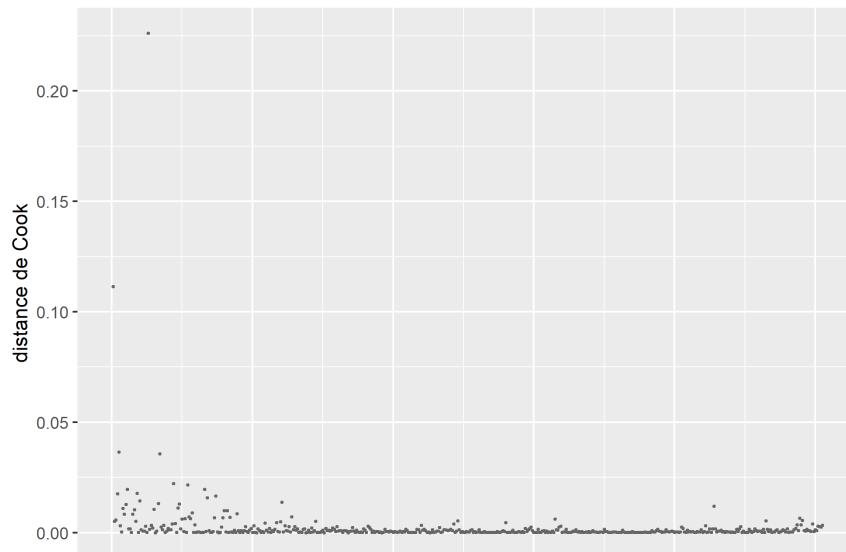


FIG. 8.31 : Distances de Cook pour le modèle binomial négatif

Nous observons, dans la figure 8.31, que quatre observations se distinguent très nettement des autres.

```

cas_etrange <- subset(data_accidents, cooksd > 0.03)
print(cas_etrange)

##   Nbr_acci Feux_auto Feux_piet Pass_piet Terreplein Apaisement EmpTotBuffer
## 1       19        1       1       1        1        0      7208.538
## 5       12        1       0       1        0        0      8585.350
## 26       7        0       0       1        0        0      1342.625
## 34       6        0       0       1        0        0     12516.410

```

```

##   Densite_pop Entropie DensiteInter Long_arterePS Artere NB_voies5 log_acci
## 1    5980.923 0.8073926    42.41597    6955.00     1      1 2.995732
## 5    8655.430 0.7607852    89.11495    6412.27     0      0 2.564949
## 26   2751.012 0.0000000    73.35344    2849.66     0      0 2.079442
## 34   8950.942 0.4300549    74.91879    8443.01     1      0 1.945910
##   catego_acci catego_acci2 Arret VAG sum_app LogEmploi AccOrdinal PopHa
## 1            1            1     0     1     4 8.883021     2 5.980923
## 5            1            1     0     1     4 9.057813     2 8.655430
## 26           1            1     1     1     3 7.202382     2 2.751012
## 34           1            1     1     0     3 9.434796     2 8.950942

```

Il s'agit à nouveau de quatre observations avec un grand nombre d'accidents. Nous décidons de les retirer du jeu de données pour ne pas fausser les résultats concernant l'ensemble des autres intersections. Dans une analyse plus détaillée, il serait judicieux de chercher à comprendre pourquoi ces quatre observations sont particulièrement accidentogènes.

```

data2 <- subset(data_accidents, cooksd < 0.03)
# Ajustement d'une première version du modèle
modelnb <- gam(Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet +
                 Terreplein + Apaisement +
                 LogEmploi + Densite_pop + Entropie + DensiteInter +
                 Long_arterePS + Artere + NB_voies5,
                 family = nb(link="log"),
                 data = data2)

# Calcul et affichage des distances de Cook
cooksd <- cooks.distance(modelnb)
df <- data.frame(
  cook = cooksd,
  oid = 1:length(cooksd)
)
ggplot(data = df) +
  geom_point(aes(x = oid, y = cook), size = 0.5, color = rgb(0.4,0.4,0.4)) +
  labs(x = "", y = "distance de Cook") +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank())

```

Après avoir retiré ces quatre observations, les distances de Cook (figure 8.32) ne révèlent plus d'observations fortement influentes dans le modèle. La prochaine étape du diagnostic est donc d'analyser les résidus simulés.

```

# Extraction de la valeur de theta
theta <- modelnb$family$getTheta(T)
nsim <- 1000
# Extraction des valeurs prédites par le modèle
mus <- predict(modelnb, type = "response")
# Calcul des simulations
cols <- lapply(1:length(mus), function(i){
  mu <- mus[[i]]
  sims <- rnbinom(n = nsim, mu = mu, size = theta)
  return(sims)
})
mat_sims <- do.call(rbind, cols)
# Calcul des résidus simulés

```

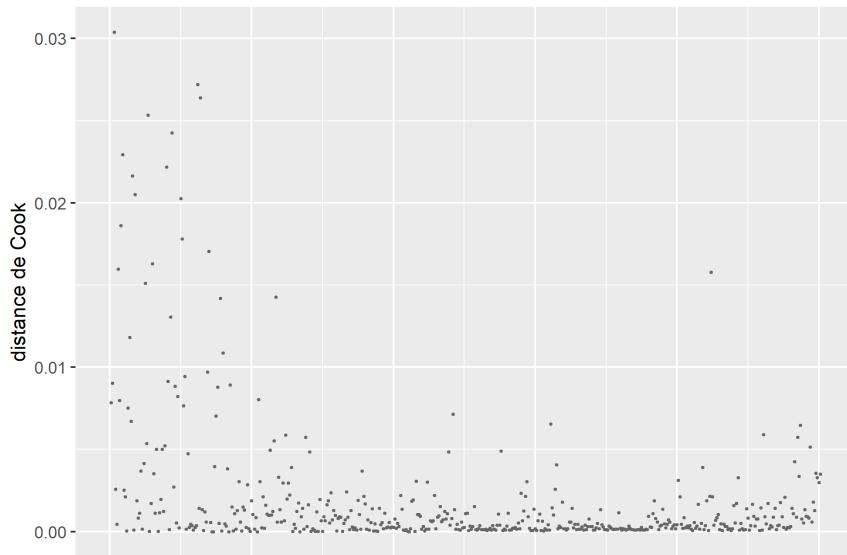


FIG. 8.32 : Distances de Cook pour le modèle binomial négatif (après avoir retiré quatre observations fortement influentes)

```
sim_res <- createDHARMA(simulatedResponse = mat_sims,
                         observedResponse = data2$Nbr_acci,
                         fittedPredictedResponse = mus,
                         integerResponse = T)

# Affichage du diagnostic
plot(sim_res)
```

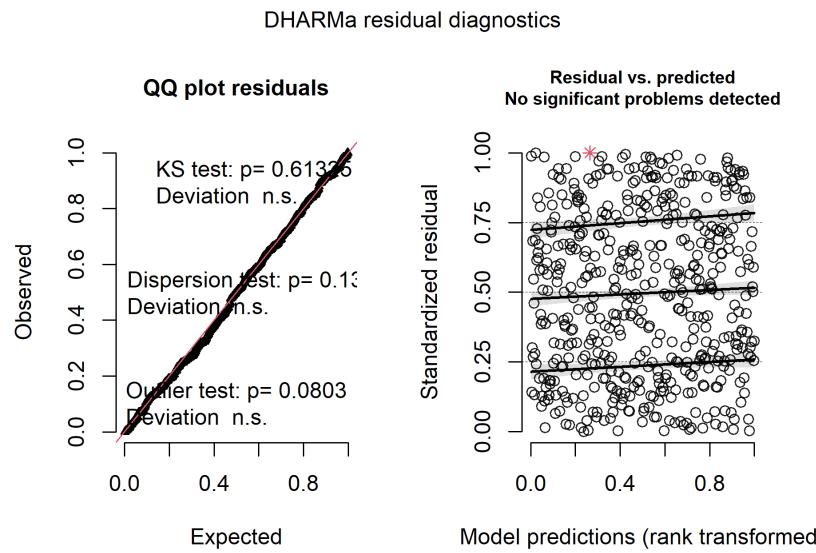


FIG. 8.33 : Diagnostic général des résidus simulés pour le modèle binomial négatif

La figure 8.33 présentant le diagnostic des résidus simulés, montre que ces derniers suivent bien une distribution uniforme et aucun problème de dispersion ni de valeurs aberrantes. La figure 8.34 permet

de comparer la distribution originale de la variable Y et les simulations issues du modèle (intervalles de confiance représentés en bleu). Nous constatons que le modèle parvient bien à reproduire la distribution originale, et ce, même pour les valeurs les plus extrêmes de la distribution.

```

# Extraction des valeurs prédites par le modèle
mus <- predict(modelnb, type="response")
# Génération de 1000 simulations pour chaque prédiction
theta <- modelnb$family$getTheta(T)
nsim <- 1000
cols <- lapply(1:length(mus), function(i){
  mu <- mus[[i]]
  sims <- round(rnbinom(n = nsim, mu = mu, size = theta))
  return(sims)
})
mat_sims <- do.call(rbind, cols)
# Préparation des données pour le graphique (valeurs réelles)
counts <- data.frame(table(data2$Nbr_acc))
names(counts) <- c("nb_accident", "frequence")
counts$nb_accident <- as.numeric(as.character(counts$nb_accident))
counts$prop <- counts$frequence / sum(counts$frequence)
# Préparation des données pour le graphique (valeurs simulées)
df1 <- data.frame(count = 0:25)
count_sims <- lapply(1:nsim, function(i){
  sim <- mat_sims[,i]
  cnt <- data.frame(table(sim))
  df2 <- merge(df1, cnt, by.x="count", by.y = "sim", all.x = T, all.y=F)
  df2$Freq <- ifelse(is.na(df2$Freq), 0, df2$Freq)
  return(df2$Freq)
})
count_sims <- do.call(cbind, count_sims)
df_sims <- data.frame(
  val = 0:25,
  med = apply(count_sims, MARGIN = 1, median),
  lower = apply(count_sims, MARGIN = 1, quantile, probs = 0.025),
  upper = apply(count_sims, MARGIN = 1, quantile, probs = 0.975)
)
# Affichage du graphique
ggplot() +
  geom_bar(aes(x=nb_accident, weight = frequence), width = 0.6, data = counts) +
  geom_errorbar(aes(x = val, ymin = lower, ymax = upper),
                data = df_sims, color = "blue", width = 0.6) +
  geom_point(aes(x = val, y = med), color = "blue", size = 1.3, data = df_sims) +
  scale_x_continuous(limits = c(-0.5,7), breaks = c(0:7)) +
  xlim(-1,12) +
  labs(subtitle = "",
       x = "nombre d'accidents",
       y = "nombre d'occurrences")

```

À titre de comparaison, nous pouvons à nouveau réaliser le graphique permettant de visualiser si la variance attendue par le modèle est proche de celle effectivement observée dans les données. Nous avons constaté avec ce graphique, lorsque nous ajustions un modèle de Poisson, que la variance des données était trop grande comparativement à celle attendue par le modèle (figure 8.26).

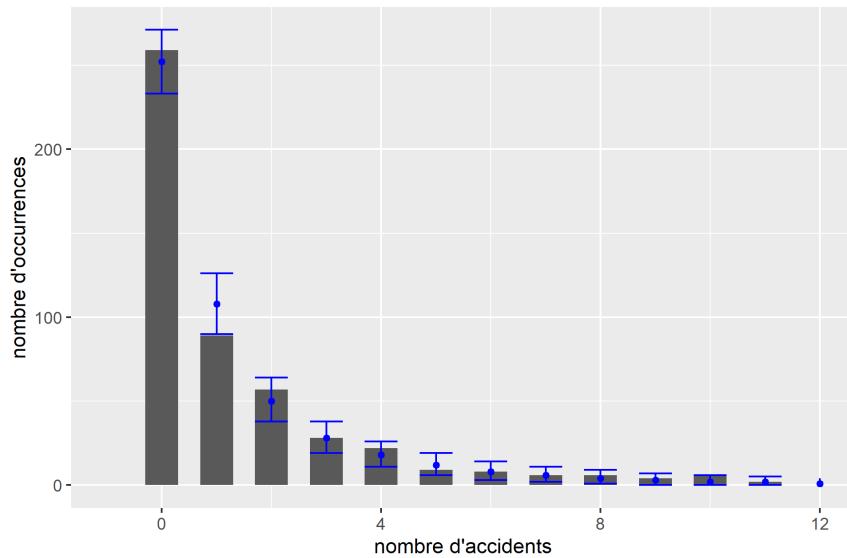


FIG. 8.34 : Comparaison de la distribution originale et des simulations pour le modèle binomial négatif

```

# Extraction des prédictions du modèle
mus <- predict(modelnb, type = "response")
# Création d'un DataFrame pour contenir la prédition et les vraies valeurs
df1 <- data.frame(
  mus = mus,
  reals = data2$Nbr_acci
)
# Calcul de l'intervalle de confiance à 95 % selon la distribution de Poisson
# et stockage dans un second DataFrame
seqa <- seq(0,round(max(mus)),1)
df2 <- data.frame(
  mus = seqa,
  lower = qnbinom(p = 0.025, mu = seqa, size = theta),
  upper = qnbinom(p = 0.975, mu = seqa, size = theta)
)
# Affichage des valeurs réelles et prédites (en rouge)
# et de leur variance selon le modèle (en noir)
ggplot() +
  geom_errorbar(data = df2,
    mapping = aes(x = mus, ymin = lower, ymax = upper),
    width = 0.2, color = rgb(0.4,0.4,0.4)) +
  geom_point(data = df1,
    mapping = aes(x = mus, y = reals),
    color = "red", size = 0.5) +
  labs(x = 'valeurs prédites',
       y = "valeurs réelles")

```

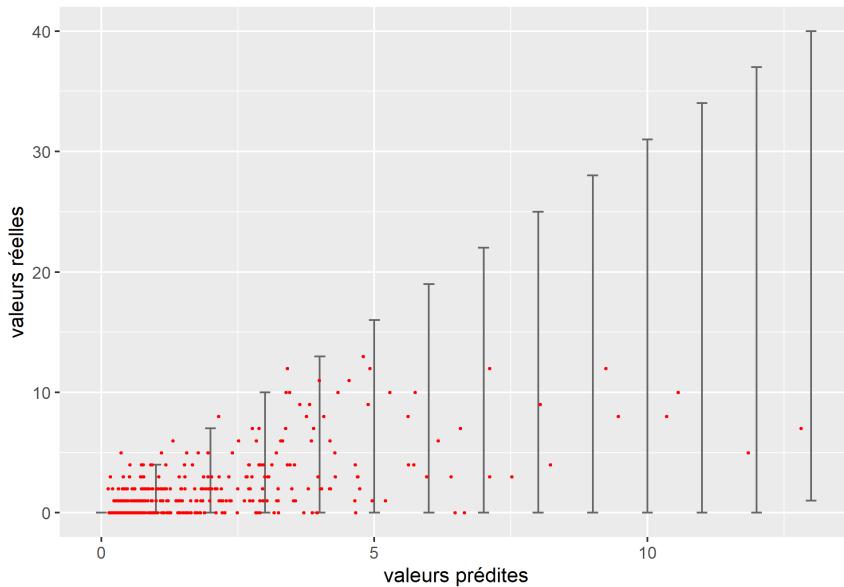


FIG. 8.35 : Représentation de la sur-dispersion des données dans le modèle de Poisson

Nous pouvons ainsi constater à la figure 8.35 que le modèle binomial négatif autorise une variance bien plus grande que le modèle de Poisson et est ainsi mieux ajusté aux données.

Vérification de la qualité d'ajustement

```
# Calcul des pseudo R2
rsqs(loglike.full = logLik(modelnb),
      loglike.null = logLik(modelnull),
      full.deviance = deviance(modelnb),
      null.deviance = modelnb$null.deviance,
      nb.params = modelnb$rank,
      n = nrow(data2))

## `$deviance expliquée`
## [1] 0.458052
##
## `$McFadden ajusté`
## 'log Lik.' 0.384353 (df=14)
##
## `$Cox and Snell`
## 'log Lik.' 0.8304773 (df=14)
##
## $Nagelkerke
## 'log Lik.' 0.8399731 (df=14)

# Calcul du RMSE
sqrt(mean((predict(modelnb, type = "response") - data2$Nbr_accu)**2))

## [1] 1.825278
```

Le modèle parvient à expliquer 45 % de la déviance. Il obtient un R^2 ajusté de McFadden de 0,14 et un R^2

de Nagelkerke de 0,42. L'erreur moyenne quadratique de la prédition est de 1,82, ce qui est identique au modèle de Poisson ajusté précédemment.

Interprétation des résultats

Il est possible d'accéder à l'ensemble des coefficients du modèle via la fonction `summary`. À nouveau, les coefficients doivent être convertis avec la fonction exponentielle (du fait de la fonction de lien `log`) et interprétés comme des effets multiplicatifs. Le tableau 8.24 présente les coefficients estimés par le modèle. Les résultats sont très similaires à ceux du modèle de quasi-Poisson original. Nous notons cependant que la variable représentant la présence d'un feu pour piéton n'est plus significative au seuil de 0,05.

TAB. 8.24 : Résultats du modèle binomial négatif

Variable	Coeff.	exp(Coeff.)	Val.p	IC 2,5 % exp(Coeff.)	IC 97,5 % exp(Coeff.)	Sign.
Constante	-3,880	0,020	0,000	0,010	0,080	***
Feux_auto	1,130	3,100	0,000	2,030	4,710	***
Feux_piet	0,350	1,420	0,016	1,060	1,900	*
Pass_piet	0,220	1,240	0,300	0,830	1,880	
Terreplein	-0,340	0,710	0,155	0,440	1,140	
Apaisement	0,240	1,270	0,315	0,790	2,030	
LogEmploi	0,230	1,260	0,025	1,030	1,550	*
Densite_pop	0,000	1,000	0,000	1,000	1,000	***
Entropie	-0,170	0,840	0,669	0,380	1,860	
DensiteInter	0,000	1,000	0,925	0,990	1,010	
Long_arterePS	0,000	1,000	0,587	1,000	1,000	
Artere	0,110	1,110	0,497	0,820	1,510	
NB_voies5	0,700	2,010	0,000	1,480	2,750	***

8.3.3 Modèle de Poisson avec excès fixe de zéros

Dans le cas où la variable Y comprendrait significativement plus de zéros que ce que suppose une distribution de Poisson, il est possible d'utiliser la distribution de Poisson avec excès de zéros. Pour rappel, cette distribution ajoute un paramètre p contrôlant pour la proportion de zéros dans la distribution. Du point de vue conceptuel, cela revient à formuler l'hypothèse suivante : dans les données que nous avons observées, deux processus distincts sont à l'œuvre. Le premier est issu d'une distribution de Poisson et l'autre produit des zéros qui s'ajoutent aux données. Les zéros produits par la distribution de Poisson sont appelés les **vrais zéros**, alors que ceux produits par le second phénomène sont appelés les **faux zéros**.

TAB. 8.25 : Carte d'identité du modèle de Poisson avec excès fixe de zéros

Type de variable dépendante	Variable de comptage
Distribution utilisée	Poisson avec excès de zéros
Formulation	$Y \sim ZIP(\mu, \theta)$ $g(\lambda) = \beta_0 + \beta X$ $g(x) = \log(x)$
Fonction de lien	\log
Paramètre modélisé	λ
Paramètres à estimer	β_0, β et p
Conditions d'application	Absence de sur-dispersion

Dans cette formulation, p est fixé. Nous n'avons donc aucune information sur ce qui produit les zéros supplémentaires, mais seulement leur proportion totale dans le jeu de données.

8.3.3.1 Interprétation des paramètres

L'interprétation des paramètres est identique à celle d'un modèle de Poisson. Le paramètre p représente la proportion de faux zéros dans la variable Y une fois que les variables indépendantes sont contrôlées.

8.3.3.2 Exemple appliqué dans R

La variable de comptage des accidents des piétons que nous avons utilisée dans les deux exemples précédents semble être une bonne candidate pour une distribution de Poisson avec excès de zéros. En effet, nous avons pu constater une sur-dispersion dans le modèle de Poisson original, ainsi qu'un nombre important d'intersections sans accident. Tentons donc d'améliorer notre modèle en ajustant un excès fixe de zéros. Nous utilisons la fonction `gamlss` du package `gamlss`.

```
library(gamlss)
modelzi <- gamlss(formula = Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet +
                    Terreplein + Apaisement + LogEmploi + Densite_pop +
                    Entropie + DensiteInter + Long_arterePS + Artere + NB_voies5,
                    sigma.formula = ~1,
                    family = ZIP(mu.link = "log", sigma.link="logit"),
                    data = data_accidents)

## GAMLSS-RS iteration 1: Global Deviance = 1516.53
## GAMLSS-RS iteration 2: Global Deviance = 1514.656
## GAMLSS-RS iteration 3: Global Deviance = 1514.52
## GAMLSS-RS iteration 4: Global Deviance = 1514.508
## GAMLSS-RS iteration 5: Global Deviance = 1514.506
## GAMLSS-RS iteration 6: Global Deviance = 1514.506

modelnull <- glm(formula = Nbr_acci ~ 1,
                  family = poisson(link="log"),
                  data = data_accidents)

# Constante pour p
coeff_p <- modelzi$sigma.coefficients
cat("Coefficient pour p =", round(coeff_p,4))

## Coefficient pour p = -1.4376

# Calcul de la déviance expliquée
1 - deviance(modelzi) / deviance(modelnull)

## [1] 0.08267513

# Calcul de la probabilité de base  $p$  d'être un faux 0
# en appliquant l'inverse de la fonction logistique
exp(-coeff_p) / (1+exp(-coeff_p))

## (Intercept)
##      0.80809
```

Nous constatons immédiatement que le modèle avec excès fixe de zéros est peu ajusté aux données. Cette version du modèle ne parvient à capter que 8 % de la déviance, ce qui s'explique facilement, car nous

n'avons donné aucune variable au modèle pour distinguer les vrais et les faux zéros. Pour cela, nous devons passer au prochain modèle : Poisson avec excès ajusté de zéros. Notons tout de même que d'après ce modèle, 81 % des observations seraient des faux zéros.

8.3.4 Modèle de Poisson avec excès ajusté de zéros

Nous avons vu dans le modèle précédent, que l'excès de zéro était conceptualisé comme la combinaison de deux phénomènes, l'un issu d'une distribution de Poisson que nous souhaitons modéliser, et l'autre générant des zéros supplémentaires. Il est possible d'aller plus loin que de simplement contrôler la proportion de zéros supplémentaires en modélisant explicitement ce second processus en ajoutant une deuxième équation au modèle. Cette deuxième équation a pour enjeu de modéliser p (la proportion de 0) à l'aide de variables indépendantes, ces dernières pouvant se retrouver dans les deux parties du modèle. L'idée étant que, pour chaque observation, le modèle évalue sa probabilité d'être un faux zéro (partie binomiale), et le nombre attendu d'accidents.

8.3.4.1 Interprétation des paramètres

L'interprétation des paramètres β_0 et β est identique à celle d'un modèle de Poisson. Les paramètres α_0 et α sont identiques à ceux d'un modèle binomial. Plus spécifiquement, ces derniers paramètres modélisent la probabilité d'observer des valeurs supérieures à zéro.

8.3.4.2 Exemple appliqué

Nous avons vu, dans l'exemple précédent, que l'utilisation du modèle avec excès fixe de zéros pour les données d'accident des piétons aux intersections ne donnait pas de résultats satisfaisants. Nous tentons ici d'améliorer le modèle en ajoutant les variables indépendantes significatives du modèle Poisson dans la seconde équation de régression destinée à détecter les faux zéros.

Vérification des conditions d'application

Pour un modèle de Poisson avec excès de zéros, il n'est pas possible de calculer les distances de Cook. Nous devons donc directement passer à l'analyse des résidus simulés.

```
# Ajuster une première version du modèle
modelza <- gamlss(formula = Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet +
                    Terreplein + Apaisement + LogEmploi + Densite_pop +
                    Entropie + DensiteInter + Long_arterePS + Artere + NB_voies5,
                    sigma.formula = ~1 + Feux_auto + Feux_piet + Densite_pop + NB_voies5,
```

TAB. 8.26 : Carte d'identité du modèle de Poisson avec excès ajusté de zéros

Type de variable dépendante	Variable de comptage
Distribution utilisée	Poisson avec excès de zéros
Formulation	$Y \sim ZIP(\mu, \theta)$ $g(\mu) = \beta_0 + \beta X$ $s(p) = \alpha_0 + \alpha X$ $g(x) = \log(x)$ $s(x) = \log\left(\frac{x}{1-x}\right)$
Fonction de lien	log et logistique
Paramètre modélisé	μ et p
Paramètres à estimer	β_0, β, α_0 et α
Conditions d'application	Absence de sur-dispersion

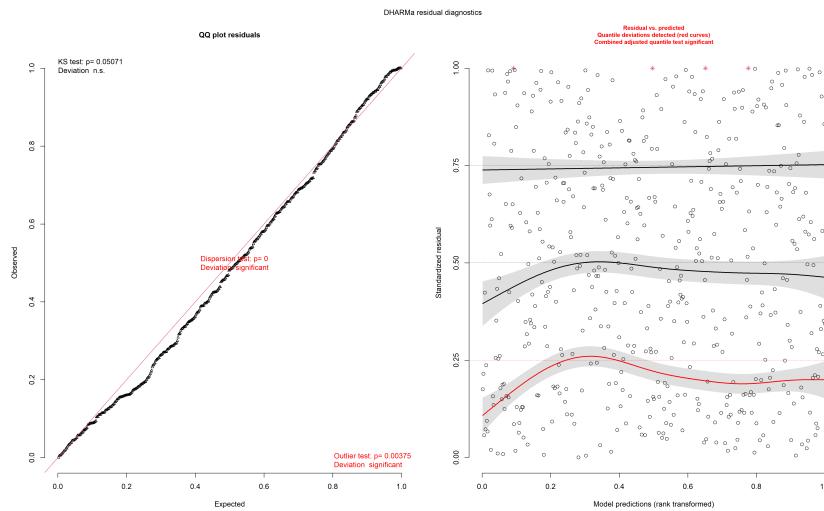


FIG. 8.36 : Diagnostic général des résidus simulés du modèle de Poisson avec excès de zéros ajusté

```
##      Densite_pop Entropie DensiteInter Long_arterePS Artere NB_voies5 log_acci
## 1      5980.923 0.8073926    42.41597   6955.00     1       1 2.995732
## 5      8655.430 0.7607852    89.11495   6412.27     0       0 2.564949
## 26     2751.012 0.0000000    73.35344   2849.66     0       0 2.079442
## 44     8090.478 0.7879618    66.86856   4517.65     0       0 1.791759
## 482    8988.260 0.4486079    60.93742   3821.78     1       0 0.000000
##      catego_acci catego_acci2 Arret VAG sum_app LogEmploi AccOrdinal PopHa
## 1          1           1     0     1      4 8.883021      2 5.980923
## 5          1           1     0     1      4 9.057813      2 8.655430
## 26         1           1     1     1      3 7.202382      2 2.751012
## 44         1           1     1     1      4 8.516897      2 8.090478
## 482        0           0     0     0      3 7.503240      0 8.988260
```

Nous retirons des données les quelques observations pouvant avoir une trop forte influence sur le modèle. Après réajustement, la figure 8.37 nous informe que nous n'avons plus de valeurs aberrantes restantes ni de fort problème de dispersion. En revanche, le premier quartile des résidus tant à être plus faible que ce que nous aurions pu nous attendre d'une distribution uniforme. Ce constat laisse penser que le modèle a du mal à bien identifier les faux zéros. Ce résultat n'est pas étonnant, car aucune variable n'avait été identifiée à cette fin dans l'article original (Cloutier et al. 2014) qui utilisait un modèle binomial négatif.

```
data2 <- subset(data_accidents, isOutlier==FALSE)
# Ajuster une première version du modèle
modelza <- gamlss(formula = Nbr_acci ~ Feux_auto + Feux_piet + Pass_piet +
  Terreplein + Apaisement + LogEmploi + Densite_pop +
  Entropie + DensiteInter + Long_arterePS + Artere + NB_voies5,
  sigma.formula = ~1 + Feux_auto + Feux_piet + Densite_pop + NB_voies5,
  family = ZIP(mu.link = "log", sigma.link="logit"),
  data = data2)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 1390.884
## GAMLSS-RS iteration 2: Global Deviance = 1381.199
## GAMLSS-RS iteration 3: Global Deviance = 1378.319
## GAMLSS-RS iteration 4: Global Deviance = 1377.422
```

```
## GAMLSS-RS iteration 5: Global Deviance = 1377.118
## GAMLSS-RS iteration 6: Global Deviance = 1377.015
## GAMLSS-RS iteration 7: Global Deviance = 1376.982
## GAMLSS-RS iteration 8: Global Deviance = 1376.972
## GAMLSS-RS iteration 9: Global Deviance = 1376.968
## GAMLSS-RS iteration 10: Global Deviance = 1376.967
## GAMLSS-RS iteration 11: Global Deviance = 1376.966
```

```
# Extraire la prédiction des valeurs lambda
lambdas <- predict(modelza, type = "response", what = "mu")
# Extraire la prédiction des valeurs p
ps <- predict(modelza, type = "response", what = "sigma")
# Calculer la combinaison de ces deux éléments
preds <- lambdas * ps
# Effectuer les 1000 simulations
nsim <- 1000
cols <- lapply(1:length(lambdas), function(i){
  lambda <- lambdas[[i]]
  p <- ps[[i]]
  sims <- rZIP(n = nsim, mu = lambda, sigma = p)
  return(sims)
})
mat_sims <- do.call(rbind, cols)
# Calculer les résidus simulés
sim_res <- createDHARMA(simulatedResponse = mat_sims,
                           observedResponse = data2$Nbr_acci,
                           fittedPredictedResponse = preds,
                           integerResponse = T)
plot(sim_res)
```

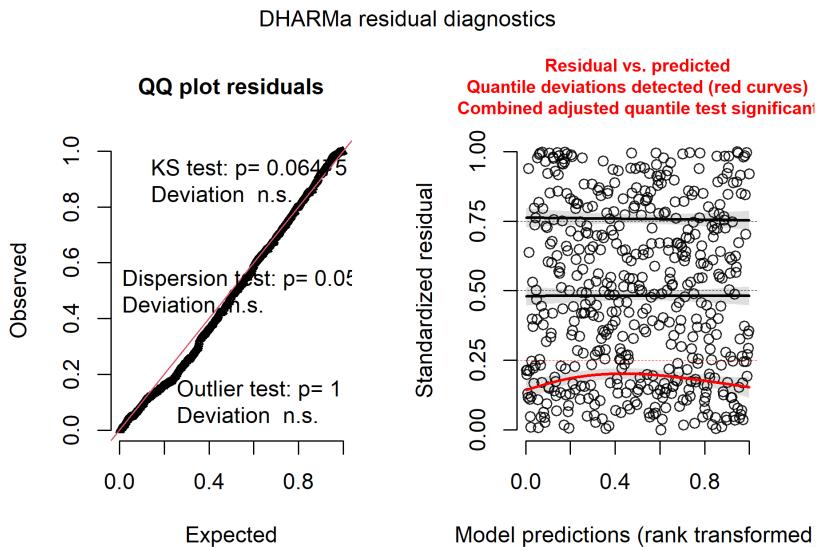


FIG. 8.37 : Diagnostic général des résidus simulés du modèle de Poisson avec excès de zéros ajusté (sans valeurs aberrantes)

Nous pouvons une fois encore comparer des simulations issues du modèle et de la distribution originale

de la variable Y . La figure 8.38 montre clairement que les simulations du modèle (en bleu) sont très éloignées dans la distribution originale (en gris), ce qui remet directement en question la pertinence de ce modèle.

```
# Extraire la prédiction des valeurs lambda
lambdas <- predict(modelza, type = "response", what = "mu")
# Extraire la prédiction des valeurs p
ps <- predict(modelza, type = "response", what = "sigma")
# Génération de 1000 simulations pour chaque prédiction
nsim <- 1000
cols <- lapply(1:length(lambdas), function(i){
  lambda <- lambdas[[i]]
  p <- ps[[1]]
  sims <- round(rZIP(nsim, mu=lambda, sigma = p))
  return(sims)
})
mat_sims <- do.call(rbind, cols)
# Préparation des données pour le graphique (valeurs réelles)
counts <- data.frame(table(data2$Nbr_acc))
names(counts) <- c("nb_accident", 'frequence')
counts$nb_accident <- as.numeric(as.character(counts$nb_accident))
counts$prop <- counts$frequence / sum(counts$frequence)
# Préparation des données pour le graphique (valeurs simulées)
df1 <- data.frame(count = 0:25)
count_sims <- lapply(1:nsim, function(i){
  sim <- mat_sims[,i]
  cnt <- data.frame(table(sim))
  df2 <- merge(df1,cnt, by.x="count", by.y = "sim", all.x = T, all.y=F)
  df2$Freq <- ifelse(is.na(df2$Freq),0,df2$Freq)
  return(df2$Freq)
})
count_sims <- do.call(cbind,count_sims)
df_sims <- data.frame(
  val = 0:25,
  med = apply(count_sims, MARGIN = 1, median),
  lower = apply(count_sims, MARGIN = 1, quantile, probs = 0.025),
  upper = apply(count_sims, MARGIN = 1, quantile, probs = 0.975)
)
# Affichage du graphique
ggplot() +
  geom_bar(aes(x=nb_accident, weight = frequence), width = 0.6, data = counts) +
  geom_errorbar(aes(x = val, ymin = lower, ymax = upper),
                data = df_sims, color = "blue", width = 0.6) +
  geom_point(aes(x = val, y = med), color = "blue", size = 1.3, data = df_sims) +
  scale_x_continuous(limits = c(-0.5,7), breaks = c(0:7)) +
  xlim(-1,12) +
  labs(subtitle = "",
       x = "nombre d'accidents",
       y = "nombre d'occurrences")
```

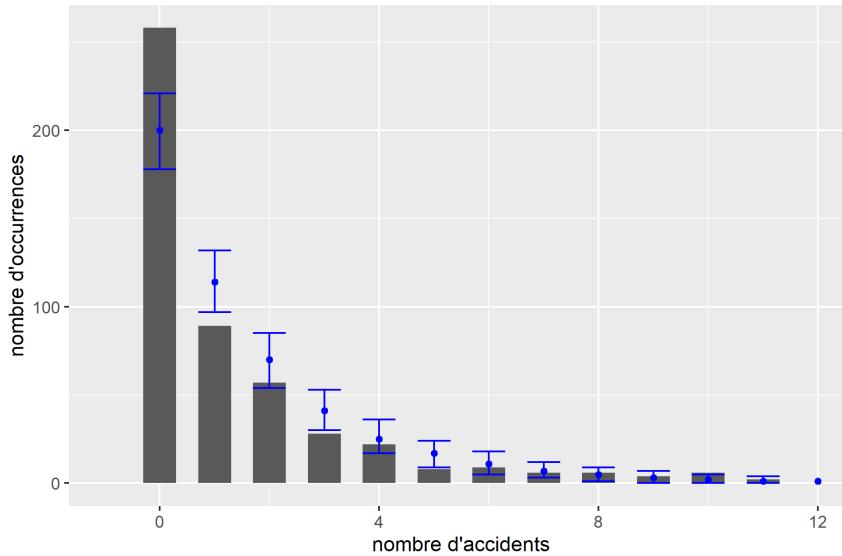


FIG. 8.38 : Comparaison de la distribution originale et des simulations pour le modèle de Poisson avec excès de zéros ajusté

Vérification la qualité d'ajustement

```

modelenull <- glm(Nbr_acci ~ 1,
  family = poisson(link="log"),
  data = data2)

# Calcul des R2
rsqs(loglike.full = loglik(modelza),
  loglike.null = logLik(modelenull),
  full.deviance = deviance(modelza),
  null.deviance = deviance(modelenull),
  nb.params = modelza$sigma.df + modelza$mu.df,
  n = nrow(data2)
)

## `$deviance expliquee`
## [1] 0.1073371
##
## `$McFadden ajuste`
## 'log Lik.' 0.3588483 (df=18)
##
## `$Cox and Snell`
## 'log Lik.' 0.8086509 (df=18)
##
## $Nagelkerke
## 'log Lik.' 0.8186255 (df=18)

# Calcul du RMSE
sqrt(mean((preds - data2$Nbr_acci)**2))

## [1] 2.635322

```

Le modèle avec excès de zéro ajusté ne parvient à expliquer que 11 % de la déviance totale. Il obtient toutefois des valeurs de R² assez hautes (McFadden ajusté : 0,36, Nagerlkerke : 0,82). Son RMSE est très élevé (2,6), comparativement à celui que nous avons obtenu avec le modèle binomial négatif (1,9). Considérant ces éléments, ce modèle est nettement moins informatif que le modèle binomial négatif et ne devrait pas être retenu. Nous montrons tout de même ici comment interpréter ces résultats.

Interprétation des résultats

L'ensemble des coefficients du modèle sont accessibles avec la fonction `summary`. Les coefficients dédiés à la partie Poisson (appelée **Mu** dans le résumé) doivent être analysés et interprétés de la même manière que s'ils provenaient d'un modèle de Poisson. Les coefficients appartenant à la partie logistique (appelé **Sigma** dans le résumé) doivent être analysés et interprétés de la même manière que s'ils provenaient d'un modèle logistique.

```
# Extraction des résultats
base_table <- summary(modelza)

## ****
## Family: c("ZIP", "Poisson Zero Inflated")
##
## Call: gamlss(formula = Nbr_acci ~ Feux_auto + Feux_piet +
##   Pass_piet + Terreplein + Apaisement + LogEmploi +
##   Densite_pop + Entropie + DensiteInter + Long_arterePS +
##   Artere + NB_voies5, sigma.formula = ~1 + Feux_auto +
##   Feux_piet + Densite_pop + NB_voies5, family = ZIP(mu.link = "log",
##   sigma.link = "logit"), data = data2)
##
## Fitting method: RS()
##
## -----
## Mu link function: log
## Mu Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.609e+00 5.419e-01 -4.814 1.98e-06 ***
## Feux_auto    5.779e-01 1.903e-01  3.036  0.00253 **
## Feux_piet    4.207e-01 1.048e-01  4.012 6.97e-05 ***
## Pass_piet    3.995e-01 1.938e-01  2.061  0.03987 *
## Terreplein   -3.348e-01 1.666e-01 -2.010  0.04502 *
## Apaisement    2.246e-01 1.535e-01  1.463  0.14405
## LogEmploi    1.663e-01 7.509e-02  2.215  0.02723 *
## Densite_pop   8.610e-05 1.501e-05  5.735 1.72e-08 ***
## Entropie     -2.894e-01 2.950e-01 -0.981  0.32698
## DensiteInter  4.035e-03 2.052e-03  1.967  0.04980 *
## Long_arterePS 1.100e-05 2.024e-05  0.544  0.58698
## Artere        8.629e-02 1.117e-01  0.772  0.44035
## NB_voies5    4.295e-01 1.017e-01  4.224 2.87e-05 ***
##
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Sigma link function: logit
```

```

## Sigma Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.0004e+00 5.530e-01 1.816 0.07005 .
## Feux_auto   -1.716e+00 6.031e-01 -2.845 0.00463 **
## Feux_piet    2.661e-01 7.228e-01 0.368 0.71292
## Densite_pop -1.170e-04 5.469e-05 -2.140 0.03286 *
## NB_voies5   -1.831e+00 1.227e+00 -1.493 0.13606
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## No. of observations in the fit: 500
## Degrees of Freedom for the fit: 18
##      Residual Deg. of Freedom: 482
##                          at cycle: 11
##
## Global Deviance: 1376.967
## AIC: 1412.967
## SBC: 1488.829
## ****

```

Multiplication par 1000 des coefficients de population

(effet pour 1000 habitants)

```

base_table[8,1] <- 1000 * base_table[8,1]
base_table[8,2] <- 1000 * base_table[8,2]
base_table[17,1] <- 1000 * base_table[17,1]
base_table[17,2] <- 1000 * base_table[17,2]

```

Multiplication par 1000 des coefficients de longueur artère

(effet pour 1 km)

```

base_table[11,1] <- 1000 * base_table[11,1]
base_table[11,2] <- 1000 * base_table[11,2]

```

Calcul des exponentiels des variables indépendantes

et des intervalles de confiance

```

expcoeff <- exp(base_table[,1])
expcoeff2.5 <- exp(base_table[,1] - 1.96 * base_table[,2])
expcoeff97.5 <- exp(base_table[,1] + 1.96 * base_table[,2])
base_table <- cbind(base_table, expcoeff, expcoeff2.5,expcoeff97.5)

```

Calculer une colonne indiquant le niveau de significativité

```

sign <- case_when(
  base_table[,4] < 0.001 ~ "***",
  base_table[,4] >= 0.001 & base_table[,4]<0.01 ~ "**",
  base_table[,4] >= 0.01 & base_table[,4]<0.05 ~ "*",
  base_table[,4] >= 0.05 & base_table[,4]<0.1 ~ ".",
  TRUE ~ ""
)

```

Arrondir à trois décimales

```

base_table <- round(base_table,3)

```

Enlever les colonnes de valeurs de t et d'erreur standard

```

base_table <- base_table[,c(1,4,5,6,7)]

```

Remplacer les 0 dans la colonne pval

```

base_table[,2] <- ifelse(base_table[,2]=="0","<0.001",base_table[,2])

```

Séparer le tout en deux tableaux

```

part_poiss <- base_table[1:13,]
part_logit <- base_table[14:18,]
# Mettre les bons noms de colonnes
colnames(part_poiss) <- c("Coeff.", "Val.p", "Exp(Coeff.)",
                           "IC 2,5 % exp(Coeff.)", "IC 97,5 % exp(Coeff.)", "Sign.")
colnames(part_logit) <- c("Coeff.", "Val.p", "RC", "IC 2,5 % RC", "IC 97,5 % RC", "Sign.")

```

Nous rapportons les résultats de ce modèle de Poisson avec excès de zéro ajusté dans les tableaux 8.27 et 8.28.

TAB. 8.27 : Résultats de la partie Poisson du modèle de Poisson avec excès de zéros ajusté

Variable	Coeff.	Val.p	Exp(Coeff.)	IC 2,5 % exp(Coeff.)	IC 97,5 % exp(Coeff.)	Sign.
(Intercept)	-2.609	<0.001	0.074	0.025	0.213	***
Feux_auto	0.578	0.003	1.782	1.227	2.588	**
Feux_piet	0.421	<0.001	1.523	1.24	1.87	***
Pass_piet	0.399	0.04	1.491	1.02	2.18	*
Terreplein	-0.335	0.045	0.715	0.516	0.992	*
Apaisement	0.225	0.144	1.252	0.927	1.691	
LogEmploi	0.166	0.027	1.181	1.019	1.368	*
Densite_pop	0.086	<0.001	1.09	1.058	1.122	***
Entropie	-0.289	0.327	0.749	0.42	1.335	
DensiteInter	0.004	0.05	1.004	1	1.008	*
Long_arterePS	0.011	0.587	1.011	0.972	1.052	
Artere	0.086	0.44	1.09	0.876	1.357	
NB_voies5	0.429	<0.001	1.536	1.259	1.875	***

TAB. 8.28 : Résultats de la partie logistique du modèle de Poisson avec excès de zéros ajusté

Variable	Coeff.	Val.p	RC	IC 2,5 % RC	IC 97,5 % RC	Sign.
(Intercept)	1.004	0.07	2.729	0.923	8.067	.
Feux_auto	-1.716	0.005	0.18	0.055	0.586	**
Feux_piet	0.266	0.713	1.305	0.316	5.381	
Densite_pop	-0.117	0.033	0.89	0.799	0.99	*
NB_voies5	-1.831	0.136	0.16	0.014	1.773	

Nous observons ainsi que la présence d'un feu de circulation divise par 5 les chances de rentrer dans la catégorie d'intersection où des accidents peuvent se produire. De même, la densité de population réduit les chances de passer dans cette catégorie de 11 %.

Concernant les coefficients pour la partie Poisson du modèle, nous observons que les présences d'un feu de circulation et d'un feu pour piéton contribuent à multiplier respectivement par 2 et 1,5 le nombre attendu d'accidents à une intersection. De même, la présence d'un axe de circulation à cinq voies augmente de 57 % le nombre d'accidents. Enfin, la densité de population est aussi associée à une augmentation du nombre d'accidents : pour 1 000 habitants supplémentaires autour de l'intersection, nous augmentons le nombre d'accidents attendu de 9 %.

8.3.5 Conclusion sur les modèles destinés à des variables de comptage

Dans cette section, nous avons vu que modéliser une variable de comptage ne doit pas toujours être réalisé avec une simple distribution de Poisson. Il est nécessaire de tenir compte de la sur ou sous-dispersion

potentielle ainsi que de l'excès de zéros. Nous n'avons cependant pas couvert tous les cas. Il est en effet possible d'ajuster des modèles avec une distribution binomiale négative avec excès de zéros (avec le package *gamlss*), ainsi que des modèles de **Hurdle**. Ces derniers ont une approche différente de celle proposée par les distributions ajustées pour tenir compte de l'excès de zéro que nous détaillons dans l'encadré « pour aller plus loin » ci-dessous. Le processus de sélection du modèle peut être résumé avec la figure 8.39. Notez que même en suivant cette procédure, rien ne garantit que votre modèle final reflète bien les données que vous étudiez. L'analyse approfondie des résidus et des prédictions du modèle est la seule façon de déterminer si oui ou non le modèle est fiable.

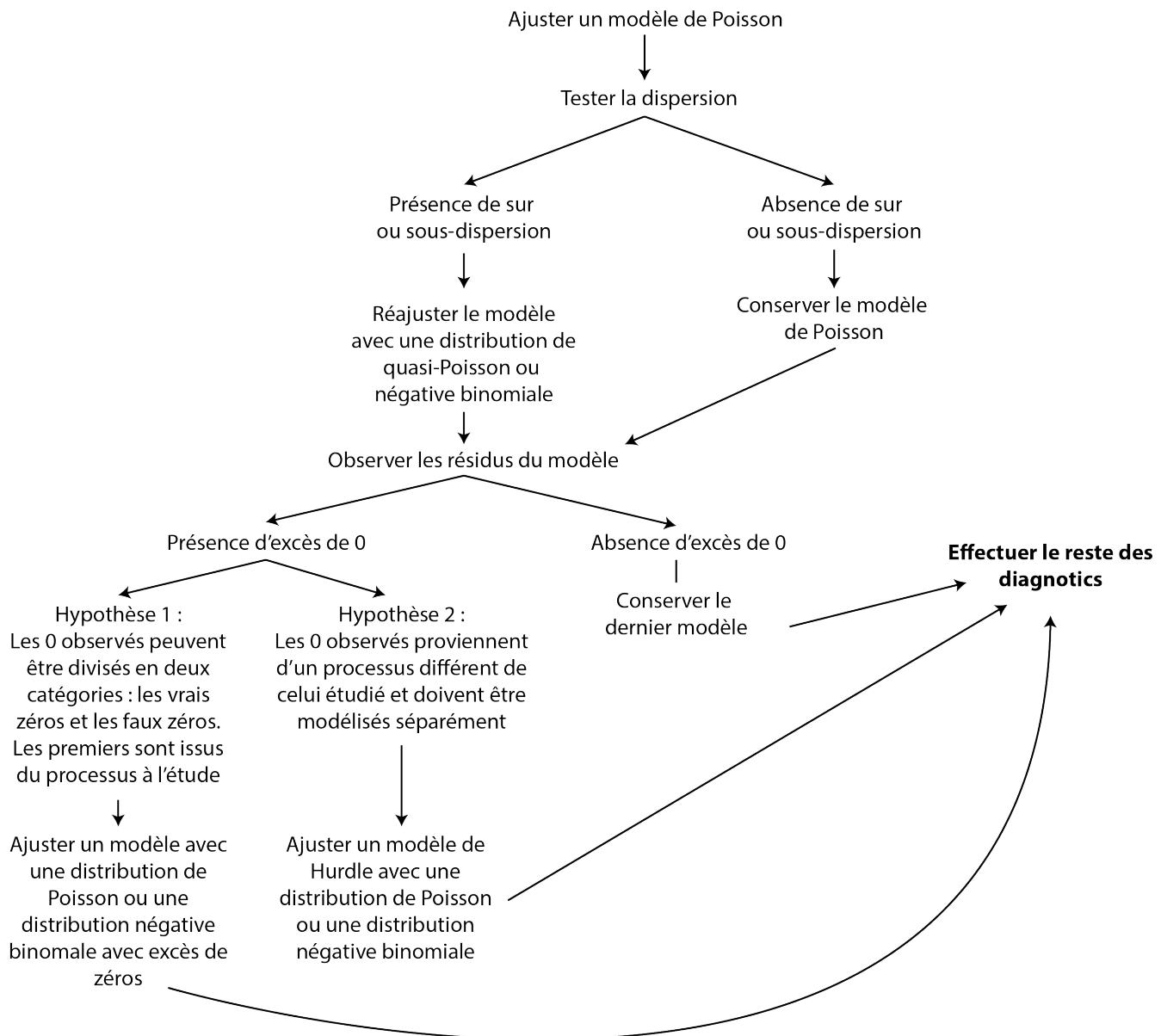


FIG. 8.39 : Processus de sélection d'un modèle pour une variable de comptage



Modèle de Hurdle versus modèle avec excès de zéro

Les modèles de Hurdle sont une autre catégorie de modèles GLM. Ils peuvent être décrits avec la formulation suivante :

$$\begin{cases} Y \sim \text{Binomial}(p) \text{ si } y = 0 \\ \text{logit}(p) = \beta_a X_a \\ Y \sim \text{TrPoisson}(\lambda) \text{ si } y > 0 \\ \log(\lambda) = \beta_b X_b \end{cases} \quad (8.23)$$

Nous constatons qu'un modèle de Hurdle utilise deux distributions, la première est une distribution binomiale dont l'objectif est de prédire si les observations sont à 0 ou au-dessus de 0. La seconde est une distribution strictement positive (supérieure à 0), il peut s'agir d'une distribution tronquée de Poisson, tronquée binomiale négative, Gamma, log-normale ou autre, dépendamment du phénomène modélisé. Puisque le modèle fait appel à deux distributions, deux équations de régression sont utilisées, l'une pour prédire p (la probabilité d'observer une valeur au-dessus de 0) et l'autre l'espérance (moyenne) de la seconde distribution.

En d'autres termes, un modèle de Hurdle modélise les données à zéro et les données au-delà de 0 comme deux processus différents (chacun avec sa propre distribution). Cette approche se distingue des modèles avec excès de zéros qui utilisent une seule distribution pour décrire l'ensemble des données. D'après un modèle avec excès de zéro, il existe de vrais et de faux zéros que l'on tente de distinguer. Dans un modèle de Hurdle, l'idée est que les zéros constituent une limite. Nous modélisons la probabilité de dépasser cette limite et ensuite la magnitude du dépassement de cette limite.

Prenons un exemple pour rendre la distinction plus concrète. Admettons que nous utilisons un capteur capable de mesurer la concentration de particules fines dans l'air. D'après les spécifications du fabricant, le capteur est capable de mesurer des taux de concentration à partir de $0,001 \mu\text{g}/\text{m}^3$. Dans une ville avec des niveaux de concentration très faibles, il est très fréquent que le capteur enregistre des valeurs à zéro. Considérant ce phénomène, il serait judicieux de modéliser le processus avec un modèle de Hurdle Gamma puisque les 0 représentent une limite qui n'a pas été franchie : le seuil de détection du capteur. Nous traitons donc différemment les secteurs au-dessous et au-dessus de ce seuil. Si nous reprenons notre exemple sur les accidents des piétons à des intersections, il est plus judicieux, dans ce cas, de modéliser le phénomène avec un modèle avec excès de zéro puisque nous pouvons observer zéro accident à une intersection dangereuse (vrai zéro) et zéro accident à une intersection sur laquelle aucun piéton ne traverse jamais (faux zéro).

8.4 Modèles GLM pour des variables continues

Comme nous l'avons vu dans la section 2.4, il existe un grand nombre de distributions permettant de décrire une grande diversité de variables continues. Il serait fastidieux de toutes les présenter, nous reviendrons donc seulement sur les plus fréquentes.

8.4.1 Modèle GLM gaussien

Comme nous l'avons vu en introduction, le modèle GLM gaussien est le plus simple puisqu'il correspond à la transposition de la régression linéaire classique (des moindres carrés) dans la forme des modèles généralisés.

TAB. 8.29 : Carte d'identité du modèle gaussien

Type de variable dépendante	Variable continue dans l'intervalle $]-\infty; +\infty[$
Distribution utilisée	Normale
Formulation	$Y \sim Normal(\mu, \sigma)$ $g(\mu) = \beta_0 + \beta X$ $g(x) = x$
Fonction de lien	Identitaire
Paramètre modélisé	μ
Paramètres à estimer	β_0, β et σ
Conditions d'application	Homoscédasticité

8.4.1.1 Conditions d'application

Les conditions d'application sont les mêmes que celles d'une régression linéaire classique. La condition de l'homoscédasticité (homogénéité de la variance) est due au fait que la variance du modèle est contrôlée par un seul paramètre fixe $var(y) = \sigma$ (l'écart-type de la distribution normale). À titre de comparaison, rappelons que dans un modèle de Poisson, la variance est égale à la moyenne ($var(y) = E(y)$) alors que dans un modèle binomial négatif, la variance est fonction de la moyenne et d'un paramètre θ ($var(y) = E(y) + E(y)^{\frac{2}{\theta}}$). Pour ces deux exemples, la variance augmente au fur et à mesure que la moyenne augmente.

8.4.1.2 Interprétation des paramètres

L'interprétation des paramètres est la même que pour une régression linéaire classique :

- β_0 : la constante, soit la moyenne attendue de la variable Y lorsque les valeurs de toutes les variables X sont 0.
- β : les coefficients de régression qui quantifient l'effet d'une augmentation d'une unité des variables X sur la moyenne de la variable Y .
- σ : l'écart-type de Y après avoir contrôlé les variables X . Il peut s'interpréter comme l'incertitude restante après modélisation de la moyenne de Y . Concrètement, si vous utilisez votre équation de régression pour prédire une nouvelle valeur de Y : \hat{Y} , l'intervalle de confiance à 95 % de cette pré-diction est $(\hat{Y} - 3\sigma; \hat{Y} + 3\sigma)$. Vous noterez donc que plus σ est grand, plus grande est l'incertitude de la prédiction.

8.4.1.3 Exemple appliqué dans R

Pour cet exemple, nous reprenons le modèle LM que nous avons présenté dans la section 7.7. À titre de rappel, l'objectif est de modéliser la densité végétale dans les secteurs de recensement de Montréal. Pour cela, nous utilisons des variables relatives aux populations vulnérables physiologiquement ou socioéconomiquement, tout en contrôlant l'effet de la forme urbaine. Parmi ces dernières, l'âge médian des bâtiments est ajouté au modèle avec une polynomiale d'ordre deux, et la densité d'habitants est transformée avec la fonction logarithmique.

Vérification des conditions d'application

La première étape de la vérification des conditions d'application est bien sûr de s'assurer de l'absence de multicolinéarité excessive.

```
# Chargement des données
load("data/lm/DataVegetation.RData")
```

```
# Calcul du VIF
library(car)
vif(glm(VegPct ~ log(HABHA)+poly(AgeMedian,2) +
        Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal))

##          GVIF Df GVIF^(1/(2*Df))
## log(HABHA)    1.289495  1     1.135559
## poly(AgeMedian, 2) 1.387429  2     1.085307
## Pct_014       1.517957  1     1.232054
## Pct_65P       1.304094  1     1.141969
## Pct_MV        1.480275  1     1.216666
## Pct_FR        1.729646  1     1.315160
```

Puisque l'ensemble des valeurs de VIF sont inférieures à deux, nos données ne sont pas marquées par une multicolinéarité problématique. La seconde étape du diagnostic consiste à calculer et à afficher les distances de Cook.

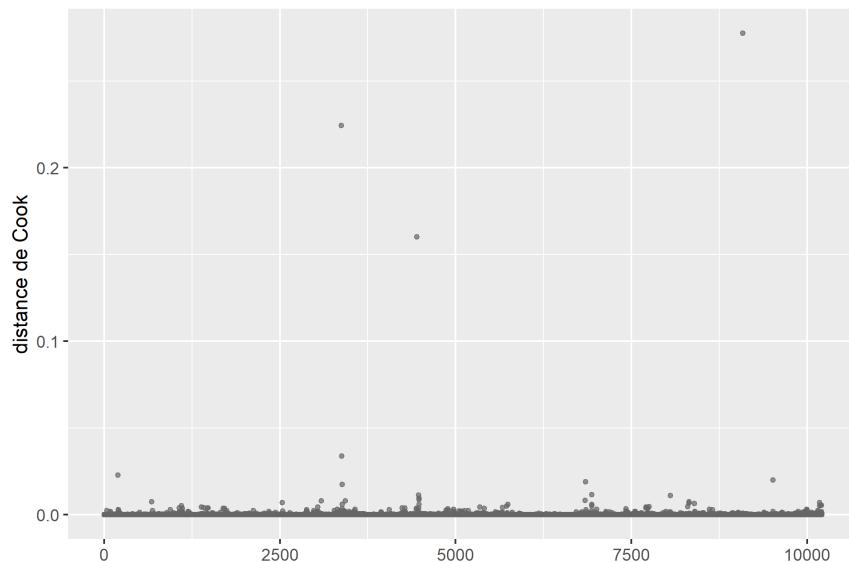


FIG. 8.40 : Distances de Cook pour le modèle gaussien

La figure 8.40 indique clairement que quatre observations sont très influentes dans le modèle.

```
# Sélection des cas étranges
cas_étranges <- subset(DataFinal, cooksd > 0.03)
print(cas_étranges)

##      VegPct ArbPct V250Pct V500Pct A250Pct A500Pct      HABHA AgeMedian Pct_014
## 3374 10.481   5.478  18.987  13.744   2.908   2.704 74.835867    226     4.76
## 3378  0.000   0.000  12.709  12.505   2.116   2.324 88.006946    206     6.25
## 4446 23.162   5.209  31.437  31.535   8.672   9.108 313.142733    206    14.40
## 9088 85.767  27.583  78.195  83.492  42.999  51.074  2.070472    207    12.00
##      Pct_65P Pct_MV Pct_FR DistCBDkm SDRNOM
## 3374    14.29  23.81   14.29     0.748 Montréal
## 3378    12.50  25.00   12.50     0.706 Montréal
```

```
## 4446   16.87  53.50  42.39      8.678 Montréal
## 9088   24.00  12.00  16.00     28.440 Montréal
```

Il s'agit de quatre îlots dans Montréal avec des logements très anciens : plus de 200 ans, alors que la moyenne est de 52 ans pour le reste de la zone d'étude. Le fait que nous ayons dans le modèle une polynomiale d'ordre 2 pour cette variable intensifie l'influence de ces valeurs extrêmes. Par conséquent, nous décidons de simplement les supprimer. Nous verrons plus tard qu'une alternative envisageable est de changer la distribution du modèle pour une distribution de Student (plus robuste aux valeurs extrêmes).

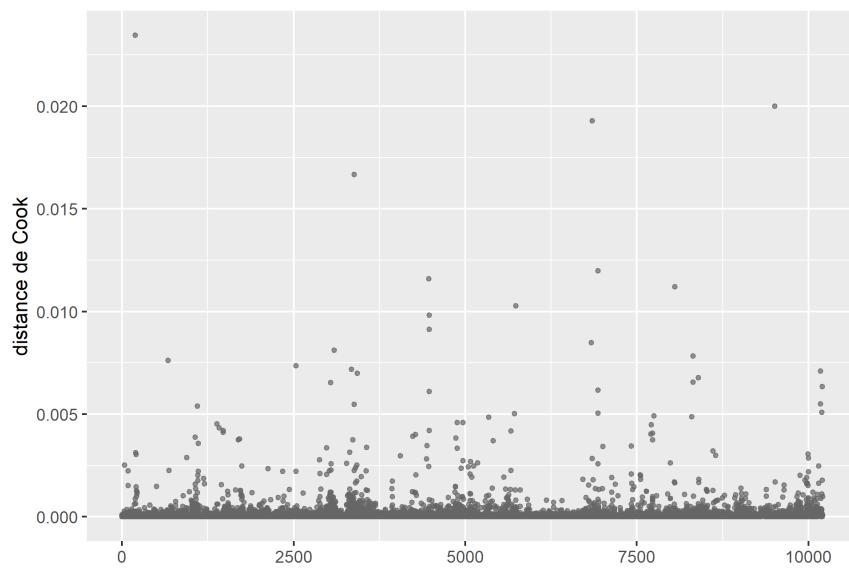


FIG. 8.41 : Distances de Cook pour le modèle gaussien après suppression des observations influentes

Une fois ces observations retirées, les nouvelles distances de Cook (figure 8.41) ne révèlent plus d'observations fortement influentes. Nous pouvons passer à l'analyse des résidus simulés. La figure 8.42 démontre que la distribution des résidus est significativement différente d'une distribution uniforme, que des valeurs aberrantes sont encore présentes et qu'il existe un lien entre résidus et prédiction dans le modèle.

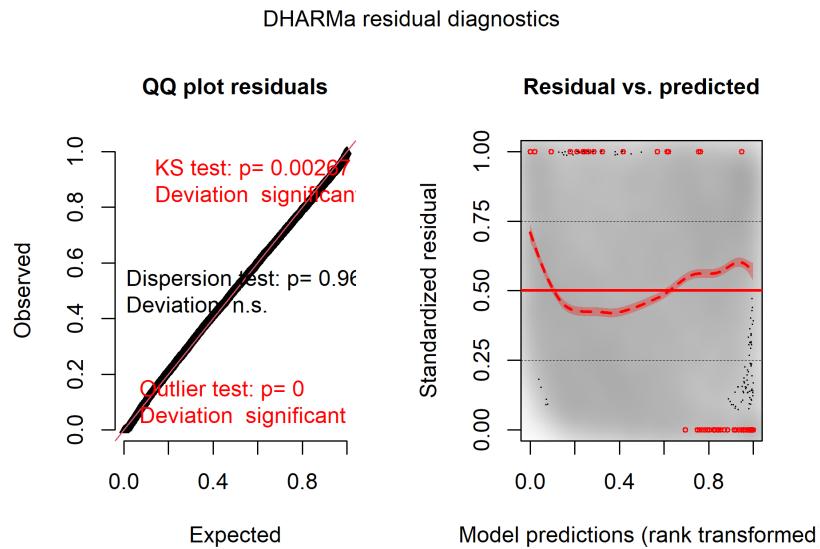


FIG. 8.42 : Diagnostic général des résidus simulés pour le modèle gaussien

Pour mieux cerner ce problème, nous pouvons, dans un premier temps, comparer la distribution originale des données et les simulations issues du modèle. La figure 8.43 montre clairement que la distribution normale est mal ajustée aux données. Ces dernières sont légèrement asymétriques et ne peuvent pas être inférieures à zéro, ce que la distribution normale ne parvient pas à reproduire.

```
df <- reshape2::melt(mat_sims[,1:30])
ggplot() +
  geom_histogram(data = DataFinal2, mapping = aes(x = VegPct, y = ..density..),
                 color = "black", fill = "white", bins = 50) +
  geom_density(data = df, aes(x = value, group = Var2),
               color = rgb(0.4,0.4,0.4,0.4), fill = rgb(0,0,0,0)) +
  labs(x = "Pourcentage de végétation dans l'îlot (%)",
       y = "Densité")
```

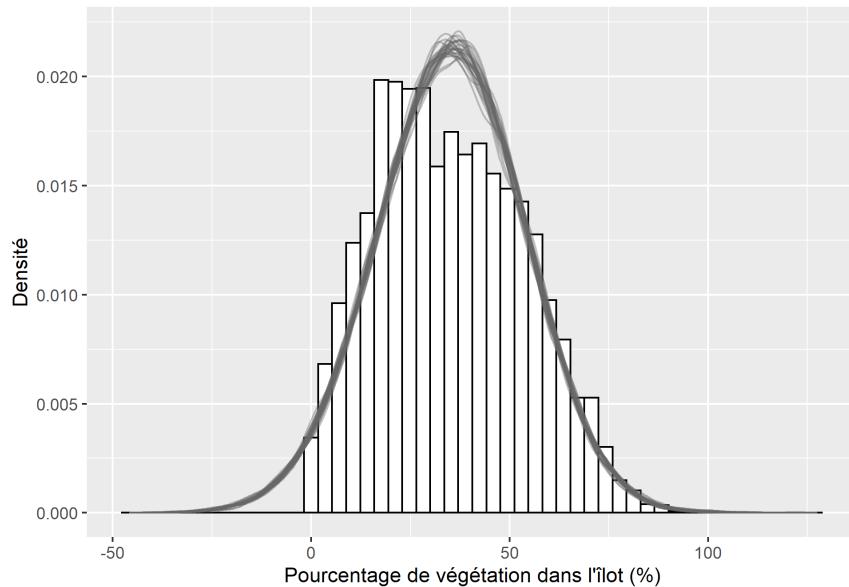


FIG. 8.43 : Comparaison de la distribution originale de la variable et des simulations issues du modèle

Il est également possible de vérifier si la condition d'homogénéité de la variance s'applique bien aux données.

```
# Extraction des prédictions du modèle
mus <- predict(modele, type = "response")
sigma_model <- sigma(modele)
# Création d'un DataFrame pour contenir les prédictions et les vraies valeurs
df1 <- data.frame(
  mus = mus,
  reals = DataFinal2$VegPct
)
# Calcul de l'intervalle de confiance à 95 % selon la distribution normale
# et stockage dans un second DataFrame
seqa <- seq(0,100,10)
df2 <- data.frame(
  mus = seqa,
  lower = qnorm(p = 0.025, mean = seqa, sd = sigma_model),
  upper = qnorm(p = 0.975, mean = seqa, sd = sigma_model)
)
# Affichage des valeurs réelles et prédites (en rouge)
# et de leur variance selon le modèle (en noir)
ggplot() +
  geom_point(data = df1,
    mapping = aes(x = mus, y = reals),
    color ="red", size = 0.5) +
  geom_errorbar(data = df2,
    mapping = aes(x = mus, ymin = lower, ymax = upper),
    width = 0.2, color = rgb(0.4,0.4,0.4)) +
  labs(x = 'valeurs prédites',
    y = "valeurs réelles")
```

À nouveau, nous constatons à la figure 8.44 que le modèle s'attend à trouver des valeurs négatives pour

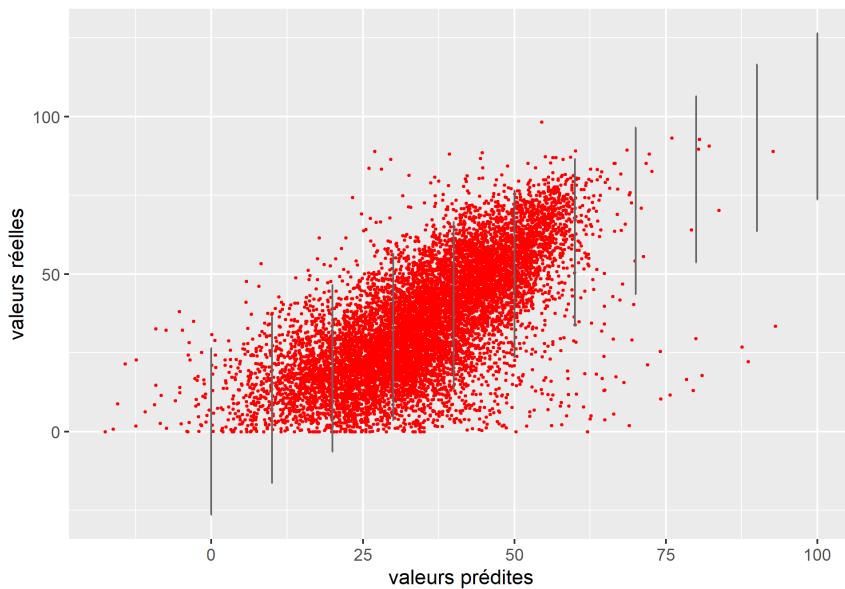


FIG. 8.44 : Comparaison de la distribution originale de la variable et des simulations issues du modèle

la concentration de végétation, ce qui n'est pas possible dans notre cas. En revanche, il semble que la variance soit bien homogène puisque la dispersion des observations semble suivre à peu près la dispersion attendue par le modèle (en noir).

Malgré ces différents constats indiquant clairement qu'un modèle gaussien est un choix sous-optimal pour ces données, nous poursuivons l'analyse de ce modèle.

Vérification de la qualité d'ajustement

```
# Ajustement d'un modèle nul
modelenull <- glm(VegPct ~ 1,
                    data = DataFinal2,
                    family = gaussian())

# Calcul des pseudo R2
rsqs(loglike.full = logLik(modele),
      loglike.null = logLik(modelenull),
      full.deviance = deviance(modele),
      null.deviance = deviance(modelenull),
      nb.params = modele$rank,
      n = nrow(DataFinal2)
    )

## `$deviance expliquee`
## [1] 0.4706321
##
## `$McFadden ajusté'
## 'log Lik.' 0.07310662 (df=9)
##
## `$Cox and Snell'
## 'log Lik.' 0.4706321 (df=9)
##
## `$Nagelkerke
```

```
## 'log Lik.' 0.4707122 (df=9)
```

Le modèle parvient à expliquer 47 % de la déviance totale, mais obtient un R^2 ajusté de McFadden de seulement 0,07.

```
# Calcul du RMSE
sqrt(mean((predict(modele, type = "response") - DataFinal2$VegPct)**2))
```

```
## [1] 13.49885
```

L'erreur quadratique moyenne et de 13,5 points de pourcentage, ce qui indique que le modèle a une assez faible capacité prédictive.

Interprétation des résultats

L'ensemble des coefficients du modèle sont accessibles via la fonction `summary`; le tableau 8.30 présente les résultats pour les coefficients du modèle.

TAB. 8.30 : Résultats du modèle gaussien

Variable	Coeff.	Err.std	Val.z	val.p	IC coeff 2,5 %	IC coeff 97,5 %	Sign.
Constante	53,606	1,000	53,640	0,000	51,647	55,565	***
AgeMedian ordre 1	2,732	15,560	0,180	0,861	-27,772	33,237	
AgeMedian ordre 2	-320,869	14,000	-22,910	0,000	-348,318	-293,420	***
Pct_014	0,915	0,030	29,310	0,000	0,853	0,976	***
Pct_65P	0,280	0,020	15,050	0,000	0,243	0,316	***
Pct_MV	-0,042	0,010	-4,190	0,000	-0,061	-0,022	***
Pct_FR	-0,340	0,010	-30,940	0,000	-0,362	-0,318	***

Les résultats de la régression linéaire multiple ont déjà été interprétés dans la section 7.7.1, nous ne commenterons pas ici les résultats du modèle GLM gaussien qui sont identiques.

8.4.2 Modèle GLM avec une distribution de Student

Pour rappel, la distribution de Student ressemble à une distribution normale (section 2.4.3.11). Elle est symétrique autour de sa moyenne et a également une forme de cloche. Cependant, elle dispose de queues lourdes, ce qui signifie qu'elle permet de représenter des phénomènes présentant davantage de valeurs extrêmes qu'une distribution normale. Pour contrôler le poids des queues, la distribution de Student intègre un troisième paramètre : ν (nu). Lorsque ν tends vers l'infini, la distribution de Student tend vers une distribution normale (figure 8.45).

Comme vous pouvez le constater dans la carte d'identité au tableau 8.31, le modèle GLM de Student est très proche du modèle GLM gaussien. Nous modélisons explicitement la moyenne de la distribution et son paramètre de dispersion (variance) est laissé fixe. Ce GLM est même souvent utilisé comme une version « robuste » du modèle gaussien du fait de sa capacité à intégrer explicitement l'effet des observations extrêmes. En effet, dans un modèle gaussien, les observations extrêmes (aussi appelées observations aberrantes) vont davantage influencer les paramètres du modèle que pour un modèle utilisant une distribution de Student.

8.4.2.1 Conditions d'application

Les conditions d'application sont les mêmes que pour un modèle GLM gaussien, à ceci près que le modèle utilisant la distribution de Student est moins sensible aux observations extrêmes.

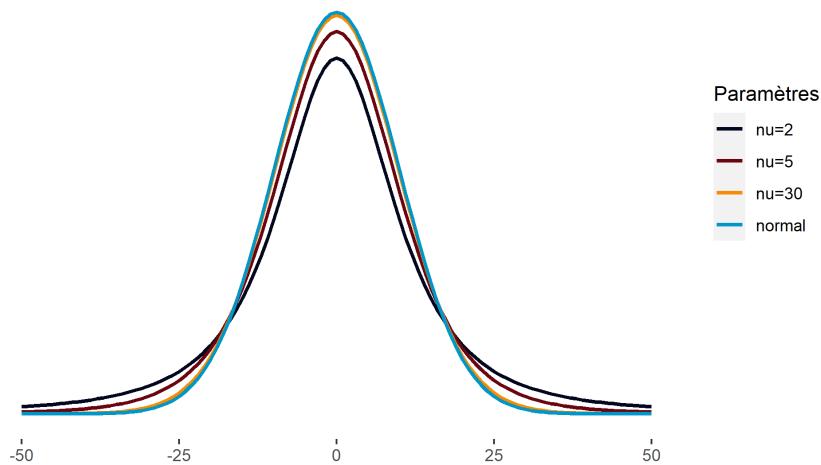


FIG. 8.45 : Effet du paramètre ν sur une distribution de Student

TAB. 8.31 : Carte d'identité du modèle de Student

Type de variable dépendante	Variable continue dans l'intervalle $]-\infty; +\infty[$
Distribution utilisée	Student
Formulation	$Y \sim Student(\mu, \sigma, \nu)$ $g(\mu) = \beta_0 + \beta X$ $g(x) = x$
Fonction de lien	Identitaire
Paramètre modélisé	μ
Paramètres à estimer	β_0, β, σ et ν
Conditions d'application	Homoscédasticité

8.4.2.2 Interprétation des paramètres

L'interprétation des paramètres est la même que pour un modèle gaussien puisque nous modélisons la moyenne de la distribution et que la fonction de lien est la fonction identitaire. Le seul paramètre supplémentaire est ν , qui n'a en soi aucune interprétation pratique. Notez simplement que si ν est supérieur à 30, un simple modèle GLM gaussien serait sûrement suffisant.

8.4.2.3 Exemple appliqué dans R

Nous proposons ici de simplement réajuster le modèle gaussien présenté dans la section précédente en utilisant une distribution de Student. Nous utilisons pour cela la fonction `gam` du package `mgcv` avec le paramètre `family=scat` pour utiliser une distribution de Student. Les valeurs de VIF ont déjà été calculées dans l'exemple précédent, nous pouvons donc passer directement au calcul des distances de Cook.

```
# Chargement des données
load("data/lm/DataVegetation.RData")
# Ajustement du modèle
modele <- gam(VegPct ~ log(HABA)+poly(AgeMedian,2)+
               Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal,
               family = scat)
```

```
# Calcul des distances de Cook
cooksd <- cooks.distance(modele)
# Affichage des valeurs
df <- data.frame(
  cook = cooksd,
  oid = 1:length(cooksd)
)
ggplot(data = df) +
  geom_point(aes(x = oid, y = cook), color = rgb(0.4,0.4,0.4,0.7), size = 1) +
  labs(x="", y = "distance de Cook")
```

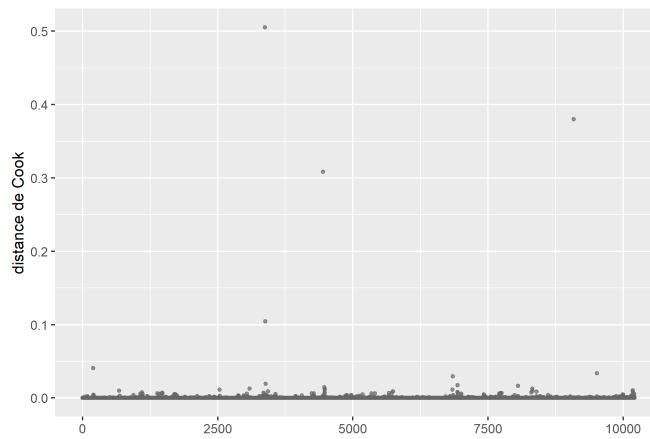


FIG. 8.46 : Distances de Cook pour un modèle GLM avec une distribution de Student

Nous retrouvons les quatre observations avec des distances de Cook très fortes que nous avons identifiées dans le modèle gaussien. Nous décidons donc de les enlever pour les mêmes raisons que précédemment.

```
# Chargement des données
DataFinal2 <- subset(DataFinal, cooksd<0.1)
# Ajustement du modèle
modele <- gam(VegPct ~ log(HABHA)+poly(AgeMedian,2)+Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal2, family = scat)

# Calcul des distances de Cook
cooksd <- cooks.distance(modele)
# Affichage des valeurs
df <- data.frame(
  cook = cooksd,
  oid = 1:length(cooksd)
)
ggplot(data = df) +
  geom_point(aes(x = oid, y = cook), color = rgb(0.4,0.4,0.4,0.7), size = 1) +
  labs(x="", y = "distance de Cook")
```

Nous pouvons à présent vérifier si les résidus simulés se comportent tel qu'attendu.

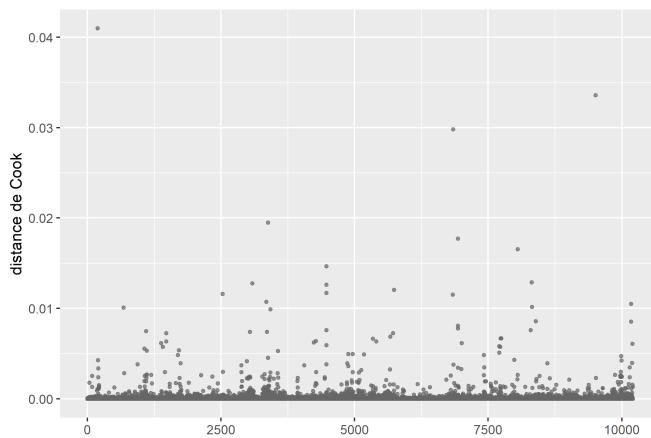


FIG. 8.47 : Distances de Cook pour un modèle GLM avec une distribution de Student après suppression des valeurs fortement influentes

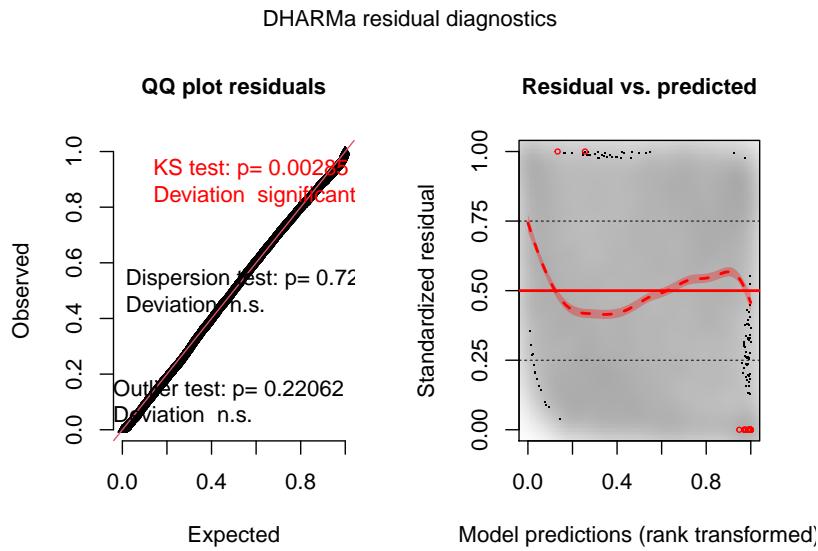


FIG. 8.48 : Diagnostic général des résidus simulés pour le GLM avec distribution de Student

Il semble que nous obtenons des résultats similaires à ceux du modèle gaussien : les résidus divergent significativement d'une distribution uniforme (figure 8.48). Le graphique quantile-quantile n'est parfois pas très adapté pour discerner une déviation de la distribution uniforme, nous pouvons dans ce cas afficher un histogramme des résidus pour en avoir le cœur net (figure 8.49).

```
ggplot() +
  geom_histogram(aes(x = residuals(sim_res)), bins = 50, color = "white") +
  labs(x = "effectifs", y = "résidus simulés")
```

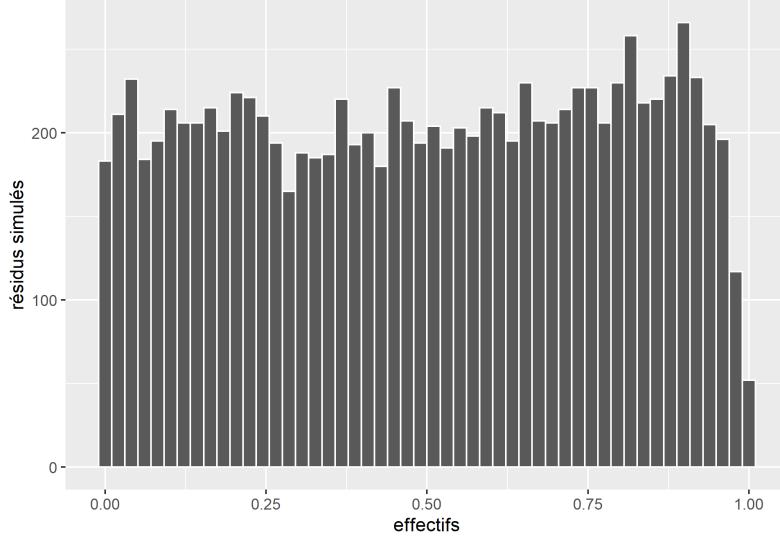


FIG. 8.49 : Distribution des résidus simulés du modèle GLM avec distribution de Student

Pour cet exercice, il est intéressant de comparer les formes des simulations issues du modèle gaussien et du modèle de Student pour bien distinguer la différence entre les deux.

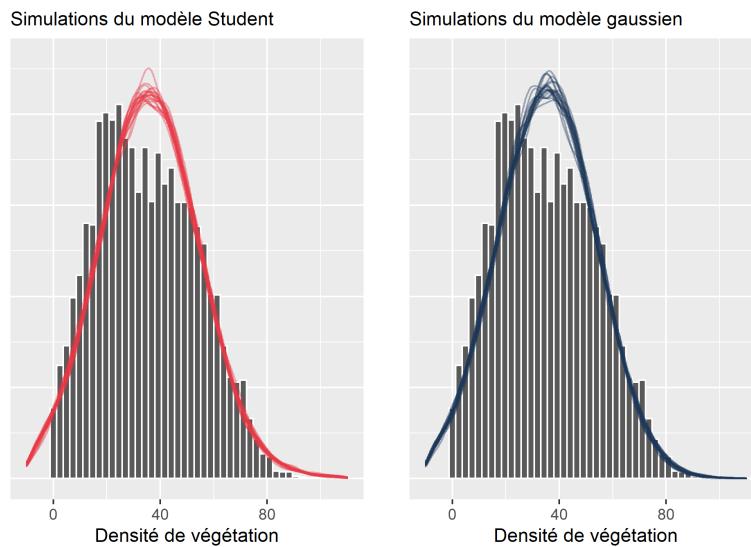


FIG. 8.50 : Simulations issues des modèles gaussien et de Student, comparées aux données originales

Nous constatons ainsi que la différence entre les deux modèles est ici très mince, voire inexistante. Le seul élément que nous pouvons noter est que le modèle de Student à une courbe (une queue de distribution) moins aplatie vers la droite. Cela lui permettrait de mieux tenir compte de cas extrêmes avec de fortes densités de végétation (ce qui concerne donc très peu d'observations puisque cette variable a un maximum de 100).

Pour déterminer si le modèle de Student est plus pertinent à retenir que le modèle gaussien, nous pouvons ajuster un second modèle de Student pour lequel nous forçons artificiellement ν à être très élevé. Pour rappel, quand ν tend vers l'infini, la distribution de Student tend vers une distribution normale. Nous forçons ici ν à être supérieur à 100 pour créer un second modèle de Student se comportant quasiment comme un modèle gaussien et calculons les AIC des deux modèles.

```
# Calcul d'un modèle de Student identitique à un modèle gaussien
modele2 <- gam(VegPct ~ log(HABHA)+poly(AgeMedian,2)+  
                  Pct_014+Pct_65P+Pct_MV+Pct_FR, data = DataFinal2,  
                  family = scat(min.df = 100))  
# Calcul des deux AIC  
AIC(modele)
```

```
## [1] 81771.92
```

```
AIC(modele2)
```

```
## [1] 82057.79
```

Le second AIC (modèle gaussien) est plus élevé, indiquant que le modèle est moins bien ajusté aux données. Dans le cas présent, il est plus pertinent de retenir le modèle de Student même si les écarts entre ces deux modèles sont minimes. Ce résultat n'est pas surprenant puisque la variable Y (pourcentage de végétation dans les îlots de l'île de Montréal) est relativement compacte et comporte peu / pas de valeurs pouvant être qualifiées de valeurs extrêmes.

Nous ne détaillons pas ici l'interprétation des coefficients du modèle (présentés au tableau 8.32) puisqu'ils s'interprètent de la même façon qu'un modèle GLM et qu'un modèle de régression linéaire multiple.

TAB. 8.32 : Résultats du modèle Student

Variable	Coeff.	Err.std	Val.z	val.p	IC coeff 2,5 %	IC coeff 97,5 %	Sign.
Constante	65,096	0,940	69,110	0,000	63,250	66,943	***
log(HABHA)	-9,502	0,160	-60,160	0,000	-9,811	-9,192	***
Pct_014	0,866	0,030	29,450	0,000	0,808	0,924	***
Pct_65P	0,237	0,020	13,540	0,000	0,203	0,272	***
Pct_MV	-0,015	0,010	-1,650	0,099	-0,034	0,003	.
Pct_FR	-0,301	0,010	-29,040	0,000	-0,321	-0,280	***

8.4.3 Modèle GLM avec distribution Gamma

Pour rappel, la distribution Gamma est strictement positive ($[0; +\infty[$), asymétrique, et a une variance proportionnelle à sa moyenne (hétéroscedastique). Dans la section sur les distributions, nous avons vu que la distribution Gamma (section 2.4.3.15) est formulée avec deux paramètres : sa forme (α ou *shape*) et son échelle (b ou *scale*). Ces deux paramètres n'ont pas une interprétation intuitive, mais il est possible avec un peu de jonglage mathématique d'arriver à une reparamétrisation intéressante. Cela est détaillé dans l'encadré ci-dessous ; notez toutefois qu'il n'est pas nécessaire de maîtriser le contenu de cet encadré pour lire la suite de cette section sur les modèles GLM avec une distribution Gamma.



Reparamétrisation d'une distribution Gamma pour un GLM

Si nous disposons d'une variable Y , suivant une distribution Gamma telle que $Y \sim \text{Gamma}(\alpha, b)$ avec α le paramètre de forme et b le paramètre d'échelle, alors, l'espérance et la variance de Y peuvent être définies comme suit :

$$\begin{aligned} E(Y) &= \alpha \times b \\ \text{Var}(Y) &= \alpha \times b^2 \end{aligned} \tag{8.24}$$

En d'autres termes, l'espérance (l'équivalent de la moyenne pour une distribution normale) de notre variable Y est égale au produit des paramètres de forme et d'échelle.

Avec ces propriétés, il est possible de redéfinir la fonction de densité de la distribution Gamma et d'arriver à une nouvelle formulation : $Y \sim \text{Gamma}(\mu, \alpha)$. μ est donc l'espérance de Y (interprétable comme sa moyenne, soit sa valeur attendue) et α permet de capturer la dispersion de la distribution Gamma. Par extension des relations présentées ci-dessus, il est possible de reformuler la variance en fonction de μ et de α .

$$\begin{aligned} \text{Var}(Y) &= \alpha \times b^2 \\ \mu &= \alpha \times b \text{ soit } b = \frac{\mu}{\alpha} \\ \text{Var}(Y) &= \alpha \times \left(\frac{\mu}{\alpha}\right)^2 \text{ soit } \text{Var}(Y) = \frac{\mu^2}{\alpha} \end{aligned} \tag{8.25}$$

Nous observons donc que la variance dans un modèle Gamma augmente de façon quadratique avec la moyenne, mais est tempérée par le paramètre de forme. Nous en concluons qu'un paramètre de forme plus grand produit une distribution moins étalée.

Dans ce contexte, μ doit être strictement positif : la valeur attendue moyenne d'une distribution Gamma doit être positive par définition puisqu'une distribution Gamma ne peut pas produire de valeurs négatives. Il est donc logique d'utiliser la fonction logarithmique comme fonction de lien, puisque sa contrepartie (la fonction exponentielle) ne produit que des résultats positifs.

Pour résumer, nous nous retrouvons donc avec un modèle qui prédit, sur une échelle logarithmique, l'espérance (~moyenne) d'une distribution Gamma. Notez qu'il existe d'autres façons de spécifier un modèle GLM avec une distribution Gamma, mais celle-ci est la plus intuitive.

8.4.3.1 Interprétation des paramètres

Puisque le modèle utilise la fonction de lien \log , alors les coefficients β expriment l'augmentation de l'espérance (la valeur attendue de Y , ce qui est proche de l'idée de moyenne) de la variable Y sur une échelle logarithmique (comme dans un modèle de Poisson). Il est possible de convertir les coefficients dans l'échelle originale de la variable Y en utilisant la fonction exponentielle (l'inverse de la fonction \log), mais ces coefficients représentent alors des effets multiplicatifs et non des effets additifs. Prenons un exemple, admettons que le coefficient β_1 , associé à la variable X_1 soit de 1,5. Cela signifie qu'une augmentation d'une unité de X_1 , augmente le log de l'espérance de Y de 1,5 unité. L'exponentielle du coefficient est 4,48, ce qui signifie qu'une augmentation d'une unité entraîne une multiplication par 4,48 de la valeur attendue de Y (l'espérance de Y). Le paramètre de forme (α) n'a pas d'interprétation pratique, bien qu'il soit utilisé dans les différents tests des conditions d'application du modèle et dans le calcul de sa déviance.

8.4.3.2 Conditions d'application

Dans un modèle GLM gaussien, la variance est capturée par un paramètre σ et est constante, produisant la condition d'homoscédasticité des résidus. Dans un modèle Gamma, la variance varie en fonction de l'espérance et du paramètre de forme selon la relation : $Var(Y) = \frac{E(Y)^2}{\alpha}$. Les résidus sont donc par nature hétéroscédistiques dans un modèle Gamma et doivent suivre cette relation.

8.4.3.3 Exemple appliqué dans R

Pour cet exemple, nous nous intéressons à la durée de déplacement en milieu urbain. Ce type d'analyse permet notamment de mieux comprendre les habitudes de déplacement de la population et d'orienter les politiques de transport. Plusieurs travaux concluent que les durées de déplacement en milieu urbain varient en fonction du motif du déplacement, du mode de transport utilisé, des caractéristiques socio-économiques de l'individu et des caractéristiques du trajet lui-même (Anastasopoulos et al. 2012; Frank et al. 2008). Nous modélisons ici la durée en minute d'un ensemble de déplacements effectués par des Montréalais en 2017 et enregistrés avec l'application MTL Trajet proposée par la Ville de Montréal. Ces données sont disponibles sur le site web des données ouvertes de Montréal⁵ et son anonymisées. Nous ne disposons donc d'aucune information individuelle. Compte tenu du très grand nombre d'observations (plus de 185 000), nous avons dû effectuer quelques opérations de tri et nous avons ainsi supprimé :

⁵<http://donnees.ville.montreal.qc.ca/dataset/mlt-trajet>

TAB. 8.33 : Carte d'identité du modèle Gamma

Type de variable dépendante	Variable continue dans l'intervalle $]0; +\infty[$
Distribution utilisée	Gamma
Formulation	$Y \sim Gamma(\mu, \alpha)$ $g(\mu) = \beta_0 + \beta X$ $g(x) = \log(x)$
Fonction de lien	\log
Paramètre modélisé	μ
Paramètres à estimer	β_0, β , et α
Conditions d'application	$Variance = \frac{\mu^2}{\alpha}$

- les trajets utilisant de multiples modes de transport (sauf en combinaison avec la marche à pied, par exemple, un trajet effectué à pied et en transport en commun a été recatégorisé comme un trajet en transport en commun uniquement). Les déplacements multimodaux se distinguent largement des déplacements unimodaux dans la littérature scientifique;
- les trajets de nuit (seuls les trajets démarrant dans l'intervalle de 7 h à 21 h ont été conservés);
- les trajets dont le point de départ est un arrondissement / municipalité pour lequel moins de 150 trajets ont été enregistrés (trop peu d'observations);
- les trajets de plus de deux heures (cas rares, considérés comme des données aberrantes);
- les trajets dont le point de départ est à moins de 100 mètres du point d'arrivée (formant des boucles plutôt que des déplacements).

Nous arrivons ainsi à un total de 24 969 observations. Pour modéliser ces durées de déplacement, nous utilisons les variables indépendantes présentées dans le tableau 8.34.

Les temps de trajet forment une variable strictement positive et très vraisemblablement asymétrique. En effet, nous nous attendons à observer une certaine concentration de valeurs autour d'une moyenne, et davantage de trajets avec de courtes durées que de trajets avec de longues durées. Pour nous en assurer, réalisons un histogramme de la distribution de notre variable Y et comparons la avec des distributions normale et Gamma.

```
# Chargement des données
dataset <- read.csv("data/glm/DureeTrajets.csv", stringsAsFactors = F)
arrondMTL <- c("Mercier-Hochelaga-Maisonneuve", "Villeray-Saint-Michel-Parc-Extension",
               "Ville-Marie", "Verdun", "Saint-Léonard", "Saint-Laurent",
               "Rosemont-La Petite-Patrie", "Rivière-des-Prairies-Pointe-aux-Trembles",
               "Pierrefonds-Roxboro", "Outremont", "Montreal-Nord", "Le Sud-Ouest",
               "Le Plateau-Mont-Royal", "Lachine" , "Ahuntsic-Cartierville",
               "Anjou" , "Cote-des-Neiges-Notre-Dame-de-Grace", "LaSalle")
)
dataset <- subset(dataset, dataset$ArrondDep %in% arrondMTL)
# Définissons 7 h du matin comme la référence pour la variable Heure de départ
dataset$HeureDep <- relevel(
  factor(dataset$HeureDep, levels = as.character(7:21)),
  ref = "7")
# Comparaison de la distribution originale avec une distribution
# normale et une distribution Gamma
```

TAB. 8.34 : Variables indépendantes utilisées dans le modèle Gamma

Nom de la variable	Signification	Type de variable	Mesure
Mode	Mode de déplacement	Variable catégorielle	Transport collectif; piéton; vélo et véhicule individuel
Motif	Motif du déplacement	Variable catégorielle	Travail; loisir; magasinage et éducation
HeureDep	Heure de départ	Variable catégorielle	De 7 h à 21 h
ArrondDep	Arrondissement de départ	Variable catégorielle	Nom de l'arrondissement dont part le trajet
LogDist	Logarithme de la distance à vol d'oiseau en km	Variable continue	Logarithme de la distance à vol d'oiseau en km entre le point de départ et d'arrivée
MemeArrond	L'arrivée du trajet se situe-t-elle dans le même arrondissement que celui du départ?	Variable binaire	Oui ou non
Semaine	Le trajet a-t-il été effectué en semaine ou en fin de semaine?	Variable binaire	Semaine ou fin de semaine

```

library(fitdistrplus)
model_gamma <- fitdist(dataset$Duree, distr = "gamma")
ggplot(data = dataset) +
  geom_histogram(aes(x=Duree, y = ..density..), bins = 40, color = "white")+
  stat_function(fun = dgamma, color = 'red', size = 0.8,
                args = as.list(model_gamma$estimate))+
  stat_function(fun = dnorm, color = 'blue', size = 0.8,
                args = list(mean = mean(dataset$Duree),
                            sd = sd(dataset$Duree)))+
  labs(x = 'Temps de déplacement (minutes)',
       y = '',
       subtitle = "modèles Gamma et gaussien")

```

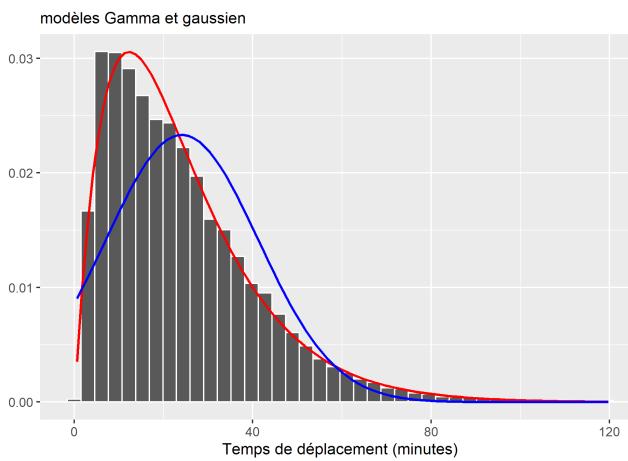


FIG. 8.51 : Distribution des temps de trajet diurne à Montréal

La figure 8.51 permet de constater l'asymétrie de la distribution des temps de trajet et qu'un modèle Gamma (ligne rouge) a plus de chance d'être adapté aux données qu'un modèle gaussien (ligne bleue).

Vérification des conditions d'application

Comme pour les modèles précédents, nous commençons par la vérification de l'absence de multicolinéarité.

```

## Calcul du VIF
vif(glm(Duree ~ Mode + Motif + HeureDep + LogDist +
         ArrondDep + MemeArrond + Jour,
         data = dataset,
         family = Gamma(link="log")))

```

	GVIF	Df	GVIF ^{1/(2*Df)}
## Mode	2.103392	3	1.131931
## Motif	1.934997	3	1.116298
## HeureDep	1.791009	14	1.021032
## LogDist	2.665998	1	1.632789
## ArrondDep	1.439499	17	1.010772
## MemeArrond	2.151113	1	1.466667
## Jour	1.330091	6	1.024055

L'ensemble des valeurs de VIF sont inférieures à trois, indiquant donc l'absence de multicolinéarité excessive. Nous pouvons donc ajuster une première version du modèle (ici avec le package VGAM et la fonction `vglm`) et calculer les distances de Cook.

```
# Calcul du modèle avec VGAM
modele <- vglm(Duree ~ Mode + Motif + HeureDep + LogDist +
                 ArrondDep+ MemeArrond + Semaine,
                 data = dataset,
                 family=gamma2(lmu = "loglink"))

# Calcul des distances de Cook
hats <- hatvalueslm(modele)[,1]
res <- residuals(modele,type = "pearson")[,1]
disp <- modele$coefficients[[2]]**-1
nbparams <- modele$rank
cooksd <- (res/(1 - hats))^2 * hats/(disp * nbparams)
df <- data.frame(
  cook = cooksd,
  oid = 1:length(cooksd)
)
# Représentation des distances de Cook
ggplot(data = df) +
  geom_point(aes(x = oid, y = cook), size = 0.5, color = rgb(0.4,0.4,0.4,0.4)) +
  geom_hline(yintercept = 0.003, color = "red") +
  labs(x = "", y = "distance de Cook")
```

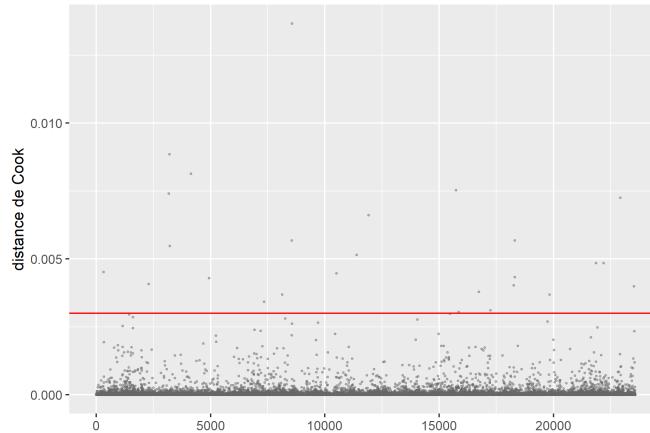


FIG. 8.52 : Distances de Cook pour le modèle Gamma

Puisque nous disposons d'un (très) grand nombre d'observations, nous pouvons nous permettre de retirer les quelques observations fortement influentes (distance de Cook > 0,003 dans notre cas) qui apparaissent dans la figure 8.52. Nous retirons ainsi 28 observations et réajustons le modèle.

```
# Retirer les valeurs influentes
dataset2 <- subset(dataset, cooksd<0.003)
# Calcul du modèle avec VGAM
modele <- vglm(Duree ~ Mode + Motif + HeureDep + LogDist +
                 ArrondDep+ MemeArrond + Semaine,
                 data = dataset2,
                 family=gamma2(lmu = "loglink"))
```

Nous constatons ainsi que dans la nouvelle version du modèle (figure 8.53), aucune valeur particulièrement influente ne semble être présente.

```
# Calcul des distances de Cook
hats <- hatvalues(lm(modele)[,1]
res <- residuals(modele,type = "pearson")[,1]
disp <- modele$coefficients[[2]]**-1
nbparams <- modele$rank
cooksdl <- (res/(1 - hats))^2 * hats/(disp * nbparams)
df <- data.frame(
  cook = cooksdl,
  oid = 1:length(cooksdl)
)
# Représentation des distances de Cook
ggplot(data = df) +
  geom_point(aes(x = oid, y = cook), size = 0.5, color = rgb(0.4,0.4,0.4)) +
  labs(x = "", y = "distance de Cook")
```

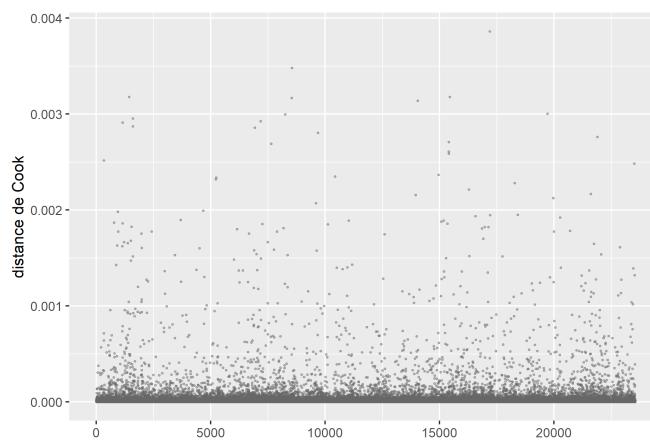


FIG. 8.53 : Distances de Cook pour le modèle Gamma (sans les observations fortement influentes)

```
# Extraction des prédictions du modèle (mu)
mus <- modele$fitted.values
# Extraction du paramètre de forme
shape <- exp(modele$coefficients[[2]])
# Calcul des simulations
nsim <- 1000
cols <- lapply(1:length(mus), function(i){
  mu <- mus[[i]]
  sims <- rgamma(n = nsim, shape = shape, scale = mu/shape)
  return(sims)
})
mat_sims <- do.call(rbind, cols)
# Représentation graphique de 20 simulations
df2 <- reshape2::melt(mat_sims[,0:20])
ggplot() +
  geom_histogram(aes(x = Duree, y = ..density..),
                 data = dataset, bins = 100, color = "black", fill = "white") +
  geom_density(aes(x = value, y=..density.., group = Var2), data = df2,
               fill = rgb(0,0,0,0), color = rgb(0.9,0.22,0.27,0.4), size = 1)+
```

```
xlim(0,200) +
  labs(X="durée (minutes)", y="densité")
```

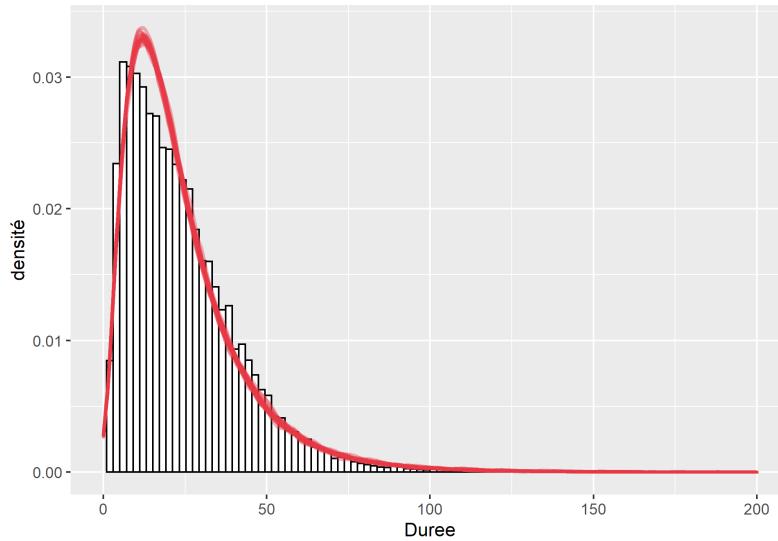


FIG. 8.54 : Comparaison de la distribution originale et de simulations issues du modèle Gamma

Avant de calculer les résidus simulés, nous comparons la distribution originale des données et des simulations issues du modèle. La figure 8.54 permet de constater que le modèle semble bien capturer l’essentiel de la forme de la variable Y originale. Nous notons un léger décalage entre la pointe des deux distributions, laissant penser que les valeurs prédites par le modèle tendent à être légèrement plus grandes que les valeurs réelles. Pour mieux appréhender ce constat, nous passons à l’analyse des résidus simulés.

```
# DHARMA tests
sim_res <- createDHARMA(simulatedResponse = mat_sims,
                           observedResponse = dataset2$Duree,
                           fittedPredictedResponse = modele@fitted.values[,1],
                           integerResponse = F)

ggplot() +
  geom_histogram(aes(x = residuals(sim_res)), bins = 100, color = "white") +
  labs(x = "résidus simulés",
       y = "effectifs")
```

Nul besoin d’un test statistique pour constater que ces résidus (figure 8.55) ne suivent pas une distribution uniforme. Nous observons une nette surreprésentation de résidus à 1 et une nette sous-représentation de résidus à 0. Il y a donc de nombreuses observations dans notre modèle pour lesquelles les simulations sont systématiquement trop fortes et il n’y en a pas assez pour lesquelles les simulations seraient systématiquement trop faibles.

```
plot(sim_res)
```

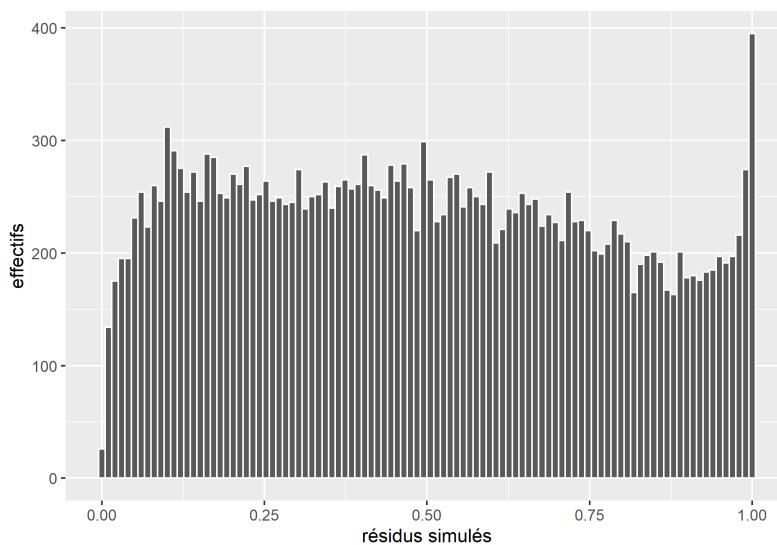


FIG. 8.55 : Distribution des résidus simulés du modèle Gamma

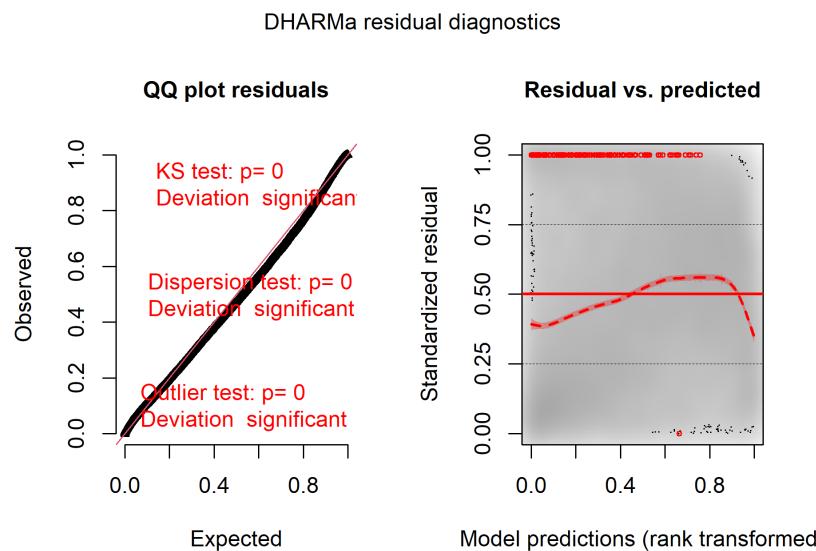


FIG. 8.56 : Diagnostic général des résidus simulés du modèle Gamma

La figure 8.56 indique que le modèle souffre à la fois d'un problème de dispersion (la relation espérance-variance n'est donc pas respectée) et est affecté par des valeurs aberrantes. Considérant que nous avons encore un très grand nombre d'observations, nous faisons le choix de retirer celles pour lesquelles la méthode des résidus simulés estime qu'elles sont des valeurs aberrantes dans au moins 1 % des simulations, soit environ 620 observations.

```
# Sélection des valeurs aberrantes au seuil 0.01
sim_outliers <- outliers(sim_res,
                           lowerQuantile = 0.01,
                           upperQuantile = 0.99,
                           return = "logical")
```

```



```

La figure 8.57 indique que les résidus simulés ne suivent toujours pas une distribution uniforme et qu'il existe une relation prononcée (panneau de droite) entre les résidus et les valeurs prédictes. Cette dernière laisse penser que des variables indépendantes importantes ont été omises dans le modèle, ce qui n'est pas surprenant compte tenu du fait que nous ne disposons d'aucune donnée socioéconomique sur les individus ayant réalisé les trajets. Nos données sont également potentiellement affectées par la présence de dépendance spatiale.

Nous pouvons comparer graphiquement la variance observée dans les données et la variance attendue par le modèle. La figure 8.58 montre clairement que la variance des données tend à être plus grande qu'attendue quand les temps de trajet sont courts, mais diminue trop vite quand les temps de trajet augmentent. D'autres distributions pourraient être envisagées pour ajuster notre modèle : Lognormal, Weibull, etc.

```

# Extraction des prédictions du modèle
mus <- predict(modele, type = "response")[,1]

```

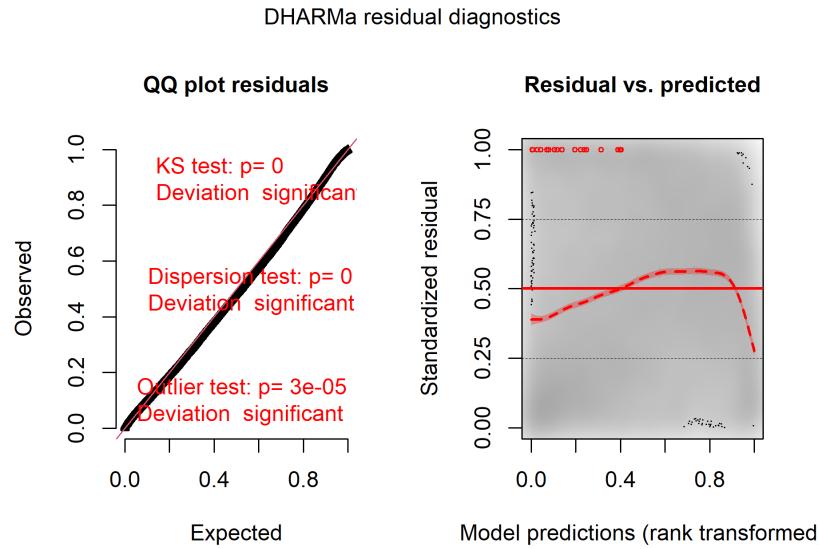


FIG. 8.57 : Diagnostic général des résidus simulés du modèle Gamma (après suppression d'environ 620 valeurs aberrantes)

```
# Création d'un DataFrame pour contenir la prédiction et les vraies valeurs
df1 <- data.frame(
  mus = mus,
  reals = dataset3$Duree
)
# Calcul de l'intervalle de confiance à 95 % selon la distribution Gamma
# et stockage dans un second DataFrame
seqa <- seq(10,120,10)
shape <- exp(modele@coefficients[[2]])
df2 <- data.frame(
  mus = seqa,
  lower = qgamma(p = 0.025, shape = shape, scale = seqa/shape),
  upper = qgamma(p = 0.975, shape = shape, scale = seqa/shape)
)
# Affichage des valeurs réelles et prédites (en rouge)
# et de leur variance selon le modèle (en noir)
ggplot() +
  geom_point(data = df1,
    mapping = aes(x = mus, y = reals),
    color =rgb(0.9,0.22,0.27,0.4), size = 0.5) +
  geom_errorbar(data = df2,
    mapping = aes(x = mus, ymin = lower, ymax = upper),
    width = 0.2, color = rgb(0.4,0.4,0.4)) +
  labs(x = 'valeurs prédites',
    y = "valeurs réelles")
```

À ce stade, nous disposons de suffisamment d'éléments pour douter des résultats du modèle. Nous poursuivons tout de même notre analyse afin d'illustrer l'estimation de la qualité d'ajustement d'un tel modèle et son interprétation.

Analyse de la qualité d'ajustement

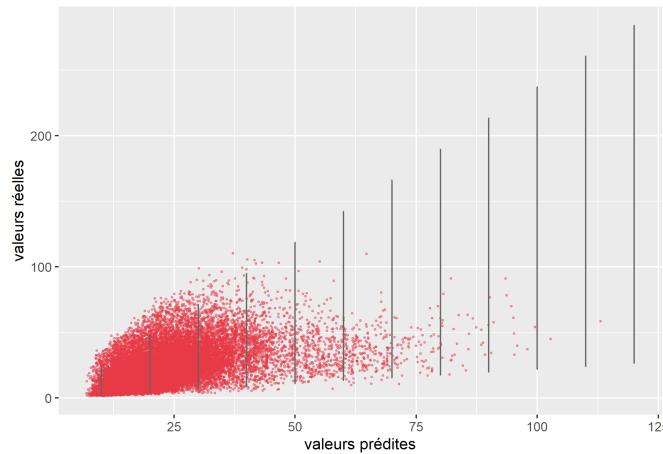


FIG. 8.58 : Comparaison de la variance attendue par le modèle et la variance observée dans les données pour le modèle Gamma

```

# Ajustement d'un modèle nul
modele.null <- vglm(Duree ~1,
                      data = dataset3,
                      model = T,
                      family=gamma2)

# Calcul des pseudo R2
rsqs(loglike.full = logLik(modele),
      loglike.null = logLik(modele.null),
      full.deviance = logLik(modele) * -2,
      null.deviance = logLik(modele.null) * -2,
      nb.params = modele2@rank,
      n = nrow(dataset3)
    )

## `$`deviance expliquee`
## [1] 0.05075475
##
## `$`McFadden ajuste`
## [1] 0.05030669
##
## `$`Cox and Snell`
## [1] 0.3332721
##
## $Nagelkerke
## [1] 0.3333855

# Calcul du RMSE
preds <- predict(modele, type="response")[,1]
sqrt(mean((preds - dataset3$Duree)**2))

## [1] 13.33195

```

Le modèle n'explique que 5 % de la déviance et obtient des valeurs de R^2 ajusté de McFadden, de Cox et Snell et de Nagelkerke de respectivement 0,05, 0,33 et 0,33. La moyenne de l'erreur quadratique est de

seulement 13,4 indiquant que le modèle se trompe en moyenne de seulement 13,4 minutes. La capacité de prédiction du modèle est donc limitée sans être catastrophique.

Interprétation des résultats

Pour rappel, la fonction de lien dans notre modèle est la fonction `log`. Chaque coefficient représente donc l'effet de l'augmentation d'une unité des variables indépendantes sur le logarithme de l'espérance de notre variable dépendante. Si nous transformons nos coefficients avec la fonction exponentielle (`exp`), nous obtenons, pour chaque augmentation d'une unité des variables indépendantes, la multiplication de l'espérance de notre variable dépendante.

Puisque nos trajets peuvent provenir de nombreux arrondissements, nous proposons de représenter l'exponentiel de leurs coefficients avec un graphique. Nous pouvons d'ailleurs comparer les exponentiels des coefficients et les effets marginaux pour simplifier l'interprétation.

```
# Extraction des coefficient du modèle
coeffs <- modèle@coefficients
# Calcul des interval de confiance des coefficients
conf <- confint(modèle)
# Passage en exponentiel
df <- exp(cbind(coeffs, conf))
# Extraction des coefficients pour les arrondissements
dfArrond <- data.frame(df[grepl("ArrondDep", row.names(df), fixed = T),])
names(dfArrond) <- c("coeff", "lower", "upper")
dfArrond$Arrondissement <- gsub("ArrondDep", "", rownames(dfArrond), fixed = T)
# Graphique des exponentiels des coefficients
P1 <- ggplot(data = dfArrond) +
  geom_vline(xintercept = 1, color = "red") +
  geom_errorbar(aes(y = reorder(Arrondissement, coeff), xmin = upper, xmax = lower),
                height = 0) +
  geom_point(aes(y = reorder(Arrondissement, coeff), x = coeff)) +
  geom_text(aes(x = upper, y = reorder(Arrondissement, coeff),
                label = paste("coeff. : ", round(coeff, 2), sep = "")),
            size = 3, nudge_x = 0.07) +
  labs(x = "Coefficient multiplicateur (ref : Ahuntsic-Cartierville)",
       y = "",
       subtitle = "Exponentiels des coefficients du modèle") +
  xlim(c(0.75, 1.46))
# Création d'un DataFrame fictif pour les effets marginaux
dfpred <- expand.grid(
  LogDist = mean(dataset3$LogDist),
  Motif = 'education',
  HeureDep = '7',
  MemeArrond = 'Different',
  ArrondDep = unique(dataset3$ArrondDep),
  Mode = 'pieton', 'velo', 'transport collectif',
  Semaine = 'lundi au vendredi')
)
# Utiliser le modèle pour effectuer des prédictions (échelle log)
lin_pred <- predict(modèle, dfpred, se = T)
mu_lin_pred <- lin_pred$fitted.values[, 1]
se_lin_pred <- lin_pred$se.fit[, 1]
dfpred2 <- data.frame(
  pred = exp(mu_lin_pred),
  lower = exp(mu_lin_pred - 1.96 * se_lin_pred),
```

```

    upper = exp(mu_lin_pred + 1.96*se_lin_pred)
)
dfpred2 <- cbind(dfpred2, dfpred)
# Réaliser le graphique des effets marginaux
P2 <- ggplot(data = dfpred2) +
  geom_col(aes(x = pred, y = ArrondDep)) +
  geom_errorbarh(aes(xmin = lower, xmax = upper, y = ArrondDep)) +
  labs(x = "Temps de déplacement prédit", y = "",
       subtitle = "Prédiction du modèle")
ggarrange(P1,P2, ncol = 1, nrow = 2)

```

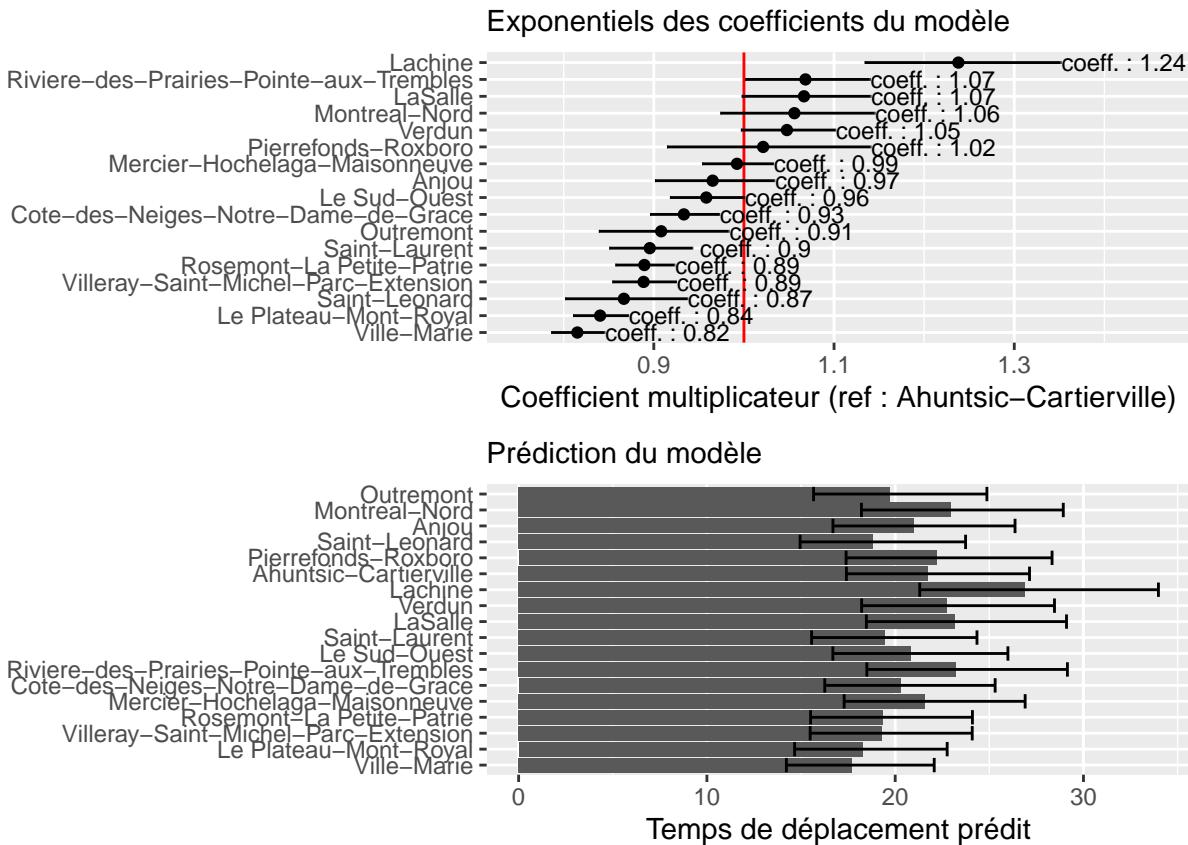


FIG. 8.59 : Effet de l'arrondissement de départ sur les temps de trajet à Montréal

La figure 8.59 permet de constater que les arrondissements Ville-Marie et Plateau-Mont-Royal se distinguent avec des trajets plus courts (environ 20 % plus courts en moyenne que les trajets partant d'Ahuntsic-Cartierville). À l'inverse, Lachine est de loin l'arrondissement avec les trajets les plus longs (25 % plus longs en moyenne que les trajets partant d'Ahuntsic-Cartierville).

Nous appliquons la même méthode de visualisation à la variable *Heure de départ* des trajets.

```

# Extraction des valeurs pour les heures de départ
dfHeures <- data.frame(df[grep("HeureDep", row.names(df), fixed = T),])
names(dfHeures) <- c("coeff", "lower", "upper")
dfHeures$Heure <- gsub("HeureDep", "", rownames(dfHeures), fixed = T)
# Rajouter des 0 et des h pour de jolies légendes
dfHeures$Heure <- paste(dfHeures$Heure, "h", sep = " ")

```

```
# Afficher le graphique
ggplot(data = dfHeures) +
  geom_hline(yintercept = 1, color = "red") +
  geom_errorbar(aes(x = Heure, ymin = upper, ymax = lower), width = 0) +
  geom_point(aes(x = Heure, y = coeff)) +
  geom_text(aes(y = upper, x = Heure, label = round(coeff,2)), size = 3, nudge_y = 0.07) +
  labs(x = "Coefficient multiplicateur (ref : 7 h)",
       y = "")
```

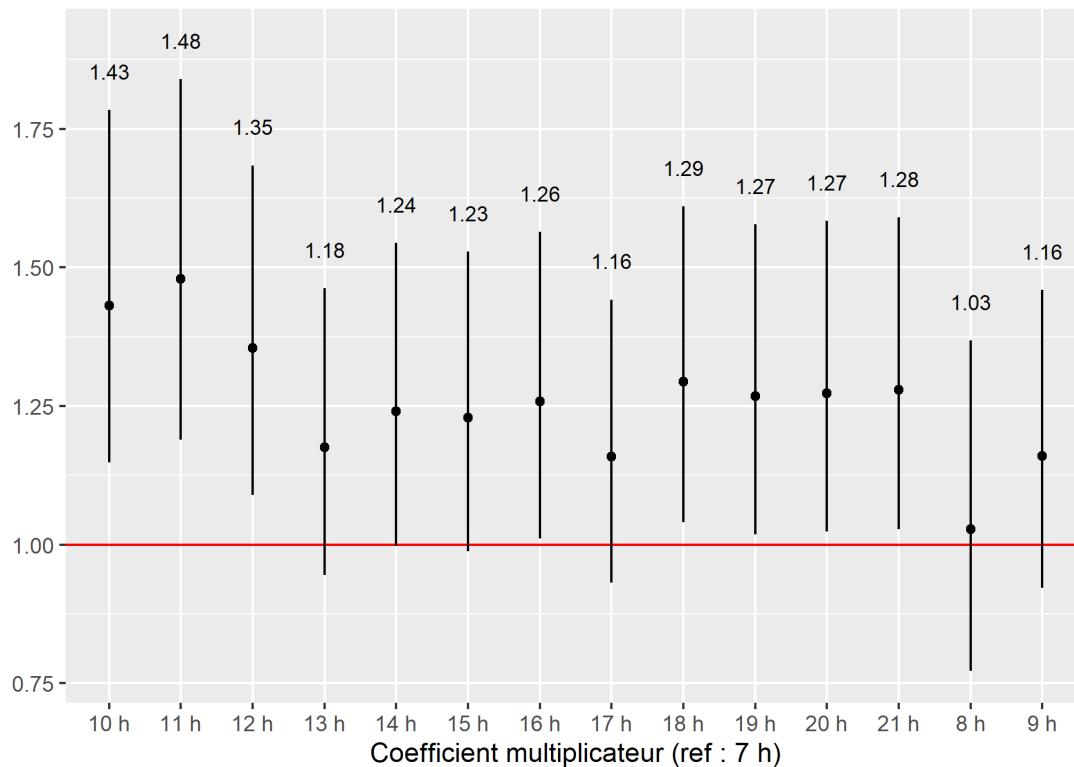


FIG. 8.60 : Effet de l'heure de départ sur les temps de trajet à Montréal

Nous pouvons ainsi observer, à la figure 8.60, que les trajets effectués à 10 h, 11 h et 12 h sont les plus longs de la journée, entre 30 et 40 % plus longs que ceux effectués à 7 h et 8 h qui constituent les trajets les plus courts.

Le reste des coefficients (ainsi que le paramètre de forme) sont affichés dans le tableau 8.35. Comparativement à un trajet effectué à pied, un trajet en transport en commun dure en moyenne 52 % plus longtemps (1,53 fois plus long), alors que les déplacements en véhicule individuel et en vélo sont respectivement 28 % et 23 % moins longs. Aucune différence n'est observable entre les déplacements effectués en semaine ou pendant la fin de semaine.

TAB. 8.35 : Résultats pour le modèle GLM Gamma

Variable	Coeff.	Exp(Coeff.)	Val.p	IC 2,5 % exp(Coeff.)	IC 97,5 % exp(Coeff.)	Sign.
Constante	2,928	18,681	0,000	14,969	23,336	***
<i>Mode</i>						
ref : pieton	—	—	—	—	—	—
transport collectif	0,421	1,523	0,000	1,484	1,562	***
vehicule individuel	-0,318	0,728	0,000	0,710	0,747	***
velo	-0,258	0,773	0,000	0,754	0,792	***
<i>Motif</i>						
ref : education	—	—	—	—	—	—
loisir	-0,012	0,988	0,495	0,956	1,022	
magasinage	-0,114	0,893	0,000	0,862	0,924	***
travail	-0,064	0,938	0,000	0,910	0,967	***
LogDist	0,334	1,396	0,000	1,381	1,411	***
<i>MemeArrond</i>						
ref : Different	—	—	—	—	—	—
Meme	-0,036	0,965	0,001	0,945	0,986	***
<i>Semaine</i>						
ref : lundi au vendredi	—	—	—	—	—	—
samedi et dimanche	0,006	1,006	0,569	0,985	1,027	
shape	1,149	3,155	0,000	3,099	3,209	***

Les déplacements ayant comme motif le magasinage et le travail ont tendance à être en moyenne plus courts de 11 % et 6 % respectivement, comparativement aux déplacements effectués pour l'éducation ou le loisir (différence non significative entre loisir et éducation). Sans surprise, la distance entre le point de départ et d'arrivée du trajet (*LogDist*) affecte sa durée de façon positive. Considérant qu'il est difficile d'interpréter des log de kilomètre (dû à une transformation de la variable originale), nous représentons l'effet de cette variable avec la prédiction du modèle à la figure. Nous utilisons pour cela le cas suivant : déplacement à pied à 7 h en semaine, ayant pour motif éducation, dont le point de départ se situe dans l'arrondissement Ahuntsic et donc le point d'arrivée est dans un autre arrondissement. Seule la distance du trajet varie de 1 à 40 km. À titre de comparaison, nous représentons aussi, pour les mêmes conditions, le cas d'une personne à vélo (en vert) et d'une personne utilisant le transport en commun (en bleu). Les lignes en pointillés représentent les intervalles de confiance à 95 % des prédictions (figure 8.61).

```
# Création d'un DataFrame fictif pour la prédiction
dfpred <- expand.grid(
  Dist = seq(1,40, 0.5),
  Motif = 'education',
  HeureDep = '7',
  MemeArrond = 'Different',
  ArrondDep = 'Ahuntsic-Cartierville',
  Mode = c('pieton','velo','transport collectif'),
  Semaine = 'lundi au vendredi'
)
# Mise en log de la variable de distance
dfpred$LogDist <- log(dfpred$Dist)
# Calcul des prédictions et de leur erreur standard (échelle log)
lin_pred <- predict(modele, dfpred, se = T)
# Calcul des intervalles de confiance et mise en exponentielle des prédictions
dfpred$pred <- exp(lin_pred$fitted.values[,1])
dfpred$lower <- exp(lin_pred$fitted.values[,1] - 1.96*lin_pred$se.fit[,1])
dfpred$upper <- exp(lin_pred$fitted.values[,1] + 1.96*lin_pred$se.fit[,1])
```

```

# Ajoutons les accents pour le graphiques
dfpred$Mode <- as.character(dfpred$Mode)
dfpred$Mode2 <- case_when(dfpred$Mode == "pieton" ~ "piéton",
                           dfpred$Mode == "velo" ~ "vélo",
                           TRUE ~ dfpred$Mode)

# Affichage des résultats
ggplot(data = dfpred) +
  geom_path(aes(x = Dist, y = lower, color = Mode2), linetype = "dashed") +
  geom_path(aes(x = Dist, y = upper, color = Mode2), linetype = "dashed") +
  geom_path(aes(x = Dist, y = pred, color = Mode2), size = 1) +
  labs(y = "temps de trajet prédit (minutes)",
       x = "distance à vol d'oiseau (km)")

```

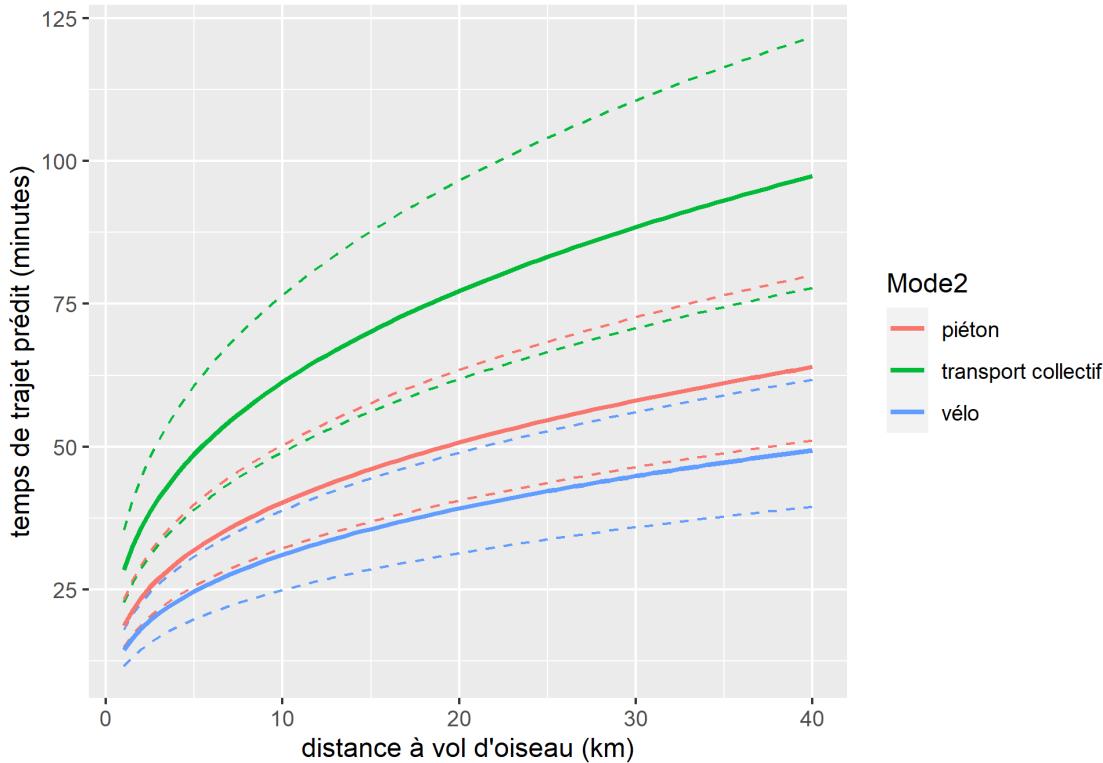


FIG. 8.61 : Effet de la distance à vol d'oiseau sur les temps de trajet à Montréal

8.4.4 Modèle GLM avec une distribution bêta

Pour rappel, la distribution bêta est une distribution définie sur l'intervalle $[0, 1]$, elle est donc particulièrement utile pour décrire des proportions, des pourcentages ou des probabilités. Dans la section 2.4.3.16 sur les distributions, nous avons présenté la paramétrisation classique de la distribution avec les paramètres a et b étant tous les deux des paramètres de forme. Ces deux paramètres n'ont pas d'interprétation pratique, mais il est possible (comme pour la distribution Gamma) de reparamétriser la distribution bêta avec un paramètre de centralité (espérance) et de dispersion.

Notez également que si la distribution bêta autorise la présence de 0 et de 1, le modèle GLM utilisant cette distribution doit les exclure des valeurs possibles s'il utilise la fonction de lien logistique. En effet, cette fonction à la forme suivante :

$$\text{logit}(x) = \log\left(\frac{x}{1-x}\right) \quad (8.26)$$

Nous pouvons constater que si $x = 1$, alors le dénominateur de la fraction est 0, or il est impossible de diviser par 0. Si $x = 0$, alors nous obtenons $\log(0)$ ce qui est également impossible au plan mathématique.

Dans le cas de figure où des 0 et/ou des 1 sont présents dans les données, quatre options sont possibles pour contourner le problème :

1. Si les observations à 0 ou 1 sont très peu nombreuses, il est envisageable de les retirer des données.
2. Si la variable mesurée le permet, il est possible de remplacer les 0 et les 1 par des valeurs très proches (0,0001 et 0,9999 par exemple) sans dénaturer excessivement les données initiales.
3. Plutôt que d'utiliser une valeur arbitraire, Smithson et Verkuilen (2006) recommande de recalculer la variable $Y \in [0; 1]$ avec la formule (8.27);
4. Employer un modèle Hurdle à trois équations, la première prédisant la probabilité d'observer $Y > 0$, la seconde, la probabilité d'observer $Y = 1$ et la dernière prédisant les valeurs de Y pour $0 > Y > 1$.

$$Y' = \frac{Y(N-1) + s}{N} \quad (8.27)$$

Avec N le nombre d'observations, Y' la variable Y transformée et s une constante. Plus cette dernière est élevée, plus la variable Y' a des valeurs éloignées de 0 et 1. la valeur de 0,5 est recommandée par les auteurs.



Reparamétrisation de la distribution bêta

Pour une distribution bêta telle que définie par $Y \sim Beta(a, b)$, l'espérance de cette distribution et sa variance sont données par :

$$\begin{aligned} E(Y) &= \frac{a}{a+b} \\ Var(Y) &= \frac{a \times b}{(a+b)^2(a+b+1)} \end{aligned} \quad (8.28)$$

Pour reparamétriser cette distribution, nous définissons un nouveau paramètre ϕ (phi) tel que :

$$\begin{aligned} a &= \phi * E(Y) \\ b &= \phi - a \\ Var(Y) &= \frac{E(Y) \times (1 - E(Y))}{1 + \phi} \end{aligned} \quad (8.29)$$

De cette manière, il est possible d'exprimer la distribution bêta en fonction de son espérance (sa valeur attendue, ce qui s'interprète approximativement comme une moyenne) et d'un paramètre ϕ intervenant dans le calcul de sa variance. Vous noterez d'ailleurs que la variance de cette distribution dépend de sa moyenne, impliquant à nouveau une hétéroscédasticité intrinsèque.

Pour résumer, nous nous retrouvons donc avec un modèle qui prédit l'espérance d'une distribution bêta avec une fonction de lien logistique. La variance de cette distribution est fonction de cette moyenne et d'un second paramètre ϕ . Ces informations sont résumées dans la fiche d'identité du modèle (tableau 8.36).

TAB. 8.36 : Carte d'identité du modèle bêta

Type de variable dépendante	Variable continue dans l'intervalle $]0, 1[$
Distribution utilisée	Student
Formulation	$Y \sim Beta(\mu, \phi)$ $g(\mu) = \beta_0 + \beta X$ $g(x) = \log\left(\frac{x}{1-x}\right)$
Fonction de lien	log
Paramètre modélisé	μ
Paramètres à estimer	β_0, β , et ϕ
Conditions d'application	$Variance = \frac{\mu \times (1-\mu)}{1+\phi}$

8.4.4.1 Conditions d'application

Comme pour un modèle Gamma, la seule condition d'application spécifique à un modèle avec distribution bêta est que la variance des résidus suit la forme attendue par la distribution bêta.

8.4.4.2 Interprétation des coefficients

Puisque le modèle utilise la fonction de lien logistique, les exponentiels des coefficients β du modèle peuvent être interprétés comme des rapports de cotes (voir la section 8.2.1 sur le modèle GLM binomial). Admettons ainsi que nous avons obtenu pour une variable indépendante X_1 le coefficient β_1 de 0,12. Puisque le coefficient est positif, cela signifie qu'une augmentation de X_1 conduit à une augmentation de l'espérance de Y . L'exponentiel de 0,12 est 1,13, ce qui signifie qu'une augmentation d'une unité de X_1 multiplie par 1,13 (augmente de 13 %) les chances d'une augmentation de Y . **Pour ce type de modèle, il est particulièrement important de calculer ses prédictions afin d'en faciliter l'interprétation.**

8.4.4.3 Exemple appliqué dans R

Afin de présenter le modèle GLM avec une distribution bêta, nous utilisons un jeu de données que nous avons construit pour l'île de Montréal. Nous nous intéressons à la question des îlots de chaleur urbains au niveau des aires de diffusion (AD – entités spatiales du recensement canadien comprenant entre 400 et 700 habitants). Pour cela, nous avons calculé dans chaque AD le pourcentage de sa surface classifiée comme îlot de chaleur dans la carte des îlots de chaleur/fraîcheur⁶ réalisée par l'INSPQ et le CERFO.

La question que nous nous posons est la suivante : les populations vulnérables socioéconomiquement et/ou physiologiquement sont-elles systématiquement plus exposées à la nuisance que représentent les îlots de chaleur ? Cette question se rattache donc au champ de la recherche sur la justice environnementale et plus spécifiquement sur sa dimension spatiale (à savoir l'équité environnementale, à distinguer des dimensions procédurale et de reconnaissance). Plusieurs études se sont d'ailleurs déjà penchées sur la question des îlots de chaleur abordée sous l'angle de l'équité environnementale (Harlan et al. 2007 ; Sanchez et Reames 2019 ; Huang, Zhou et Cadenasso 2011). Nous modélisons donc pour chaque AD ($n=3\,158$) de l'île de Montréal la proportion de sa surface couverte par des îlots de chaleur. Nos variables indépendantes sont divisées en deux catégories : variables environnementales et variables socio-économiques. Les premières sont des variables de contrôle, il s'agit de la densité de végétation dans l'AD (ajoutée avec une polynomiale d'ordre deux) et de l'arrondissement dans lequel elle se situe. Ces deux paramètres affectent directement les chances d'observer des îlots de chaleur, mais nous souhaitons isoler leurs effets (toutes choses étant égales par ailleurs) de ceux des variables socio-économiques. Ces dernières ont pour objectif de cibler les populations vulnérables sur le plan physiologique (personnes âgées et enfants de moins de 14 ans) ou socio-économique (minorités visibles et faible revenu). L'ensemble de ces variables

⁶<https://www.donneesquebec.ca/recherche/fr/dataset/ilot-de-chaleur-fraicheur-urbains-et-temperature-de-surface>

sont présentées dans le tableau 8.37. Notez que, puisque le modèle avec distribution bêta ne peut pas prendre en compte des valeurs exactes de 1 ou 0, nous les avons remplacées respectivement par 0,99 et 0,01. Cette légère modification n'altère que marginalement les données, surtout si nous considérons qu'elles sont agrégées au niveau des AD et proviennent originellement d'imagerie satellitaire.

TAB. 8.37 : Variables indépendantes utilisées dans le modèle bêta

Nom de la variable	Signification	Type de variable	Mesure
A65PlusPct	Population de 65 ans et plus	Variable continue	Pourcentage de la population ayant 65 ans et plus
A014Pct	Population de 14 ans et moins	Variable continue	Pourcentage de la population ayant 14 ans et moins
PopFRPct	Population à faible revenu	Variable continue	Pourcentage de la population à faible revenu
PopMVPct	Minorités visibles	Variable continue	Pourcentage de la population faisant partie des minorités visibles
VegPct	Végétation	Variable continue	Pourcentage de la surface de l'AD couverte par de la végétation
Arrond	Arrondissements	Variable continue	Arrondissement de l'Île de Montréal

Vérification des conditions d'application

Sans surprise, nous commençons par charger nos données et nous nous assurons de l'absence de multicolinéarité excessive entre nos variables indépendantes.

```
## Chargement des données
dataset <- read.csv("data/glm/data_chaleur.csv", fileEncoding = "utf8")
## Calcul des valeurs de vif
vif(glm(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  poly(prt_veg, degree = 2) + Arrond,
  data = dataset))
```

	GVIF	Df	GVIF^(1/(2*Df))
## A65Pct	1.609917	1	1.268825
## A014Pct	2.206072	1	1.485285
## PopFRPct	2.162036	1	1.470386
## PopMVPct	2.370269	1	1.539568
## poly(prt_veg, degree = 2)	2.619552	2	1.272204
## Arrond	7.899208	32	1.032820

La seule variable semblant poser un problème de multicolinéarité est la variable `Arrond`. Cependant, du fait de sa nature multinomiale, elle regroupe en réalité 32 coefficients (voir la colonne `Df`). Il faut donc utiliser la règle habituelle de 5 sur le carré de la troisième colonne (`GVIF^(1/(2*Df))`) du tableau (Fox et Monette 1992), soit $1,032820^2 = 1,066717$, ce qui est bien inférieur à la limite de 5. Nous n'avons donc pas de problème de multicolinéarité excessive. Nous pouvons passer au calcul des distances de Cook. Pour ajuster notre modèle, nous utilisons le package `mgcv` et la fonction `gam` avec le paramètre `family = betar(link = "logit")`.

```
# Ajustement d'une première version du modèle
modele <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
```

```

    poly(prt_veg, degree=2) + Arrond,
    data = dataset, family = betar(link = "logit"))
# Calcul des distances de Cook
df <- data.frame(
  cooksd = cooks.distance(modele),
  oid = 1:nrow(dataset)
)
# Affichage des distances de Cook
ggplot(data = df)+ 
  geom_point(aes(x = oid, y = cooksd),
             color = rgb(0.4,0.4,0.4,0.4), size = 0.5)+ 
  labs(x = "", y = "Distance de Cook")

```

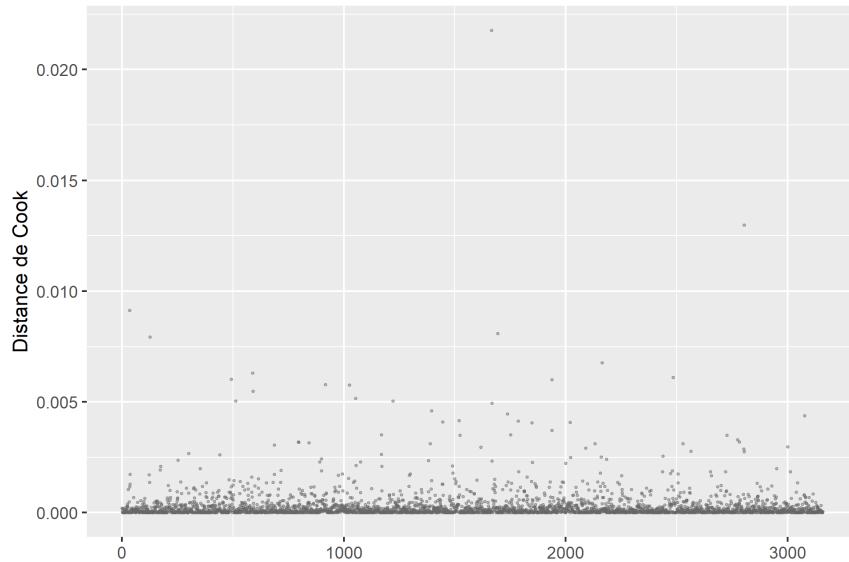


FIG. 8.62 : Distances de Cook pour le modèle GLM bêta

Nous pouvons observer à la figure 8.62 que seulement deux observations se distinguent très nettement des autres. Nous les isolons donc dans un premier temps.

```

cas_étranges <- subset(dataset, df$cooksd >= 0.01)
print(cas_étranges[,23:ncol(cas_étranges)])

```

```

##      A014Pct A65Pct PopFRPct PopMVPct      Km2     HabKm2 Shape_Leng Shape_Area
## 1666    11.78  26.77      6.38     11.65 6.458460    72.3083  21153.055  6458456.6
## 2803    15.54  31.07     24.17     52.17 0.134013  5283.0879   1651.471  134012.5
##      prt_hot  prt_veg dist_cntr      Arrond   hot
## 1666  1.824483 90.11691 29.181345 Senneville 0.02
## 2803 40.117994 60.15745  4.178823 Westmount 0.40

```

Ces deux observations n'ont pas de points communs marqués, et ne semblent pas avoir de valeurs particulièrement fortes sur les différentes variables indépendantes ou la variable dépendante. Nous décidons donc de les supprimer et de recalculer les distances de Cook.

```

# Suppression des deux observations très influentes
dataset2 <- subset(dataset, df$cooksd < 0.01)
modele2 <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  I(prt_veg**2) + prt_veg + Arrond,
  data = dataset2, family = betar(link = "logit"), methode = "REML")
# Calcul des distances de Cook
df2 <- data.frame(
  cooksd = cooks.distance(modele2),
  oid = 1:nrow(dataset2)
)
# Affichage des distances de Cook
ggplot(data = df2)+ 
  geom_point(aes(x = oid, y = cooksd),
  color = rgb(0.4,0.4,0.4,0.4), size = 0.5)+ 
  labs(x = "", y = "Distance de Cook")

```

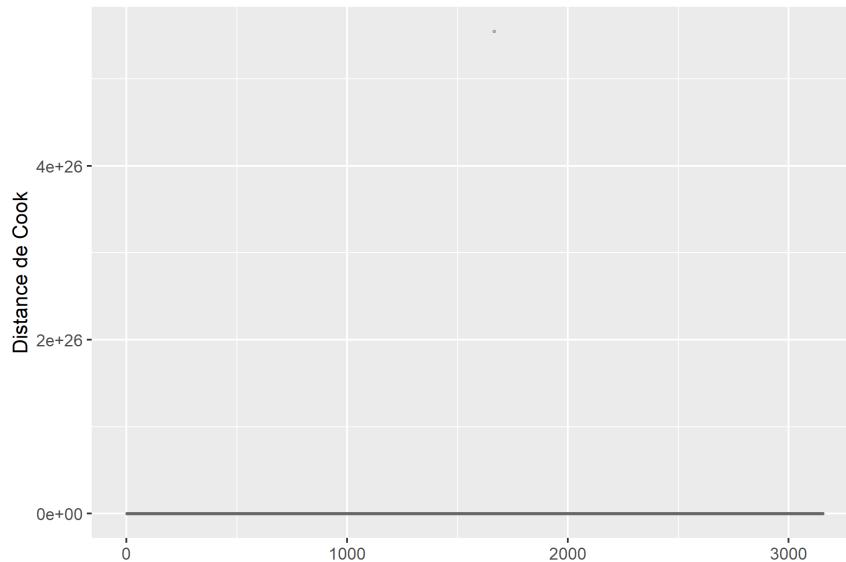


FIG. 8.63 : Distances de Cook pour le modèle GLM bêta (suppression de deux observations influentes)

Après réajustement (figure 8.63) nous constatons à nouveau qu'une observation est **extrêmement** éloignée des autres. Nous la retirons également, car cette différence est si forte qu'elle risque de polluer le modèle.

```

# Suppression de l'observation très étonnante
dataset3 <- subset(dataset2, df2$cooksd < max(df2$cooksd))
# Réajustement du modèle
modele3 <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  I(prt_veg**2) + prt_veg + Arrond,
  data = dataset3, family = betar(link = "logit"), methode = "REML")
# Calcul des distances de Cook
df3 <- data.frame(
  cooksd = cooks.distance(modele3),
  oid = 1:nrow(dataset3)
)

```

```
)  
# Affichage des distances de Cook  
ggplot(data = df3)+  
  geom_point(aes(x = oid, y = cooksd),  
             color = rgb(0.4,0.4,0.4,0.4), size = 0.5)+  
  labs(x = "", y = "Distance de Cook")
```

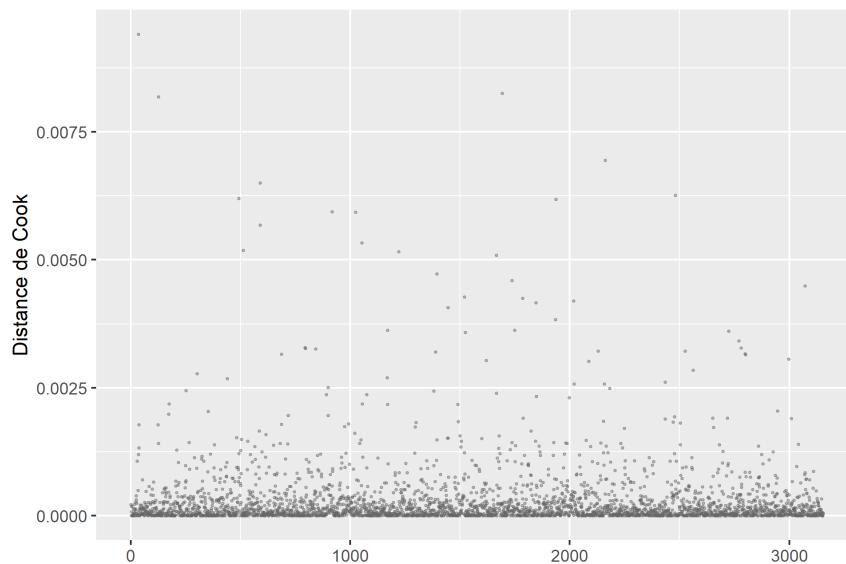


FIG. 8.64 : Distances de Cook pour le modèle GLM bêta (suppression de trois observations influentes)

Tout semble aller pour le mieux après ce second passage (figure 8.64). Si nous avions continué à observer des valeurs aussi influentes, nous aurions dû commencer à sérieusement questionner nos données ou notre modèle. La prochaine étape du diagnostic est donc l'analyse des résidus simulés.

```
fittedPredictedResponse = modele3$fitted.values,
integerResponse = F)
plot(sim_res)
```

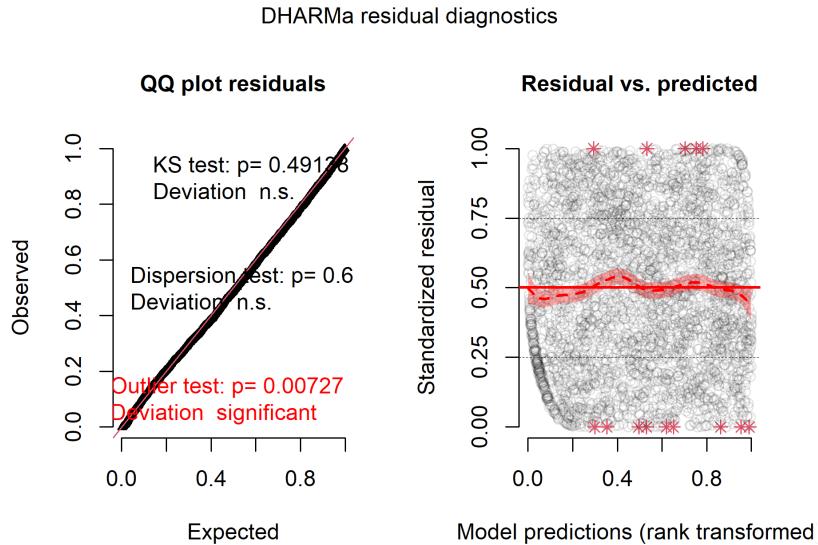


FIG. 8.65 : Diagnostic général des résidus simulés du modèle bêta

La figure 8.65 indique que les résidus suivent bien une distribution uniforme. Le test des valeurs aberrantes n'est pas significatif au seuil de 0,01 (nous retenons ce seuil considérant le grand nombre de simulations et d'observations de notre jeu de données), nous décidons donc de ne pas supprimer davantage d'observations. Le panneau de droite indique une relation non linéaire instable, mais essentiellement centrée sur la ligne droite attendue. Pour plus de détails, nous calculons ces résidus simulés avec chacune des variables indépendantes.

```
# Préparons un plot multiple
par(mfrow=c(2,3))
vars <- c("A65Pct", "A014Pct", "PopFRPct", "PopMV_pct", "prt_veg")
for(v in vars){
  plotResiduals(sim_res, dataset3[[v]], main = "", xlab = v)
}
plotResiduals(sim_res, dataset3[["prt_veg"]]**2, xlab = "prt_veg^2", main = "")
```

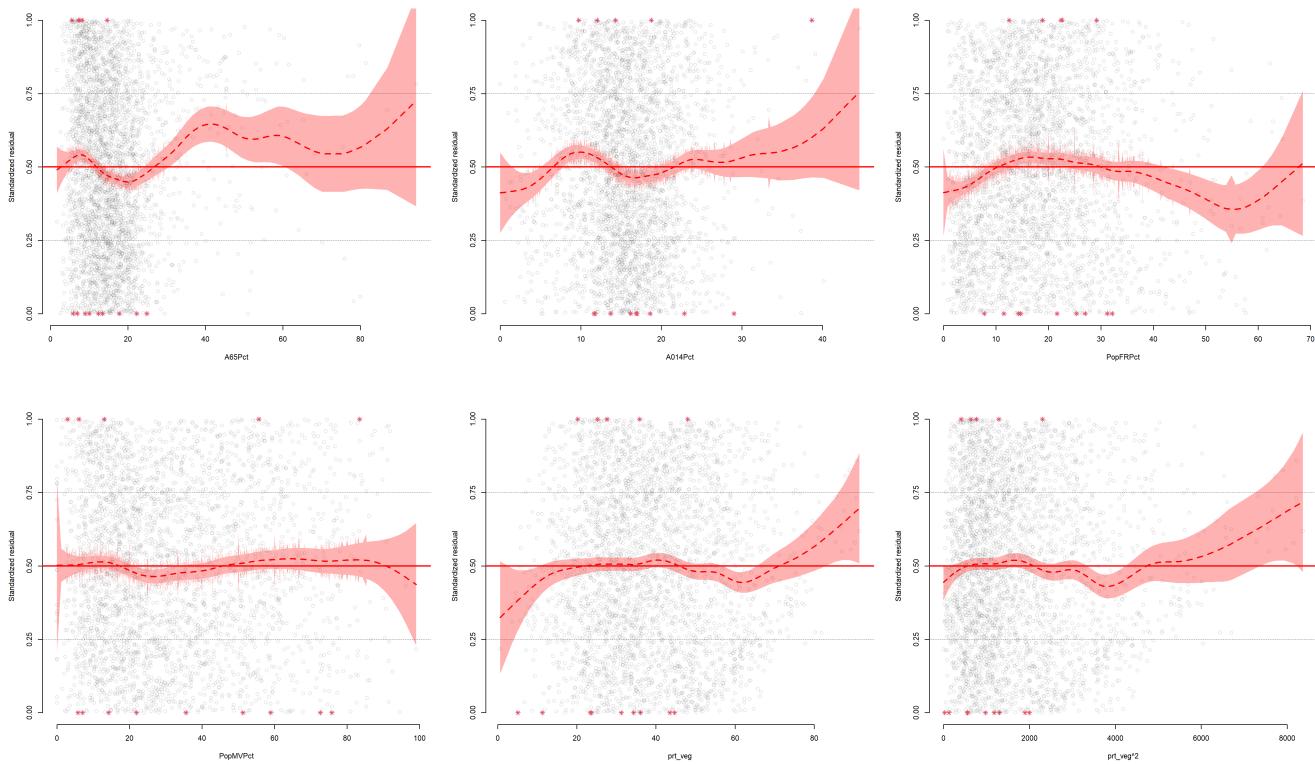


FIG. 8.66 : Relation entre chaque variable indépendante et les résidus simulés du modèle bêta

La figure 8.66 indique des relations marginales et négligeables entre nos variables indépendantes et nos résidus simulés. Concernant la variable Arrond (figure 8.67), nous observons une situation plus particulière. Pour quelques arrondissements, les résidus simulés sont nettement plus forts ou plus faibles. Notre hypothèse est que cet effet est provoqué par l'introduction de cette variable dans notre modèle comme un effet fixe alors que sa nature devrait nous inciter à l'introduire comme un effet aléatoire. Nous n'avons pas encore présenté ces concepts ici, mais nous le ferons dans le chapitre 9. En attendant, nous conservons le modèle tel quel et passons à l'analyse de sa qualité d'ajustement.

```
df <- data.frame(
  resid = residuals(sim_res),
  Arrond = dataset3$Arrond
)
ggplot(data = df) +
  geom_boxplot(aes(x = Arrond, y = resid))+
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()
      )+
  labs(x = "Arrondissements", y = "Résidus simulés")
```

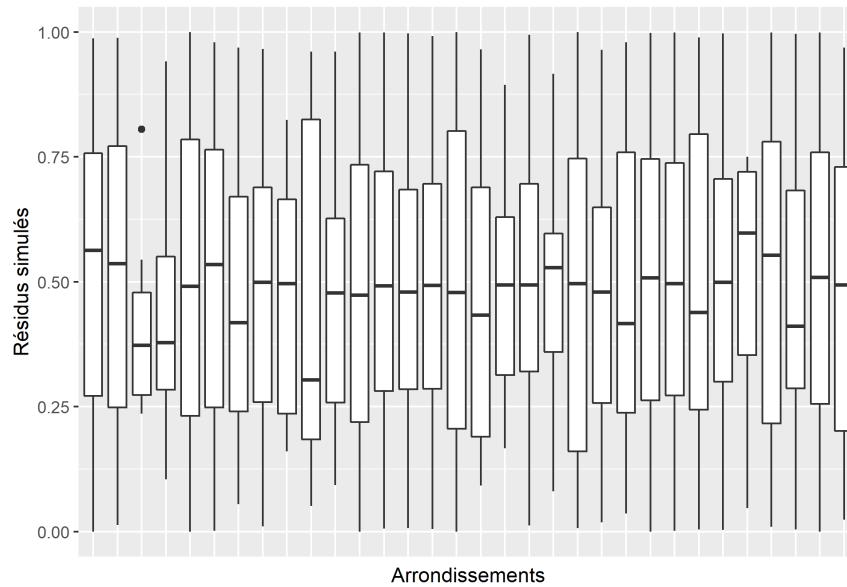


FIG. 8.67 : Relation entre la variable Arrondissement et les résidus simulés du modèle bêta

Analyse de la qualité d'ajustement

Dans un premier temps, nous comparons la distribution originale des données à des simulations issues du modèle.

```
# Extraction de 20 simulations
df2 <- data.frame(mat_sims[,1:20])
df3 <- reshape2::melt(df2)
ggplot() +
  geom_histogram(aes(x = hot, y = ..density..),
                 data = dataset3, bins = 100, color = "black", fill = "white") +
  geom_density(aes(x = value, y=..density.., group = variable), data = df3,
               fill = rgb(0,0,0,0), color = rgb(0.2,0.2,0.2,0.3), size = 1) +
  labs(x = "résidus simulés", y = "densité")
```

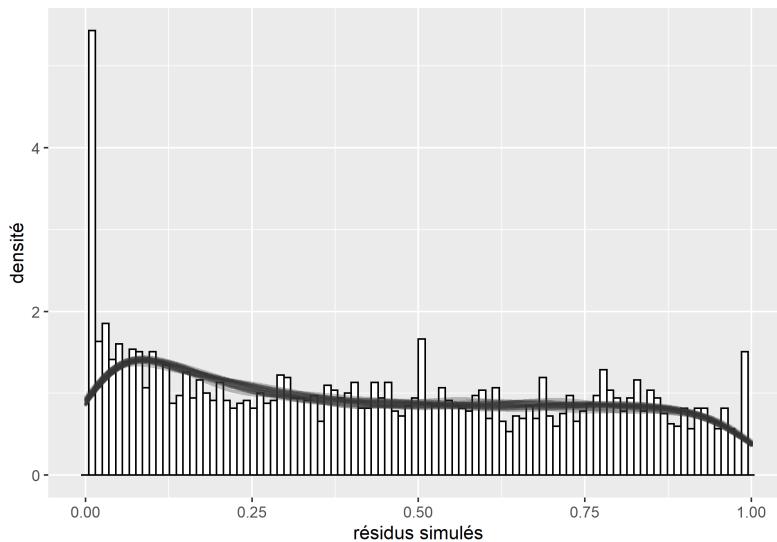


FIG. 8.68 : Comparaison entre la distribution originale et les simulations issues du modèle

Nous constatons à la figure 8.68 que le modèle est parvenu à reproduire la forme générale de la distribution originale : un plus grand nombre de valeurs proches de zéro, suivies d'une répartition presque homogène dans les valeurs comprises entre 0,15 et 0,8, suivies par un plus faible nombre de valeurs quand Y est supérieur à 0,8. Il semble en revanche manquer un certain nombre de valeurs extrêmes proches de 0 (absence d'îlot de chaleur) et proches de 1 (couverture à 100 % par des îlots de chaleur).

```
# Calcul des pseudo R2
rsqs(loglike.full = modele3$deviance/-2,
      loglike.null = modele3>null.deviance/-2,
      full.deviance = modele3$deviance,
      null.deviance = modele3>null.deviance,
      nb.params = modele3$rank,
      n = nrow(dataset3))
```

```
## `$deviance expliquee`
## [1] 0.9017396
##
## `$McFadden ajuste'
## [1] 0.8992329
##
## `$Cox and Snell'
## [1] 0.999828
##
## $Nagelkerke
## [1] 0.9998949
```

```
# Calcul du RMSE
sqrt(mean((modele3$fitted.values - modele3$y)**2))
```

```
## [1] 0.1025719
```

Le modèle parvient à expliquer 90 % de la déviance totale et obtient des pseudo- R^2 très élevés. Il obtient

cependant un RMSE de 0,10 soit une erreur quadratique moyenne de 10 % dans la prédiction, ce qui est tout de même important. Le modèle ne semble pas souffrir particulièrement de sur-ajustement comme les pseudo-R² auraient pu nous le laisser penser.

L'ensemble des coefficients du modèle sont accessibles via la fonction `summary`. Pour rappel, il est nécessaire de les convertir avec la fonction exponentielle pour pouvoir les interpréter en termes de rapport de cotes. À nouveau, nous proposons de construire dans un premier temps une figure pour observer l'effet des arrondissements.

```
# Identifier les coefficients pour les arrondissements
test <- grepl("Arrond", names(modele3$coefficients), fixed = T)
# Extraire les coefficients et les erreurs standards
coeffs <- modele3$coefficients[test]
err.std <- summary(modele3)$se[test]
# Créer un DataFrame avec les rapports de cote et les intervalles de confiance
df <- data.frame(
  Arrond = gsub("Arrond", "", names(coeffs), fixed = T),
  coeffs = coeffs,
  err.std = err.std,
  RC = exp(coeffs),
  lowerRC = exp(coeffs - 1.96 * err.std),
  upperRC = exp(coeffs + 1.96 * err.std)
)
# Retrouver l'arrondissement de référence
allArrond <- unique(dataset3$Arrond)
refArrond <- setdiff(allArrond, df$Arrond)
# Créer le graphique
ggplot(data = df) +
  geom_errorbarh(aes(xmin = lowerRC, xmax = upperRC, y = reorder(Arrond, RC)))+
  geom_point(aes(x = RC, y = reorder(Arrond, RC)))+
  geom_vline(xintercept = 1, color = "red")+
  geom_text(aes(x = upperRC, y = reorder(Arrond, RC),
    label = paste("RC : ", round(RC, 2), sep = "")), size = 3, nudge_x = 0.3)+
  labs(x = paste("Rapport de cote (rouge : ", refArrond, ')', sep = ''),
    y = 'Arrondissement')
```

Nous constatons ainsi que seuls quelques arrondissements ont une différence d'exposition aux îlots de chaleur significative au seuil de 0,05 comparativement à Ahuntsic-Cartierville (figure 8.69). Pour l'essentiel, il s'agit d'arrondissements pour lesquels nous observons des rapports de cotes supérieurs à 1. Verdun, Lasalle et le Plateau-Mont-Royal sont les arrondissements les plus touchés avec des chances d'observer des niveaux supérieurs de densité d'îlots de chaleur multipliés par 3,19, 2,89 et 2,74. Le reste des coefficients sont affichés dans le tableau 8.38.

Nous notons ainsi que le seul groupe associé avec une augmentation significative des chances d'observer une augmentation de la densité d'îlot de chaleur est le groupe des personnes à faible revenu (1,4 % de chance supplémentaire à chaque augmentation d'un point de pourcentage de la variable indépendante). Pour mieux cerner la taille de cet effet, nous représentons l'effet marginal de ce coefficient en maintenant toutes les autres variables à leur moyenne. Nous calculons également ces effets marginaux pour trois arrondissements différents : Verdun (RC le plus fort), Ahuntsic-Cartierville (la référence) et Dollard-des-Ormeaux (RC le plus faible). Nous réalisons également un second graphique pour visualiser l'effet non linéaire de la variable *pourcentage de végétation*. La figure 8.70 nous indique ainsi que le rôle de l'arrondissement est plus important que celui du pourcentage de personnes à faible revenu. Cependant, nous constatons que passer de 0 % de personnes à faible revenu dans une AD à 75 % est associé avec une multiplication de la surface couverte par des îlots de chaleur par environ 1,5 (toutes choses égales par

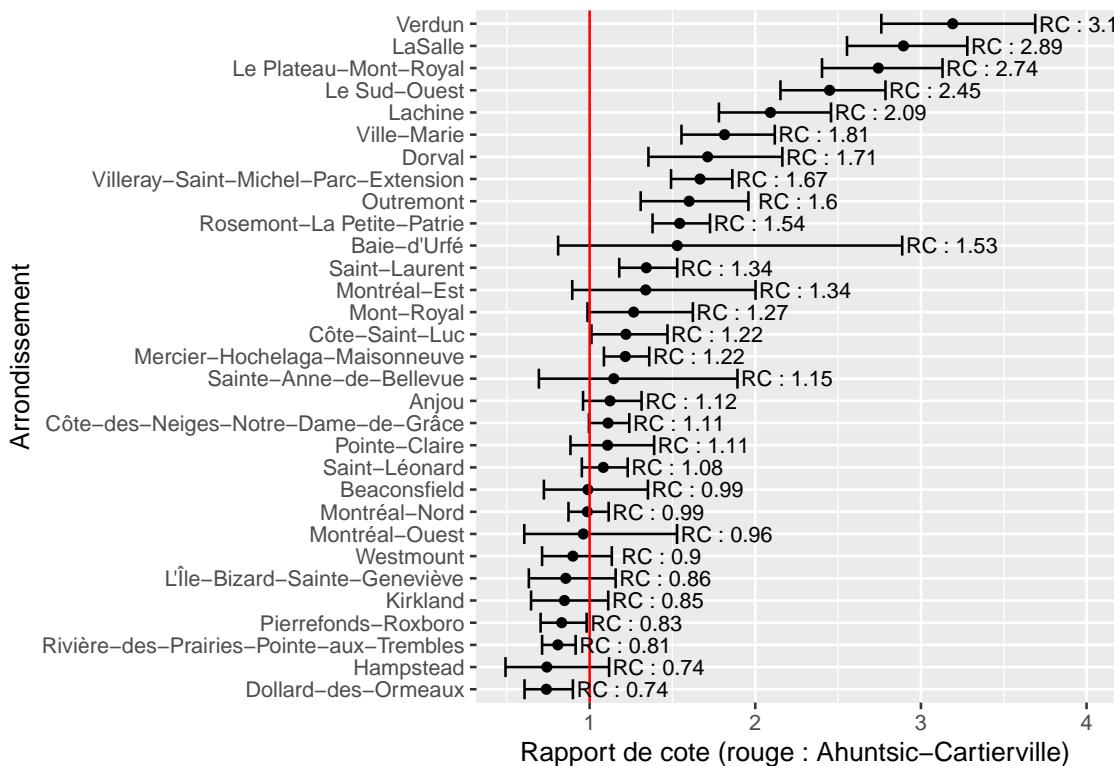


FIG. 8.69 : Rapports de cote pour les arrondissements dans le modèle bêta

ailleurs). Le rôle de la végétation dans la réduction de la surface des îlots de chaleur est très net et non linéaire. L'essentiel de la réduction est observé entre 0 et 50 % de végétation dans une AD, au-delà de ce seuil, la réduction des îlots de chaleur par la végétation est moins flagrante. Il semblerait donc exister à Montréal une forme d'iniquité systématique pour les populations à faible revenu, qui seraient davantage exposées aux îlots de chaleur. Cependant, compte tenu de la dépendance spatiale et de l'hétéroscésadité observées plus haut, des ajustements devraient être apportés au modèle pour confirmer ou infirmer ce résultat.

```
# Créer un DataFrame pour la prédiction
df <- expand_grid(
  A65Pct = mean(dataset3$A65Pct),
  A014Pct = mean(dataset3$A014Pct),
  PopFRPct = seq(0,75, 1),
  PopMVPct = mean(dataset3$PopMVPct),
  prt_veg = mean(dataset3$prt_veg),
```

TAB. 8.38 : Résultats pour le modèle GLM bêta

Variable	Coeff.	RC	Val.p	IC 2,5 % RC	IC 97,5 % RC	Sign.
Constante	3,468	32,059	0,000	25,636	40,085	***
A65Pct	-0,002	0,998	0,243	0,996	1,001	
A014Pct	-0,006	0,994	0,035	0,988	1,000	*
PopFRPct	0,013	1,014	0,000	1,011	1,016	***
PopMVPct	-0,003	0,997	0,000	0,995	0,998	***
prt_veg	-0,137	0,872	0,000	0,865	0,879	***

```

Arrond = c("Verdun", 'Ahuntsic-Cartierville', 'Dollard-des-Ormeaux')
)
# Effectuer les prédition sur l'échelle log
pred <- predict(modele3, df, se=T, type = "link")
# Calculer les prédictions et leurs intervalles de confiance
ilink <- modele3$family$linkinv
df$pred <- ilink(pred$fit)
df$lower <- ilink(pred$fit - 1.96* pred$se.fit)
df$upper <- ilink(pred$fit + 1.96* pred$se.fit)
# Afficher le résultat
P1 <- ggplot(data = df) +
  geom_path(aes(x = PopFRPct, y = pred, color = Arrond), size =1) +
  geom_path(aes(x = PopFRPct, y = lower, color = Arrond), linetype="dashed") +
  geom_path(aes(x = PopFRPct, y = upper, color = Arrond), linetype="dashed")+
  labs(x = "Personnes à faible revenu (%)",
       y = "Surface de l'AD couverte par des îlots de chaleur (%)",
       color = 'Arrondissement')+
  ylim(0,1)
# Pour la végétation
df2 <- expand.grid(
  A65Pct = mean(dataset3$A65Pct),
  A014Pct = mean(dataset3$A014Pct),
  PopFRPct = mean(dataset3$PopFRPct),
  PopMVPct = mean(dataset3$PopMVPct),
  prt_veg = seq(0,95,1),
  Arrond = c("Verdun", 'Ahuntsic-Cartierville', 'Dollard-des-Ormeaux')
)
# Effectuer les prédition sur l'échelle log
pred2 <- predict(modele3, df2, se=T, type = "link")
# Calculer les prédictions et leurs intervalles de confiance
df2$pred <- ilink(pred2$fit)
df2$lower <- ilink(pred2$fit - 1.96* pred2$se.fit)
df2$upper <- ilink(pred2$fit + 1.96* pred2$se.fit)
# Afficher le résultat
P2 <- ggplot(data = df2) +
  geom_path(aes(x = prt_veg, y = pred, color = Arrond), size =1) +
  geom_path(aes(x = prt_veg, y = lower, color = Arrond), linetype="dashed") +
  geom_path(aes(x = prt_veg, y = upper, color = Arrond), linetype="dashed")+
  labs(x = "Couverture végétale (%)", y = '', color = 'Arrondissement')+
  ylim(0,1)
ggarrange(P1,P2, common.legend = T)

```

8.5 Conclusion sur les modèles linéaires généralisés

Comme vous avez dû le remarquer, les modèles linéaires généralisés constituent un monde à part entière et tout un livre pourrait être rédigé à leur sujet. Leur grande flexibilité les rend extrêmement utiles dans de nombreux contextes, mais complique leur mise en œuvre, chaque modèle ayant ses propres spécificités théoriques. Ils partagent cependant tous une base commune : le choix d'une distribution et d'une fonction de lien. L'ensemble de leurs spécificités découle directement de ces deux choix.

La figure 8.71 résume les choix de modèles présentés au cours de ce chapitre pour des variables qualitatives, de comptage et continues. Notez bien qu'il ne s'agit que de la partie émergée de l'iceberg, car il existe de nombreuses autres distributions plus ou moins complexes (skew-normale, log-normal, beta-

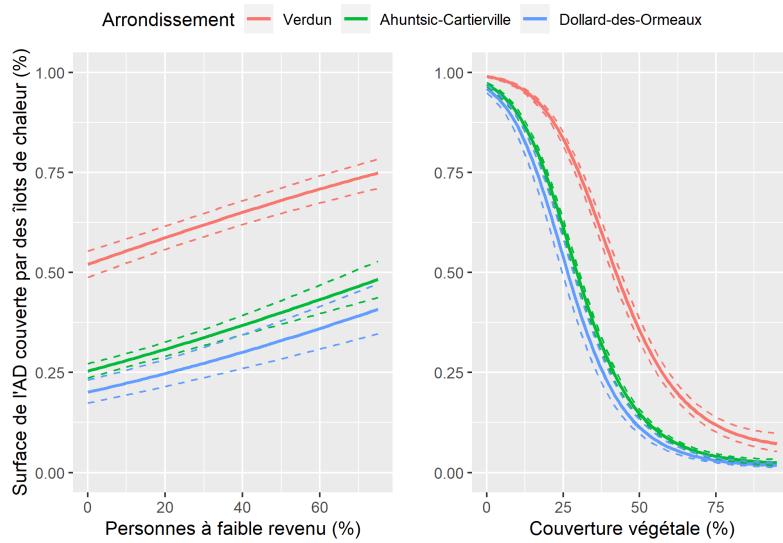


FIG. 8.70 : Effets marginaux des variables pourcentage de personnes à faible revenu et densité de végétation

binomiale, Box-Cox, Gumbel etc.). D'autres pistes pourraient aussi être explorées pour aller plus loin avec les GLM, notamment les modèles Hurdle (combinant un modèle binomial et un modèle avec une distribution continue), les modèles tronqués ou censurés (tenant compte d'une limite nette dans la variable dépendante) ou encore les modèles distributionnels ajustant une équation de régression pour chaque paramètre de la distribution utilisée.

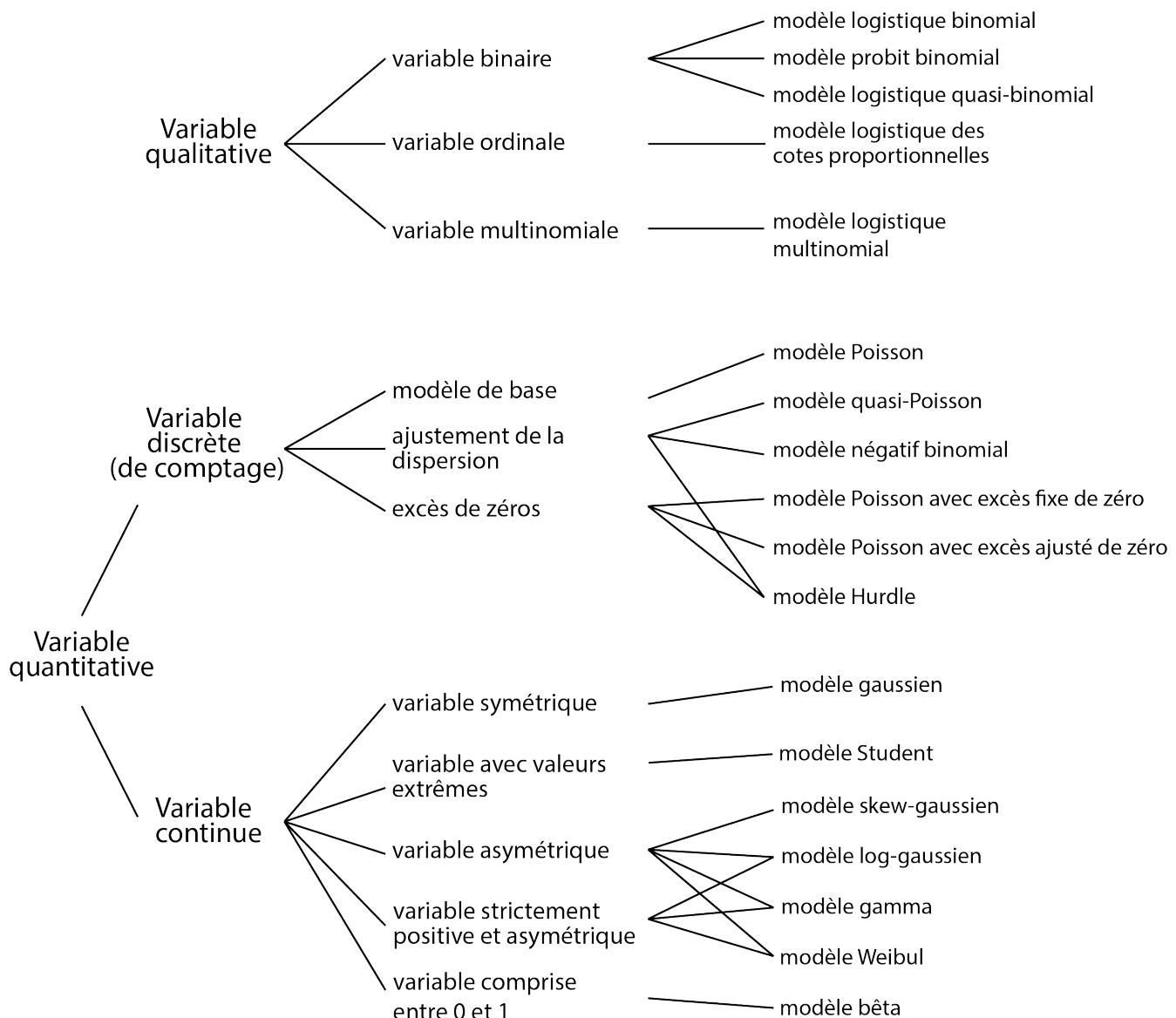


FIG. 8.71 : Résumé graphique des principaux GLM abordés

8.6 Quiz de révision du chapitre

Questions

- Contrairement à un modèle LM, un GLM permet de choisir :
 - la distribution conditionnelle de la variable Y
 - une fonction de lien appliquée aux coefficients du modèle
 - une fonction de lien appliquée à l'équation de régression du modèle
 - la distribution à priori de la variable Y
 - la distribution des résidus du modèle

Relisez au besoin la section 8.1.

- La fonction de lien est utilisée pour transformer :
 - les paramètres d'un modèle et les rendre plus interprétables

- le résultat de l'équation d'un modèle pour le contraindre à un intervalle de valeurs cohérent avec le paramètre de la distribution modélisée
- le résultat de l'équation d'un modèle pour obtenir un effet multiplicatif plutôt qu'additif des coefficients
- la variable Y pour qu'elle suive une distribution normale

Relisez au besoin la section [8.1.2](#)

- **Un modèle GLM est ajusté par la méthode des moindres carrés.**

- Vrai
- Faux

Relisez au besoin la section [8.1.1](#).

- **Dans un modèle GLM la vraisemblance (likelihood) est :**

- le produit des probabilités d'observer chacune des observations selon le modèle
- la somme des résidus du modèle
- un test de significativité du modèle
- impossible à calculer en pratique, nous lui préférons le log-likelihood

Relisez au besoin la section [11.3](#).

- **Idéalement, les résidus simulés d'un modèle GLM devraient :**

- suivre une distribution normale
- ne suivre aucune distribution particulière
- suivre une distribution uniforme
- être corrélés avec la variable Y

Relisez au besoin la section [8.1.4](#).

- **L'AIC**

- mesure la qualité d'ajustement d'un modèle
- tient compte du nombre de paramètres dans un modèle
- mesure le niveau de significativité des coefficients d'un modèle
- pénalise les modèles avec un plus petit nombre de paramètres
- tient compte du nombre d'observations dans le modèle

Relisez au besoin la section [8.1.5](#).

- **Pour un modèle utilisant une distribution binomiale, le paramètre modélisé par l'équation de régression est :**

- mu, la moyenne de y
- lambda, le nombre d'occurrence attendu pour y
- sigma, la variance de y
- p, la probabilité d'observer y = 1

Relisez au besoin la section [8.2.1](#).

- **Si la fonction de lien logistique est utilisée dans un modèle binomial, comment peuvent être interprétés les coefficients ?**

- En appliquant la fonction log aux coefficients, nous obtenons l'effet linéaire de ceux-ci sur la probabilité p

- En appliquant la fonction \exp aux coefficients, nous obtenons l'effet linéaire de ceux-ci sur la probabilité p
- En appliquant la fonction \exp aux coefficients, nous obtenons leur effet sous forme de rapport de côte sur la probabilité p
- En appliquant la fonction $1/\exp$ aux coefficients, nous obtenons leur effet sous forme de rapport de côte sur la probabilité p

Relisez au besoin la section 8.2.1.

• **Pour choisir la distribution la mieux adaptée pour un modèle GLM, il est possible de :**

- consulter la littérature et retenir les distributions utilisées habituellement
- comparer plusieurs distributions et conserver le modèle le mieux ajusté selon les AIC
- utiliser systématiquement la distribution normale
- demander à un service de voyance

Relisez au besoin la section 8.5.

Réponses

- Contrairement à un modèle LM, un GLM permet de choisir :
 - la distribution conditionnelle de la variable Y
 - une fonction de lien appliquée à l'équation de régression du modèle
- La fonction de lien est utilisée pour transformer :
 - le résultat de l'équation d'un modèle pour le contraindre à un intervalle de valeurs cohérent avec le paramètre de la distribution modélisée
 - le résultat de l'équation d'un modèle pour obtenir un effet multiplicatif plutôt qu'additif des coefficients
- Un modèle GLM est ajusté par la méthode des moindres carrés.
 - Faux
- Dans un modèle GLM la vraisemblance (likelihood) est :
 - le produit des probabilités d'observer chacune des observations selon le modèle
 - impossible à calculer en pratique, nous lui préférerons le log-likelihood
- Idéalement, les résidus simulés d'un modèle GLM devraient :
 - suivre une distribution uniforme
- L'AIC
 - mesure la qualité d'ajustement d'un modèle
 - tient compte du nombre de paramètres dans un modèle
- Pour un modèle utilisant une distribution binomiale, le paramètre modélisé par l'équation de régression est :
 - p , la probabilité d'observer $y = 1$
- Si la fonction de lien logistique est utilisée dans un modèle binomial, comment peuvent être interprétés les coefficients ?
 - En appliquant la fonction \exp aux coefficients, nous obtenons leur effet sous forme de rapport de côte sur la probabilité p
- Pour choisir la distribution la mieux adaptée pour un modèle GLM, il est possible de :
 - consulter la littérature et retenir les distributions utilisées habituellement
 - comparer plusieurs distributions et conserver le modèle le mieux ajusté selon les AIC

Chapitre 9

Régressions à effets mixtes (GLMM)

Dans les deux chapitres précédents, nous avons consécutivement présenté la méthode de la régression linéaire multiple (LM) ainsi qu'une de ses extensions, soit les modèles linéaires généralisés (GLM). Dans ce chapitre, nous poursuivons sur cette voie avec une nouvelle extension : les modèles généralisés à effet mixtes (GLMM). À la fin de cette section, vous serez en mesure de :

- comprendre la distinction entre un modèle GLM et un modèle GLMM;
- distinguer un effet fixe d'un effet aléatoire ;
- formuler des modèles GLMM avec des constantes et/ou des pentes aléatoires ;
- effectuer les diagnostics d'un GLMM.



Dans ce chapitre, nous utilisons principalement les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2`, le seul, l'unique !
 - * `ggpubr` pour combiner des graphiques et réaliser des diagrammes.
 - * `ellipse` pour représenter des ellipses sur certains graphiques.
- Pour ajuster des modèles GLMM :
 - * `lme4`, offrant une interface simple pour ajuster des GLMM.
- Pour analyser des modèles GLM :
 - * `car`, essentiellement pour la fonction `vif`.
 - * `DHARMa` pour le diagnostic des résidus simulés.
 - * `merTools` pour explorer les résultats d'un GLMM.
 - * `lmerTest` pour obtenir des tests de significativité pour les coefficients d'un GLMM.
 - * `MuMin` pour calculer les R^2 conditionnel et marginal.
 - * `performance` pour calculer l'ICC et d'autres mesures d'ajustement.

9.1 Introduction

9.1.1 Indépendance des observations et effets de groupes

Nous avons vu dans les précédents chapitres que l'indépendance des observations est une condition d'application commune à l'ensemble des modèles de régression. Cette condition implique ainsi que chaque unité d'observation de notre jeu de données est indépendante des autres ; en d'autres termes, qu'elle ne soit associée à aucune autre observation par un lien de dépendance. Prenons un exemple concret pour illustrer cette notion. Admettons que nous nous intéressons à la performance scolaire d'élèves du secondaire à Montréal. Pour cela, nous collectons la moyenne des résultats aux examens du Ministère de

tous les élèves des différentes commissions scolaires de l'île de Montréal. Chaque élève appartient à une classe spécifique, et chaque classe se situe dans une école spécifique. Les classes constituent des environnements particuliers, la performance des élèves y est influencée par un ensemble de facteurs comme la personne qui enseigne et les relations entre les élèves d'une même classe. Deux élèves provenant d'une même classe sont donc lié(e)s par une forme de structure propre à leur classe et ne peuvent pas être considéré(e)s comme indépendant(e)s. De même, l'école constitue un environnement particulier pouvant influencer la performance des élèves du fait de moyens financiers plus importants, de la mise en place de programmes spéciaux, de la qualité des infrastructures (bâtiment, gymnase, cour d'école) ou d'une localisation minimisant certaines nuisances à l'apprentissage comme le bruit. À nouveau, deux élèves provenant d'une même école partagent une forme de structure qui, cette fois-ci, est propre à leur école. Si nous collectons des données pour l'ensemble du Canada, nous pourrions étendre ce raisonnement aux villes dans lesquelles les écoles se situent et aux provinces.

Dans cet exemple, la dépendance entre les données est provoquée par un effet de groupe : il est possible de rassembler les observations dans des ensembles (classes et écoles) influençant vraisemblablement la variable étudiée (performance scolaire). Les effets des classes et des écoles ne sont cependant pas intrinsèques aux élèves. En effet, il est possible de changer un ou une élève de classe ou d'école, mais pas de changer son âge ou sa situation familiale. Il est ainsi possible de distinguer la population des élèves, la population des classes, et la population des écoles (figure 9.1). Ces effets de groupes sont plus la règle que l'exception dans l'analyse de données en sciences sociales, ce qui met à mal l'hypothèse d'indépendance des observations. Notez que les effets de groupes ne sont pas les seules formes de structures remettant en cause l'indépendance des observations. Il existe également des structures temporelles (deux observations proches dans le temps ont plus de chances de se ressembler) et spatiales (deux observations proches dans l'espace ont plus de chances de se ressembler) ; cependant, les cas de la dépendance temporelle et spatiale ne sont pas couverts dans ce livre, car ils sont complexes et méritent un ouvrage dédié.

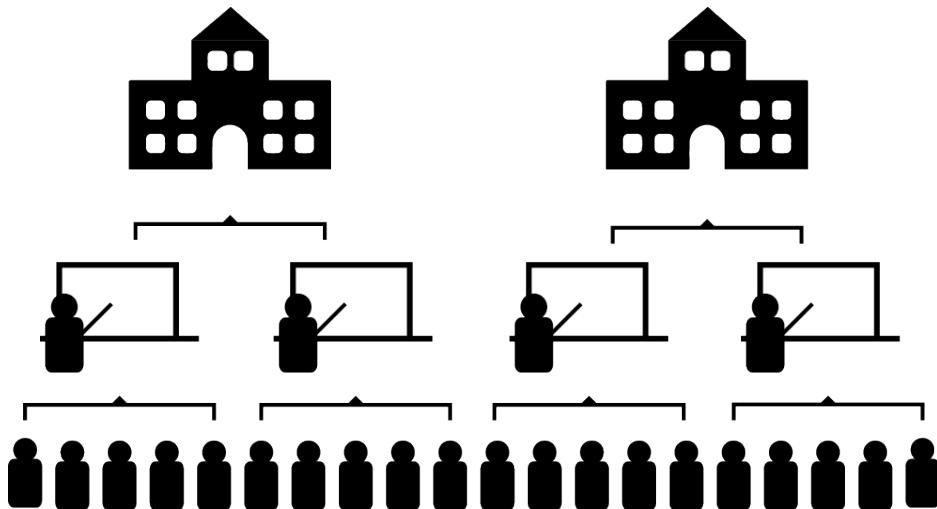


FIG. 9.1 : Structure hiérarchique entre élèves, classes et écoles



La notion de pseudo-réPLICATION

Les effets de dépendance causés par des structures de groupe, temporelles ou spatiales, sont regroupés sous le terme de pseudo-réPLICATION. Il est intéressant de se pencher sur la signification de ce mot pour comprendre le problème intrinsèque causé par la dépendance entre les observations et son influence sur l'inférence.

Reprendons l'exemple des élèves et de la performance scolaire et admettons que nous souhaitons estimer la moyenne générale de l'ensemble des élèves sur l'île de Montréal, mais que nous ne disposons pas du jeu de données complet. Nous devons donc collecter un échantillon suffisamment grand pour estimer la moyenne

pour l'ensemble de cette population. Raisonnons en termes de quantité d'informations. Si nous ne disposons d'aucune observation (nous n'avons pas encore interrogé d'élèves), cette quantité est de 0. Si nous interrogeons un premier ou une première élève, nous obtenons une donnée supplémentaire, et donc un point d'information supplémentaire (+1). Admettons maintenant que nous collectons 30 observations dans une école, 10 dans une seconde et 5 dans une troisième. A priori, nous pourrions dire que nous avons ajouté 45 points d'information à notre total de connaissance. Ce serait le cas si les observations étaient indépendantes les unes des autres. Dans un tel contexte, chaque observation ajoute la même quantité d'information. Cependant, puisque les personnes étudiant dans la même école ont plus de chance de se ressembler, interroger les élèves d'une même école apporte moins d'information. Notez que plus la ressemblance entre les élèves d'une même école est forte, plus la quantité d'information est réduite. Nous sommes donc loin de disposer d'une quantité d'information égale à 45. Chaque réplication de l'expérience (demander à un ou une élève sa moyenne annuelle) n'apporte pas autant d'information qu'attendu si les observations étaient indépendantes, c'est pourquoi on parle de **pseudo-réPLICATION**.

La pseudo-réPLICATION influence directement l'inférence statistique puisque le calcul des différents tests statistiques assume que chaque observation apporte autant d'information que les autres. En cas de présence de pseudo-réPLICATION, la quantité d'information présente dans l'échantillon est plus petite qu'attendu. Il est possible de voir cela comme une forme de surestimation de la taille de l'échantillon. En cas de pseudo-réPLICATION, nous disposons en réalité de moins de données que ce que l'on attendrait d'un échantillon de cette taille, si les observations étaient indépendantes. La conséquence est la sous-estimation de la variabilité réelle des données et l'augmentation des risques de trouver des effets significatifs dans l'échantillon alors qu'ils ne le sont pas pour l'ensemble de la population.

9.1.2 Terminologie : effets fixes et effets aléatoires

Puisque les effets des classes et des écoles ne sont pas propres aux élèves, il convient de les introduire différemment dans les modèles de régression. Nous appelons un effet fixe, un effet qui est propre aux observations que nous étudions et un effet aléatoire, un effet provoqué par une structure externe (effet de groupe, effet temporel et/ou effet spatial). Un modèle combinant à la fois des effets fixes et des effets aléatoires est appelé un **modèle à effets mixtes**, ou GLMM pour *Generalized Linear Mixed Model*. Tous les modèles que nous avons ajustés dans les sections précédentes ne comprenaient que des effets fixes alors qu'à plusieurs reprises, des effets aléatoires induits par l'existence de structure de groupe auraient pu (dû) être utilisés. Prenons pour exemple le modèle logistique binomial visant à prédire la probabilité d'utiliser le vélo comme mode de transport pour son trajet le plus fréquent. La variable multinomiale *Pays*, représentant le pays dans lequel les personnes interrogées résident, a été introduite comme un effet fixe. Cependant, l'effet du pays ne constitue pas une caractéristique propre aux individus ; il s'agit plutôt d'un agrégat complexe mêlant culture, météorologie, politiques publiques et formes urbaines. À l'inverse, le sexe ou l'âge sont bien des caractéristiques intrinsèques des individus et peuvent être considérés comme des effets fixes.

Notez que l'utilisation du terme *effet aléatoire* peut porter à confusion, car il est utilisé différemment en fonction du champ d'études. Parmi les différentes définitions relevées par Gelman (2005) d'un effet aléatoire, citons les suivantes :

- Les effets fixes sont identiques pour tous les individus, alors que les effets aléatoires varient (définition 1).
- Les effets sont fixes s'ils sont intéressants en eux-mêmes, et les effets sont aléatoires si nous nous intéressons à la population dont ils sont issus (définition 2).
- Lorsqu'un échantillon couvre une grande part de la population, la variable correspondante est un effet fixe. Si l'échantillon couvre une faible part de la population, l'effet est aléatoire (définition 3).
- Si l'effet est censé provenir d'une variable aléatoire, alors il s'agit d'un effet aléatoire (définition 4).

- Les effets fixes sont estimés par la méthode des moindres carrés ou par le maximum de vraisemblance, alors que les effets aléatoires sont estimés avec régularisation (*shrinkage*) (définition 5).

Il est ainsi possible de se retrouver dans des cas où un effet serait classé comme fixe selon une définition et aléatoire selon une autre. La deuxième définition suppose même qu'un effet peut être aléatoire ou fixe selon l'objectif de l'étude. La dernière définition a l'avantage d'être mathématique, mais ne permet pas de décider si un effet doit être traité comme aléatoire ou fixe. Nous ne proposons pas ici de clore le débat, mais plutôt de donner quelques pistes de réflexion pour décider si un effet doit être modélisé comme fixe ou aléatoire :

- Est-ce que l'effet en question est propre aux individus étudiés ou est externe aux individus ? S'il est propre aux individus, il s'agit plus certainement d'un effet fixe. À titre d'exemple, il n'est pas possible de changer l'âge d'une personne, mais il est certainement possible changer sa ville de résidence.
- Existe-t-il un nombre bien arrêté de catégories possibles pour l'effet en question ? Si oui, il s'agit plus certainement d'un effet fixe. Il y a un nombre bien arrêté de catégories pour la variable sexe, mais pour la variable pays, de nombreuses autres valeurs peuvent être ajoutées. Il est également possible de se demander s'il semble cohérent d'effectuer un échantillonnage sur les catégories en question. Dans le cas des pays, nous pourrions mener une étude à l'échelle des pays et collecter des données sur un échantillon de l'ensemble des pays. Il existe donc une population de pays, ce que nous ne pouvons pas affirmer pour la variable sexe.
- L'effet en question est direct ou indirect ? Dans le second cas, l'effet en question est un agglomérat complexe découlant de plusieurs processus n'ayant pas été mesurés directement, ce qui correspond davantage à un effet aléatoire. Ainsi, l'effet du pays de résidence des individus sur leur probabilité d'utiliser le vélo est bien une agglomération complexe d'effets (culture, météorologie, orientation des politiques publiques, formes urbaines, etc.) n'ayant pas tous été mesurés. À l'inverse, l'âge d'un individu a bien un effet direct sur sa probabilité d'utiliser le vélo.
- L'effet est-il le même pour tous les individus, ou doit-il varier selon le groupe dans lequel l'individu se situe ? Si un effet doit varier en fonction d'un groupe, il s'apparente davantage à un effet aléatoire. Pour reprendre l'exemple de l'âge, nous pourrions décider que cette caractéristique des individus n'a peut-être pas le même effet en fonction du pays dans lequel vit l'individu et l'ajouter au modèle comme un effet aléatoire.

Vous comprendrez donc qu'une partie non négligeable du choix entre effet fixe ou un effet aléatoire réside dans le cadre théorique à l'origine du modèle. Maintenant que cette distinction conceptuelle a été détaillée, nous pouvons passer à la présentation statistique des modèles GLMM.

9.2 Principes de base des GLMM

Un GLMM est donc un modèle GLM introduisant à la fois des effets fixes et des effets aléatoires. Si nous ne considérons que les effets de groupes, un GLMM peut avoir trois formes : constantes aléatoires, pentes aléatoires et constantes et pentes aléatoires. Nous présentons ici ces trois formes en reprenant l'exemple ci-dessus avec des élèves intégré(e)s dans des classes et pour lesquel(le)s le niveau de performance à l'examen ministériel de mathématique nous intéresse.

9.2.1 GLMM avec constantes aléatoires

Il s'agit de la forme la plus simple d'un GLMM. Plus spécifiquement, elle autorise le modèle à avoir une constante différente pour chaque catégorie d'une variable multinomiale. En d'autres termes, si nous reprenons l'exemple des élèves dans leurs classes, nous tentons d'ajouter dans le modèle l'idée que chaque classe a une moyenne différente en termes de performance à l'examen de mathématique. Il est assez facile

de visualiser ce que cela signifie à l'aide d'un graphique. Admettons que nous modélisons la note obtenue par des élèves du secondaire à l'examen ministériel de mathématique à partir d'une autre variable continue représentant le temps de travail moyen par semaine en dehors des heures de classe et d'une variable catégorielle représentant dans quelle classe se trouve chaque élève. Notez qu'il ne s'agit pas ici de vraies données, mais de simples simulations utilisées à titre d'illustration. Si nous ne tenons pas compte des classes, nous pouvons ajuster une régression linéaire simple entre nos deux variables continues comme le propose la figure 9.2.

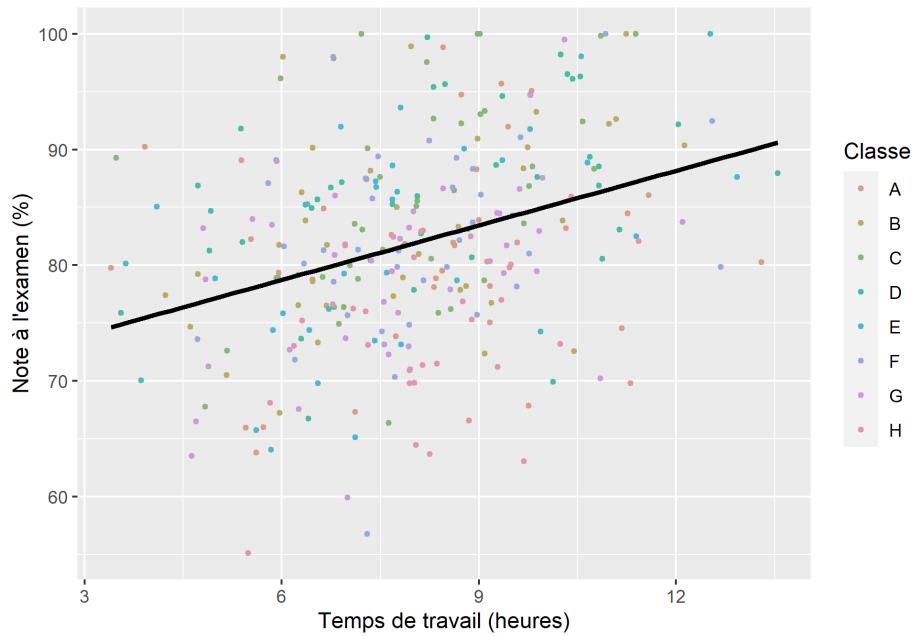


FIG. 9.2 : Influence du temps de travail sur la performance scolaire d'élèves

Nous constatons que notre modèle semble bien identifier la relation positive entre le temps de travail et le niveau de performance, mais la droite de régression est très éloignée de chaque point; nous avons ainsi énormément d'erreurs de prédiction, et donc des résidus importants. Jusqu'ici, nous avons vu que nous pouvons ajouter un prédicteur et intégrer l'effet des classes comme un effet fixe (figure 9.3).

Cet ajustement constitue une nette amélioration du modèle. Prenons un instant pour reformuler clairement notre modèle à effets fixes :

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma) \\
 g(\mu) &= \beta_0 + \beta_1 x_1 + \sum_{j=1}^k \beta_j x_{2j} \\
 g(x) &= x
 \end{aligned} \tag{9.1}$$

avec x_1 le temps de travail et x_2 la classe ayant $k-1$ modalités (puisque une modalité est la référence). Nous ajustons ainsi un coefficient pour chaque classe, ce qui a pour effet de tirer vers le haut ou vers le bas la prédiction du modèle en fonction de la classe. Cet effet est pour l'instant fixe, mais nous avons déterminé dans les sections précédentes qu'il serait conceptuellement plus approprié de le traiter comme un effet aléatoire.

Passons à présent à la reformulation de ce modèle en transformant l'effet fixe de la classe en effet aléatoire.

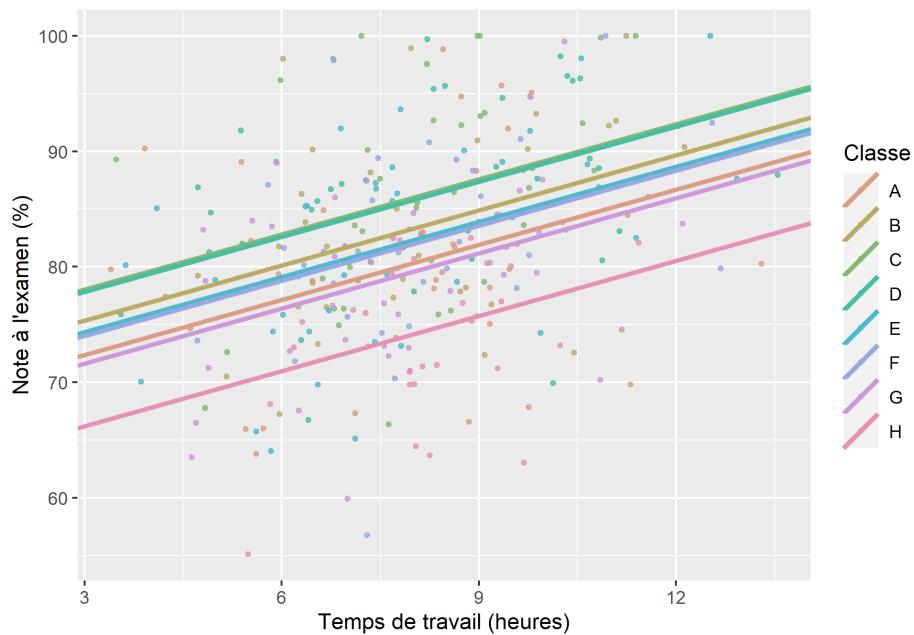


FIG. 9.3 : Influence du temps de travail sur la performance scolaire d’élèves en tenant compte de l’effet de leur classe (effet fixe)

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma_e) \\
 g(\mu) &= \beta_0 + \beta_1 x_1 + v \\
 v &\sim Normal(0, \sigma_v) \\
 g(x) &= x
 \end{aligned} \tag{9.2}$$

Remarquez que l’effet fixe de la classe $\sum_{j=1}^k \beta_j x_{2j}$ a été remplacé par v qui est un terme aléatoire propre aux classes et qui suit une distribution normale centrée sur 0 ($v \sim Normal(0, \sigma_v)$). En d’autres termes, cela signifie que l’effet des classes sur la performance des élèves suit une distribution normale et que si nous moyennons l’effet de toutes les classes, cet effet serait de 0. Nous ne modélisons donc plus l’effet moyen de chaque classe comme dans le modèle à effets fixes, mais la variabilité de l’effet des classes, soit σ_v . Notre modèle a donc deux variances, une au niveau des élèves (σ_e) et une au niveau des classes (σ_v). Cette particularité explique souvent pourquoi ce type de modèle est appelé un modèle hiérarchique ou un modèle de partition de la variance. Cette information est particulièrement intéressante, car elle permet de calculer la part de la variance présente au niveau des élèves et celle au niveau des classes.

Selon cette formulation, les constantes propres à chaque classe sont issues d’une distribution normale (nous reviendrons d’ailleurs sur ce choix plus tard), mais elles n’apparaissent pas directement dans le modèle. Ces paramètres ne sont plus estimés directement dans le modèle, mais a posteriori à partir des prédictions du modèle, et sont appelés *Best Linear Unbiased Predictor* (BLUP). Ces dernières précisions devraient d’ailleurs mieux vous aider à comprendre l’origine des définitions 1, 2 et 4 que nous avons mentionnées précédemment.

En comparant les figures 9.3 et 9.4, la différence ne saute pas aux yeux ; vous pourriez alors légitimement vous demander pourquoi tous ces efforts et cette complexité théorique pour une différence d’ajustement minime ? Trois arguments permettent de justifier l’utilisation de constantes aléatoires plutôt que d’effets fixes dans notre cas.

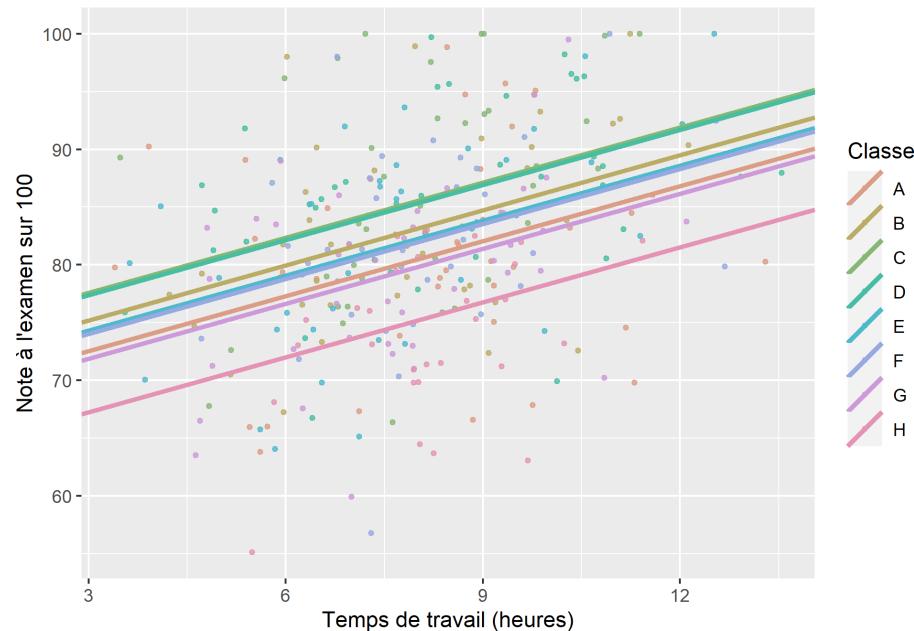


FIG. 9.4 : Influence du temps de travail sur la performance scolaire d'élèves en tenant compte de l'effet de leur classe (effet aléatoire)

9.2.1.1 Resserrement (*shrinkage*) et mutualisation (*partial pooling*)

Le premier intérêt d'utiliser un effet aléatoire réside dans sa méthode d'estimation qui diffère largement d'un effet fixe. Il est assez facile de se représenter intuitivement la différence entre les deux. Dans le cas de nos élèves et de nos classes, lorsque l'effet des classes est estimé avec un effet fixe, l'effet de chaque classe est déterminé de façon totalement indépendante des autres classes. En d'autres termes, il n'est possible d'en apprendre plus sur une classe qu'en collectant des données dans cette classe (*separate pooling*). Si l'effet des classes est estimé comme un effet aléatoire, alors l'information entre les classes est mutualisée (*partial pooling*). L'idée étant que l'information que nous apprenons sur des élèves dans une classe est au moins en partie valide dans les autres classes. Cette méthode d'estimation est particulièrement intéressante si nous ne disposons que de peu d'observations dans certaines classes, puisque nous pouvons apprendre au moins une partie de l'effet de cette classe à partir des données des autres classes. Cela n'est pas possible dans le cas d'un effet fixe où l'on traite chaque classe en silo. McElreath (2020) écrit à ce sujet qu'un effet fixe « n'a pas de mémoire » et qu'il oublie tout ce qu'il a appris sur les classes lorsqu'il passe à une nouvelle classe. La conséquence de cette mutualisation de l'information est un resserrement (*shrinkage*) des effets des classes autour de leur moyenne. Cela signifie que les tailles des effets de chaque classe sont plus petites dans le cas d'un effet aléatoire que d'un effet fixe. Utiliser des effets aléatoires conduit donc à une estimation plus conservatrice de l'effet des classes. Nous pouvons le visualiser en comparant les effets de classes dans le modèle à effets mixtes et le modèle à effets fixes. La figure 9.5 montre clairement que les effets aléatoires tendent à se rapprocher (resserrement) de leur moyenne (ligne noire), et donc à identifier des effets moins extrêmes pour chaque classe. Cette explication est directement en lien avec la définition 5 d'un effet aléatoire vu précédemment.

9.2.1.2 Prédiction pour de nouveaux groupes

Une autre retombée directe de la mutualisation de l'information est la capacité du modèle à envisager les effets plausibles pour de nouvelles classes. En effet, puisque nous avons approximé l'effet des classes sous forme d'une distribution normale dont nous connaissons la moyenne (0) et l'écart-type (σ_v), nous

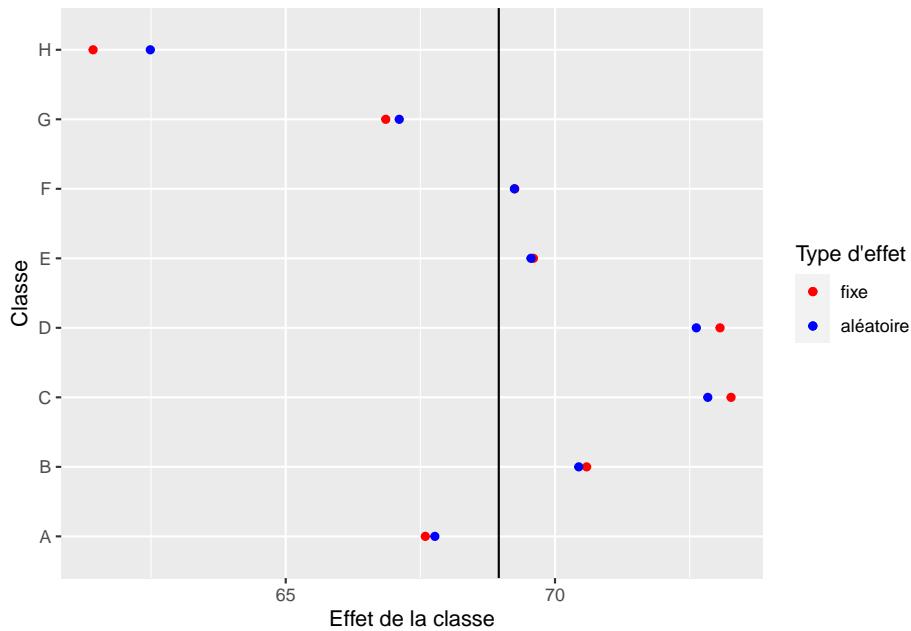


FIG. 9.5 : Comparaison des effets des classes pour le modèle à effets fixes versus le modèle à effets aléatoires

pouvons **simuler** des données pour de nouvelles classes, ce que ne permet pas un effet fixe. Ce constat est d'ailleurs directement lié à la définition 3 des effets aléatoires vue précédemment. Dans notre cas, $\sigma_v = 3,542$, ce qui nous permet d'affirmer que dans 95 % des classes, l'effet de la classe sur la performance scolaire doit se trouver entre $-1,96 \times 3,542$ et $+1,96 \times 3,542$, soit l'intervalle $[-6,942, 6,942]$.

9.2.1.3 Partition de la variance

Un autre avantage net de l'effet aléatoire est l'estimation du paramètre σ_v , soit la variance au niveau des écoles. Ce dernier permet de calculer un indicateur très intéressant, soit le **coefficients de corrélation intraclassé (ICC)** :

$$ICC = \frac{\sigma_v}{\sigma_v + \sigma_e} \quad (9.3)$$

Il s'agit donc du pourcentage de la variance présente au niveau des classes, qui peut être interprétée comme le niveau de corrélation (de ressemblance) entre les élèves d'une même classe.

Dans notre cas, l'écart-type est de 3,542 au niveau des classes et de 7,734 au niveau des élèves. Nous pouvons donc calculer l'ICC au niveau des classes avec la formule précédente : $3,542 / (3,542 + 7,734) = 0,314$. Cela signifie que le niveau de corrélation entre deux élèves d'une même classe est de 0,314 ou encore que 31,4 % de la variance de Y se situe au niveau des classes, ce qui est conséquent. Une telle information ne peut être extraite d'un modèle avec uniquement des effets fixes. Notez ici que l'ICC peut être calculé pour chaque niveau d'un modèle à effet mixte. Dans notre exemple, nous n'avons qu'un seul niveau au-dessus des élèves, soit les classes, mais nous pourrions étendre cette logique à des écoles, par exemple. Notez également que cette formule de l'ICC n'est valide que pour un modèle pour lequel la distribution de la variable Y est normale. Des développements apparaissent pour proposer d'autres formulations adaptées à d'autres distributions, mais il est également possible d'estimer l'ICC à partir des simulations issues du modèle (Nakagawa, Johnson et Schielzeth 2017; Aly et al. 2014; Stryhn et al. 2006; Wu, Crespi et Wong 2012). L'idée générale reste d'expliquer la partition de la variance dans le modèle.

En plus de l'ICC, il est également possible de calculer les **R² marginal et conditionnel** du modèle. Le premier représente la variance expliquée par le modèle si seulement les effets fixes sont pris en compte, et le second si les effets fixes et aléatoires sont pris en compte. Distinguer les deux sources d'information permet de mieux cerner l'importance du rôle des écoles dans la performance des élèves. Dans notre cas, nous obtenons un R² marginal de 0,115 et un R² conditionnel de 0,269, ce qui nous confirme à nouveau que le rôle joué par la classe dans le niveau de performance est loin d'être négligeable.

9.2.2 GLMM avec pentes aléatoires

Dans cette seconde version du GLMM, nous n'envisageons plus de faire varier une constante en fonction des classes, mais un coefficient en fonction des classes. Admettons que nous voulons tester ici si l'effet du temps de travail (x_1) sur la performance scolaire (Y) n'est pas constant partout. En d'autres termes, nous supposons que, dans certaines classes, le temps de travail hebdomadaire en dehors de l'école est plus ou moins efficace que d'autres classes. L'idée sous-jacente est que nous n'observons pas de différence en termes de moyenne entre deux classes, mais en termes d'effet pour notre variable x_1 . À nouveau, nous pouvons nous contenter d'un effet fixe pour intégrer cette idée dans notre modèle. Pour cela, nous avons simplement à ajouter une interaction entre notre variable quantitative *temps de travail* et notre variable qualitative *classe*. Nous obtenons le résultat décrit par la figure 9.6. Notez ici que la constante est bien la même pour chaque classe (les lignes s'intersectent à 0 sur l'axe des x), et que seule la pente change.

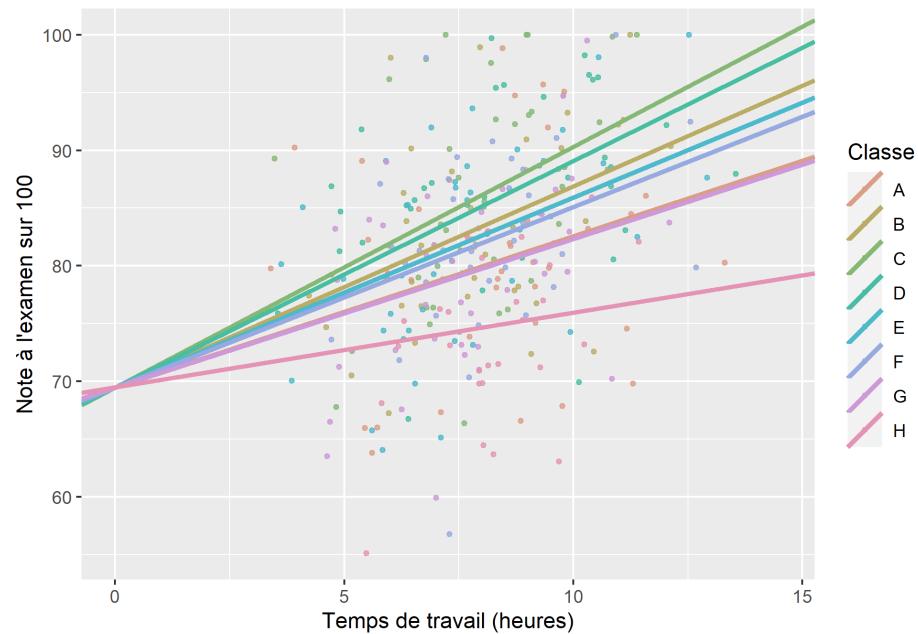


FIG. 9.6 : Influence du temps de travail sur la performance scolaire d'élèves en interaction avec la classe (effet fixe)

La formulation de ce modèle à effets fixes seulement est la suivante :

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma) \\
 g(\mu) &= \beta_0 + \beta_1 x_1 + \sum_{j=1}^k \beta_j x_{2j} x_1 \\
 g(x) &= x
 \end{aligned} \tag{9.4}$$

Nous constatons donc que nous avons un effet principal β_1 décrivant le lien entre le temps de travail et

la note obtenue à l'examen pour l'ensemble des élèves, ainsi qu'un bonus ou un malus sur cet effet β_j s'appliquant en fonction de la classe. Nous pouvons reformuler ce modèle pour inclure cet effet spécifique par classe comme un effet aléatoire :

$$\begin{aligned} Y &\sim \text{Normal}(\mu, \sigma_e) \\ g(\mu) &= \beta_0 + \beta_1 x_1 + \nu x_1 \\ \nu &\sim \text{Normal}(0, \sigma_\nu) \\ g(x) &= x \end{aligned} \tag{9.5}$$

Nous formulons ici un modèle dans lequel la classe modifie l'effet de la variable *temps d'étude* sur la variable *note à l'examen*. L'effet moyen de x_1 (propre aux individus) est capté par le coefficient β_1 , les bonus ou malus ajoutés à cet effet par la classe sont issus d'une distribution normale centrée sur 0 avec un écart-type, soit σ_ν . À nouveau, l'idée est que si nous moyennons l'effet de toutes les classes, nous obtenons 0. Aussi, le fait de modéliser cet effet comme un effet aléatoire nous permet de partitionner la variance, de mutualiser l'information entre les classes et de resserrer l'estimation des effets des classes.

Les résultats pour ce second modèle sont présentés à la figure 9.7, et une comparaison entre les estimations des effets fixes et des effets aléatoires est présentée à la figure 9.8. Nous pouvons ainsi constater à nouveau l'effet de resserrement provoqué par l'effet aléatoire.

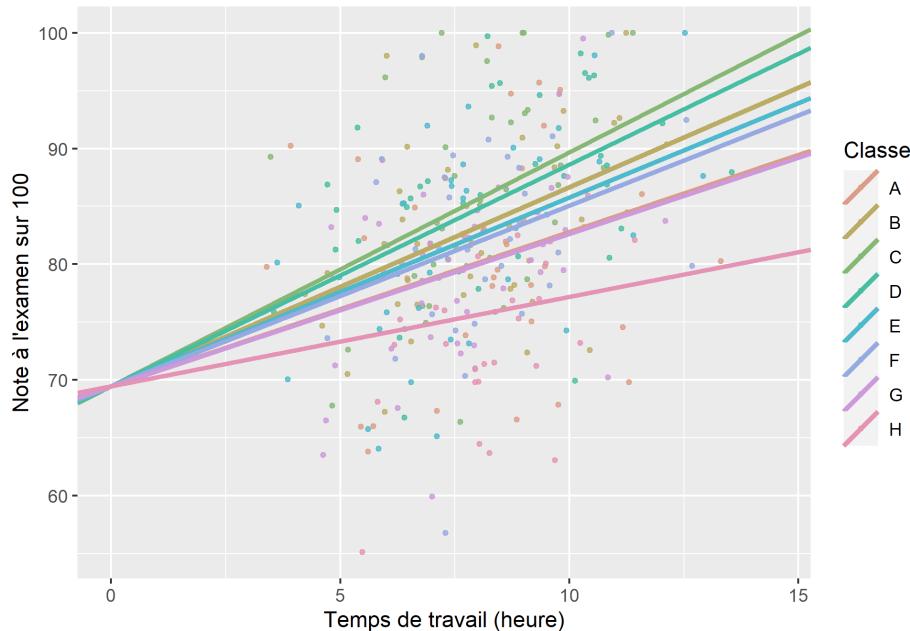


FIG. 9.7 : Influence du temps de travail sur la réussite scolaire d'élèves en interaction avec la classe (effet aléatoire)

Lorsque nous intégrons des pentes aléatoires dans un modèle, nous faisons face au problème suivant : la variance associée aux pentes aléatoires n'est pas fixe, mais proportionnelle à la variable X autorisée à varier. Si nous comparons la figure 9.4 (constantes aléatoires) et la figure 9.7 (pentes aléatoires), nous constatons bien que la dispersion des prédictions du modèle (représentées par les lignes) augmente dans le cas de pentes aléatoires et reste identique dans le cas des constantes aléatoires. La conséquence pratique est qu'il existe un nombre infini de valeurs possibles pour l'ICC. Dans ce contexte, il est préférable de laisser de côté cet indicateur et de ne reporter que les R^2 marginal et conditionnel. Dans notre cas, nous obtenons les valeurs 0,109 et 0,258, ce qui confirme une fois encore que le rôle joué par la classe est loin d'être négligeable.

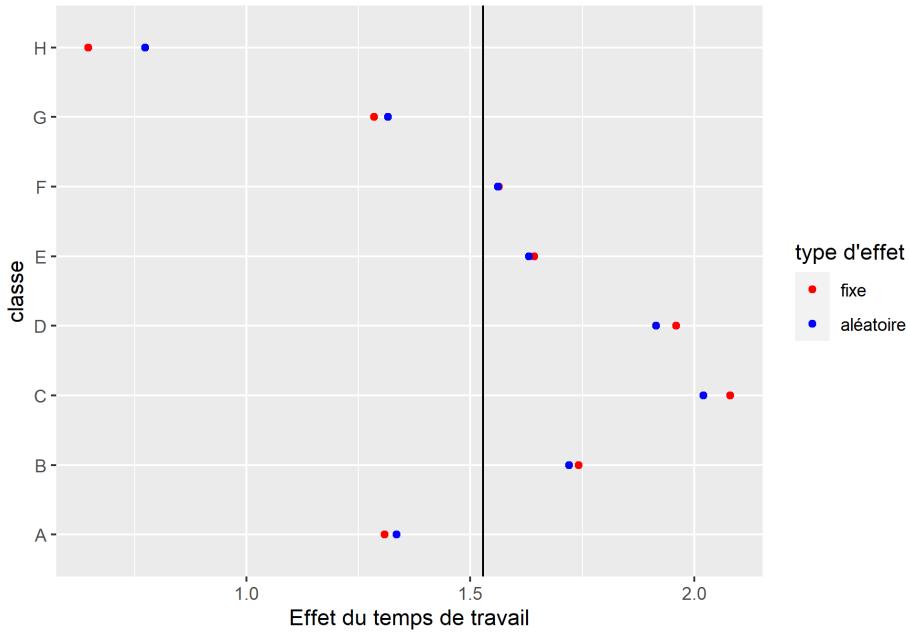


FIG. 9.8 : Influence du temps de travail sur la réussite scolaire d’élèves en interaction avec la classe (effet aléatoire)

9.2.3 GLMM avec constantes et pentes aléatoires

Vous l’aurez certainement deviné en lisant le titre de cette section : il est tout à fait possible de combiner à la fois des constantes et des pentes aléatoires dans un modèle. Cela augmente bien sûr la complexité du modèle et introduit quelques subtilités comme la notion de distribution normale multivariée, mais chaque chose en son temps.

Si nous reprenons notre exemple avec nos élèves et nos classes, combiner à la fois des constantes et des pentes aléatoires revient à formuler l’hypothèse que chaque classe a un effet sur la moyenne de la performance de ses élèves, mais également un effet sur l’efficacité du temps de travail. Il est possible de créer un modèle avec uniquement des effets fixes tenant compte de ces deux aspects en ajoutant dans le modèle la variable multinomiale *classe* ainsi que son interaction avec la variable *temps de travail*. La formulation de ce modèle à effets fixes est la suivante :

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma) \\
 g(\mu) &= \beta_0 + \beta_1 x_1 + \sum_{j=1}^k \beta_{2j} x_{2j} + \beta_{3j} x_{2j} x_1 \\
 g(x) &= x
 \end{aligned} \tag{9.6}$$

Nous pouvons représenter les résultats de ce modèle avec la figure 9.9.

Nous reformulons à présent ce modèle pour intégrer l’effet moyen de chaque classe (constante) et l’effet des classes sur l’efficacité du temps de travail (pente) comme deux effets aléatoires :

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma) \\
 g(\mu) &= \beta_0 + v_1 + (\beta_1 + v_2)x_1 \\
 \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} &\sim \mathcal{N} \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{v_1} & \sigma_{v_1 v_2} \\ \sigma_{v_1 v_2} & \sigma_{v_2} \end{pmatrix} \right) \\
 g(x) &= x
 \end{aligned} \tag{9.7}$$

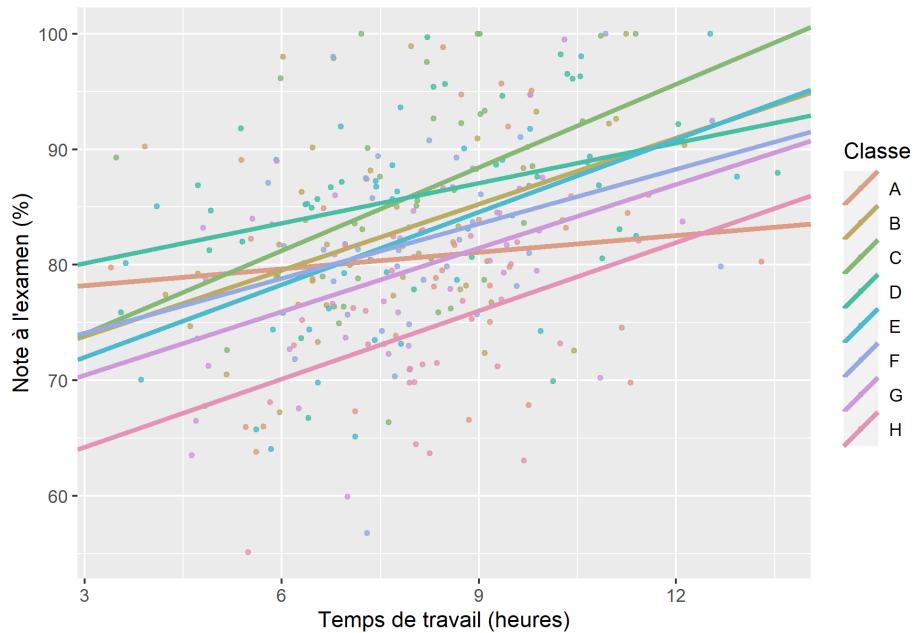


FIG. 9.9 : Influence du temps de travail sur la performance scolaire d'élèves en tenant compte de l'effet de leur classe et de l'effet de la classe sur l'efficacité du temps de travail (effet fixe)

Pas de panique ! Cette écriture peut être interprétée de la façon suivante :

Le modèle a deux effets aléatoires, l'un faisant varier la constante en fonction de la classe (v_1) et l'autre l'effet de la classe sur l'efficacité du temps de travail (v_2). Ces deux effets sont issus d'une distribution normale bivariée (une dimension par effet aléatoire). Cette distribution normale bivariée a donc deux moyennes et ces deux moyennes sont à 0 (les effets s'annulent si nous considérons toutes les classes ensemble). Elle dispose également d'une variance par effet aléatoire (σ_{v_1} et σ_{v_2}) et d'une covariance entre les deux effets aléatoires ($\sigma_{v_1 v_2}$). Cette covariance permet de tenir compte du fait que, potentiellement, les classes avec une constante plus élevée pourraient systématiquement avoir une efficacité du temps de travail plus faible ou plus élevée. Cette formulation implique donc d'ajuster trois paramètres de variance : σ_{v_1} , σ_{v_2} et $\sigma_{v_1 v_2}$. Il peut arriver que nous n'ayons pas assez de données pour estimer ces trois paramètres, ou que nous décidions, pour des raisons théoriques, qu'aucune corrélation ne soit attendue entre σ_{v_1} et σ_{v_2} . Dans ce cas, il est possible de fixer $\sigma_{v_1 v_2}$ à 0, ce qui revient à indiquer au modèle que v_1 et v_2 proviennent de deux distributions normales distinctes, nous pouvons donc écrire :

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma) \\
 g(\mu) &= \beta_0 + v_1 + (\beta_1 + v_2)x_1 \\
 \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{v_1} & 0 \\ 0 & \sigma_{v_2} \end{pmatrix}\right) \\
 g(x) &= x
 \end{aligned} \tag{9.8}$$

Ce qui est identique à :

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma) \\
 g(\mu) &= \beta_0 + v_1 + (\beta_1 + v_2)x_1 \\
 v_1 &\sim Normal(0, \sigma_{v_1}) \\
 v_2 &\sim Normal(0, \sigma_{v_2}) \\
 g(x) &= x
 \end{aligned} \tag{9.9}$$

Nous avons déjà abordé la notion de covariance dans la section 4.2. Pour rappel, la covariance dépend de l'unité de base des deux variables sur laquelle elle est calculée. Ici, il s'agit d'un coefficient et d'une constante. Il est donc préférable de la standardiser pour obtenir la corrélation entre les deux effets :

$$\text{corr}(v_1; v_2) = \frac{\sigma_{v_1 v_2}}{\sqrt{\sigma_{v_1}} \sqrt{\sigma_{v_2}}} \quad (9.10)$$

Si cette corrélation est positive, cela signifie que les classes ayant tendance à avoir un effet positif sur la performance scolaire ont également tendance à influencer positivement l'efficacité du temps de travail. À l'inverse, une corrélation négative signifie que l'efficacité du temps de travail a tendance à être plus faible dans les classes où la performance scolaire moyenne est élevée. Si la corrélation n'est pas significative, c'est que les deux effets sont indépendants l'un de l'autre.

Pour cet exemple, nous conservons la première formulation afin de montrer comment interpréter $\sigma_{v_1 v_2}$, mais nous ne disposons probablement pas de suffisamment de classes différentes pour estimer correctement ces trois paramètres. Les résultats de ce modèle sont représentés à la figure 9.10.

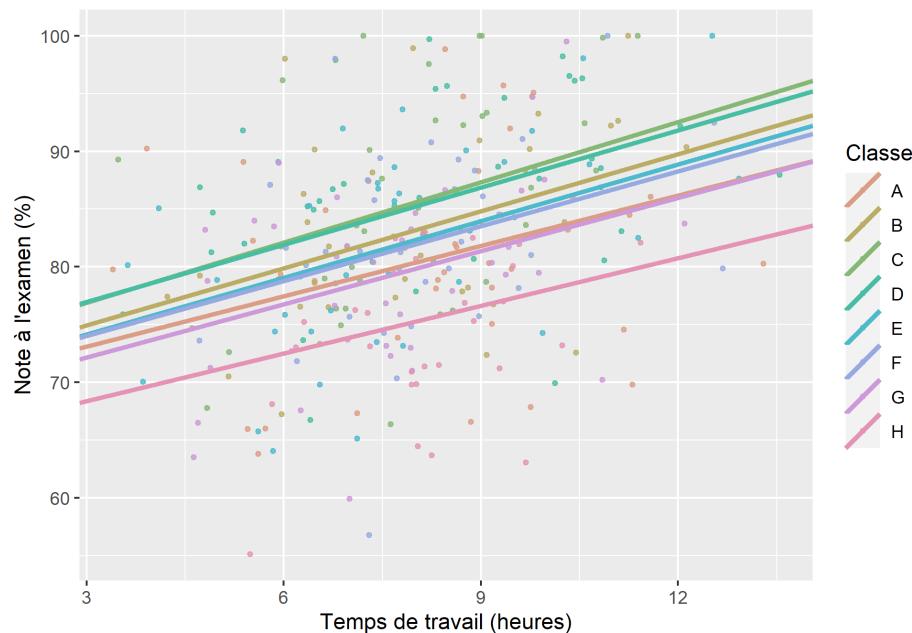


FIG. 9.10 : Influence du temps de travail sur la performance scolaire d'élèves en tenant compte de l'effet de leur classe et de l'effet de la classe sur l'efficacité du temps de travail (effet aléatoire)

Nous pouvons ainsi constater que pour ce troisième modèle, l'effet de resserrement est bien plus prononcé que pour les modèles précédents (figure 9.11). Si nous nous fions au modèle à effets fixes (figure 9.9), alors l'effet de l'école sur l'efficacité du temps de travail est très important. En revanche, le modèle à effet aléatoire identifie que la différence de moyenne entre les écoles est importante, mais la différence en termes d'efficacité du temps de travail est beaucoup plus anecdotique.

Notre modèle estime les valeurs de σ_{v_1} à 8,563, de σ_{v_2} à 0,042 et de $\sigma_{v_1 v_2}$ à 0,073. La corrélation entre les deux effets est donc de 0,122, ce qui est relativement faible (pour l'anecdote, notez que la valeur originale de corrélation entre ces deux effets était de 0,1 lorsque nous avons simulé ces données, notre modèle a donc bien été capable de retrouver le paramètre original). À nouveau, puisque nous avons des pentes aléatoires dans ce modèle, nous ne pouvons pas calculer l'ICC; nous pouvons cependant rapporter les R² marginal et conditionnel. Leurs valeurs respectives sont 0,115 et 0,269, ce qui nous confirme une nouvelle fois que l'ajout d'effets aléatoires contribue à expliquer une partie importante de la variance de la

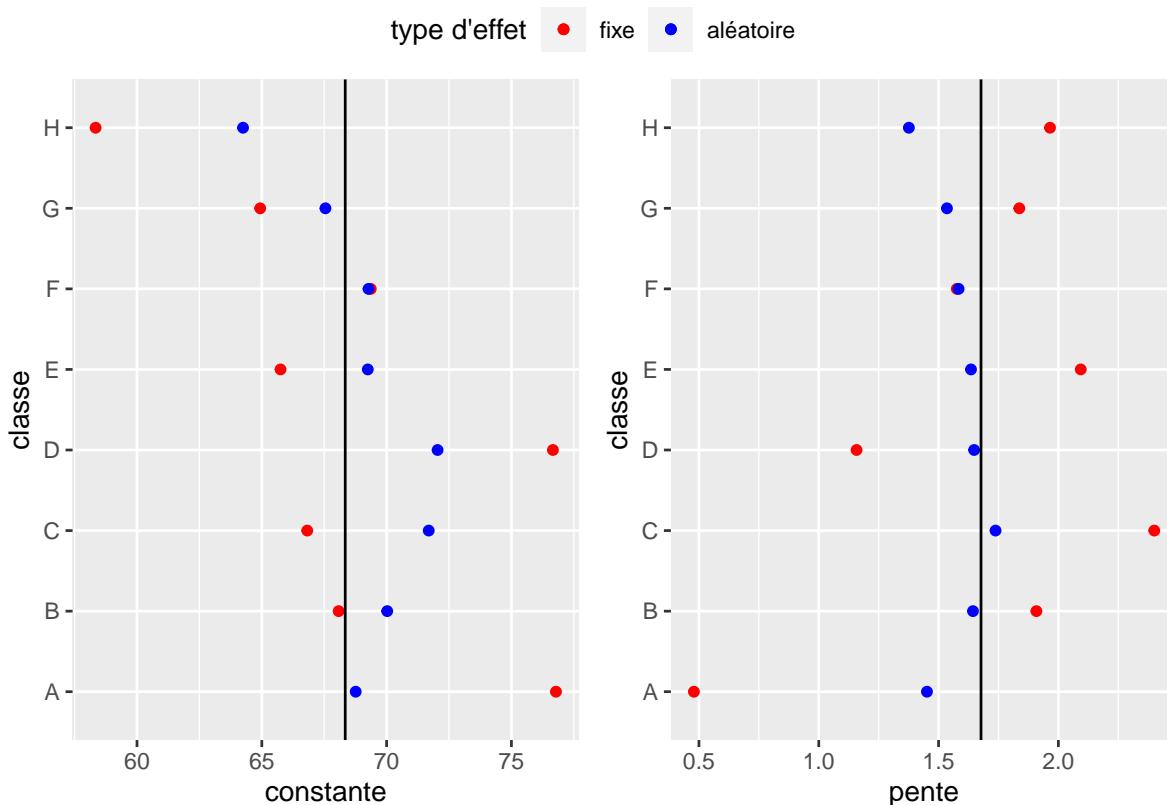


FIG. 9.11 : Comparaison des effets fixes et aléatoires pour le modèle intégrant l’effet des classes et l’interaction entre les classes et le temps de travail

performance scolaire.

Pour terminer cette section, comparons brièvement les trois modèles (constantes aléatoires, pentes aléatoires, constantes et pentes aléatoires) pour déterminer lequel est le mieux ajusté à nos données. Nous ajoutons également un quatrième modèle dans lequel les deux effets aléatoires sont présents, mais non corrélés ($\sigma_{v_1 v_2} = 0$). Le tableau 9.1 nous permet de constater que l’ajout des constantes aléatoires joue un rôle essentiel dans le premier modèle : le R^2 conditionnel est plus que deux fois supérieur au R^2 marginal. Cependant, l’ajout des pentes aléatoires dans les trois autres modèles apporte finalement très peu d’information, nous laissant penser que l’effet de la classe sur le temps de travail est faible, voire inexistant.



Modèles à effets mixtes avec des structures croisées

Jusqu’à présent, nous avons abordé des modèles GLMM comprenant des structures imbriquées (*nested* en anglais), c'est-à-dire qu'une observation d'un niveau 1 est incluse dans un et un seul groupe du niveau 2.

TAB. 9.1 : Comparaison des trois modèles à effets aléatoires

modèle	AIC	R2 marginal	R2 conditionnel
Constantes aléatoires	2 100,9	0,12	0,27
Pentes aléatoires	2 101,6	0,11	0,26
Pentes et constantes aléatoires corrélées	2 104,7	0,11	0,27
Pentes et constantes aléatoires non-correlées	2 102,7	0,11	0,27

Comme structure imbriquée à trois niveaux, nous avons vu comme exemple des élèves intégrés dans des classes elles-mêmes intégrées dans des écoles (figure 9.1) : un ou une élève appartient à une et une seule classe qui est elle-même localisée dans une et une seule école (élève / classe / école).

Notez qu'il est aussi possible d'avoir des structures des données croisées (*crossed*).

Admettons à présent que nous ne nous intéressons pas à la classe dans laquelle se situe l'élève, mais plutôt à la personne qui enseigne. Admettons également que ces personnes peuvent donner des cours dans plusieurs écoles. Nous nous retrouvons dans un cas de figure où une personne qui enseigne peut se situer dans plusieurs écoles, ce qui diffère du cas précédent où chaque classe appartient à une seule école. Dans ce second cas, on parle d'une **structure croisée** plutôt qu'imbriquée.

Si les personnes enseignent dans toutes les écoles, il est possible de dire que le design d'étude est croisé complet ou croisé partiel si elles n'enseignent que dans certaines écoles. La figure 9.12 résume ces trois situations.

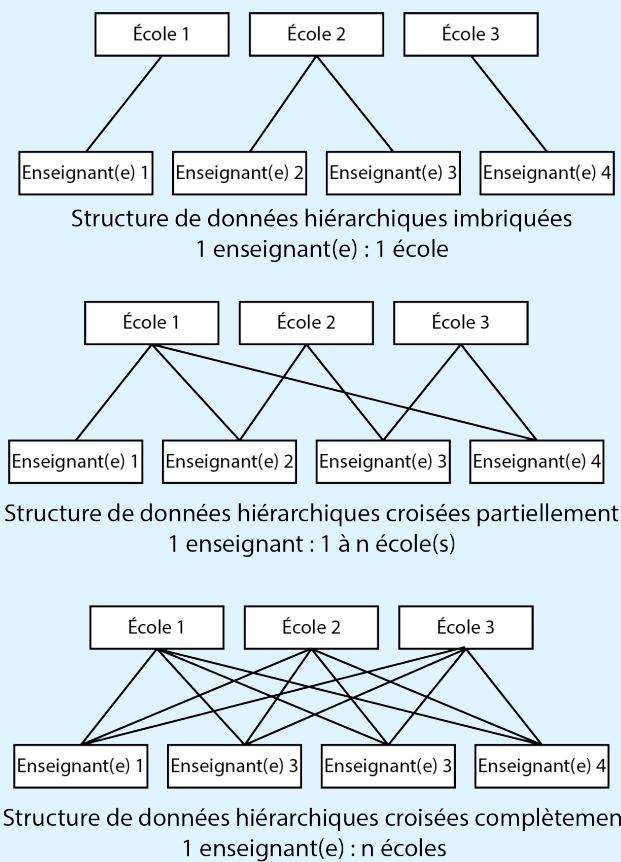


FIG. 9.12 : Différentes structures de données hiérarchiques (imbriquée versus croisée)

Il est important de bien saisir la structure de son jeu de données, car l'estimation d'un modèle avec effets imbriqués ou croisés peut donner des résultats parfois significativement différents. De plus, un modèle imbriqué est généralement moins difficile à ajuster qu'un modèle croisé. En effet, dans un modèle imbriqué, deux personnes étudiant dans deux écoles différentes sont jugées indépendantes. Dans un modèle croisé, deux élèves provenant de deux écoles différentes peuvent tout de même partager une dépendance du fait qu'ils ou elles ont pu avoir le même professeur ou la même professeure. La structure de dépendance (et donc de la matrice de covariance des effets aléatoires) est ainsi plus complexe pour un modèle croisé.

9.3 Conditions d'application des GLMM

Puisque les GLMM sont une extension des GLM, ils partagent l'essentiel des conditions d'application de ces derniers. Pour simplifier, si vous ajustez un modèle GLMM avec une distribution Gamma, vous devez réaliser les mêmes tests que ceux pour un simple GLM avec une distribution Gamma.

Une question importante se pose souvent lorsque nous ajustons des modèles GLMM : **combien de groupes faut-il au minimum aux différents niveaux?** En effet, pour estimer les différentes variances, nous devons disposer de suffisamment de groupes différents. Dans le cas d'un modèle avec uniquement une constante aléatoire, il est fréquent de lire que nous devons disposer au minimum de cinq groupes différents (Gelman et Hill 2006), en dessous de ce minimum, traiter l'effet comme aléatoire plutôt que fixe apporte très peu d'information. De plus, l'estimation des variances pour chaque niveau est très imprécise, donnant potentiellement des valeurs inexactes pour l'ICC et polluant l'interprétation. Avec cinq groupes ou moins, il est certainement plus judicieux d'ajuster seulement un effet fixe. Dans un modèle avec plusieurs effets aléatoires et plusieurs variances / covariances à estimer, ce nombre doit être augmenté proportionnellement, à moins que les effets aléatoires ne soient estimés indépendamment les uns des autres. Notez ici que, si l'enjeu du modèle était d'estimer avec une grande précision les paramètres de variances, il faudrait compter au minimum une centaine de groupes. Il n'est pas nécessaire d'avoir le même nombre d'observations par groupe, car les modèles GLMM partagent l'information entre les groupes. Cependant, dans les groupes avec peu d'observations (inférieur à 15), l'estimation de leur effet propre (BLUP) est très incertaine.

Puisque les GLMM font intervenir la distribution normale aux niveaux supérieurs du modèle, il est nécessaire de vérifier si les hypothèses qu'elle implique sont respectées. Il s'agit essentiellement de deux hypothèses : les effets aléatoires suivent bien une distribution normale (univariée ou multivariée), et la variance au sein des groupes est bien homogène.

9.3.1 Vérification de la distribution des effets aléatoires

Reprendons la formulation d'un modèle simple avec seulement deux niveaux et seulement une constante aléatoire :

$$\begin{aligned} Y &\sim \text{Normal}(\mu, \sigma_e) \\ g(\mu) &= \beta_0 + \beta_1 x_1 + v \\ v &\sim \text{Normal}(0, \sigma_v) \\ g(x) &= x \end{aligned} \tag{9.11}$$

Ce modèle formule l'hypothèse que les constantes aléatoires v proviennent d'une distribution normale avec une moyenne de 0 et un écart-type σ_v . La première étape du diagnostic est donc de vérifier si les constantes aléatoires suivent bien une distribution normale, ce que nous pouvons faire habituellement avec un diagramme quantile-quantile. Si nous reprenons notre exemple avec nos données de performance scolaire des sections précédentes, nous obtenons la figure 9.13. Puisque les points tombent bien approximativement sur la ligne rouge, nous pouvons conclure que cette condition d'application est bien respectée. Notez qu'il est également possible d'utiliser ici un des tests vus dans le chapitre 2 pour tester formellement la distribution des constantes aléatoires, mais nous disposons rarement de suffisamment de valeurs différentes pour qu'un tel test soit pertinent.

Cette vérification est bien sûr à appliquer à chacun des niveaux (en dehors du niveau de base) du modèle étudié.

Si nous nous intéressons maintenant au modèle avec constantes et pentes aléatoires, nous avons deux cas de figure :

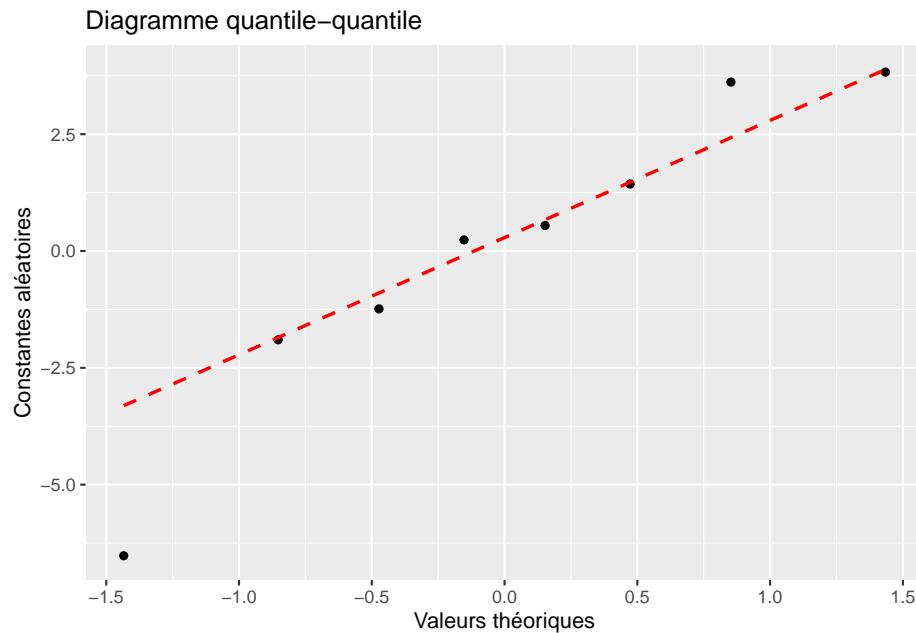


FIG. 9.13 : Distribution normale univariée des constantes aléatoires

- notre modèle inclut une covariance entre les constantes et les pentes ; elles proviennent donc d'une distribution normale bivariée.
- notre modèle considère les pentes et les constantes comme indépendantes ; elles proviennent donc de deux distributions normales distinctes.

Le second cas est de loin le plus simple puisqu'il nous suffit de réaliser un graphique de type quantile-quantile pour les deux effets aléatoires séparément. Dans le premier cas, il nous faut adapter notre stratégie pour vérifier si les deux effets aléatoires suivent conjointement une distribution normale multivariée. Pour cela, nous devons, dans un premier temps, observer séparément la distribution des pentes et des constantes, puisque chaque variable provenant d'une distribution normale multivariée suit elle-même une distribution normale univariée (Burdenski 2000). Nous pouvons, dans un second temps, construire un graphique nous permettant de juger si nos pentes et nos constantes suivent bien la distribution normale bivariée attendue par le modèle. Pour l'illustrer, nous reprenons le modèle sur la performance scolaire intégrant des pentes et des constantes aléatoires avec une covariance estimée entre les deux.

La figure 9.14 représente donc les deux graphiques quantile-quantile univariés. Les deux semblent indiquer que nos effets aléatoires suivent bien chacun une distribution normale. La figure 9.15 montre la distribution normale bivariée attendue par le modèle avec des ellipses représentant différents pourcentiles de cette distribution. Les valeurs des effets aléatoires sont représentées par des points noirs. Seulement 5 % des points noirs devraient se trouver dans la première ellipse et 95 % des points devraient se trouver dans la quatrième ellipse. En revanche, seulement 20 % des points devraient se trouver dans le dernier anneau et seulement 5 % des points en dehors de cet anneau. Il faut donc évaluer si les points sont plus ou moins centrés que ce que nous attendons. Pour simplifier la lecture, il est possible de rajouter des points grisés en arrière-plan représentant des réalisations possibles de cette distribution normale bivariée. Les vrais points noirs devraient avoir une dispersion similaire à celle des points grisés. Dans notre cas, ils semblent suivre un patron cohérent avec notre distribution normale bivariée. Dans le cas contraire, cela signifierait que le modèle doit être révisé.

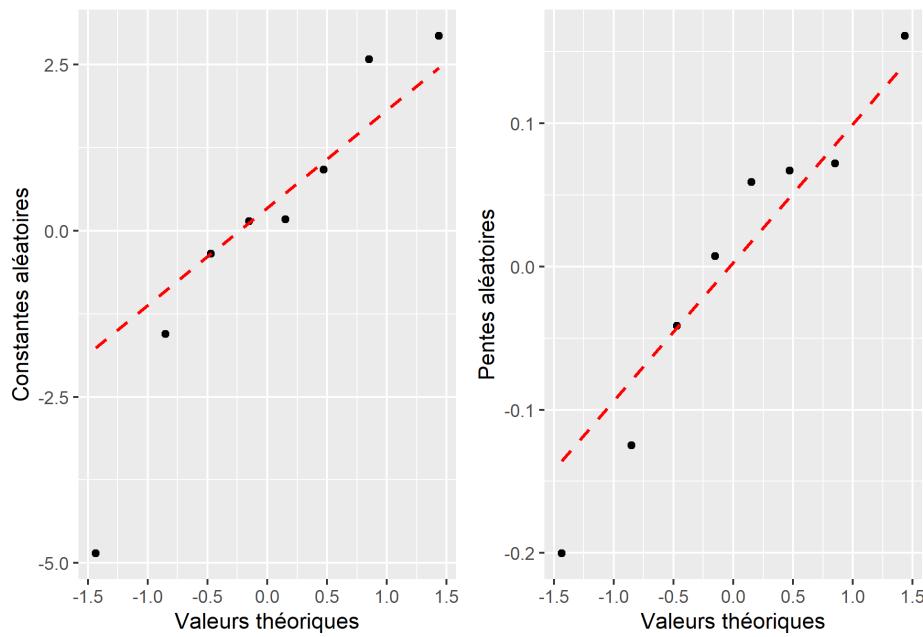


FIG. 9.14 : Multiples distributions normales univariées des constantes et pentes aléatoires

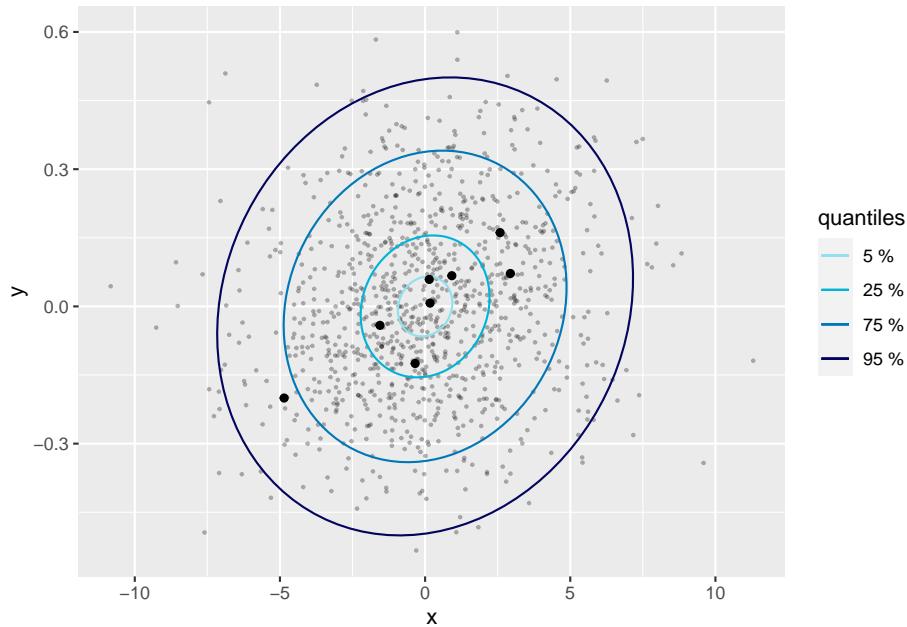


FIG. 9.15 : Distribution normale bivariée des constantes et des pentes aléatoires

9.3.2 Homogénéité des variances au sein des groupes

Dans le chapitre 8 sur les GLM, nous avons vu que chaque distribution a sa propre définition de la variance. Pour rappel, un modèle gaussien assume une variance constante, un modèle de Poisson assume une variance égale à son espérance, alors qu'un modèle Gamma assume une variance proportionnelle au carré de son espérance divisée par un paramètre de forme, etc. Nous devions donc, pour chaque GLM, vérifier graphiquement si la variance présente dans les données originales était proche de la variance

attendue par le modèle. Dans un modèle GLMM, le même exercice doit être fait pour chaque groupe aux différents niveaux du modèle.

Dans notre exemple sur la performance scolaire, notre variable Y a été modélisée avec une distribution normale. Le modèle assume donc une uniformité de sa variance (homoscédasticité). La figure 9.16 nous montre ainsi que, qu'elle que soit la classe, la dispersion des points (notes des élèves) semble bien respecter la variance attendue par le modèle (représentée par les lignes noires).

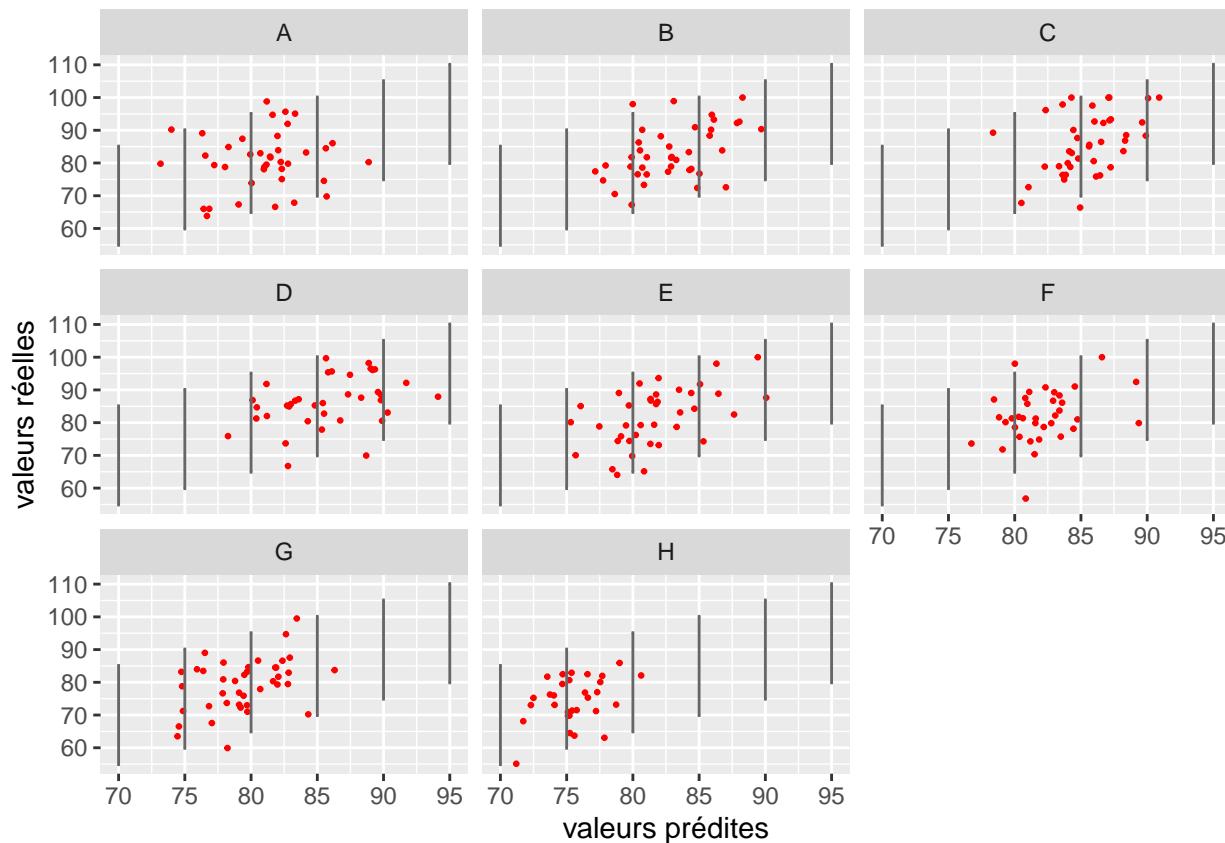


FIG. 9.16 : Homogénéité de la variance pour les différents groupes d'un modèle GLMM gaussien

9.4 Inférence dans les modèles GLMM

Une des questions importantes à se poser lorsque nous construisons un modèle est toujours : est-ce que les différents effets présents dans le modèle ont un effet significativement différent de zéro sur la variable dépendante ? Cette étape d'inférence est plus compliquée pour les modèles GLMM que dans les modèles GLM à cause de la présence d'effets aléatoires. Ces derniers brouillent le comptage du nombre de paramètres et, par extension, du nombre de degrés de liberté des modèles. Pour un effet aléatoire, il est possible de déterminer que le nombre de degrés de liberté est de 1 puisque nous ajustons un seul paramètre supplémentaire (la variance de cet effet aléatoire). Selon un autre point de vue, il serait possible d'affirmer que le nombre de degrés de liberté est de $k - 1$ (avec k le nombre de groupes dans cet effet aléatoire), ce que nous utilisons habituellement pour un effet fixe. La vraie valeur du nombre de degrés de liberté se situe quelque part entre ces deux extrêmes. L'enjeu du nombre de degrés de liberté est crucial, car il influence directement l'estimation des valeurs de p pour l'ensemble des coefficients du modèle. Avec un nombre de degrés de liberté plus petit, les valeurs de p sont plus faibles et les effets plus significatifs. Le sujet est d'ailleurs l'objet d'une telle controverse que les auteurs de certains packages comme `lme4`

(un des *packages* les plus utilisés pour ajuster des GLMM) ont fait le choix de ne renvoyer aucune valeur de p dans les résultats des modèles. L'article de Bolker et al. (2009) propose une explication détaillée et relativement accessible du problème (en plus d'une excellente introduction aux GLMM) : en se basant sur leurs recommandations, il est possible de séparer le problème de l'inférence dans les GLMM en trois sous problèmes :

- Quel est le degré de significativité des effets fixes ?
- Quel est le degré de significativité de l'effet aléatoire dans le modèle ?
- Quels sont les degrés de significativité de chaque constante / pente aléatoire ?

9.4.1 Inférence pour les effets fixes

Trois approches peuvent être envisagées pour déterminer si un effet fixe est significatif ou non. Elles font appel à trois approches théoriques différentes (test classique, comparaison de modèles et *bootstrapping*) et peuvent donc donner des résultats différents. À titre exploratoire, il peut être intéressant de toutes les tester, mais certaines peuvent être préférées en fonction de votre champ de recherche.

9.4.1.1 Test classique

Nous avons vu, pour les modèles LM et GLM, que les valeurs de p sont calculées à partir de scores obtenus en divisant les coefficients par leurs erreurs standards. Une approche similaire peut être utilisée pour les modèles GLMM. Cependant, la question du nombre de degrés de liberté à utiliser reste un problème. L'approche la plus flexible est certainement l'approximation par la méthode Satterthwaite proposant une estimation de ce nombre de degrés de liberté et, par extension, des valeurs de p .

9.4.1.2 Rapports de vraisemblance

Si le modèle comprend suffisamment d'observations (par suffisamment, comprenez au moins une centaine d'observations par paramètre), il est également possible d'utiliser une série de tests de rapports de vraisemblance pour vérifier si l'apport de chaque variable indépendante contribue à améliorer significativement le modèle. Cette approche correspond à une analyse de type 3, comme nous l'avons mentionné dans la section 8.2.4 pour le modèle logistique multinomial.

9.4.1.3 Bootstrapping

L'approche par *bootstrapping* (*parametric-bootstrap* ou *semi-parametric-bootstrap*) permet de calculer, pour les différents paramètres d'un modèle, un intervalle de confiance. L'idée étant de réajuster un grand nombre de fois le modèle sur des sous-échantillons de données pour saisir la variabilité des différents paramètres du modèle. Si les intervalles de confiance ainsi construits ne comprennent pas de zéro, il est possible de dire que cet effet est significatif. À nouveau, cette méthode n'est valide que si le jeu de données comporte suffisamment d'observations. L'intérêt de cette approche est qu'elle ne postule pas d'hypothèse sur la distribution des paramètres qui ont la fâcheuse tendance à ne pas suivre une distribution normale dans le cas des GLMM. Elle est d'ailleurs considérée comme la plus robuste bien que coûteuse en termes de temps de calcul.

9.4.2 Inférence pour les effets aléatoires, effet global

Pour déterminer si un effet aléatoire est significatif dans un modèle, il est recommandé d'utiliser un test de rapport de vraisemblance entre un modèle sans l'effet aléatoire et un modèle avec l'effet aléatoire. L'analyse des différences entre les valeurs de déviance, l'AIC et le BIC peut également aider à déterminer si l'ajout de l'effet aléatoire est justifié. Il est également possible de considérer les valeurs de l'ICC

et du R^2 conditionnel. Notez ici que si vous avez une très bonne raison théorique d'ajouter l'effet aléatoire dans votre modèle et suffisamment d'observations / groupes pour l'ajuster, il peut être pertinent de laisser l'effet aléatoire dans le modèle même si tous les indicateurs mentionnés précédemment indiquent qu'il contribue faiblement au modèle. Le retirer risquerait en effet de donner l'impression que les autres paramètres du modèle sont plus significatifs qu'ils ne le sont en réalité.

Notez que l'approche par *bootstrapping* décrite pour les effets fixes peut aussi être utilisée ici pour obtenir un intervalle de confiance pour l'ICC, le R^2 conditionnel et les différents paramètres de variance et covariance.

9.4.3 Inférence pour les effets aléatoires, des constantes et des pentes

Pour rappel, dans l'approche fréquentiste présentée ici, les valeurs des constantes et des pentes aléatoires ne sont pas à proprement parler des paramètres du modèle : elles sont estimées à posteriori (BLUP). Pour déterminer si ces constantes et des pentes sont significativement différentes de zéro et significativement différentes les unes des autres, il est possible de calculer les intervalles de confiance de chacune d'entre elles par *bootstrap*, par profilage ou par simulation à partir du modèle. Si la constante du groupe j a zéro dans son intervalle de confiance, nous pouvons alors déclarer que le groupe j en question ne semble pas varier du reste de la population en termes de moyenne. Si la pente l du groupe j a zéro dans son intervalle de confiance, nous pouvons alors déclarer que le groupe j en question ne semble pas varier du reste de la population pour l'effet l . Notez que la méthode par simulation est bien plus rapide que les deux autres, mais que l'approche par *bootstrapping* reste la plus fiable.

9.5 Conclusion sur les GLMM

Les GLMM sont donc une extension des GLM offrant une grande flexibilité de modélisation (variabilité des pentes et des constantes en fonction de groupes) et nous permettant d'analyser la partition de la variance entre plusieurs niveaux de nos données. Cependant, cette flexibilité implique des modèles plus complexes avec un travail de diagnostic et d'interprétation plus long et potentiellement plus ardu.

9.6 Mise en œuvre des GLMM dans R

Pour cet exemple de GLMM, nous proposons d'analyser à nouveau les données présentées dans la section 6.2.1.1 sur le modèle logistique binomial. Pour rappel, nous modélisons la probabilité qu'un individu utilise le vélo comme mode de transport pour son trajet le plus fréquent en utilisant une enquête réalisée auprès d'environ 26 000 Européens. Initialement, nous avons intégré les pays comme un effet fixe. Or, nous savons à présent qu'il serait plus judicieux de les traiter comme un effet aléatoire. Nous comparons deux modèles, un pour lequel seulement la constante varie par pays et un second dans lequel la pente pour l'âge varie également par pays. L'hypothèse étant que l'effet de l'âge sur l'utilisation du vélo pourrait être réduit dans certains pays où la culture du vélo est plus présente. Cette hypothèse implique également la présence potentielle d'une corrélation inverse entre la constante et la pente de chaque pays : dans un pays où la probabilité de base d'utiliser le vélo est plus élevée, l'effet de l'effet de l'âge est probablement réduit.

Pour ajuster ces modèles, nous utilisons le package `lme4`, permettant d'ajuster des modèles GLMM avec des distributions gaussienne, Gamma, de Poisson et binomial. Lorsque d'autres distributions sont nécessaires, il est possible de se tourner vers le package `gamlss`. Notez cependant que les effets aléatoires de `gamlss` sont estimés avec une méthode appelée PQL très flexible, mais qui peut produire des résultats erronés dans certains cas (Bolker et al. 2009).

Afin de limiter les répétitions, nous ne recalculons pas ici le VIF et nous excluons d'emblée les observations aberrantes (provenant de Malte ou de Chypre ou avec des temps de trajets supérieurs à 400 minutes).

9.6.1 Ajustement du modèle avec uniquement une constante aléatoire

Nous commençons donc par ajuster un premier modèle avec une constante aléatoire en fonction du pays. Dans la plupart des *packages* intégrant des effets aléatoires, la syntaxe suivante est utilisée pour stipuler une constante aléatoire : $+ (1 | \text{Pays})$. Concrètement, nous tentons d'ajuster le modèle décrit par l'équation (9.12).

$$\begin{aligned} Y &\sim \text{Binomial}(p) \\ g(p) &= \beta_0 + \beta_1 x_1 + v \\ v &\sim \text{Normal}(0, \sigma_v) \\ g(x) &= \log\left(\frac{x}{1-x}\right) \end{aligned} \tag{9.12}$$

Il s'agit simplement d'un modèle logistique binomial dans lequel nous avons ajouté une constante aléatoire : v . Dans notre cas, elle varie avec la variable Pays. La syntaxe dans R pour produire ce modèle est la suivante.

```
# Chargement des données
dfenquete <- read.csv("data/glm/enquete_transport_UE.csv", encoding = "UTF-8")
dfenquete$Pays <- relevel(as.factor(dfenquete$Pays), ref = "Allemagne")
# Retirer les observations aberrantes
dfenquete2 <- subset(dfenquete, (dfenquete$Pays %in% c("Malte", "Chypre")) == F &
                      dfenquete$Duree < 400)

# Ajustement du modèle
library(lme4)
# Nécessité ici de centrer et réduire ces variables pour permettre au modèle de converger
dfenquete2$Age2 <- scale(dfenquete2$Age, center = T, scale = T)
dfenquete2$Duree2 <- scale(dfenquete2$Duree, center = T, scale = T)
modele1 <- glmer(y ~ Sexe + Age2 + Education + StatutEmploi + Revenu +
                  Residence + Duree2 + ConsEnv + (1|Pays),
                  family = binomial(link="logit"),
                  control = glmerControl(optimizer = "bobyqa"),
                  data = dfenquete2)
```

Nous nous concentrons ici sur l'interprétation des résultats du modèle et nous réalisons l'ensemble des diagnostics dans une section dédiée en fin de chapitre. Notez cependant que le diagnostic **devrait précéder l'interprétation** comme nous l'avons vu dans le chapitre sur les modèles GLM.

Vous constaterez que nous avons centré-réduit les variables Age et Duree. Il est souvent nécessaire de réaliser cette étape en amont pour s'assurer que le modèle converge sans trop de difficulté. Dans notre cas, si ces deux variables sont laissées dans leur échelle d'origine, la fonction `glmer` ne parvient pas à ajuster le modèle. Notez que cette transformation nous permet d'obtenir les coefficients standardisés, s'exprimant alors en écarts-types. La fonction `summary` nous donne accès à un premier ensemble d'informations.

```
summary(modele1)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
```

```

## Family: binomial ( logit )
## Formula: y ~ Sexe + Age2 + Education + StatutEmploi + Revenu + Residence +
##           Duree2 + ConsEnv + (1 | Pays)
## Data: dfenquete2
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC  logLik deviance df.resid
## 19176.1 19322.8 -9570.1 19140.1    25529
##
## Scaled residuals:
##      Min     1Q Median     3Q    Max
## -1.1989 -0.4418 -0.3212 -0.2134  7.2461
##
## Random effects:
## Groups Name        Variance Std.Dev.
## Pays   (Intercept) 0.5949   0.7713
## Number of obs: 25547, groups: Pays, 26
##
## Fixed effects:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -3.368942  0.212577 -15.848 < 2e-16 ***
## Sexehomme                  0.370864  0.037921   9.780 < 2e-16 ***
## Age2                        -0.102388  0.018777  -5.453 4.95e-08 ***
## Educationsecondaire         0.188096  0.103204   1.823  0.06837 .
## Educationsecondaire inferieur 0.297591  0.111352   2.673  0.00753 **
## Educationuniversite        0.138670  0.106518   1.302  0.19297
## StatutEmploisans emploi     0.256476  0.042284   6.066 1.31e-09 ***
## Revenufaible                 0.073836  0.071653   1.030  0.30279
## Revenumoyen                  0.039404  0.065243   0.604  0.54587
## Revenusans reponse          0.215706  0.102308   2.108  0.03500 *
## Revenutres eleve            -0.121285  0.185352  -0.654  0.51288
## Revenutres faible           0.237388  0.085712   2.770  0.00561 **
## Residencegrande ville       0.272277  0.069280   3.930 8.49e-05 ***
## Residencepetite-moyenne ville 0.276282  0.061496   4.493 7.03e-06 ***
## Residencezone rurale        -0.118967  0.069096  -1.722  0.08511 .
## Duree2                      -0.018718  0.019241  -0.973  0.33065
## ConsEnv                      0.101757  0.009277  10.969 < 2e-16 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

La première partie de ce résumé nous rappelle la formule utilisée pour le modèle et nous indique différents indicateurs de qualité d'ajustement comme l'AIC, le BIC et la déviance. Nous avons ensuite une partie dédiée aux effets aléatoires (`Random Effects`) et une partie dédiée aux effets fixes (`Fixed effects`). Cette dernière s'interprète de la même manière que pour un modèle à effets fixes, n'oubliez cependant pas d'utiliser la fonction exponentielle pour obtenir les rapports de cotes (fonction de lien logistique).

9.6.1.1 Rôle joué par l'effet aléatoire

Comme vous pouvez le constater, la section `Random Effects` ne comprend qu'un seul paramètre : la variance de l'effet pays. Nous pouvons ainsi écrire que l'effet du pays suit une distribution normale avec

une moyenne de 0 et une variance σ^2 de 0,595. Pour aller plus loin dans cette analyse, nous pouvons calculer le coefficient de corrélation intraclasse (ICC). Cependant, puisque notre modèle est binomial et non gaussien, nous ne disposons pas d'un paramètre de variance au niveau des individus, il est donc possible, à la place, d'utiliser la variance théorique du modèle : $\frac{\pi^2}{3}$. Nous calculons ainsi notre ICC :

```
# Extraction de la variance des Pays
var_pays <- VarCorr(modele1)[[1]][[1]]
# Calcul de l'ICC
var_pays / (((pi**2)/3) + var_pays)

## [1] 0.153141
```

Nous pouvons parvenir au même résultat en utilisant la fonction `icc` du *package* `performance`.

```
library(performance)
# Calcul de l'ICC
icc(modele1)

## # IntraClass Correlation Coefficient
##
##      Adjusted ICC: 0.153
##      Conditional ICC: 0.148
```

Notez que cette fonction distingue un ICC ajusté et un ICC conditionnel. Le premier correspond à l'ICC que nous avons présenté jusqu'ici et que nous avons calculé à la main. L'ICC conditionnel inclut dans son estimation la variance présente dans les effets fixes. Un fort écart entre ces deux ICC indique que les effets fixes sont capables de capturer une très forte variance dans les données, ce qui pourrait remettre en cause la pertinence de l'effet aléatoire. Dans notre cas, la différence entre les deux est très faible.

En plus du ICC, nous pouvons calculer les R^2 marginal et conditionnel. Pour cela, nous utilisons la fonction `r.squaredGLMM` du *package* `MuMin`.

```
library(MuMin)
r.squaredGLMM(modele1)

##          R2m       R2c
## theoretical 0.03492136 0.18271447
## delta      0.01554518 0.08133502
```

Cette fonction nous renvoie à la fois les R^2 obtenus en utilisant la variance théorique du modèle ($\frac{\pi^2}{3}$ dans notre cas) et la variance estimée par la méthode delta. La seconde est plus conservative, mais les deux résultats indiquent que les effets aléatoires expliquent une part importante de la variance comparativement aux effets fixes. Notez également que la fonction `r2` du *package* `performance` peut calculer ces deux R^2 , mais seulement en utilisant la variance théorique.

9.6.1.2 Significativité de l'effet aléatoire

Nous souhaitons déterminer ici si notre effet aléatoire contribue à significativement améliorer le modèle. Pour cela, nous effectuons un test de rapport de vraisemblance entre le modèle sans l'effet aléatoire (un simple GLM ici) et le modèle complet. Nous utilisons pour cela la fonction `anova` :

```

# Ajustement d'un modèle sans l'effet aléatoire
model_simple <- glm(y ~ Sexe + Age2 + Education + StatutEmploi + Revenu +
  Residence + Duree2 + ConsEnv,
  family = binomial(link="logit"),
  data = dfenquete2)
# Comparaison des deux modèles
anova(modele1,model_simple)

## Data: dfenquete2
## Models:
## model_simple: y ~ Sexe + Age2 + Education + StatutEmploi + Revenu + Residence +
## model_simple: Duree2 + ConsEnv
## modele1: y ~ Sexe + Age2 + Education + StatutEmploi + Revenu + Residence +
## modele1: Duree2 + ConsEnv + (1 | Pays)
##          npar   AIC    BIC   logLik deviance Chisq Df Pr(>Chisq)
## model_simple 17 20521 20660 -10243.6     20487
## modele1      18 19176 19323 -9570.1     19140  1347  1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Le test indique clairement que le modèle complet est mieux ajusté : les valeurs de l'AIC, du BIC et de la déviance sont toutes grandement réduites et le test est largement significatif.

Pour aller plus loin, nous pouvons utiliser une approche par *bootstrap* pour calculer un intervalle de confiance pour la variance de l'effet aléatoire, l'ICC et le R² conditionnel. Nous utilisons pour cela la fonction **bootMer**. Si vous essayez de lancer cette syntaxe, vous constaterez qu'elle prend énormément de temps, ce qui s'explique par le grand nombre de fois où le modèle doit être réajusté. Nous vous recommandons donc de bien enregistrer vos résultats après l'exécution de la fonction avec la fonction **save**. Notez que pour réduire significativement le temps de calcul, il est possible d'utiliser simultanément plusieurs coeurs de votre processeur, ce que nous faisons ici avec le package **snow**.

```

# Définition d'une fonction pour extraire les valeurs qui nous intéressent
extractor <- function(mod){
  vari <- VarCorr(mod)[[1]][[1]]
  ICC <- vari / (vari + (pi**2/3))
  r2cond <- performance::r2(mod)[[1]]
  return(c("vari"=vari,"icc"=ICC,"r2cond"=r2cond))
}

# Préparation d'un environnement multitraiteme pour accélérer le calcul
library(snow)
# Préparation de huit coeurs (attention si votre machine en a moins!)
cl <- makeCluster(8)
clusterEvalQ(cl,library("lme4"))
valeurs <- bootMer(modele1,FUN = extractor,nsim = 1000,
  use.u = F, type="parametric", ncpus = 8,
  parallel="snow",
  cl = cl)
# Sauvegarde des résultats
save(valeurs,file = 'data/glmm/boot_binom.rda')

```

Nous pouvons à présent analyser l'incertitude de ces différents paramètres. Pour cela, nous devons commencer par observer graphiquement leurs distributions obtenues par *bootstrap* avec la figure 9.17.

```
# Chargement de nos valeurs préalablement enregistrées
load('data/glmm/boot_binom.rda')
# Construction de trois graphiques de distribution
df <- data.frame(valeurs$t)
names(df) <- c("variance", "icc", "R2cond")
breaks1 <- as.vector(quantile(df$variance, probs = c(0.001, 0.15, 0.5, 0.85, 0.999)))
labs1 <- round(breaks1, 2)
p1 <- ggplot(df) +
  geom_histogram(aes(x = variance), bins = 50, fill = "#e63946", color = "black") +
  geom_vline(xintercept = median(df$variance),
             color = "black", linetype="dashed", size = 1) +
  scale_x_continuous(breaks = breaks1, labels = labs1) +
  theme(axis.ticks.y = element_blank(),
        axis.text.y = element_blank(), axis.title.y = element_blank())
breaks2 <- as.vector(quantile(df$icc, probs = c(0.001, 0.15, 0.5, 0.85, 0.999)))
labs2 <- round(breaks2, 2)
p2 <- ggplot(df) +
  geom_histogram(aes(x = icc), bins = 50, fill = "#a8dadc", color = "black") +
  geom_vline(xintercept = median(df$icc),
             color = "black", linetype="dashed", size = 1) +
  scale_x_continuous(breaks = breaks2, labels = labs2) +
  theme(axis.ticks.y = element_blank(),
        axis.text.y = element_blank(), axis.title.y = element_blank())
breaks3 <- as.vector(quantile(df$R2cond, probs = c(0.001, 0.15, 0.5, 0.85, 0.999)))
labs3 <- round(breaks3, 3)
p3 <- ggplot(df) +
  geom_histogram(aes(x = R2cond), bins = 50, fill = "#1d3557", color = "black") +
  geom_vline(xintercept = median(df$R2cond),
             color = "black", linetype="dashed", size = 1) +
  scale_x_continuous(breaks = breaks3, labels = labs3) +
  theme(axis.ticks.y = element_blank(),
        axis.text.y = element_blank(), axis.title.y = element_blank())
ggarrange(p1, p2, p3, nrow = 2, ncol = 2)
```

Les trois distributions sont toutes suffisamment éloignées de zéro pour que nous puissions en conclure que ces différentes valeurs sont toutes différentes de zéro. Notez également que les distributions sont relativement symétriques, indiquant que nous disposons de probablement suffisamment d'information dans nos données pour inclure notre effet aléatoire. Des distributions fortement asymétriques indiquerait, au contraire, une forte difficulté du modèle à estimer le paramètre de variance à partir des données. Dans un article, il n'est pas nécessaire de reporter ces graphiques, mais plus simplement les intervalles de confiance à 95 % et les médianes.

```
# Intervalle de confiance pour la variance
quantile(df$variance, probs = c(0.0275, 0.5, 0.975))
```

```
##      2.75%      50%     97.5%
## 0.3059400 0.5578986 0.9200472
```

```
# Intervalle de confiance pour l'ICC
quantile(df$icc, probs = c(0.0275, 0.5, 0.975))
```

```
##      2.75%      50%     97.5%
```

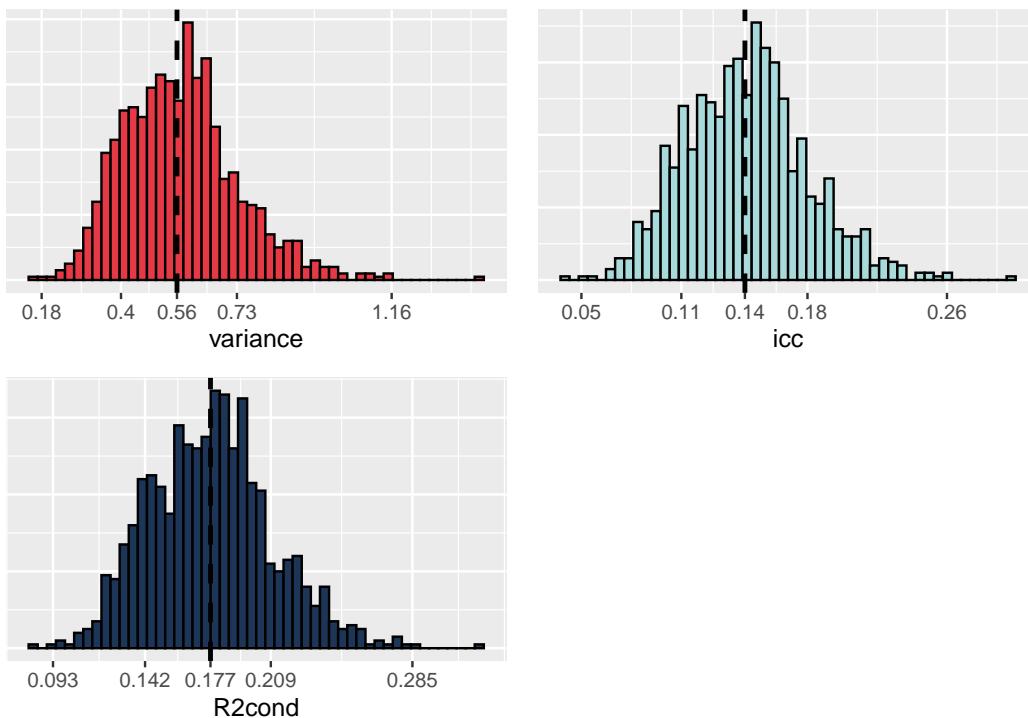


FIG. 9.17 : Distributions obtenues par bootstrap de la variance de l'effet aléatoire, de l'ICC et du R carré conditionnel

```
## 0.0850824 0.1449928 0.2185428
```

```
# Intervalle de confiance pour le R2 conditionnel
quantile(df$R2cond,probs = c(0.0275,0.5,0.975))
```

```
##      2.75%      50%     97.5%
## 0.1208577 0.1772778 0.2462622
```

9.6.1.3 Significativité des différentes constantes

Puisque nous avons conclu que l'effet aléatoire contribue significativement au modèle, nous pouvons à présent vérifier si les constantes ajustées pour chaque pays varient significativement les unes des autres. Pour rappel, les pentes et les constantes aléatoires ne sont pas directement estimées par le modèle, mais à posteriori. Il en résulte qu'il n'y a pas de moyen direct de mesurer l'incertitude de ces paramètres, et donc de construire des intervalles de confiance. Une première option pour contourner ce problème est d'effectuer des simulations à partir de la distribution postérieure du modèle. Notez que cette approche s'inspire largement de l'approche statistique bayésienne. Nous utilisons ici le package `merTools` pour effectuer 1000 simulations et obtenir une erreur standard pour chaque constante aléatoire de chaque pays.

```
# Simulations et extraction des effets aléatoires
library(merTools)
simsRE <- REsim(modele1,n.sims = 1000, oddsRatio = F)
# Calcul des intervalles de confiance
simsRE$lower <- simsRE$mean - 1.96 * simsRE$sd
simsRE$upper <- simsRE$mean + 1.96 * simsRE$sd
```

```
# Variable binaire pour la significativité
simsRE$sign <- case_when(
  simsRE$lower<0 & simsRE$upper<0 ~ "inf",
  simsRE$lower>0 & simsRE$upper>0 ~ "sup",
  TRUE ~ "not"
)
# Représentation des intervalles de confiance
ggplot(simsRE) +
  geom_errorbarh(aes(xmin = lower, xmax = upper,
                      y = reorder(groupID,mean)), size = 0.5, height = 0.5) +
  geom_point(aes(x = mean, y = reorder(groupID,mean),
                 color = sign)) +
  scale_color_manual(values = c("inf" = "#0077b6", "sup" = "#e63946", "not"="#000000"),
                     labels = c("sign. < 0", "sign. > 0", "non sign.")) +
  labs(x = "Constante aléatoire", y = "Pays")
```

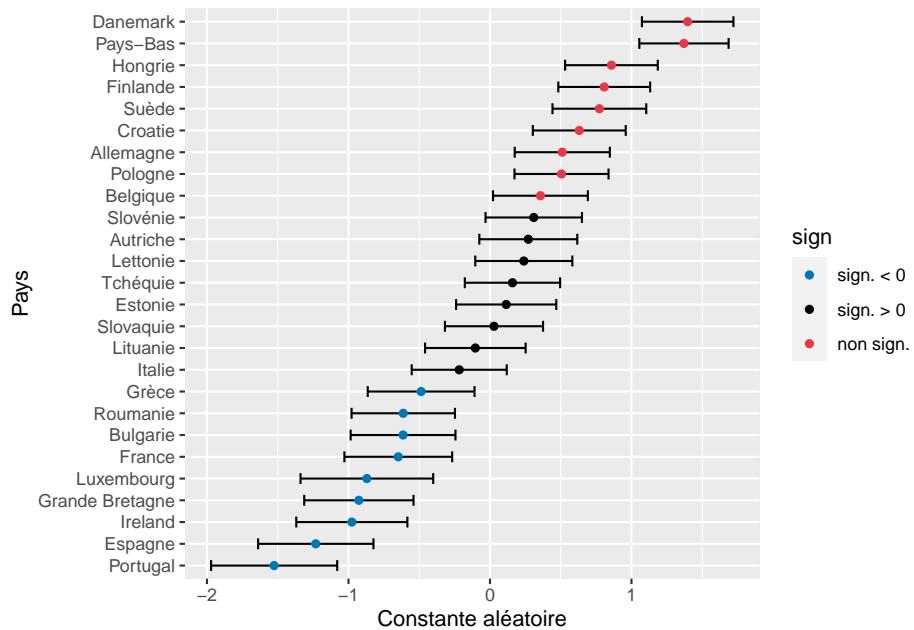


FIG. 9.18 : Constantes aléatoires estimées par Pays (IC par simulations)

La figure 9.18 permet de repérer en un coup d'oeil les pays pour lesquels la probabilité d'utiliser le vélo comme moyen de transport pour le trajet le plus fréquent est la plus élevée ou la plus faible. Notez cependant que les valeurs représentées sont pour l'instant des logarithmes de rapport de cotes. Nous devons donc les convertir en rapports de cotes avec la fonction exponentielle pour faciliter leur interprétation.

```
# Conversion en rapports de cote (et arrondissement à trois décimales)
mat <- round(exp(simsRE[,c("mean","lower","upper")]),3)
rownames(mat) <- simsRE$groupID
names(mat) <- c("RC", "RC.025", "RC.975")
print(mat)
```

	##	RC	RC.025	RC.975
## Allemagne		1.667	1.191	2.333
## Autriche		1.311	0.927	1.853

	1.428	1.022	1.997
## Belgique	0.541	0.374	0.784
## Bulgarie	1.880	1.354	2.610
## Croatie	4.043	2.926	5.587
## Danemark	0.292	0.194	0.439
## Espagne	1.121	0.787	1.598
## Estonie	2.242	1.621	3.102
## Finlande	0.523	0.357	0.766
## France	0.396	0.269	0.583
## Grande Bretagne	0.615	0.422	0.897
## Grèce	2.359	1.700	3.273
## Hongrie	0.377	0.254	0.558
## Ireland	0.805	0.575	1.126
## Italie	1.270	0.901	1.790
## Lettonie	0.902	0.632	1.287
## Lituanie	0.419	0.262	0.670
## Luxembourg	3.939	2.873	5.400
## Pays-Bas	1.658	1.189	2.311
## Pologne	0.218	0.139	0.340
## Portugal	0.542	0.376	0.781
## Roumanie	1.029	0.727	1.455
## Slovaquie	1.362	0.969	1.916
## Slovénie	2.166	1.555	3.018
## Suède	1.172	0.837	1.641
## Tchéquie			

Nous observons ainsi qu'une personne vivant en Finlande voit ses chances multipliées par 2,25 d'utiliser le vélo comme mode de transport pour son trajet le plus fréquent comparativement à la moyenne des pays européens. À l'inverse, une personne résidant en France a 47 % de chances de moins d'utiliser le vélo.

Notez cependant que cette approche basée sur des simulations peut poser des problèmes, car elle ne renvoie qu'une erreur standard pour mesurer l'incertitude de nos constantes. Dans les cas où nous ne disposons pas de beaucoup d'observations par groupe, la distribution à posteriori des constantes peut être asymétrique, rendant l'estimation des intervalles de confiance par les erreurs standards inutiles. Il est possible de détecter ce cas de figure quand les médianes et les moyennes renvoyées par la fonction `simsRE` diffèrent nettement. Une alternative plus robuste est à nouveau d'estimer la variabilité des effets aléatoires par `bootstrap`. Cette méthode requiert bien plus de temps de calcul que la précédente, nous vous recommandons donc de commencer par la méthode par simulations pour disposer d'un premier aperçu des résultats et d'utiliser ensuite la méthode `bootstrap` quand votre modèle est dans sa forme finale.

```
# Création de la fonction d'extraction
extractor2 <- function(mod){
  elements <- ranef(mod)$Pays
  vec <- elements[,1]
  names(vec) <- rownames(elements)
  return(vec)
}

# Préparation de l'opération en multitraitement
cl <- makeCluster(8)
clusterEvalQ(cl,library("lme4"))

# Calcul des effets aléatoires en bootstrap
```

```

valeurs <- bootMer(modele1, FUN = extractor2, nsim = 1000,
                     use.u = T, type="parametric", ncpus = 8,
                     parallel="snow",
                     cl = cl)
# Sauvegarder des résultats!
save(valeurs, file = 'data/glmm/boot_binom2.rda')

```

Puisque nous disposons des distributions *bootstrapées* des différents effets aléatoires, nous pouvons directement les représenter dans un graphique (figure 9.19). Les résultats sont très similaires à ceux de la figure 9.18, ce qui s'explique par le grand nombre d'observations et de groupes. Avec moins d'observations, il est recommandé de privilégier l'approche par *bootstrap*.

```

# Chargement de nos valeurs bootstrapées
load('data/glmm/boot_binom2.rda')
# Conversion des bootstraps en intervalle de confiance
q025 <- function(x){return(quantile(x,probs = 0.025))}
q975 <- function(x){return(quantile(x,probs = 0.975))}
df <- reshape2::melt(valeurs$t)
df_med <- df %>% group_by(Var2) %>% summarise(
  med = median(value),
  lower = q025(value),
  upper = q975(value))
# Ajout d'une variable pour la couleur si significatif
df_med$sign <- case_when(
  df_med$lower<0 & df_med$upper<0 ~ "inf",
  df_med$lower>0 & df_med$upper>0 ~ "sup",
  TRUE ~ "not")
)
# Affichage des résultats
ggplot(df_med) +
  geom_errorbar(aes(xmin = lower, xmax = upper, y = reorder(Var2,med)), width = 0.5) +
  geom_point(aes(x = med, y = reorder(Var2,med), color = sign)) +
  scale_color_manual(values = c("inf" = "#0077b6", "sup" = "#e63946", "not"="#000000"),
                     labels = c("sign. < 0", "sign. > 0", "non sign."))
  labs(x = "Constante aléatoire", y = "Pays")

```

9.6.2 Ajustement du modèle avec constantes et pentes aléatoires

Dans le modèle précédent, nous avons ajusté, pour chaque pays, une constante aléatoire afin de vérifier si la probabilité d'utiliser le vélo comme mode de transport principal changeait d'un pays d'Europe à l'autre. Nous souhaitons à présent tester l'hypothèse que l'effet de l'âge sur la probabilité d'utiliser le vélo varie d'un pays à l'autre. Pour cela, nous ajustons des constantes aléatoires par pays. Nous comparons trois modèles, triés ici selon leur niveau de complexité (nombre de paramètres) :

- le modèle avec uniquement des constantes aléatoires;
- le modèle avec des constantes et des pentes aléatoires indépendantes;
- le modèle avec des constantes et des pentes aléatoires corrélées.

Dans le package `lme4`, les syntaxes pour ajuster ces trois modèles sont les suivantes :

- constantes aléatoires : `+(1|Pays)` ;
- constantes et pentes aléatoires indépendantes : `+(1 + Age|Pays)` ;
- constantes et pentes aléatoires corrélées : `+(1 + Age|Pays)`.

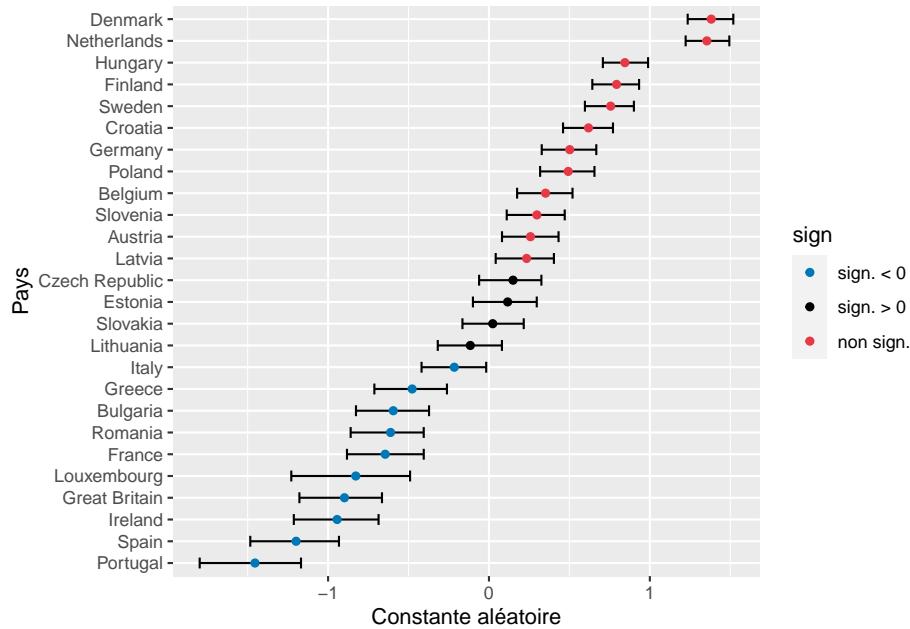


FIG. 9.19 : Constantes aléatoires estimées par Pays (IC par bootstrap)

Notez qu'il est aussi possible d'ajuster un modèle avec uniquement des pentes aléatoires avec la syntaxe : `+(-1 + Age | Pays)`. Le paramètre `-1` sert à retirer explicitement la constante aléatoire du modèle. Ajustons donc nos deux modèles avec pentes et constantes aléatoires.

```
# Constantes et pentes aléatoires indépendantes
modele2 <- glmer(y ~ Sexe + Age2 + Education + StatutEmploi + Revenu +
  Residence + Duree2 + ConsEnv + (1 + Age2 || Pays),
  family = binomial(link="logit"),
  control = glmerControl(optimizer = "bobyqa"),
  data = dfenquete2)

# Constantes et pentes aléatoires corrélées
modele3 <- glmer(y ~ Sexe + Age2 + Education + StatutEmploi + Revenu +
  Residence + Duree2 + ConsEnv + (1 + Age2 | Pays),
  family = binomial(link="logit"),
  control = glmerControl(optimizer = "bobyqa"),
  data = dfenquete2)
```

9.6.2.1 Significativité de l'effet aléatoire

Puisque les trois modèles sont imbriqués, la première étape est de vérifier si les ajouts successifs au modèle de base sont significatifs, ce que nous pouvons tester avec un rapport de vraisemblance.

```
anova(modele1,modele2,modele3)
```

```
## Data: dfenquete2
## Models:
## modele1: y ~ Sexe + Age2 + Education + StatutEmploi + Revenu + Residence +
## modele1:      Duree2 + ConsEnv + (1 | Pays)
## modele2: y ~ Sexe + Age2 + Education + StatutEmploi + Revenu + Residence +
```

```

## modele2: Duree2 + ConsEnv + (1 + Age2 || Pays)
## modele3: y ~ Sexe + Age2 + Education + StatutEmploi + Revenu + Residence +
## modele3: Duree2 + ConsEnv + (1 + Age2 | Pays)
##      npar   AIC   BIC logLik deviance Chisq Df Pr(>Chisq)
## modele1 18 19176 19323 -9570.1    19140
## modele2 19 19171 19325 -9566.3    19133 7.5754  1   0.005917 **
## modele3 20 19172 19335 -9566.1    19132 0.4033  1   0.525385
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Nous constatons ainsi que l'ajout des pentes aléatoires permet d'améliorer significativement le modèle, mais que l'ajout de la corrélation entre les pentes et les constantes aléatoires a un apport très marginal. Nous décidons tout de même de le garder dans un premier temps, car ce paramètre a un intérêt théorique. Affichons le résumé du modèle 3.

```
summary(modele3)
```

```

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: y ~ Sexe + Age2 + Education + StatutEmploi + Revenu + Residence +
##          Duree2 + ConsEnv + (1 + Age2 | Pays)
## Data: dfenquete2
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC logLik deviance df.resid
## 19172.2 19335.1 -9566.1  19132.2    25527
##
## Scaled residuals:
##      Min      1Q Median      3Q     Max
## -1.2087 -0.4392 -0.3218 -0.2137  7.1677
##
## Random effects:
## Groups Name        Variance Std.Dev. Corr
## Pays   (Intercept) 0.594976 0.77135
##       Age2        0.007326 0.08559 -0.22
## Number of obs: 25547, groups: Pays, 26
##
## Fixed effects:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -3.357922  0.213122 -15.756 < 2e-16 ***
## Sexehomme                 0.372886  0.037961   9.823 < 2e-16 ***
## Age2                     -0.087748  0.027796  -3.157 0.001595 **
## Educationsecondaire       0.179620  0.103838   1.730 0.083664 .
## Educationsecondaire inferieur 0.286993  0.111904   2.565 0.010329 *
## Educationuniversite       0.130114  0.107149   1.214 0.224622
## StatutEmploisans emploi    0.256253  0.042508   6.028 1.66e-09 ***
## Revenufaible                0.070578  0.071820   0.983 0.325754
## Revenumoyen                  0.038414  0.065347   0.588 0.556630
## Revenusans reponse          0.203615  0.102658   1.983 0.047319 *

```

```

## Revenutres eleve          -0.124202  0.185684 -0.669 0.503564
## Revenutres faible         0.235617  0.085949  2.741 0.006118 ***
## Residencegrande ville    0.269577  0.069366  3.886 0.000102 ***
## Residencepetite-moyenne ville 0.275698  0.061575  4.477 7.56e-06 ***
## Residencezone rurale     -0.118979  0.069177 -1.720 0.085446 .
## Duree2                     -0.018889  0.019274 -0.980 0.327067
## ConsEnv                    0.101888  0.009291  10.967 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

À nouveau, nous nous intéressons ici principalement à la section `Random Effect`, puisque les effets fixes s'interprètent exactement comme dans les modèles présentés dans le chapitre 6. Les constantes ont une variance de 0,595 et les pentes de 0,007. La corrélation entre les deux effets est de -0,22. Cette corrélation est négative et relativement faible, ce qui signifie que les pays dans lesquels la constante est forte tendent à avoir un coefficient plus petit pour l'âge, et donc une réduction accrue de la probabilité d'utiliser le vélo avec l'âge. Nous devons cependant encore nous assurer qu'elle est significativement différente de 0. Pour cela, nous devons calculer l'intervalle de confiance des trois paramètres de variance du modèle. Nous utilisons à nouveau une approche par `bootstrap` et nous enregistrons les résultats.

```

# Fonction d'extraction des trois paramètres de variance
extractor3 <- function(mod){
  vari1 <- VarCorr(mod)[[1]][[1]]
  vari2 <- VarCorr(mod)[[1]][[4]]
  covari <- VarCorr(mod)[[1]][[2]]
  return(c("vari1"=vari1,"vari2"=vari2,"covari"=covari))
}

# Lancement du bootstrap
valeurs <- bootMer(modele3,FUN = extractor3,nsim = 1000,
                     use.u = F, type="parametric", ncpus = 8,
                     parallel="snow",
                     cl = cl,
                     .progress="txt",PBarg=list(style=3))

# Enregistrement des résultats
save(valeurs,file = 'data/glmm/boot_binom3.rda')

```

À partir des valeurs *bootstrapées*, nous pouvons représenter les distributions de ces trois paramètres (variance des constantes, variance des pentes et corrélation entre les deux).

```

# Chargement des résultats
load('data/glmm/boot_binom3.rda')
# Conversion des valeurs de covariance en corrélation
df <- data.frame(
  corr_values = valeurs$t[,3] / (sqrt(valeurs$t[,1]) * sqrt(valeurs$t[,2])),
  vari_const = valeurs$t[,1],
  vari_pente = valeurs$t[,2]
)
# Histogramme pour la variance des constantes
breaks1 <- quantile(df$vari_const,probs=c(0.025,0.5,0.975,0.999))
label1 <- round(breaks1,3)
p1 <- ggplot(df) +
  geom_histogram(aes(x = vari_const), color = "black", fill = "white", bins = 30) +
  geom_vline(xintercept = median(df$vari_const), color = "red", size = 1, linetype="dashed") +
  geom_vline(xintercept = quantile(df$vari_const, probs = 0.025),

```

```

        color = "blue", size = 0.5, linetype="dashed") +
geom_vline(xintercept = quantile(df$vari_const, probs = 0.975),
            color = "blue", size = 0.5, linetype="dashed") +
labs(x = "Variance des constantes", y="")+
scale_x_continuous(breaks = breaks1, labels = label1)
# Histogramme pour la variance des pentes
breaks2 <- quantile(df$vari_pente,probs=c(0.025,0.5,0.975,0.999))
label2 <- round(breaks2,3)
p2 <- ggplot(df) +
  geom_histogram(aes(x = vari_pente), color = "black", fill = "white", bins = 30) +
  geom_vline(xintercept = median(df$vari_pente), color = "red", size = 1, linetype="dashed") +
  geom_vline(xintercept = quantile(df$vari_pente, probs = 0.025),
              color = "blue", size = 0.5, linetype="dashed") +
  geom_vline(xintercept = quantile(df$vari_pente, probs = 0.975),
              color = "blue", size = 0.5, linetype="dashed") +
  labs(x = "Variance des pentes", y="")+
  scale_x_continuous(breaks = breaks2, labels = label2)
# Histogramme pour la corrélation
breaks3 <- c(-1,-0.5,0,0.5,1,median(df$corr_values))
label3 <- round(breaks3,3)
p3 <- ggplot(df) +
  geom_histogram(aes(x = corr_values), color = "black", fill = "white", bins = 30) +
  geom_vline(xintercept = median(df$corr_values), color = "red", size = 1, linetype="dashed") +
  geom_vline(xintercept = quantile(df$corr_values, probs = 0.025),
              color = "blue", size = 0.5, linetype="dashed") +
  geom_vline(xintercept = quantile(df$corr_values, probs = 0.975),
              color = "blue", size = 0.5, linetype="dashed") +
  labs(x = "Corrélation pentes/constantes", y="") +
  scale_x_continuous(breaks = breaks3, labels = label3)
ggarrange(p1,p2, p3, ncol = 2, nrow = 2)

```

Nous constatons ainsi, à la figure 9.20, que la variance des constantes aléatoires est significativement différente de zéro (cette valeur n'est pas dans l'intervalle de confiance à 95 % représenté par les lignes verticales bleues) et une médiane de 0,56 (ligne verticale rouge). Pour les pentes, zéro est également à la limite de l'intervalle de confiance, et la distribution asymétrique et étalée nous indique que ce paramètre est fortement incertain dans le modèle. Enfin, la corrélation entre les pentes et les constantes est de loin le paramètre le plus incertain et son intervalle de confiance est franchement à cheval sur zéro, ce qui devrait nous amener à privilégier un modèle sans ce paramètre.

Pour terminer, nous pouvons calculer les R^2 marginal et conditionnel du modèle afin de mieux cerner le rôle joué par les effets fixes et les effets aléatoires.

```
r.squaredGLMM(modele3)
```

```

##                   R2m          R2c
## theoretical  0.03413720  0.18360159
## delta        0.01419945  0.07636951

```

Les valeurs des R^2 marginal et conditionnel du modèle sont similaires à celles que nous avons obtenus avec seulement des constantes aléatoires dans la section précédente, signalant l'apport relativement faible des pentes aléatoires.

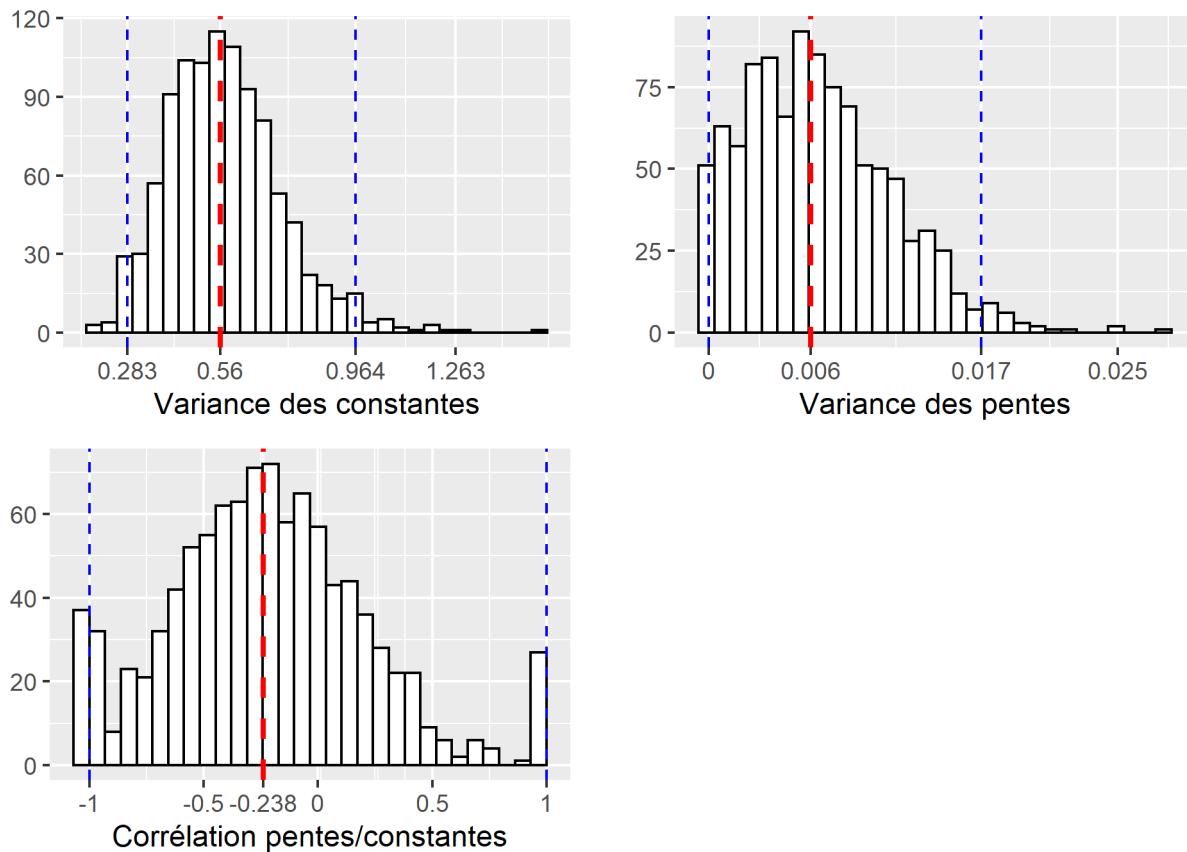


FIG. 9.20 : Incertitude autour des paramètres de variance obtenue par bootstrap

9.6.2.2 Analyse des effets aléatoires

Pour analyser facilement les constantes et les pentes aléatoires de chaque pays, nous pouvons représenter graphiquement leurs intervalles de confiance construits à partir des simulations tirées de la distribution a posteriori du modèle.

```
# Simulations et extraction des effets aléatoires
library(merTools)
simsRE <- REsim(modele3,n.sims = 1000, oddsRatio = F)
# Calcul des intervalles de confiance
simsRE$lower <- simsRE$mean - 1.96 * simsRE$sd
simsRE$upper <- simsRE$mean + 1.96 * simsRE$sd
# Variable binaire pour la significativité
simsRE$sign <- case_when(
  simsRE$lower<0 & simsRE$upper<0 ~ "inf",
  simsRE$lower>0 & simsRE$upper>0 ~ "sup",
  TRUE ~ "not"
)
df1 <- subset(simsRE, grepl("Intercept",simsRE$term,fixed = T))
df2 <- subset(simsRE, grepl("Age2",simsRE$term,fixed = T))
# Représentation des intervalles de confiance
p1 <- ggplot(df1) +
  geom_errorbarh(aes(xmin = lower, xmax = upper,
                     y = reorder(groupID,mean)), size = 0.5, height = 0.5) +
  geom_point(aes(x = mean, y = reorder(groupID,mean),
```

```

        color = sign)) +
scale_color_manual(values = c("inf" = "#0077b6", "sup" = "#e63946", "not"="#000000"),
                   labels = c("sign. < 0", "non sign.", "sign. > 0")) +
labs(x = "Constante aléatoire", y = "Pays")
p2 <- ggplot(df2) +
geom_errorbar(aes(xmin = lower, xmax = upper,
                  y = reorder(groupID,mean)), size = 0.5, height = 0.5) +
geom_point(aes(x = mean, y = reorder(groupID,mean),
               color = sign)) +
scale_color_manual(values = c("inf" = "#0077b6", "sup" = "#e63946", "not"="#000000"),
                   labels = c("sign. < 0", "non sign.", "sign. > 0")) +
labs(x = "Pente aléatoire (âge)", y = "Pays")
ggarrange(p1,p2, common.legend = T, nrow = 1, ncol = 2)

```

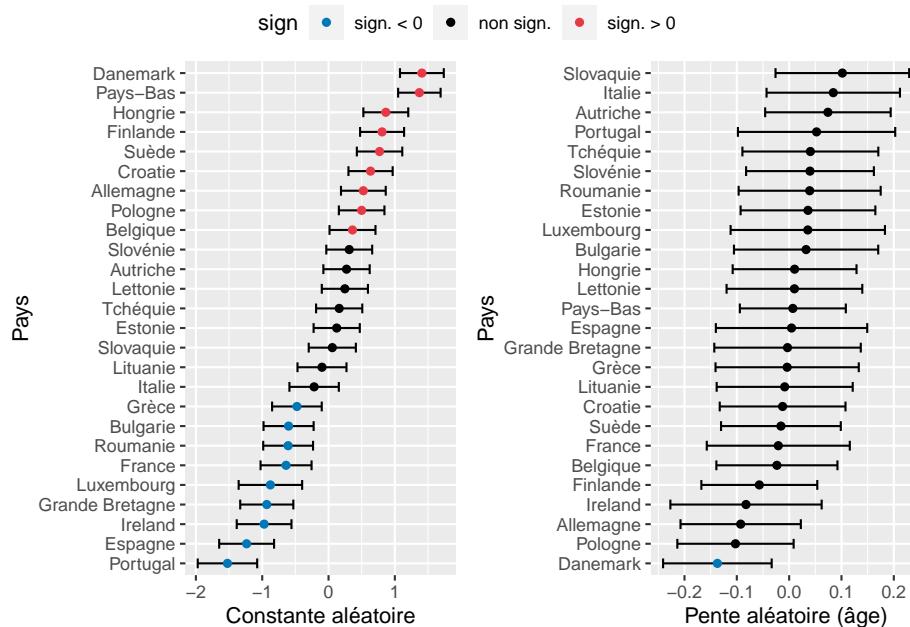


FIG. 9.21 : Constantes aléatoires estimées par pays (intervalles de confiance obtenus par simulations)

La figure 9.21 nous permet ainsi de constater que l'effet des pays sur les pentes est presque toujours non significatif, sauf pour le Danemark. Son effet négatif (-0,136) indique un renforcement de l'effet général, lui-même négatif (-0,088). Une interprétation possible est qu'au Danemark, l'utilisation du vélo est proportionnellement plus courante par les jeunes que dans le reste des pays de l'Europe.

Pour l'interprétation finale, il est nécessaire d'afficher les valeurs exactes de ces différents paramètres et, dans notre cas, de les convertir en rapports de cotes avec la fonction exponentielle. Pour les pentes aléatoires, il peut être plus facile d'interpréter la somme de l'effet fixe et de l'effet aléatoire.

```

# Extraction des effets aléatoires obtenus par simulation
mat <- simsRE[c("mean", "lower", "upper")]
mat$Pays <- simsRE$groupID
mat$effet <- simsRE$term
# Séparation des pentes et des constantes
df1 <- subset(mat, grepl("Intercept", mat$effet, fixed = TRUE))
df2 <- subset(mat, grepl("Age2", mat$effet, fixed = TRUE))

```

```
# Conversion en rapports de cotes pour les pentes (+ effet fixe)
df2$RC <- round(exp(df2$mean + fixef(modele3)[[3]]),3)
df2$RC025 <- round(exp(df2$lower + fixef(modele3)[[3]]),3)
df2$RC975 <- round(exp(df2$upper + fixef(modele3)[[3]]),3)
print(head(df2[c("Pays","RC","RC025","RC975")],10))
```

```
##          Pays      RC RC025 RC975
## 27 Allemagne 0.835 0.744 0.937
## 28 Autriche 0.986 0.875 1.112
## 29 Belgique 0.895 0.797 1.005
## 30 Bulgarie 0.946 0.824 1.086
## 31 Croatie 0.905 0.802 1.020
## 32 Danemark 0.799 0.720 0.886
## 33 Espagne 0.920 0.796 1.063
## 34 Estonie 0.950 0.835 1.080
## 35 Finlande 0.865 0.774 0.967
## 36 France 0.897 0.783 1.029
```

Nous constatons ainsi qu'au Danemark, les chances pour un individu d'utiliser le vélo sont réduites de 20 % à chaque augmentation de l'âge d'un écart-type, contre seulement 1,5 % en Autriche (non significatif pour ce dernier). Notons ici que l'écart-type de la variable Age est de 11 ans. Nous pouvons à présent analyser les constantes.

```
# Conversion en rapports de cotes pour les constantes
df1$RC <- round(exp(df1$mean),3)
df1$RC025 <- round(exp(df1$lower),3)
df1$RC975 <- round(exp(df1$upper),3)
print(head(df1[c("Pays","RC","RC025","RC975")],10))
```

```
##          Pays      RC RC025 RC975
## 1 Allemagne 1.691 1.207 2.369
## 2 Autriche 1.312 0.925 1.860
## 3 Belgique 1.435 1.015 2.029
## 4 Bulgarie 0.547 0.375 0.799
## 5 Croatie 1.885 1.350 2.630
## 6 Danemark 4.087 2.937 5.686
## 7 Espagne 0.291 0.192 0.440
## 8 Estonie 1.131 0.798 1.602
## 9 Finlande 2.243 1.609 3.125
## 10 France 0.527 0.359 0.774
```

En revanche, les chances pour un individu d'utiliser le vélo comme mode de transport pour son trajet le plus fréquent sont 4 fois supérieures à la moyenne européenne, contre seulement 1,3 fois en Autriche. Notez à nouveau que les intervalles de confiance pour ces pentes et ces constantes pourraient être estimés plus fiablement par *bootstrap*.

9.6.2.3 Diagnostic des effets aléatoires

Pour rappel, dans un modèle GLMM, les effets aléatoires sont modélisés comme provenant de distributions normales. Nous devons donc vérifier qu'ils respectent cette condition d'application. La figure 9.22 (graphique quantile-quantile) nous permet de constater que les constantes suivent bien une distribution

normale, ce qui ne semble pas vraiment être le cas pour les pentes. Considérant que leurs effets sont petits, il serait plus pertinent ici de les retirer du modèle.

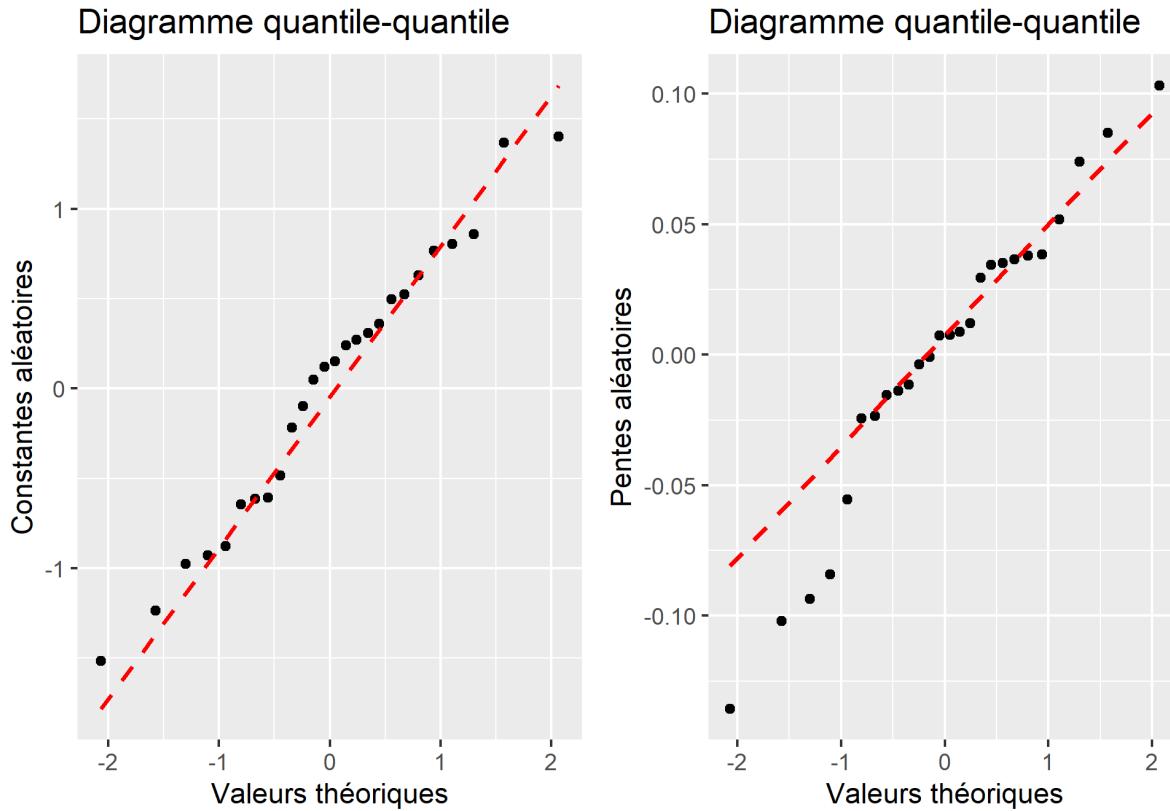


FIG. 9.22 : Distribution normale univariée des constantes et des pentes aléatoires

Considérant que ce modèle inclut une corrélation entre les constantes et les pentes aléatoires, il est également nécessaire de vérifier si elles suivent conjointement une distribution normale bivariée. La figure 9.23 semble indiquer que c'est le cas.

```
cor_mat <- VarCorr(modele3)[[1]]
re_effects <- data.frame(ranef(modele3)$Pays)
names(re_effects) <- c("constante", "pente")
library(ellipse)
levels <- c(0.05, 0.25, 0.75, 0.95)
els <- lapply(levels, function(i){
  el <- data.frame(ellipse(cor_mat, center = c(0,0), level = i))
  names(el) <- c("x", "y")
  return(el)
})
ref_points <- data.frame(data.frame(MASS::mvrnorm(n = 1000, mu = c(0,0), Sigma = cor_mat)))
names(ref_points) <- c("x", "y")
ggplot() +
  geom_point(aes(x = x, y = y), data = ref_points, alpha = 0.3, size = 0.4) +
  geom_path(data = els[[1]], aes(x = x, y = y, color = "a")) +
  geom_path(data = els[[2]], aes(x = x, y = y, color = "b")) +
  geom_path(data = els[[3]], aes(x = x, y = y, color = "c")) +
  geom_path(data = els[[4]], aes(x = x, y = y, color = "d")) +
  geom_point(data = re_effects, aes(x = constante, y = pente)) +
```

```
scale_color_manual(values = c("a"="#90e0ef",
                             "b"="#00b4d8",
                             "c"="#0077b6",
                             "d"="#03045e"),
                   labels = c("5 %", "25 %", "75 %", "95 %"))+
  labs(x = "Constantes", y = "Pentes", color = "quantiles")
```

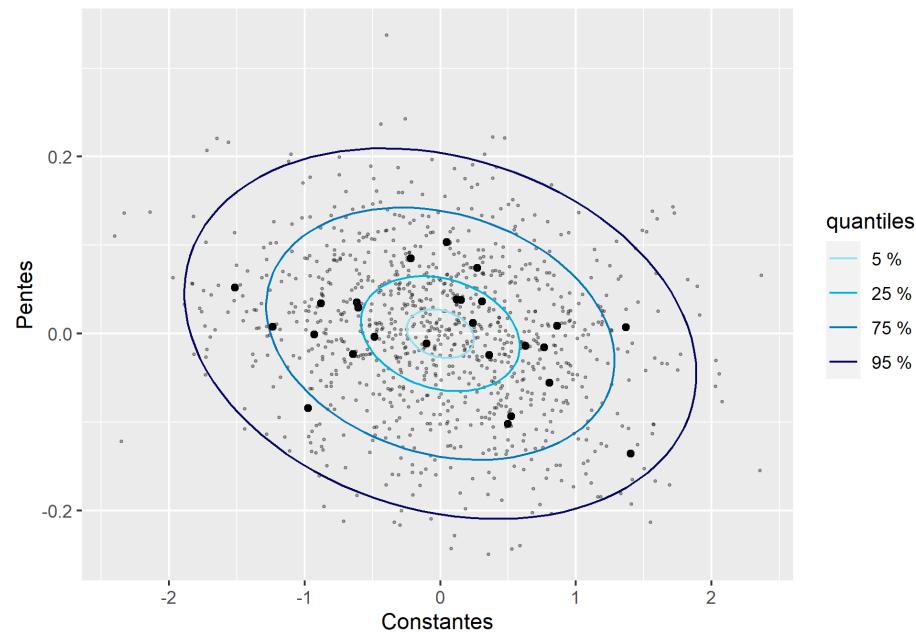


FIG. 9.23 : Distribution normale bivariée des constantes et des pentes aléatoires

9.6.2.4 Inférence pour les effets fixes

Nous avons mentionné, dans les sections précédentes, que le calcul de valeurs de p pour les effets fixes fait l'objet de controverses pour les modèles GLMM. La méthode offrant le meilleur compromis entre rapidité de calcul et fiabilité est la méthode Satterthwaite implémentée dans le package `lmerTest`. Pour l'utiliser, il suffit de charger le package `lmerTest` après `lme4`, ce qui modifie la fonction `summary` pour qu'elle utilise directement cette approche.

```
library(lmerTest)
round(summary(modele3)$coefficients, 3)
```

	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	-3.358	0.213	-15.756	0.000
## Sexehomme	0.373	0.038	9.823	0.000
## Age2	-0.088	0.028	-3.157	0.002
## Educationsecondaire	0.180	0.104	1.730	0.084
## Educationsecondaire inferieur	0.287	0.112	2.565	0.010
## Educationuniversite	0.130	0.107	1.214	0.225
## StatutEmploisans emploi	0.256	0.043	6.028	0.000
## Revenuifaible	0.071	0.072	0.983	0.326
## Revenumoyen	0.038	0.065	0.588	0.557

```
## Revenusans reponse      0.204   0.103   1.983   0.047  
## Revenutres eleve       -0.124   0.186  -0.669   0.504  
## Revenutres faible      0.236   0.086   2.741   0.006  
## Residencegrande ville  0.270   0.069   3.886   0.000  
## Residencepetite-moyenne ville  0.276   0.062   4.477   0.000  
## Residencezone rurale    -0.119   0.069  -1.720   0.085  
## Duree2                  -0.019   0.019  -0.980   0.327  
## ConsEnv                 0.102   0.009  10.967   0.000
```

Les deux autres options envisageables sont : effectuer une analyse de type 3 ou calculer les intervalles de confiance par *bootstrap*. Cependant, elles requièrent beaucoup plus de temps de calcul. Par conséquent, elles ne sont pas présentées ici.

9.7 Quiz de révision du chapitre

Questions

- GLMM est un acronyme signifiant :

- modèles Linéaires Généralisés à effets Mixtes
- modèles Linéaires Gaussiens à effets Mixtes
- modèles Linéaire Généralisé Multinomial
- modèles Linéaire Généralisés Multiniveau

Relisez au besoin la section [9.1.2](#)

- Les modèles GLMM, comparativement aux GLM, permettent de tenir compte de :

- la distribution spécifique de la variable Y
- les distributions spécifiques des variables X
- l'hétéroscédasticité des résidus
- la non-indépendance des observations

Relisez au besoin la section [9.1.1](#).

- Un effet aléatoire se distingue d'un effet fixe, car :

- Un effet aléatoire n'est pas propre aux individus (unités d'observations), mais provient de facteurs externes.
- Un effet aléatoire induit une forme de hiérarchie, de regroupement d'individus dans des groupes au seins desquels les observations ont plus tendance à se ressembler.
- Un effet aléatoire provient d'une autre population que les unités d'observations et peut donc être également échantillonné.
- Un effet aléatoire implique un partage de l'information dans son estimation. Comparative-ment à un effet fixe, la conséquence est notamment la réduction des tailles d'effet (shrinkage).

Relisez au besoin la section [9.1.2](#).

- Quels sont les principaux types d'effets que nous pouvons modéliser avec des effets aléatoires ?

- Des constantes aléatoires, suggérant des écarts absous entre les groupes
- Des pentes aléatoires, suggérant des écarts en termes d'effets pour certains prédicteurs en fonction des groupes
- Des pentes et des constantes aléatoires

Relisez au besoin la section [9.2](#).

- Quels indicateurs peuvent être utilisés pour analyser la part de la variance / déviance expliquée aux différents niveaux d'un GLMM ?

- L'ICC, soit le coefficient de corrélation intraclassé
- Le BIC, soit le critère d'information Bayésien
- L'AIC
- La déviance expliquée
- Les R² marginal et conditionnel

Relisez au besoin la section [9.2.1.3](#).

- Toutes les variables catégorielles devraient être incluses dans un modèle comme des effets aléatoires.

- Vrai
- Faux

Relisez au besoin la section [9.1.2](#).

- **Lors de l'estimation d'un effet aléatoire, le modèle estime exactement :**

- $k + 1$ paramètres, soit le nombre de catégories plus un paramètre pour la variance de l'effet aléatoire en question
- un seul paramètre, la variance de la distribution normale associée à l'effet aléatoire en question
- k paramètres, soit le nombre de catégories pour l'effet aléatoire en question
- $k - 1$ paramètres, soit le nombre de catégories moins un pour la catégorie de référence

Relisez au besoin la section [9.2.1](#).

Réponses

- GLMM est un acronyme signifiant :
 - modèles Linéaires Généralisés à effets Mixtes
- Les modèles GLMM, comparativement aux GLM, permettent de tenir compte de :
 - la non-indépendance des observations
- Un effet aléatoire se distingue d'un effet fixe, car :
 - Un effet aléatoire n'est pas propre aux individus (unités d'observations), mais provient de facteurs externes.
 - Un effet aléatoire induit une forme de hiérarchie, de regroupement d'individus dans des groupes au seins desquels les observations ont plus tendance à se ressembler.
 - Un effet aléatoire provient d'une autre population que les unités d'observations et peut donc être également échantillonné.
 - Un effet aléatoire implique un partage de l'information dans son estimation. Comparativement à un effet fixe, la conséquence est notamment la réduction des tailles d'effet (shrinkage).
- Quels sont les principaux types d'effets que nous pouvons modéliser avec des effets aléatoires ?
 - Des constantes aléatoires, suggérant des écarts absous entre les groupes
 - Des pentes aléatoires, suggérant des écarts en termes d'effets pour certains prédicteurs en fonction des groupes
 - Des pentes et des constantes aléatoires
- Quels indicateurs peuvent être utilisés pour analyser la part de la variance / déviance expliquée aux différents niveaux d'un GLMM ?
 - L'ICC, soit le coefficient de corrélation intraclassé
 - Les R² marginal et conditionnel
- Toutes les variables catégorielles devraient être incluses dans un modèle comme des effets aléatoires.
 - Faux
- Lors de l'estimation d'un effet aléatoire, le modèle estime exactement :
 - un seul paramètre, la variance de la distribution normale associée à l'effet aléatoire en question

Chapitre 10

Régressions multiniveaux

Dans le précédent chapitre, nous avons abordé les modèles à effets mixtes qui permettent d'introduire à la fois des effets fixes et aléatoires (GLMM). Dans ce chapitre, nous poursuivons sur cette voie avec une nouvelle extension des modèles GLM : les modèles multiniveaux. Ces modèles sont simplement une extension des modèles à effets mixtes et permettent de modéliser un phénomène avec une structure hiérarchique des données, tel que décrit dans le chapitre précédent.

Rappel de la structure hiérarchique des données

Exemple à deux niveaux : il s'agit de modéliser un phénomène y_{ij} , soit une variable dépendante Y pour un individu i (niveau 1) niché dans un groupe j (niveau 2). Par exemple, modéliser l'indice de masse corporelle (IMC) de 5000 individus résidant dans 100 quartiers différents.

Exemple à trois niveaux : il s'agit de modéliser un phénomène y_{ijk} , soit une variable dépendante Y pour un individu i (niveau 1), niché dans un groupe j (niveau 2) appartenant à un groupe k (niveau 3). Par exemple, modéliser les notes à un examen de mathématiques d'élèves (niveau 1) nichés dans des classes (niveau 2) nichées dans des écoles (niveau 3).

Nous avons largement décrit précédemment trois principaux types de modèles d'effets mixtes (GLMM) :

- Les GLMM avec constantes aléatoires qui permettent d'avoir une constante différente pour chacun des groupes (niveau 2).
- Les GLMM avec pentes aléatoires qui permettent de faire varier une variable indépendante au niveau 1 (coefficients) en fonction des groupes au niveau 2.
- Les GLMM avec constantes et pentes aléatoires.

Les modèles multiniveaux se différencient des modèles à effets mixtes puisqu'ils permettent d'introduire des variables indépendantes mesurées aux niveaux supérieurs (2 et 3).



Dans ce chapitre, nous utilisons les *packages* suivants :

- `lme4` pour en œuvre des modèles multiniveaux avec une variable dépendante continue.
- `performance` pour obtenir le coefficient intraclass (ICC).
- `MuMIN` pour obtenir les pseudo R^2 .

10.1 Modèles multiniveaux : deux intérêts majeurs

Les modèles multiniveaux ont deux principaux avantages : analyser la répartition de la variance entre les différents niveaux et introduire des variables explicatives aux différents niveaux du modèle.

10.1.1 Répartition de la variance entre les différents niveaux

Les modèles multiniveaux permettent d'estimer comment se répartit la variance entre les différents niveaux du jeu de données. Dans les deux exemples de l'encadré précédent, ils permettraient de répondre aux questions suivantes :

- Quel niveau explique le plus l'IMC, le niveau individuel (niveau 1) ou le niveau contextuel (niveau 2)?
- Comment se répartit la variance des notes à l'examen de mathématiques entre les trois niveaux ? A-t-on plus de variance pour les individus (niveau 1) ou au sein des classes (niveau 2) ou entre les différentes écoles (niveau 3) ?

10.1.2 Estimation des coefficients aux différents niveaux

Les modèles multiniveaux permettent d'estimer simultanément les coefficients de plusieurs variables indépendantes introduites à chacun des niveaux du modèle. Autrement dit, de voir comment les variables indépendantes introduites aux différents niveaux influencent la variable dépendante (Y) mesurée au niveau 1. Si nous reprenons l'exemple à trois niveaux (élèves/classes/écoles), plusieurs facteurs peuvent influencer la réussite ou la performance scolaire des élèves aux différents niveaux :

- **Variables indépendantes au niveau 1** (élève) : âge, sexe, statut socioéconomique, langue maternelle autre que la langue d'enseignement...
- **Variables indépendantes au niveau 2** (classe) : nombre d'élèves par classe, programme spécialisé ou pas...
- **Variables indépendantes au niveau 3** (école) : indice de défavorisation de l'école, école publique ou privée, qualité des infrastructures de l'école (bâtiment, gymnase, cour d'école)...

Dans la même veine, afin d'illustrer l'apport des modèles multiniveaux dans le champ de la géographie de la santé, Philibert et Apparicio (2007, 129) signalent que « pour un modèle à deux niveaux, il s'agit de modéliser y_{ij} , par exemple l'IMC d'un individu i (niveau 1) résidant dans un quartier j (niveau 2). Il est alors possible de mettre des variables explicatives tant au niveau 1 (âge, sexe, revenu, niveau d'éducation, etc.) qu'au niveau 2 (niveau de défavorisation sociale du quartier, offre de services et d'équipements sportifs et récréatifs, caractéristiques de l'environnement urbain, etc.). Dans cet exemple, nous pouvons voir comment la modélisation multiniveaux permet d'estimer simultanément les effets environnementaux et individuels de manière à distinguer la contribution de chacun des niveaux (ex. : l'effet du revenu des individus et celui de la défavorisation du quartier) dans l'explication des variations géographiques observées ».



Évaluer les effets de milieu avec des analyses multiniveaux

En santé des populations et en études urbaines, les modèles multiniveaux sont largement mobilisés pour évaluer les effets de milieu (*neighbourhoods effects* ou *area effects* en anglais).

Atkinson et Kintrea (2001, 2278) définissent les « effets de milieu comme le changement net dans les potentialités de l'existence (*life chances*) attribuable au fait de vivre dans un quartier (ou une zone) plutôt qu'un autre » [traduction libre]. Les effets de milieu peuvent être positifs ou négatifs et concerner aussi bien les enfants que les adultes.

Les analyses multiniveaux sont particulièrement adaptées à l'évaluation des effets de milieu. En effet, plusieurs phénomènes — état de santé, comportement ou choix individuels — peuvent être influencés à la fois par des caractéristiques individuelles (âge, sexe, niveau de revenu, niveau d'éducation, etc.) et par des caractéristiques contextuelles (caractéristiques du quartier).

Avec un modèle multiniveau, une fois contrôlées les caractéristiques individuelles (variables indépendantes mesurées au niveau 1), il est alors possible d'évaluer l'effet des caractéristiques du quartier (variables indépendantes mesurées au niveau 2) sur un phénomène y_{ij} mesuré pour un individu i résidant dans un quartier j .

10.2 Différents types de modèles multiniveaux

10.2.1 Description du jeu de données utilisé

Dans le cadre de cette section, nous présentons uniquement les modèles à deux niveaux, soit celui pour modéliser un phénomène y_{ij} . Pour ce faire, nous utilisons des données tirées d'une étude de Pham et al. (2017). Dans cet article, les auteurs souhaitent évaluer les effets des caractéristiques de la forme urbaine et des caractéristiques socioéconomiques sur la couverture des arbres de rue, et ce, à partir d'un modèle multiniveau. Ils disposent ainsi d'une structure hiérarchique de données avec deux niveaux : les tronçons de rue (niveau 1, $n = 10\,814$) inclus dans un et un seul secteur de recensement (niveau 2, $n = 312$). La variable dépendante (y_{ij}) est le pourcentage de la superficie du tronçon de rue qui est couverte par des arbres, calculé à partir d'images satellites à haute résolution (Quickbird, 60 cm, septembre 2008). L'ensemble des variables utilisées pour les modèles sont reportées au tableau 10.1.

Sept variables indépendantes relatives à la forme urbaine sont mesurées pour les tronçons de rue, soit la largeur et la longueur de la rue, l'âge médian des bâtiments (introduit également au carré pour vérifier l'existence d'un effet curvilinéaire ; voir la section 7.5.1.1), les pourcentages de bâtiments résidentiels, de duplex et de triplex, le nombre de bâtiments et finalement la distance moyenne entre le bâtiment et la rue. Les variables indépendantes pour les 312 secteurs de recensement (niveau 2) sont extraites du recensement canadien de 2006 (tableau 10.1).

Tab. 10.1 : Statistiques descriptives pour les variables des modèles multiniveaux

Nom	Intitulé	Type	Niveau	Moy.	Écart type
PCTArb	Arbres sur le tronçon de rue (%)	VD	1	7,2	10,7
Width	Largeur des rues	VI	1	16,0	7,3
Length	Longueur de rues	VI	1	136,0	87,8
AgeMed	Âge médian des bâtiments	VI	1	1 952,7	28,3
ResiPCT	Bâtiments résidentiels (%)	VI	1	83,5	28,0
DuTriPct	Duplex ou triplex (%)	VI	1	41,8	39,3
NoLog	Nombre de bâtiments	VI	1	14,0	14,4
Setback	Distance entre le bâtiment et la rue	VI	1	7,2	4,3
ValLog	Valeur moyenne des logements (milliers de dollars)	VI	2	267,6	80,0
UDipPCT	Diplômés universitaires (%)	VI	2	16,9	9,6
PCTFRAVI	Personnes à faible revenu (%)	VI	2	30,3	11,5
PCTIMGRE	Immigrants récents (%)	VI	2	10,0	7,3
AvecEnf	Ménages avec enfants (%)	VI	2	34,8	12,6
FranPCT	Langue maternelle française (%)	VI	2	66,9	24,1

10.2.2 Démarche classique pour les modèles multiniveaux

La démarche habituelle en analyse multiniveau est de réaliser plusieurs modèles, allant du plus simple au plus complexe. Cette stratégie permet habituellement de bien cerner la répartition de la variance entre les différents niveaux et l'apport des variables explicatives introduites aux différents niveaux. De la sorte, cinq types de modèles peuvent être construits :

1. Le modèle vide (appelé aussi modèle inconditionnel) qui comprend des constantes aléatoires au niveau 2, mais aucune variable explicative.
2. Le modèle avec uniquement les variables indépendantes au niveau 1 et des constantes aléatoires au niveau 2.
3. Le modèle complet avec les variables indépendantes aux deux niveaux et des constantes aléatoires.
4. Le modèle complet avec les variables indépendantes aux deux niveaux, incluant une interaction entre une variable indépendante mesurée au niveau 1 et une autre mesurée au niveau 2.
5. Le modèle avec les variables indépendantes aux deux niveaux et des constantes et pentes aléatoires.

Dans les sous-sections suivantes, nous détaillons chacun de ces cinq modèles en prenant soin de montrer les similitudes qu'ils partagent avec les modèles à effets mixtes vus précédemment. Notez d'emblée que les trois premiers modèles sont les plus fréquemment utilisés.

10.2.2.1 Modèle vide

Comme son nom l'indique, le modèle vide ne comprend aucune variable explicative. Il consiste simplement à faire varier la constante du niveau 1 avec des effets aléatoires au niveau 2, ce qui explique qu'il est souvent comparé à une ANOVA avec des effets aléatoires. En d'autres termes, ce modèle correspond à un GLMM avec constantes aléatoires dans lequel aucune variable indépendante n'est incluse au niveau 1. D'ailleurs, si vous comparez l'équation (10.1) avec l'équation (9.2) au chapitre précédent, vous constaterez que seul le paramètre $\beta_1 x_1$ a été ôté et qu'il comprend aussi deux variances : l'une fixe au niveau 1 (σ_e) et l'autre aléatoire (stochastique) au niveau 2 (σ_v).

$$\begin{aligned} Y &\sim \text{Normal}(\mu, \sigma_e) \\ g(\mu) &= \beta_0 + v \\ v &\sim \text{Normal}(0, \sigma_v) \\ g(x) &= x \end{aligned} \tag{10.1}$$

Quel est alors l'intérêt de réaliser un modèle si simple ? À partir des deux variances, il est possible de calculer le **coefficient de corrélation intraclasse** (*intraclass-correlation* (ICC) en anglais) qui est le rapport entre la variance aléatoire et la somme des variances des deux niveaux, soit fixe et aléatoire (équation (10.2)). Ce coefficient varie ainsi de 0 à 1 et indique la proportion de la variance de la variable dépendante qui est imputable au niveau 2. Tous(tes) les auteur(e)s s'entendent sur le fait qu'il est impératif de commencer une analyse de multiniveau en calculant ce modèle vide qui nous informe de la répartition de la variance entre les deux niveaux (Raudenbush et Bryk 2002; Gelman et Hill 2006; Tabachnick, Fidell et Ullman 2007; Bressoux 2010). Nous pourrons ensuite analyser l'évolution de ce coefficient dans les modèles subséquents.

$$\rho = \frac{\sigma_v}{\sigma_v + \sigma_e} \tag{10.2}$$

Les résultats du modèle vide (inconditionnel) à partir des données de Pham et al. (2017) sont présentés au tableau 10.2. La variance du niveau 1 est de 92,93 contre 19,82 au niveau 2. Le coefficient de corrélation intraclasse vaut alors : $19,82 / (19,82 + 92,93) = 0,1758$. Cela signifie que 18 % de la variance de la variable

dépendante sont imputables au niveau 2 (des secteurs de recensement) et 82 % au niveau 1 (des tronçons). Nous verrons comment évolue ce coefficient dans les modèles subséquents.

TAB. 10.2 : Résultats du modèle vide (modèle 1)

Paramètre	Coefficient	Erreur type	Valeur de T
Effets fixes (niveau 1)			
Constante	7,337	0,277	314,918
Répartition de la variance			
Variance (niveau 1)	19,818		
Variance (niveau 2)	92,925		
Coefficient de corrélation intraclasse	0,176		
Qualité d'ajustement du modèle			
AIC	80 305,219		
R ² marginal	0,000		
R ² conditionnel	0,176		

10.2.2.2 Modèle avec les variables indépendantes du niveau 1

Dans ce second modèle, nous introduisons uniquement les variables explicatives au niveau 1. Par conséquent, ce modèle est tout simplement un modèle à effets mixtes (GLMM) avec des constantes aléatoires largement décrit à la section 9.2.1). Si vous comparez l'équation du modèle vide (équation (10.2)) avec l'équation de ce modèle (équation (10.3)), vous constaterez que le paramètre $\beta_1 x_1$ a été ajouté. Il correspond au coefficient pour la variable indépendante X_1 mesurée au niveau 1 (effet fixe). Nous pourrions alors ajouter d'autres paramètres pour les autres variables indépendantes du modèle, soit $\beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$ (k étant le nombre de variables explicatives mesurées au niveau 1, effets fixes).

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma_e) \\
 g(\mu) &= \beta_0 + \beta_1 x_1 + \nu \\
 \nu &\sim Normal(0, \sigma_\nu) \\
 g(x) &= x
 \end{aligned} \tag{10.3}$$

Les résultats du second modèle sont présentés au tableau 10.3.

La répartition de la variance entre les deux niveaux. La variance du niveau 1 est désormais de 15,263 contre 80,317 au niveau 2, ce qui permet d'obtenir un coefficient de corrélation intraclasse de 0,1597. Cela signifie que de 16 % de la variance de la variable dépendante sont imputables au niveau 2 (des secteurs de recensement), une fois contrôlées les caractéristiques des tronçons.

La qualité d'ajustement du modèle. Dans le chapitre précédent sur les GLMM, nous avons largement décrit plusieurs mesures de la qualité d'ajustement du modèle, notamment l'AIC et les R² marginal et conditionnel. À titre de rappel, voici comment interpréter ces mesures :

- Plus la valeur de l'AIC est faible, mieux le modèle est ajusté. En comparant les valeurs d'AIC du modèle vide et du modèle avec les variables explicatives du niveau 1 (80 305 *versus* 78 785), nous constatons, sans surprise, que ce dernier modèle est plus performant.
- Le R² marginal indique la proportion de la variance expliquée uniquement si les effets fixes sont pris en compte (ici, 0,129). Quant au R² conditionnel, il indique la proportion de la variance expliquée à la fois par les effets fixes et aléatoires (ici, 0,268). L'écart important entre les deux R² signale que les secteurs de recensement (effets aléatoires, niveau 2) jouent un rôle important dans le modèle.

Quelles informations peut-on tirer des coefficients de régression ? Les variables indépendantes relatives à la forme urbaine les plus importantes sont : le pourcentage de bâtiments résidentiels (ResiPCT), la largeur

de la rue (`Width`) et le nombre de bâtiments (`NoLog`). Aussi, la distance entre le bâtiment et la rue (`Setback`) est associée positivement avec la variable dépendante. En effet, à chaque ajout d'un mètre de la distance moyenne entre les bâtiments et le tronçon de rue, la couverture des arbres sur le tronçon augmente de 0,202 point de pourcentage, toutes choses étant égales par ailleurs.

TAB. 10.3 : Résultats du modèle avec les variables indépendantes au niveau 1 (modèle 2)

Paramètre	Coefficient	Erreurs type	Valeur de T
Effets fixes (niveau 1)			
Constante	-1 028,618	179,736	10 799,64
Width	-0,129	0,013	10 754,96
Length	0,011	0,002	10 733,36
AgeMed	1,103	0,186	10 799,74
AgeMed2	0,000	0,000	10 799,55
ResiPCT	0,047	0,003	10 804,92
DuTriPct	-0,013	0,003	10 703,70
NoLog	0,147	0,011	10 797,85
Setback	0,202	0,023	10 797,86
Répartition de la variance			
Variance (niveau 1)	15,263		
Variance (niveau 2)	80,317		
Coefficient de corrélation intraclassé	0,160		
Qualité d'ajustement du modèle			
AIC	78 785,827		
R ² marginal	0,129		
R ² conditionnel	0,268		

Remarquez la valeur de la constante : -1028,618. À titre de rappel, la constante est la valeur que prend la variable dépendante quand toutes les variables indépendantes sont égales à 0. Or, il est impossible qu'elles soient toutes égales à zéro.



Centrage des variables quantitatives mesurées au niveau 1

En analyse multiniveau, il est très courant et souvent recommandé de centrer les variables explicatives quantitatives au niveau 1. Deux options sont alors possibles :

1. Pour une variable indépendante donnée, les observations sont centrées sur leur moyenne générale, c'est-à-dire la moyenne de l'ensemble des observations du jeu de données, soit $X_{ij} - \bar{X}$. Dans ce cas, la constante est donc la valeur que prend la variable Y quand toutes les variables indépendantes sont égales à leur moyenne respective.
2. Chaque observation est centrée sur la moyenne de son groupe respectif, soit $X_{ij} - \bar{X}_{\cdot j}$.

Tel que signalé par Bressoux (2010, 328), « dans le premier cas, la variance des pentes sera estimée pour l'individu moyen dans la distribution générale, tandis que dans le second elle est estimée pour l'individu moyen de chaque groupe ».

Autrement dit, comparativement à un modèle sans centrage, les valeurs des coefficients pour les variables indépendantes sont les mêmes dans le premier cas (seule la valeur de la constante va changer) tandis qu'elles sont différentes dans le second cas.

Attention, il ne faut pas appliquer de centrage sur une variable qualitative, qu'elle soit dichotomique, nominale ou ordinale.

Pourquoi la pratique du centrage en analyse multiniveau est si courante ?

Dans la plupart des livres sur les régressions multiniveaux, le centrage est recommandé, notamment dans l'ouvrage classique de Raudenbush et Bryk (2002). Rappelons que ces modèles sont largement utilisés en

éducation avec une structure hiérarchique classique élève/école/classe. Nous nous intéressons alors à l'individu moyen (l'élève), ce qui explique que le centrage est habituellement appliqué. Par exemple, ne pas centrer l'âge des élèves fait que la constante qui est obtenue est peu interprétable : difficile d'évaluer la note moyenne à un examen quand la variable *âge de l'élève* a la valeur de 0, tout comme les autres variables explicatives quantitatives relatives à l'élève.

Centrage et réduction de l'ensemble des variables du modèle

Il est à noter que certains auteurs centrent et réduisent l'ensemble des variables du modèle. À titre de rappel, le centrage consiste à soustraire à chaque valeur la moyenne de la variable; la réduction, à la diviser par l'écart-type de la variable (section 2.5.5.2). Pour chaque variable, la moyenne est alors égale à 0 et l'écart-type à 1. Les coefficients s'interprètent alors en termes d'augmentation d'une unité d'écart-type tant pour la VI que la VD. Ils correspondent alors à des coefficients de régression standardisés (abordés dans la section 7.4.2). Ce processus de centrage et de réduction des variables peut être motivé par des problèmes de convergence du modèle (lorsque l'algorithme d'optimisation n'arrive pas à trouver une solution pour produire les coefficients).

Par conséquent, nous vous proposons de centrer les variables du niveau 1 de notre jeu de données. Si vous comparez les tableaux 10.3 et 10.4, vous constaterez que les valeurs relatives aux coefficients, aux mesures de la répartition de la variance et à la qualité d'ajustement du modèle sont les mêmes. Seule la valeur de la constante change : elle passe de -1028,618 à 7,228. Elle s'interprète désormais de la façon suivante : si toutes les variables explicatives sont égales à leurs moyennes respectives, alors le pourcentage de la superficie du tronçon couverte par des arbres est égal à 7,228 %.

TAB. 10.4 : Résultats du modèle avec les variables indépendantes centrées au niveau 1 (modèle 2)

Paramètre	Coefficient	Erreur type	Valeur de T
Effets fixes (niveau 1)			
Constante	7,228	0,248	318,313
Width.c	-0,129	0,013	10 754,961
Length.c	0,011	0,002	10 733,361
AgeMed.c	1,103	0,186	10 801,399
AgeMed2.c	0,000	0,000	10 801,243
ResiPCT.c	0,047	0,003	10 804,923
DuTriPct.c	-0,013	0,003	10 703,746
NoLog.c	0,147	0,011	10 797,848
Setback.c	0,202	0,023	10 797,856
Répartition de la variance			
Variance (niveau 1)	15,263		
Variance (niveau 2)	80,317		
Coefficient de corrélation intraclassé	0,160		
Qualité d'ajustement du modèle			
AIC	78 785,827		
R ² marginal	0,129		
R ² conditionnel	0,268		

10.2.2.3 Modèle complet avec les variables indépendantes aux niveaux 1 et 2

Le troisième type de modèle consiste à introduire à la fois les variables explicatives mesurées au niveau 1 et au niveau 2 (équation (10.4)). Il est communément appelé le modèle complet. Les variables explicatives du niveau 2 sont aussi considérées comme des effets fixes.

$$\begin{aligned}
 Y &\sim Normal(\mu, \sigma_e) \\
 g(\mu) &= \underbrace{\beta_0 + \beta_1 x_1}_{\text{effets fixes (niveau 1)}} + \underbrace{\beta_2 z_2}_{\text{effets fixes (niveau 2)}} + \epsilon \\
 v &\sim Normal(0, \sigma_v) \\
 g(x) &= x
 \end{aligned} \tag{10.4}$$

Les résultats du troisième modèle sont présentés au tableau 10.5. Ce modèle permet d'évaluer les effets des caractéristiques socioéconomiques (mesurés au niveau des secteurs de recensement) sur la couverture des arbres des îlots, une fois contrôlées les caractéristiques de la forme urbaine des tronçons. Rappelons, que dans ce modèle, les constantes sont aléatoires et les variables indépendantes au niveau 1 sont centrées.

Quelles informations peut-on tirer des coefficients de régression du niveau 2? D'emblée, deux caractéristiques n'ont pas d'effet significatif sur la variable dépendante, soit les pourcentages de diplômés universitaires et de ménages avec enfants. Par contre, toutes choses étant égales par ailleurs, la valeur moyenne des logements et le pourcentage d'immigrants récents sont associés à une augmentation de la couverture végétale. À l'inverse, le pourcentage de personnes à faible revenu est associé à une diminution de la couverture végétale.

TAB. 10.5 : Résultats du modèle avec les variables indépendantes aux niveaux 1 et 2 (modèle 3)

Paramètre	Coefficient	Erreur type	Valeur de T
Effets fixes (niveau 1 : tronçons)			
Constante	-0,518	3,227	313,586
Width.c	-0,132	0,013	10 762,184
Length.c	0,011	0,002	10 728,713
AgeMed.c	1,097	0,185	10 783,965
AgeMed2.c	0,000	0,000	10 782,352
ResiPCT.c	0,046	0,003	10 778,050
DuTriPct.c	-0,013	0,003	10 608,724
NoLog.c	0,148	0,011	10 793,143
Setback.c	0,194	0,023	10 793,303
Effets fixes (niveau 2 : secteurs de recensement)			
ValLog	0,016	0,004	312,695
UDipPCT	0,014	0,035	329,191
PCTFRAVI	-0,088	0,030	328,151
PCTIMGRE	0,237	0,049	321,723
AvecEnf	0,001	0,032	314,079
FranPCT	0,052	0,016	316,295
Répartition de la variance			
Variance (niveau 1)	12,121		
Variance (niveau 2)	80,347		
Coefficient de corrélation intraclassé			
Qualité d'ajustement du modèle	0,131		
AIC	78 776,845		
R ² marginal	0,160		
R ² conditionnel	0,270		

10.2.2.4 Modèle avec une interaction entre deux niveaux

Dans la section 7.5.4, nous avons vu comment introduire des variables d'interaction dans une régression linéaire multiple, soit entre deux variables continues (section 7.5.4.1), soit entre une variable continue et une variable dichotomique (section 7.5.4.2), soit entre deux variables dichotomiques (section 7.5.4.3). En analyse multivariée, il peut être pertinent d'introduire une interaction entre une variable mesurée au niveau 1 et une autre mesurée au niveau 2 (équation (10.5)).

$$\begin{aligned}
Y &\sim Normal(\mu, \sigma_e) \\
g(\mu) &= \underbrace{\beta_0 + \beta_1 x_1}_{\text{effets fixes (niv. 1)}} + \underbrace{\beta_2 z_2}_{\text{effets fixes (niv. 2)}} + \underbrace{\beta_3(x_1 \times z_2)}_{\text{interaction (niv. 1 et 2)}} + \epsilon \\
v &\sim Normal(0, \sigma_v) \\
g(x) &= x
\end{aligned} \tag{10.5}$$

Dans le tableau 10.6, nous introduisons une variable d'interaction entre la *distance entre le bâtiment et la rue* (Setback.c) et le *pourcentage de personnes à faible revenu* (PCTFRAVI). On constate alors que PCTFRAVI est associé négativement avec la variable dépendante ($\beta = -0,079$, $t = -2,684$). Toutefois, lorsqu'elle est mise en interaction avec la variable Setback.c, cette variable est significative et positive ($\beta = 0,008$, $t = 4,591$).

Tab. 10.6 : Résultats du modèle avec une variable d'interaction entre les deux niveaux 1 et 2 (modèle 4)

Paramètre	Coefficient	Erreur type	Valeur de T
Effets fixes (niveau 1 : tronçons)			
Constante	-0,009	3,198	313,170
Width.c	-0,136	0,013	10 763,142
Length.c	0,011	0,002	10 730,154
AgeMed.c	1,092	0,185	10 781,041
AgeMed2.c	0,000	0,000	10 779,196
ResiPCT.c	0,046	0,003	10 778,148
DuTriPct.c	-0,013	0,003	10 592,140
NoLog.c	0,145	0,011	10 793,158
Setback.c	0,003	0,048	10 757,245
Effets fixes (niveau 2 : secteurs de recensement)			
ValLog	0,016	0,004	311,944
UDipPCT	0,009	0,035	328,840
PCTFRAVI	-0,079	0,029	332,194
PCTIMGRE	0,219	0,048	326,135
AvecEnf	-0,007	0,032	313,551
FranPCT	0,050	0,016	316,142
Variable d'interaction (niv. 1 et 2)			
Setback X PCTFRAVI	0,008	0,002	10 470,185
Répartition de la variance			
Variance (niveau 1)	11,829		
Variance (niveau 2)	80,239		
Coefficient de corrélation intraclassé	0,128		
Qualité d'ajustement du modèle			
AIC	78 768,659		
R ² marginal	0,163		
R ² conditionnel	0,270		

10.3 Conditions d'application des régressions multiniveaux

Puisque les modèles multiniveaux sont une extension des modèles à effets mixtes (GLMM), nous retrouvons globalement les mêmes conditions d'application (voir la section 9.3), dont les principales sont :

1. l'absence de multicolinéarité excessive,
2. la normalité des résidus,
3. l'absence d'observations trop influentes dans le modèle.

Combien de groupes au niveau 2? Dans la section 9.3, nous avons vu que dans un modèle GLMM, plusieurs auteur(e)s, notamment Gelman et Hill (2006), préconisent un minimum de cinq groupes dans

un modèle à effets mixtes. Toutefois, dans un modèle complet d'une régression multiniveau, nous introduisons aussi des variables indépendantes au niveau 2. Par conséquent, le nombre de groupes doit être augmenté significativement, et ce, idéalement proportionnellement au nombre des variables indépendantes ajoutées au niveau 2. En ce sens, Bressoux (2010, 325) conseille d'avoir au moins 10 groupes pour chaque variable indépendante mesurée au niveau 2. Toujours selon Bressoux (2010, 325), certains auteurs recommandent même 20 groupes par variable indépendante au niveau 2. En conséquence, bien qu'aucune règle de pouce soit clairement admise, il est clair qu'un modèle complet multiniveau nécessite un nombre de groupes conséquent.

10.4 Mise en œuvre dans R

Pour mettre en œuvre des modèles multiniveaux avec une variable dépendante continue, nous utilisons la fonction `lmer` du package `lme4`. Pour d'autres distributions, nous pouvons utiliser la fonction `glmer` implémentant différentes familles de modèles GLM, notamment binomiale (modèle multiniveau logistique), gaussien, Gamma, inverse gaussien, Poisson, Quasi-poisson, etc. Comme pour les modèles GLMM, lorsque d'autres distributions sont nécessaires, il est possible d'utiliser le package `gamlss`.

10.4.1 Le modèle vide

Dans le code R ci-dessous, la syntaxe `lmer(PCTArb ~ 1 + (1| SRNOM), data = Multiniveau)` permet de construire le modèle vide avec la variable indépendante `PCTArb` et `SRNOM` comme variable définissant les groupes au niveau 2, soit les 312 secteurs de recensement. À titre de rappel, le modèle vide ne comprend aucune variable indépendante.

```
library("lme4")
library("MuMin")
# chargement du jeu de données
load("data/multiniveau/dataArbres.RData")

# MODÈLE 1 : modèle vide (sans prédicteurs)
#-----
# Écrire Y ~ 1 signifie que le modèle est vide
# 1| SRNOM : signifie que l'on fait varier la constante avec la variable SRNOM
Modèle1 <- lmer(PCTArb ~ 1 + (1| SRNOM), data = Multiniveau)

# Nombre de groupes
cat("nombre de groupes =", length(unique(Multiniveau$SRNOM)))

## nombre de groupes = 312
```

La fonction `summary(Modèle1)` permet d'afficher les résultats du modèle. Dans la section intitulée `Random effects`, la variance pour le niveau 2 (`SRNOM (Intercept)`) est de 19,82 contre 92,93 pour le niveau 1 (`Residual`). Le coefficient de corrélation intraclasse (ICC) est donc égal à $19,82 / (19,82+92,93) \times 100 = 17,58\%$.

```
# Résultats du modèle
summary(Modèle1)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PCTArb ~ 1 + (1 | SRNOM)
```

```

##      Data: Multiniveau
##
## REML criterion at convergence: 80299.2
##
## Scaled residuals:
##      Min       1Q   Median      3Q      Max
## -1.9413 -0.5295 -0.2235  0.2175  8.4695
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## SRNOM    (Intercept) 19.82     4.452
## Residual             92.93     9.640
## Number of obs: 10814, groups: SRNOM, 312
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)  7.3373    0.2772 314.9183  26.47   <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Notez qu'il est possible d'obtenir directement la valeur de l'ICC avec la fonction `icc(Modele1)` du package `performance` et les statistiques d'ajustement du modèle avec les fonctions `logLik`, `AIC` et `BIC`.

```
# Calcul de l'ICC (coefficient intraclasse)
performance::icc(Modele1)
```

```
## # Intraclass Correlation Coefficient
##
##      Adjusted ICC: 0.176
##      Conditional ICC: 0.176
```

```
ICC1 <- performance::icc(Modele1)
cat("Part de la variance de la variable dépendante imputable au niveau 2 : ",
    round(ICC1$ICC_adjusted*100,2), "%", sep="")
```

```
## Part de la variance de la variable dépendante imputable au niveau 2 : 17.58%
```

```
# Qualité d'ajustement du modèle
cat("Statistiques d'ajustement du modèle :",
    "\n-2 Log V = ", -2*logLik(Modele1),
    "\nAIC =", AIC(Modele1),
    "\nBIC =", BIC(Modele1))
```

```
## Statistiques d'ajustement du modèle :
## -2 Log V = 80299.22
## AIC = 80305.22
## BIC = 80327.08
```

10.4.2 Modèle avec les variables indépendantes du niveau 1

Le second modèle consiste à introduire les variables indépendantes mesurées pour les tronçons de rue (niveau 1). Notez comment sont centrées préalablement les variables explicatives.

```
# Centrage des variables indépendantes
VINiv1 <- c("Width", "Length", "AgeMed", "AgeMed2", "ResiPCT", "DuTriPct", "NoLog", "Setback")
for (e in VINiv1){
  e.c <- paste(e, ".c", sep="")
  Multiniveau[[e.c]] <- Multiniveau[[e]] - mean(Multiniveau[[e]])
}

# MODÈLE 2 : modèle avec les prédicteurs au niveau 1 (rues)
# -----
Modele2 <- lmer(PCTArb ~
  # Variables indépendantes au niveau 1
  Width.c+Length.c+AgeMed.c+AgeMed2.c+ResiPCT.c+DuTriPct.c+NoLog.c+Setback.c+
  (1| SRNOM), data = Multiniveau)

# Résultats du modèle
summary(Modele2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PCTArb ~ Width.c + Length.c + AgeMed.c + AgeMed2.c + ResiPCT.c +
##           DuTriPct.c + NoLog.c + Setback.c + (1 | SRNOM)
## Data: Multiniveau
##
## REML criterion at convergence: 78763.8
##
## Scaled residuals:
##      Min     1Q Median     3Q    Max 
## -2.9065 -0.5536 -0.1941  0.2569  9.4205 
##
## Random effects:
##   Groups   Name        Variance Std.Dev. 
##   SRNOM    (Intercept) 15.26    3.907  
##   Residual             80.32    8.962  
## Number of obs: 10814, groups:  SRNOM, 312
##
## Fixed effects:
##              Estimate Std. Error          df t value Pr(>|t|)    
## (Intercept) 7.228e+00 2.479e-01 3.183e+02 29.151 < 2e-16 ***
## Width.c     -1.292e-01 1.272e-02 1.075e+04 -10.160 < 2e-16 ***
## Length.c    1.085e-02 1.717e-03 1.073e+04  6.322 2.69e-10 ***
## AgeMed.c    1.103e+00 1.856e-01 1.080e+04  5.946 2.83e-09 ***
## AgeMed2.c   -2.950e-04 4.791e-05 1.080e+04 -6.158 7.62e-10 ***
## ResiPCT.c   4.699e-02 3.466e-03 1.080e+04 13.558 < 2e-16 ***
## DuTriPct.c -1.299e-02 2.683e-03 1.070e+04 -4.842 1.30e-06 ***
## NoLog.c     1.473e-01 1.057e-02 1.080e+04 13.938 < 2e-16 ***
## Setback.c   2.018e-01 2.295e-02 1.080e+04  8.792 < 2e-16 ***
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) Wdth.c Lngth. AgMd.c AgMd2. RsPCT. DTrPc. NoLg.c
## Width.c    -0.003
## Length.c    0.011 -0.216
## AgeMed.c    0.000  0.010 -0.014
## AgeMed2.c   0.002 -0.011  0.013 -1.000
## ResiPCT.c   0.056  0.095  0.208  0.023 -0.024
## DuTriPct.c -0.010  0.022  0.086 -0.074  0.077  0.025
## NoLog.c     -0.030  0.156 -0.785 -0.008  0.009 -0.269 -0.127
## Setback.c    0.048 -0.018 -0.146  0.007 -0.008 -0.014  0.035  0.038
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling

```

```

# Calcul de l'ICC (coefficient intraclasse)
performance::icc(Modele2)

```

```

## # IntraClass Correlation Coefficient
##
##      Adjusted ICC: 0.160
## Conditional ICC: 0.139

```

```

ICC2 <- performance::icc(Modele2)
cat("Part de la variance de la variable dépendante ",
    "\nimputable au niveau 2 : ", round(ICC2$ICC_adjusted*100,2), "%", sep="")

```

```

## Part de la variance de la variable dépendante
## imputable au niveau 2 : 15.97%

```

```

# Calcul des R2 conditionnel et marginal avec les fonctions
# r.squaredGLMM ou r2_nakagawa du package performance
r.squaredGLMM(Modele2)

```

```

##           R2m         R2c
## [1,] 0.1292872 0.2683329

```

```

r2_nakagawa(Modele2)

```

```

## # R2 for Mixed Models
##
## Conditional R2: 0.268
## Marginal R2: 0.129

```

```

# Qualité d'ajustement du modèle
cat("Statistiques d'ajustement du modèle",
    "\n-2 Log L = ", -2*logLik(Modele2),
    "\nAIC =", AIC(Modele2),
    "\nBIC =", BIC(Modele2))

```

```
## Statistiques d'ajustement du modèle
## -2 Log L = 78763.83
## AIC = 78785.83
## BIC = 78866
```

10.4.3 Modèle avec les variables indépendantes aux niveaux 1 et 2

Le troisième modèle comprend à la fois les variables indépendantes mesurées aux deux niveaux (tronçons et secteurs de recensement).

```
# MODÈLE 3 : modèle complet avec les prédicteurs aux niveaux 1 et 2
# -----
Modele3 <- lmer(PCTArb ~
  # Variables indépendantes au niveau 1
  Width.c+Length.c+AgeMed.c+AgeMed2.c+ResiPCT.c+DuTriPct.c+NoLog.c+Setback.c+
  # Variables indépendantes au niveau 2
  ValLog+UDipPCT+PCTFRAVI+PCTIMGRE+AvecEnf+FranPCT+
  (1| SRNOM), data = Multiniveau)

# Résultats du modèle
summary(Modele3)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PCTArb ~ Width.c + Length.c + AgeMed.c + AgeMed2.c + ResiPCT.c +
##           DuTriPct.c + NoLog.c + Setback.c + ValLog + UDipPCT + PCTFRAVI +
##           PCTIMGRE + AvecEnf + FranPCT + (1 | SRNOM)
## Data: Multiniveau
##
## REML criterion at convergence: 78742.8
##
## Scaled residuals:
##      Min     1Q   Median     3Q    Max
## -3.0461 -0.5558 -0.1939  0.2622  9.4190
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   SRNOM   (Intercept) 12.12     3.482
##   Residual          80.35     8.964
## Number of obs: 10814, groups:  SRNOM, 312
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) -5.175e-01 3.227e+00 3.136e+02 -0.160 0.87271
## Width.c     -1.319e-01 1.271e-02 1.076e+04 -10.375 < 2e-16 ***
## Length.c     1.076e-02 1.717e-03 1.073e+04   6.265 3.87e-10 ***
## AgeMed.c     1.097e+00 1.854e-01 1.078e+04   5.920 3.31e-09 ***
## AgeMed2.c    -2.936e-04 4.785e-05 1.078e+04  -6.136 8.75e-10 ***
## ResiPCT.c    4.649e-02 3.482e-03 1.078e+04  13.352 < 2e-16 ***
## DuTriPct.c   -1.268e-02 2.677e-03 1.061e+04  -4.737 2.20e-06 ***
## NoLog.c      1.478e-01 1.057e-02 1.079e+04  13.985 < 2e-16 ***
```

```

## Setback.c    1.944e-01  2.307e-02  1.079e+04   8.428 < 2e-16 ***
## ValLog       1.591e-02  3.856e-03  3.127e+02   4.126 4.75e-05 ***
## UDipPCT      1.405e-02  3.546e-02  3.292e+02   0.396  0.69221
## PCTFRAVI    -8.837e-02  2.958e-02  3.282e+02  -2.988  0.00302 **
## PCTIMGRE     2.367e-01  4.860e-02  3.217e+02   4.870 1.76e-06 ***
## AvecEnf      5.778e-04  3.226e-02  3.141e+02   0.018  0.98572
## FranPCT      5.213e-02  1.638e-02  3.163e+02   3.183  0.00160 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling

```

```

# Qualité d'ajustement du modèle
cat("Statistiques d'ajustement du modèle",
  "\n-2 Log L = ", -2*logLik(Modele3),
  "\nAIC =", AIC(Modele3), "\nBIC =", BIC(Modele3))

```

```

## Statistiques d'ajustement du modèle
## -2 Log L = 78742.85
## AIC = 78776.85
## BIC = 78900.75

```

```

# Calcul de l'ICC (coefficient intraclasse)
performance:::icc(Modele3)

```

```

## # Intraclass Correlation Coefficient
##
##      Adjusted ICC: 0.131
## Conditional ICC: 0.110

```

```

ICC3 <- performance:::icc(Modele3)
cat("Part de la variance de la variable dépendante ",
  "\nimputable au niveau 2 : ", round(ICC3$ICC_adjusted*100,2), "%", sep="")

```

```

## Part de la variance de la variable dépendante
## imputable au niveau 2 : 13.11%

```

```

# Calcul des R2 conditionnel et marginal avec les fonctions
# r.squaredGLMM ou r2_nakagawa du package performance
r.squaredGLMM(Modele3)

```

```

##          R2m        R2c
## [1,] 0.1598477 0.269979

```

```
r2_nakagawa(Modele3)
```

```

## # R2 for Mixed Models
##
## Conditional R2: 0.270
## Marginal R2: 0.160

```

10.4.4 Modèle complet avec une interaction

Le quatrième modèle consiste à ajouter au modèle complet une interaction entre deux variables des deux niveaux.

```
# Variance d'interaction
Multiniveau$PCTFRAVI_Setback <- Multiniveau$PCTFRAVI * Multiniveau$Setback.c

# MODÈLE 4 : interaction aux deux niveaux
# -----
Modele4 <- lmer(PCTArb ~
  # Variables indépendantes au niveau 1
  Width.c+Length.c+AgeMed.c+AgeMed2.c+ResiPCT.c+DuTriPct.c+NoLog.c+Setback.c+
  # Variables indépendantes au niveau 2
  ValLog+UDipPCT+PCTFRAVI+PCTIMGRE+AvecEnf+FranPCT+
  # Variable d'interaction
  PCTFRAVI_Setback+
  (1 | SRNOM), data = Multiniveau)

# Résultats du modèle
summary(Modele4)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: PCTArb ~ Width.c + Length.c + AgeMed.c + AgeMed2.c + ResiPCT.c +
##   DuTriPct.c + NoLog.c + Setback.c + ValLog + UDipPCT + PCTFRAVI +
##   PCTIMGRE + AvecEnf + FranPCT + PCTFRAVI_Setback + (1 | SRNOM)
## Data: Multiniveau
##
## REML criterion at convergence: 78732.7
##
## Scaled residuals:
##    Min     1Q   Median     3Q    Max
## -3.0148 -0.5568 -0.1922  0.2598  9.4261
##
## Random effects:
## Groups   Name        Variance Std.Dev.
## SRNOM    (Intercept) 11.83    3.439
## Residual           80.24    8.958
## Number of obs: 10814, groups: SRNOM, 312
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) -9.484e-03 3.198e+00 3.132e+02 -0.003 0.99764
## Width.c     -1.357e-01 1.273e-02 1.076e+04 -10.660 < 2e-16 ***
## Length.c    1.079e-02 1.716e-03 1.073e+04  6.289 3.32e-10 ***
## AgeMed.c    1.092e+00 1.852e-01 1.078e+04  5.894 3.88e-09 ***
## AgeMed2.c   -2.923e-04 4.780e-05 1.078e+04 -6.114 1.00e-09 ***
## ResiPCT.c   4.608e-02 3.480e-03 1.078e+04 13.239 < 2e-16 ***
## DuTriPct.c -1.268e-02 2.674e-03 1.059e+04 -4.742 2.14e-06 ***
## NoLog.c     1.454e-01 1.057e-02 1.079e+04 13.747 < 2e-16 ***
## Setback.c   3.443e-03 4.759e-02 1.076e+04  0.072 0.94233
```

```

## ValLog           1.582e-02  3.820e-03  3.119e+02   4.141 4.46e-05 ***
## UDipPCT          9.404e-03  3.515e-02  3.288e+02   0.268  0.78920
## PCTFRAVI        -7.883e-02  2.937e-02  3.322e+02  -2.684  0.00764 **
## PCTIMGRE         2.194e-01  4.829e-02  3.261e+02   4.543 7.82e-06 ***
## AvecEnf          -7.063e-03  3.199e-02  3.136e+02  -0.221  0.82542
## FranPCT          4.987e-02  1.623e-02  3.161e+02   3.072  0.00231 **
## PCTFRAVI_Setback 8.169e-03  1.779e-03  1.047e+04   4.591 4.46e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling

# Qualité d'ajustement du modèle
cat("Statistiques d'ajustement du modèle",
  "\n-2 Log L = ", -2*logLik(Modele4),
  "\nAIC =", AIC(Modele3), "\nBIC =", BIC(Modele4))

## Statistiques d'ajustement du modèle
## -2 Log L = 78732.66
## AIC = 78776.85
## BIC = 78899.85

# Calcul de l'ICC (coefficient intraclasse)
performance:::icc(Modele4)

## # Intraclass Correlation Coefficient
##
##      Adjusted ICC: 0.128
##      Conditional ICC: 0.108

ICC4 <- performance:::icc(Modele4)
cat("Part de la variance de la variable dépendante ",
  "\nimputable au niveau 2 : ", round(ICC4$ICC_adjusted*100,2), "%", sep="")

## Part de la variance de la variable dépendante
## imputable au niveau 2 : 12.85%

# Calcul des R2 conditionnel et marginal avec les fonctions
# r.squaredGLMM ou r2_nakagawa du package performance
r.squaredGLMM(Modele4)

##          R2m      R2c
## [1,] 0.1628372 0.270394

r2_nakagawa(Modele4)

## # R2 for Mixed Models
##
##      Conditional R2: 0.270
##      Marginal R2: 0.163

```

10.4.5 Comparaison des quatre modèles

Pour comparer les modèles, nous utilisons habituellement les statistiques d'ajustement du modèle vues plus haut, soit le maximum de vraisemblance ($-2 \text{ Log-likelihood}$), l'AIC, l'ICC et les R^2 marginal et conditionnel.

```
c_logLik <- c(logLik(Modele1), logLik(Modele2), logLik(Modele3), logLik(Modele4))
ICC <- c(performance::icc(Modele1)$ICC_adjusted,
         performance::icc(Modele2)$ICC_adjusted,
         performance::icc(Modele3)$ICC_adjusted,
         performance::icc(Modele4)$ICC_adjusted)

R2m <- c(r.squaredGLMM(Modele1)[1],
          r.squaredGLMM(Modele2)[1],
          r.squaredGLMM(Modele3)[1],
          r.squaredGLMM(Modele4)[1])

R2c <- c(r.squaredGLMM(Modele1)[2],
          r.squaredGLMM(Modele2)[2],
          r.squaredGLMM(Modele3)[2],
          r.squaredGLMM(Modele4)[2])

print(data.frame(
  Modele = c("Modèle 1 (vide)",
             "Modèle 2 (VI : niv. 1)",
             "Modèle 3 (VI : niv. 1 et 2)",
             "Modèle 4 (interaction niv. 1 et 2"),
  dl = AIC(Modele1, Modele2, Modele3, Modele4)$df,
  Moins2LogLik = round(-2*c_logLik,0),
  AIC = round(AIC(Modele1, Modele2, Modele3, Modele4)$AIC,0),
  ICC = round(ICC,4),
  R2marg = round(R2m,3),
  R2cond = round(R2c,3)
))

##                                     Modele dl Moins2LogLik   AIC     ICC R2marg R2cond
## 1                      Modèle 1 (vide)  3      80299 80305 0.1758  0.000  0.176
## 2          Modèle 2 (VI : niv. 1) 11      78764 78786 0.1597  0.129  0.268
## 3          Modèle 3 (VI : niv. 1 et 2) 17      78743 78777 0.1311  0.160  0.270
## 4 Modèle 4 (interaction niv. 1 et 2) 18      78733 78769 0.1285  0.163  0.270
```

Vous constaterez ci-dessus que les valeurs d'AIC et de $-2 \log$ de vraisemblance diminuent des modèles 1 à 4, signalant une amélioration progressive des modèles. Cela se traduit aussi par une augmentation du R^2 conditionnel incluant à la fois les effets fixes et aléatoires. Sans surprise, la valeur du coefficient de corrélation intraclasse diminue du modèle vide au modèle complet : plus nous ajoutons de variables dépendantes, plus la capacité explicative du niveau 2 diminue.

Il est également judicieux de vérifier si un modèle est significativement différent du modèle précédent avec la fonction `anova` qui compare les différences de leurs déviances. En guise d'exemple, la différence de déviance de 59 ($78\ 625 - 78\ 684 = 59$) entre les modèles 3 et 2 (modèle complet *versus* modèle GLMM) avec six degrés de liberté – puisque le modèle 3 inclut six variables indépendantes de plus que le précédent ($17 - 11 = 6$) – est significative ($p < 0,001$). Cela indique que le modèle 3 est plus performant que le précédent.

```
anova(Modele1, Modele2, Modele3, Modele4)

## Data: Multiniveau
## Models:
## Modele1: PCTArb ~ 1 + (1 | SRNOM)
## Modele2: PCTArb ~ Width.c + Length.c + AgeMed.c + AgeMed2.c + ResiPCT.c +
## Modele2:     DuTriPct.c + NoLog.c + Setback.c + (1 | SRNOM)
## Modele3: PCTArb ~ Width.c + Length.c + AgeMed.c + AgeMed2.c + ResiPCT.c +
## Modele3:     DuTriPct.c + NoLog.c + Setback.c + ValLog + UDipPCT + PCTFRAVI +
## Modele3:     PCTIMGRE + AvecEnf + FranPCT + (1 | SRNOM)
## Modele4: PCTArb ~ Width.c + Length.c + AgeMed.c + AgeMed2.c + ResiPCT.c +
## Modele4:     DuTriPct.c + NoLog.c + Setback.c + ValLog + UDipPCT + PCTFRAVI +
## Modele4:     PCTIMGRE + AvecEnf + FranPCT + PCTFRAVI_Setback + (1 | SRNOM)
##          npar   AIC   BIC logLik deviance    Chisq Df Pr(>Chisq)
## Modele1    3 80304 80326 -40149      80298
## Modele2   11 78706 78786 -39342      78684 1614.351   8 < 2.2e-16 ***
## Modele3   17 78659 78783 -39313      78625   59.131   6  6.758e-11 ***
## Modele4   18 78640 78771 -39302      78604   21.166   1  4.213e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.5 Quiz de révision du chapitre

Questions

- Quels sont les deux intérêts majeurs des modèles multiniveaux ?

- Analyser la répartition de la variance entre les différents niveaux
- Introduire des variables explicatives aux différents niveaux du modèle
- construire des splines

Relisez au besoin la section [10.1](#).

- Le modèle vide comprend :

- aucune variable explicative
- plusieurs variables explicatives aux niveaux 1 et 2

Relisez au besoin la section [10.2.2.1](#).

- Un modèle avec uniquement les variables indépendantes du niveau 1 est un modèle à effets mixtes (GLMM).

- Vrai
- Faux

Relisez au besoin la section [10.2.2.2](#).

- Quelle mesure permet d'analyser la répartition de la variance entre les deux niveaux ?

- Coefficients de régression
- Coefficient de corrélation intraclassé
- AIC

Relisez au besoin la section [10.2.2.1](#).

- Un modèle complet comprend des variables explicatives aux deux niveaux.

- Vrai
- Faux

Relisez au besoin la section [10.2.2.3](#).

- Est-ce possible d'introduire une interaction entre une variable mesurée au niveau 1 et une autre mesurée au niveau 2 ?

- Vrai
- Faux

Relisez au besoin la section [10.2.2.4](#).

Réponses

- Quels sont les deux intérêts majeurs des modèles multiniveaux ?

- Analyser la répartition de la variance entre les différents niveaux
- Introduire des variables explicatives aux différents niveaux du modèle

- Le modèle vide comprend :

- aucune variable explicative

- Un modèle avec uniquement les variables indépendantes du niveau 1 est un modèle à effets mixtes (GLMM).

- Vrai

- Quelle mesure permet d'analyser la répartition de la variance entre les deux niveaux ?
 - Coefficient de corrélation intraclassé
- Un modèle complet comprend des variables explicatives aux deux niveaux.
 - Vrai
- Est-ce possible d'introduire une interaction entre une variable mesurée au niveau 1 et une autre mesurée au niveau 2 ?
 - Vrai

Chapitre 11

Modèles généralisés additifs

Dans les précédents chapitres, nous avons eu l'occasion d'explorer toute une panoplie de modèles : régressions linéaires, modèles généralisés, modèles généralisés à effets mixtes et modèles multiniveaux. Dans ce chapitre, nous abordons une nouvelle extension dans le monde des régressions : les modèles généralisés additifs (*Generalized additive model* en anglais — GAM). Cette extension a pour but de permettre de modéliser des relations non linéaires entre les variables indépendantes et la variable dépendante.



Dans ce chapitre, nous utilisons principalement les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2` le seul, l'unique!
 - * `ggsurvplot` pour combiner des graphiques et réaliser des diagrammes.
 - * `metR` pour placer des étiquettes sur des isolignes.
- Pour jouer avec des *splines* :
 - * `splines2` pour construire les fonctions de base de nombreuses *splines*.
 - * `segmented` pour ajuster des modèles avec des coefficients variant par segment.
- Pour ajuster des modèles GAM :
 - * `mgcv`, le *package* de référence pour ajuster des GAM dans R!
 - * `gamlss`, un second *package* très flexible pour ajuster des GAM.
 - * `gamlss.add`, une extension de `gamlss` ajoutant des distributions supplémentaires.
- Pour analyser des modèles GAM :
 - * `itsadug` pour notamment extraire certains résultats d'un GAM.
 - * `mixedup` pour notamment extraire les effets aléatoires d'un GAM.
 - * `DHARMa` pour le diagnostic des résidus simulés.

11.1 Introduction

Puisque les modèles GAM sont une extension des modèles GLM, ils peuvent s'appliquer à des modèles pour des variables indépendantes qualitatives, de comptage ou continues. Nous l'appliquons ici, à titre d'illustration, à une variable indépendante continue. Pour rappel, la formule décrivant un modèle linéaire généralisé (GLM) utilisant une distribution normale et une fonction de lien identitaire est la suivante :

$$\begin{aligned} Y &\sim \text{Normal}(\mu, \sigma) \\ g(\mu) &= \beta_0 + \beta X \\ g(x) &= x \end{aligned} \tag{11.1}$$

Les coefficients β permettent de quantifier l'effet des variables indépendantes (X) sur la moyenne (l'es-

pérance) (μ) de la variable dépendante (Y). Un coefficient β_k négatif indique que, si la variable X_k augmente, alors la variable Y tend à diminuer et inversement, si le coefficient est positif. L'inconvénient de cette formulation est que le modèle est capable de capturer uniquement des relations linéaires entre ces variables. Or, il existe de nombreuses situations dans lesquelles une variable indépendante a un lien non linéaire avec une variable dépendante ; voici quelques exemples :

- Si nous mesurons le niveau de bruit émis par une source sonore (variable dépendante) à plusieurs endroits et que nous tentons de prédire l'intensité sonore en fonction de la distance à la source (variable indépendante), nous pouvons nous attendre à observer une relation non linéaire entre les deux. En effet, le son étant une énergie se dispersant selon une sphère dans l'espace, son intensité est inversement proportionnelle au carré de la distance avec la source sonore.
- La concentration de la pollution atmosphérique en ville suit généralement des patrons temporels et spatiaux influencés directement par la météorologie et les activités humaines. Autrement dit, il serait absurde d'introduire l'espace de façon linéaire (avec un gradient nord-sud ou est-ouest), ou le moment de la journée de façon linéaire (comme si la pollution augmentait du matin au soir ou inversement). En guise d'exemple, la figure 11.1, tirée de Gelb et Apparicio (2020), illustre bien ces variations temporelles pour deux polluants (le dioxyde d'azote et l'ozone).

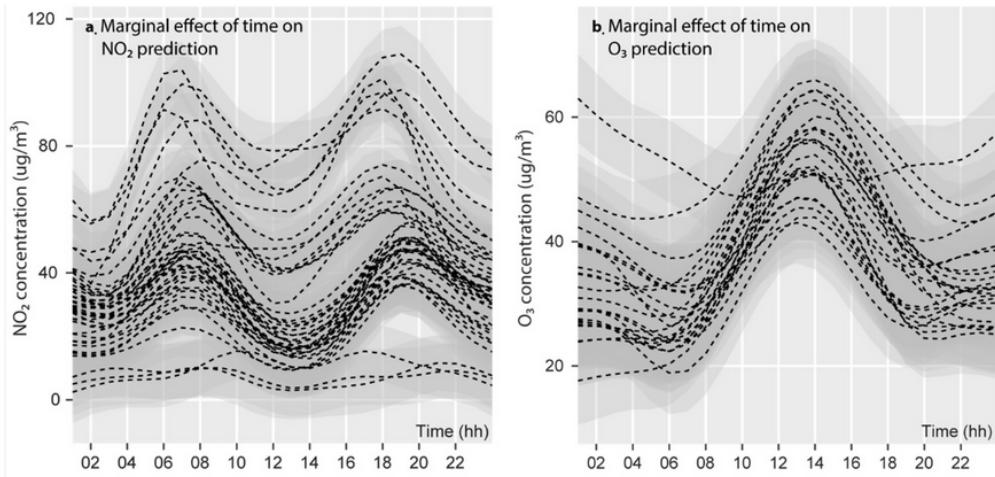


FIG. 11.1 : Patron journalier du dioxyde d'azote et de l'ozone à Paris

11.1.1 Non linéarité fonctionnelle

Il existe de nombreuses façons d'introduire des relations non linéaires dans un modèle. La première et la plus simple à mettre en œuvre est de transformer la variable indépendante à l'aide d'une fonction inverse, exponentielle, logarithmique ou autre.

Prenons un premier exemple avec une variable Y que nous tentons de prédire avec une variable X , présenté à la figure 11.2. Si nous ajustons une droite de régression à ces données (en bleu), nous constatons que l'augmentation de X est associée à une augmentation de Y . Cependant, la droite de régression est très éloignée des données et ne capte qu'une petite partie de la relation. Une lecture attentive permet de constater que l'effet de X sur Y augmente de plus en plus rapidement à mesure que X augmente. Cette forme est caractéristique d'une relation exponentielle. Nous pouvons donc transformer la variable X avec la fonction exponentielle afin d'obtenir un meilleur ajustement (en rouge).

La figure 11.3 illustre trois autres situations avec les fonctions logarithmique, logistique inverse et racine carrée. Cette approche peut donner des résultats intéressants si vous disposez d'une bonne justification théorique sur la forme attendue de la relation entre X et Y .

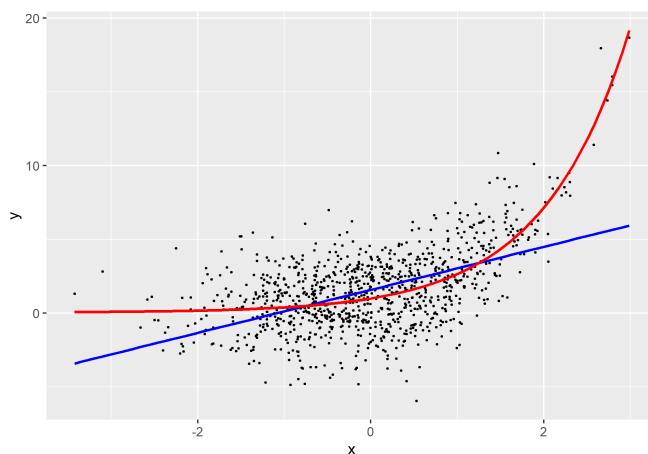


FIG. 11.2 : Relation non linéaire exponentielle

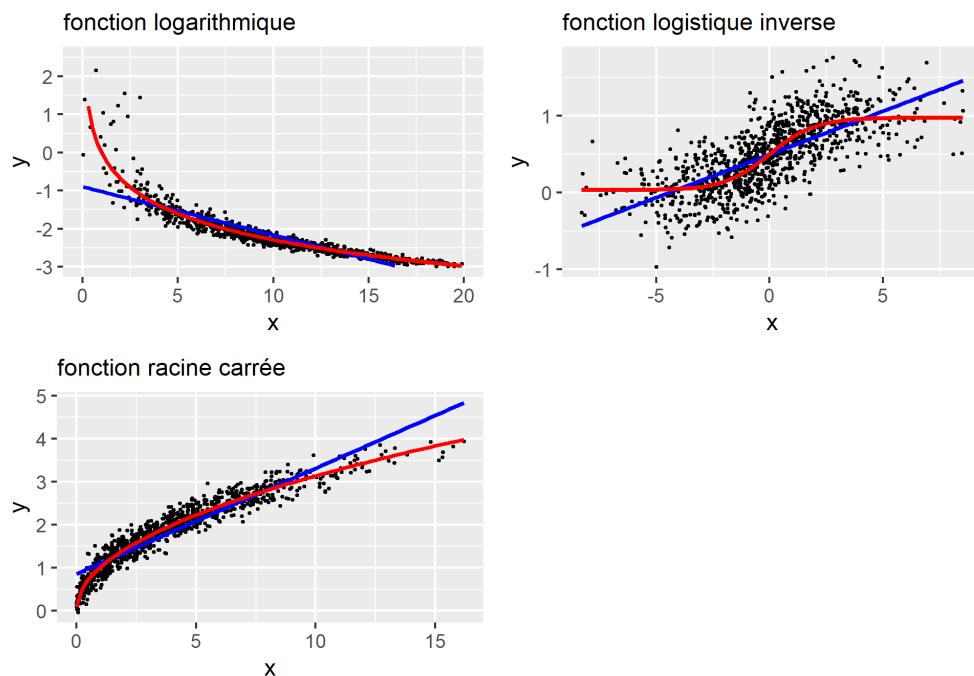


FIG. 11.3 : Autres relations non linéaires

Il existe également des cas de figure dans lesquels aucune fonction ne donne de résultats pertinents, tel qu'illustré à la figure 11.4. Nous constatons facilement qu'aucune des fonctions proposées n'est capable de bien capter la relation entre les deux variables. Puisque cette relation est complexe, il convient alors d'utiliser une autre stratégie pour la modéliser.

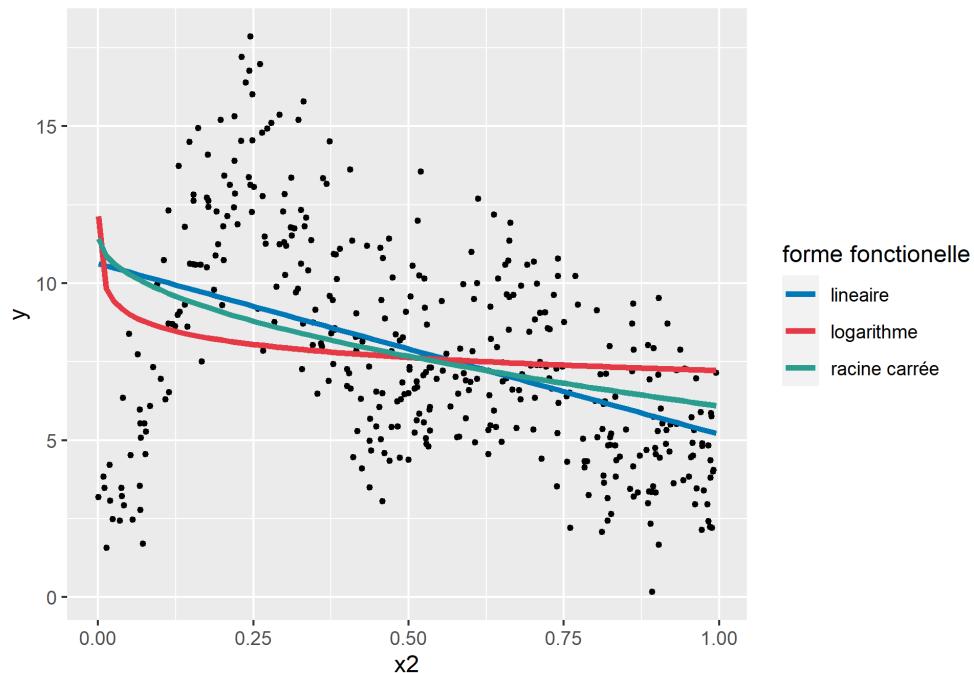


FIG. 11.4 : Relation non linéaire plus complexe

11.1.2 Non linéarité avec des polynomiales

Nous avons vu, dans le chapitre sur la régression simple (section 7.5.1.1), qu'il est possible d'utiliser des polynomiales pour ajuster des relations non linéaires. Pour rappel, il s'agit simplement d'ajouter à un modèle la variable X à différents exposants ($X + X^2 + \dots + X^k$). Chaque exposant supplémentaire (chaque ordre supplémentaire) permet au modèle d'ajuster une relation plus complexe. Rien de tel qu'un graphique pour illustrer le tout (figure 11.5).

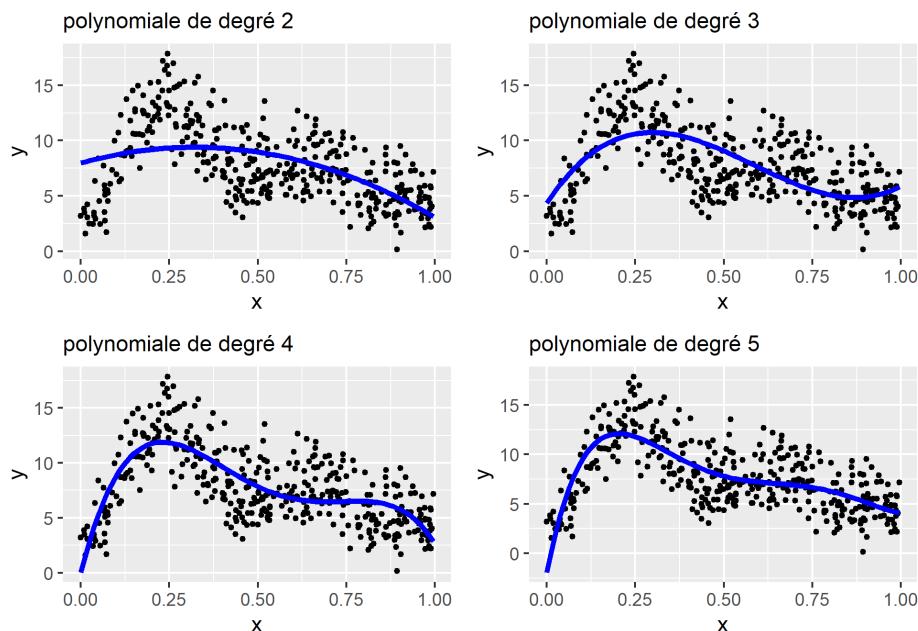


FIG. 11.5 : Visualisation de plusieurs polynomiales

L'enjeu est de sélectionner le bon nombre de degrés de la polynomiale pour le modèle. Chaque degré supplémentaire constitue une nouvelle variable dans le modèle, et donc un paramètre supplémentaire. Un trop faible nombre de degrés produit des courbes trop simplistes, alors qu'un nombre trop élevé conduit à un surajustement (*overfitting* en anglais) du modèle. La figure 11.6 illustre ces deux situations.

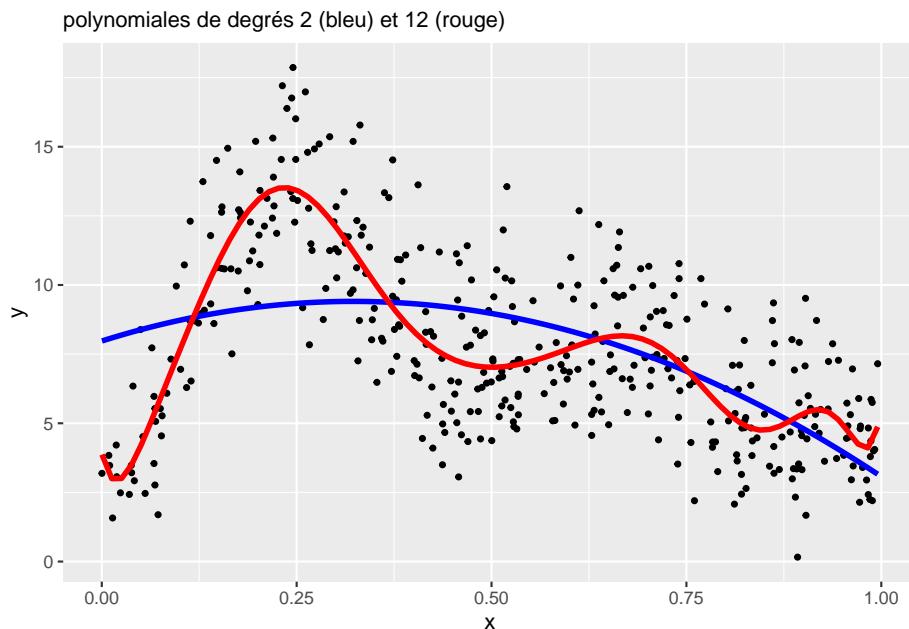


FIG. 11.6 : Sur et sous-ajustement d'une polynomiale

Un des problèmes inhérents à l'approche des polynomiales est la difficulté d'interprétation. En effet, les coefficients ne sont pas directement interprétables et seule une figure représentant les prédictions du modèle permet d'avoir une idée de l'effet de la variable X sur la variable Y.

11.1.3 Non linéarité par segments

Un compromis intéressant offrant une interprétation simple et une relation potentiellement complexe consiste à découper la variable X en segments, puis d'ajuster un coefficient pour chacun de ces segments. Nous obtenons ainsi une ligne brisée et des coefficients faciles à interpréter (figure 11.7). Nous ne présentons pas d'exemple d'application dans R, mais sachez que le package `segmented` permet d'ajuster ce type de modèle.

L'enjeu est alors de déterminer le nombre de points et la localisation de points de rupture. L'inconvénient majeur de cette approche est qu'en réalité, peu de phénomènes sont marqués par des ruptures très nettes.

À la figure 11.7, nous avons divisé la variable X en trois segments (k_1 , k_2 et k_3), définis respectivement avec les intervalles suivants : [0,00-0,22], [0,22-0,41] et [0,41-1,00]. Concrètement, cela revient à diviser la variable X en trois nouvelles variables X_{k1} , X_{k2} , et X_{k3} . La valeur de X_{ik} est égale à x_i si x_i se trouve dans l'intervalle propre à k , et à 0 autrement. Ici, nous obtenons trois coefficients :

- le premier est positif, une augmentation de X sur le premier segment est associée à une augmentation de Y;
- le second est négatif, une augmentation de X sur le second segment est associée à une diminution de Y;
- le troisième est aussi négatif, une augmentation de X sur le troisième segment est associée à une diminution de Y, mais moins forte que pour le second segment.

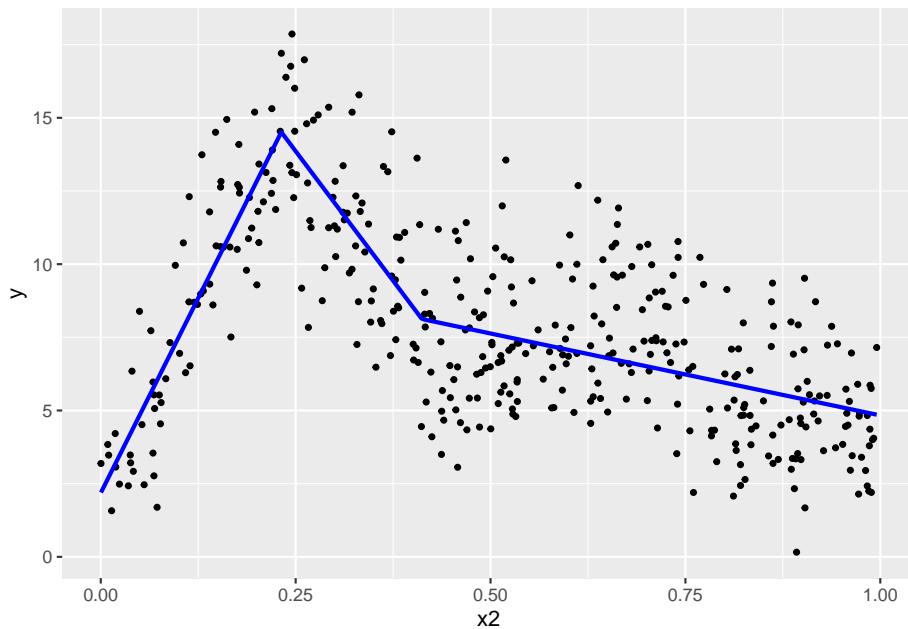


FIG. 11.7 : Régression par segment

11.1.4 Non linéarité avec des *splines*

La dernière approche, et certainement la plus flexible, est d'utiliser ce que l'on appelle une *spline* pour capter des relations non linéaires. Une *spline* est une fonction créant des variables supplémentaires à partir d'une variable X et d'une fonction de base. Ces variables supplémentaires, appelées bases (*basis* en anglais), sont ajoutées au modèle ; la sommation de leurs valeurs multipliées par leurs coefficients permet de capter les relations non linéaires entre une variable dépendante et une variable indépendante. Le nombre de bases et leur localisation (plus souvent appelé nœuds) permettent de contrôler la complexité de la fonction non linéaire.

Prenons un premier exemple simple avec une fonction de base triangulaire (*tent basis* en anglais). Nous créons ici une *spline* avec sept nœuds répartis équitablement sur l'intervalle de valeurs de la variable X. Les sept bases qui en résultent sont présentées à la figure 11.8. Dans cette figure, chaque sommet d'un triangle correspond à un nœud et chaque triangle correspond à une base.

En ajoutant ces bases dans notre modèle de régression, nous pouvons ajuster un coefficient pour chacune et le représenter en multipliant ces bases par les coefficients obtenus avec une simple régression linéaire (figure 11.9).

Nous remarquons ainsi que les bases correspondant à des valeurs plus fortes de Y ont reçu des coefficients plus élevés. Pour reconstituer la fonction non linéaire, il suffit d'additionner ces bases multipliées par leurs coefficients, soit la ligne bleue à la figure 11.10.

La fonction de base triangulaire est intéressante pour présenter la logique qui sous-tend les *splines*, mais elle est rarement utilisée en pratique. On lui préfère généralement d'autres formes donnant des résultats plus lisses comme les *B-spline* quadratiques, *B-spline* cubiques, *M-spline*, *Duchon spline*, etc.

Les approches que nous venons de décrire sont regroupées sous l'appellation de modèles additifs. Dans les prochaines sous-sections, nous nous concentrerons davantage sur les *splines* du fait de leur plus grande flexibilité.

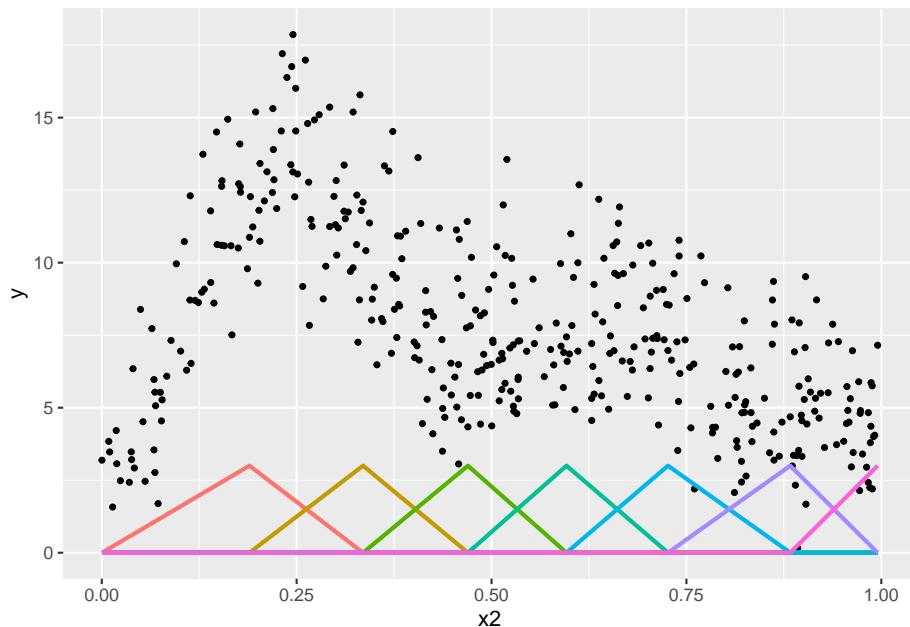


FIG. 11.8 : Bases de la spline triangulaire

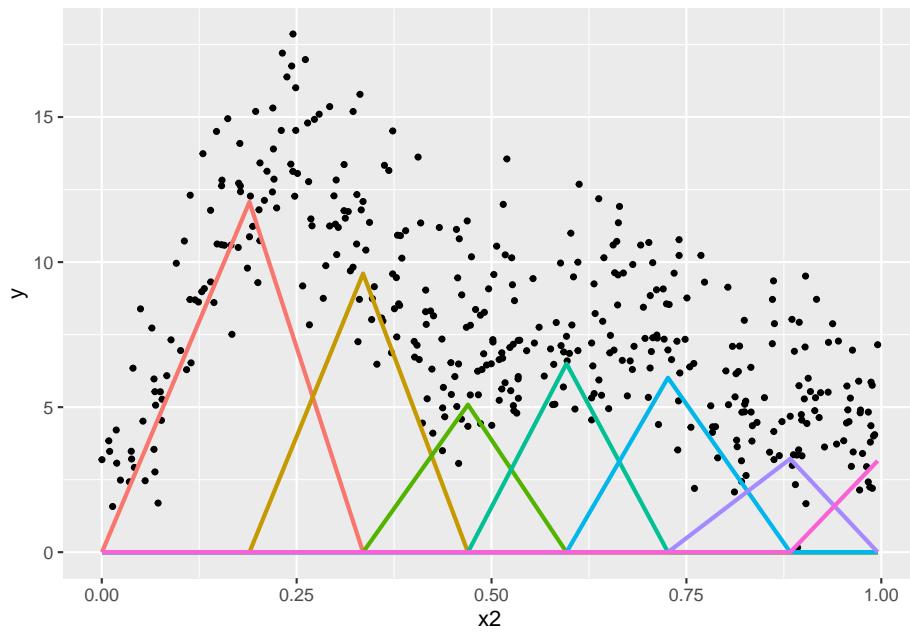


FIG. 11.9 : Spline triangulaire multipliée par ces coefficients

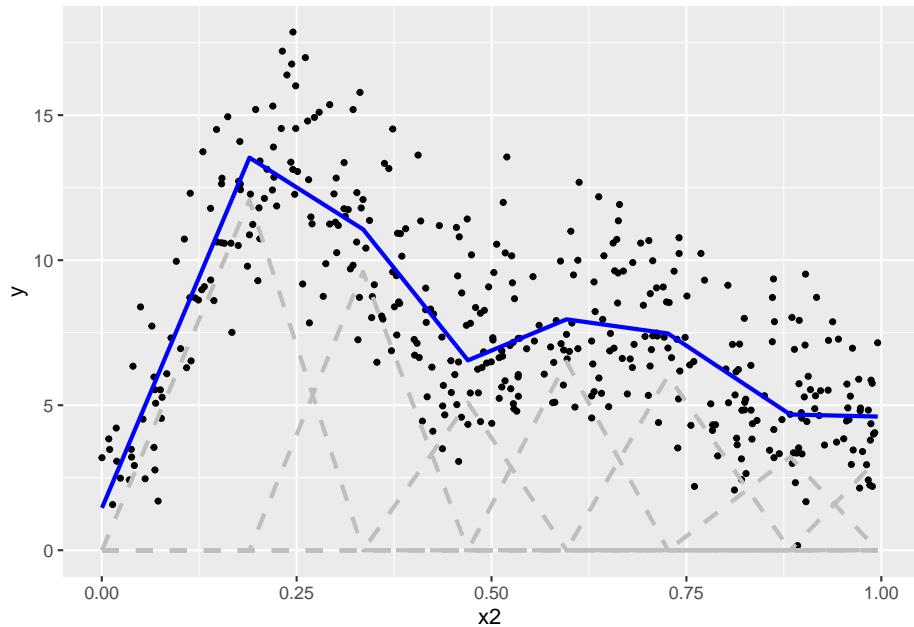


FIG. 11.10 : Spline triangulaire

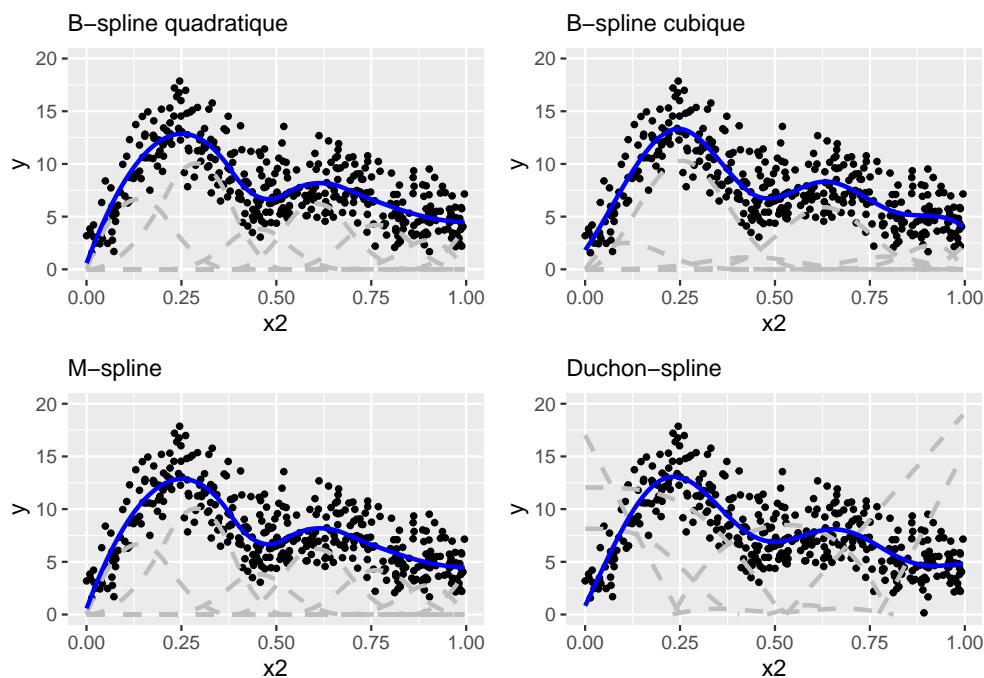


FIG. 11.11 : Comparaison de différentes bases

11.2 Spline de régression et spline de lissage

Dans les exemples précédents, nous avons vu que la construction d'une *spline* nécessite d'effectuer deux choix importants : le nombre de nœuds et leur localisation. Un trop grand nombre de nœuds conduit à un surajustement du modèle alors qu'un trop faible nombre de nœuds conduit à un sous-ajustement. Lorsque ces choix sont effectués par l'utilisateur et que les bases sont ajoutées manuellement dans le modèle tel que décrit précédemment, nous parlons alors de **splines de régression** (*Regression Spline* en anglais).

Une approche a été proposée pour faciliter le choix du nombre de nœuds, il s'agit de **splines de lissage** (*smoothing spline* en anglais). L'idée derrière cette approche est d'introduire dans le modèle une pénalisation associée avec le nombre de nœuds (ou degré de liberté) de la *spline*, dans un souci de parcimonie : chaque noeud supplémentaire doit suffisamment contribuer au modèle pour être conservé. Il n'est pas nécessaire ici de rentrer dans le détail mathématique de cette pénalisation qui est un peu complexe. Retenez simplement qu'elle dépend d'un paramètre appelé λ :

- plus λ tend vers 0, plus la pénalisation est faible et plus la *spline* de lissage devient une simple *spline* de régression ;
- à l'inverse, plus elle est forte, plus la pénalité est importante, au point que la *spline* peut se résumer à une simple ligne droite.

Cela est illustré à la figure 11.12 comprenant trois *splines* avec 20 nœuds et des valeurs λ différentes contrôlant la force de la pénalité.

Bien évidemment, nous constatons qu'avec la *spline* de régression (non pénalisée), 20 nœuds conduisent à un fort surajustement du modèle. En revanche, les *splines* de lissage (pénalisées) permettent de corriger ce problème de surajustement. Toutefois, une valeur trop importante de λ conduit à un sous-ajustement du modèle (ici $\lambda = 3$ et $\lambda = 100$, lignes verte et bleue).

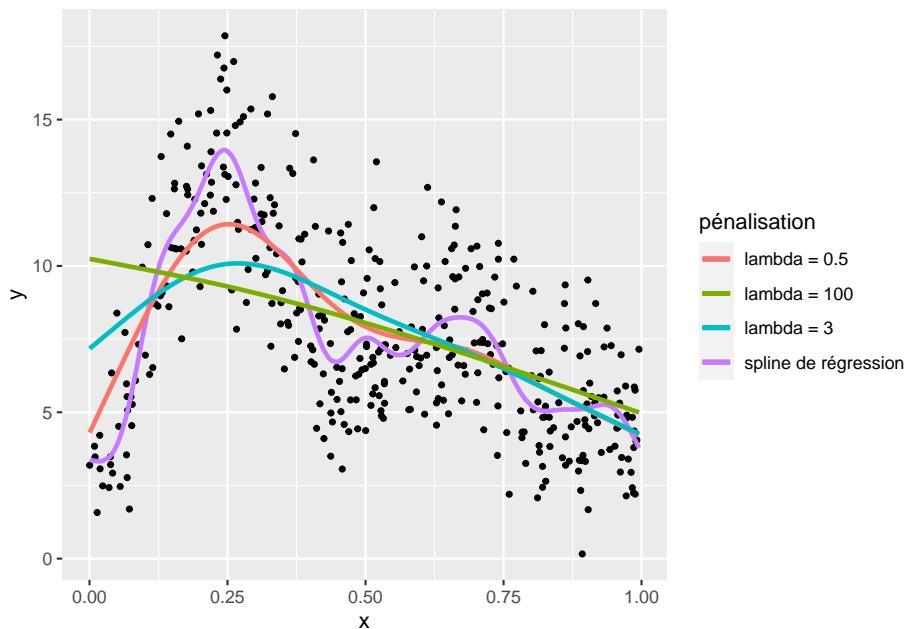


FIG. 11.12 : Pénalisation des splines

Avec les *splines* de lissage, l'enjeu est de sélectionner une valeur optimale de λ . Le plus souvent, les *packages R* **estiment eux-mêmes** ce paramètre à partir des données utilisées dans le modèle. Toutefois, gardez en mémoire que vous pouvez modifier ce paramètre. Mentionnons également que les *splines* de

lissage peuvent être reparamétrées dans un modèle pour être intégrées comme des effets aléatoires. Dans ce cas-ci, λ est remplacé par un simple paramètre de variance directement estimé dans le modèle (Wood 2004).

11.3 Interprétation d'une *spline*

L'interprétation d'une *spline* se fait à l'aide de graphiques. En effet, puisqu'elle est composée d'un ensemble de coefficients appliqués à des bases, il est difficile d'interpréter directement ces derniers. Nous préférerons alors représenter la fonction obtenue à l'aide d'un graphique, illustrant son **effet marginal**. Ce graphique est construit en trois étapes :

1. Créer un jeu de données fictif dans lequel l'ensemble des variables indépendantes sont fixées à leurs moyennes respectives, sauf la variable pour laquelle nous souhaitons représenter la *spline*. Pour cette dernière, un ensemble de valeurs allant de son minimum à son maximum est utilisé ;
2. Utiliser le modèle pour prédire les valeurs attendues de la variable dépendante pour chacune des observations fictives ainsi créées ;
3. Afficher les prédictions obtenues dans un graphique.

Notez ici qu'un graphique des effets marginaux se base sur les prédictions du modèle. Si un modèle est mal ajusté, les prédictions ne seront pas fiables et il sera inutile d'interpréter la *spline* obtenue.

Il est aussi possible, dans le cas des *splines* de lissage, d'interpréter les *estimated degrees of freedom* (EDF) qui constituent une approximation du nombre de noeuds de la *spline*. S'ils ne nous renseignent pas sur la forme de la *spline*, ils nous indiquent son niveau de complexité. Une *spline* avec un EDF de 1 est en réalité un simple terme linéaire. Plus l'EDF augmente, plus la *spline* est complexe.

11.4 Multicolinéarité non linéaire

Lorsque des *splines* sont ajoutées dans un modèle, il est nécessaire de vérifier si ces dernières ne posent pas un problème de multicolinéarité. Cependant, le VIF ne peut plus être utilisé du fait de la non-linéarité des relations modélisées. Il est alors nécessaire d'utiliser une autre mesure : la concurvité (*concurvity*) permettant de mesurer sur une échelle allant de 0 à 1 à quel point deux *splines* ont en réalité capturé le même effet et se substituent l'une à l'autre. Une valeur de 0 indique une absence totale de concurvité alors qu'une valeur de 1 indique que deux *splines* sont rigoureusement identiques (modèle non identifiable).

11.5 *Splines* avancées

Jusqu'ici, nous avons seulement présenté le cas le plus simple pour lequel une *spline* est construite à partir d'une seule variable dépendante continue, mais les *splines* peuvent être utilisées dans de nombreux autres contextes et ont une incroyable flexibilité. Nous détaillons ici trois exemples fréquents : les *splines* cycliques, les *splines* variant par groupe et les *splines* multivariées. Pour une description complète des effets non linéaires possibles avec `mgcv`, n'hésitez pas à consulter sa documentation¹.

11.5.1 *Splines* cycliques

Une *spline* cyclique est une extension d'une *spline* classique dont les bases aux extrémités sont spécifiées de telle sorte que la valeur au départ de la *spline* soit la même que celle à la fin de la *spline*. Cela permet à la *spline* de former une boucle, ce qui est particulièrement intéressant pour des variables dont le 0 et la

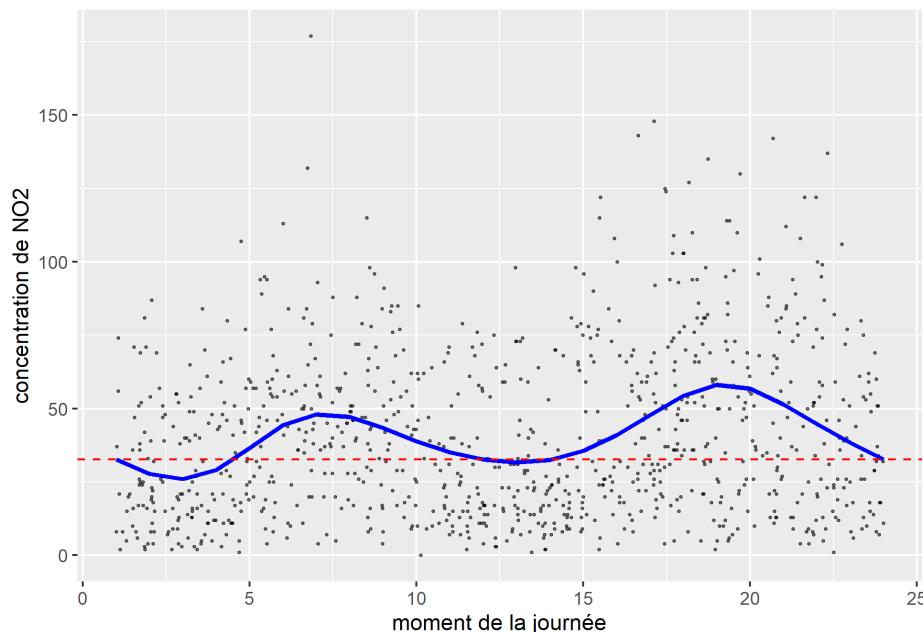
¹<https://stat.ethz.ch/R-manual/R-patched/library/mgcv/html/smooth.terms.html>

TAB. 11.1 : Exemples de splines avancées

Type	Code	Description
spline cyclique	<code>s(x, bs = 'cc')</code>	Une spline cyclique doit être utilisée si le 0 de la variable X correspond également à sa valeur maximum. Un bon exemple est le temps dans une journée car 24 h est équivalent à 0 h
spline variant par groupe	<code>s(x, by = x2)</code>	Une spline variant par groupe permet d'ajuster une spline à une variable X1 différente pour chaque groupe identifié par une variable qualitative X2
spline bivariée	<code>s(x1,x2)</code>	Une spline bivariée est utilisée pour modéliser l'interaction non linéaire de deux variables X1 et X2 s'exprimant dans la même unité (typiquement des coordonnées géographiques cartésienne)
spline d'interaction complète	<code>te(x1,x2)</code>	Une spline d'interaction permet de modéliser l'interaction non linéaire de deux variables continues pouvant s'exprimer dans des unités différentes, elle combine les effets spécifiques de chacune des deux variables et leur interaction
spline d'interaction partielle	<code>s(x1) + s(x2) + ti(x1,x2)</code>	Une spline d'interaction partielle permet de distinguer les effets non linéaires individuels de deux variables de leur interaction non linéaire

valeur maximale correspondent en réalité à la même valeur. L'exemple le plus parlant est certainement le cas d'une variable représentant la mesure d'un angle en degrés. Les valeurs de 0 et 360 sont identiques et les valeurs 350 et 10 sont toutes les deux à une distance de 10 degrés de 0. Un autre exemple possible serait de considérer l'heure comme une variable continue; dans ce cas, 24 h et 0 h signifient la même chose.

Prenons un exemple concret. Nous souhaitons modéliser la concentration de dioxyde d'azote (NO_2) à Paris, mesurée par un ensemble de stations fixes. Nous pourrions nous attendre à ce que le NO_2 suive chaque jour un certain patron. Concrètement, à proximité d'axes routiers majeurs, nous nous attendons à observer des pics suivant les flux pendulaires. À la figure 11.13, nous retrouvons bien les deux pics attendus correspondant aux heures de pointe du matin et du soir. Aussi, tel qu'indiqué par la ligne rouge, la valeur prédite par la *spline* est la même à 24 h et à 0 h.

**FIG. 11.13 :** Spline cyclique pour modéliser la concentration de dioxyde d'azote

11.5.2 Splines par groupe

Tel qu'abordé dans les chapitres précédents, il arrive régulièrement que les observations appartiennent à différents groupes. Dans ce cas de figure, nous pouvons être amené à vérifier si la relation décrite par une *spline* est identique pour chacun des groupes d'observations. Il s'agit alors d'ajuster une *spline* différente par groupe. Dans l'exemple précédent, chaque valeur de NO₂ a été mesurée par une station fixe de mesure spécifique. Compte tenu du fait que l'environnement autour de chaque station est particulier, nous pourrions s'attendre à ce que les valeurs de NO₂ ne présentent pas exactement les mêmes patrons journaliers pour chaque station.

À la figure 11.14, il est possible de constater que le NO₂ suit globalement le même patron temporel pour l'ensemble des stations à l'exception de trois d'entre-elles. Il s'agit en réalité de stations situées dans des secteurs ruraux de la région parisienne, et donc moins impactées par le trafic routier.

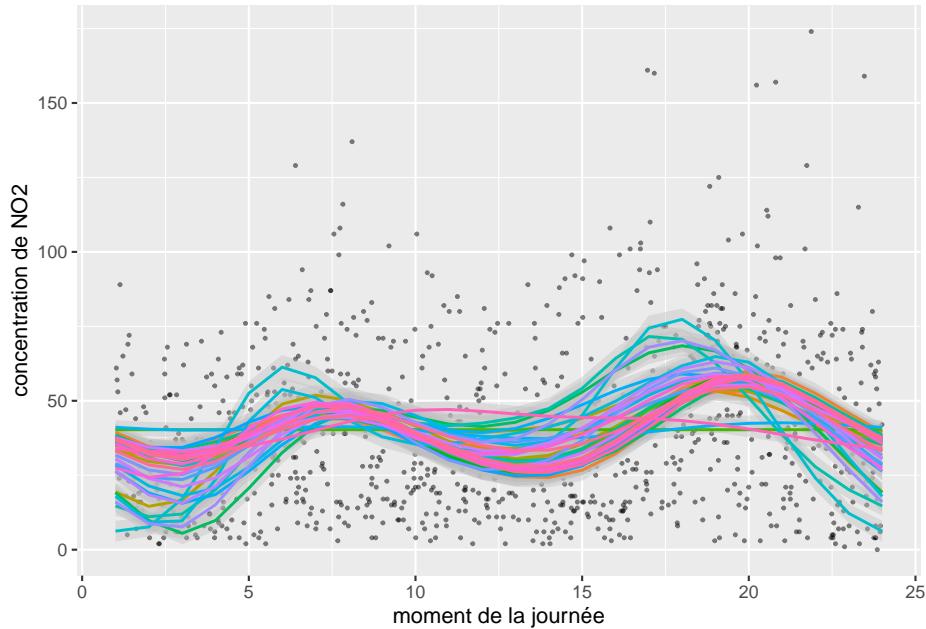


FIG. 11.14 : Spline cyclique variant par groupe

11.5.3 Splines multivariées et splines d'interaction

Jusqu'ici, nous n'avons considéré que des *splines* ne s'appliquant qu'à une seule variable indépendante ; cependant, il est possible de construire des *splines* multivariées s'ajustant simultanément sur plusieurs variables indépendantes. L'objectif est alors de modéliser les potentielles interactions non linéaires entre les variables indépendantes combinées dans une même *spline*. Prenons un exemple concret, dans la section sur les modèles GLM, nous avons modélisé la couverture des aires de diffusion (AD) à Montréal par des îlots de chaleur. Parmi les variables indépendantes, nous avons notamment utilisé la distance au centre-ville ainsi que la part de la surface végétalisée des AD. Nous pourrions formuler l'hypothèse que ces deux variables influencent conjointement et de façon non linéaire la proportion de la surface d'îlot de chaleur dans chaque AD. Pour représenter une *spline* sur plusieurs dimensions, nous utilisons alors une carte de chaleur dont la couleur représente la valeur de la variable dépendante prédite en fonction des deux variables indépendantes.

Il est important de distinguer la **spline d'interaction** et la **spline multivariée**. La première est utilisée lorsque les variables indépendantes introduites dans la *spline* ne sont pas exprimées sur la même échelle et n'évoluent pas conjointement. L'exemple donné ci-dessus avec les variables de végétation et de distance

au centre-ville est un exemple de *spline* d'interaction, la première variable étant exprimée en pourcentage et l'autre en mètres. De plus, ces deux variables ne sont pas conjointes, mais bien distinctes l'une de l'autre. Un cas typique où une *spline* multivariée serait à privilégier est le cas de l'ajout des coordonnées spatiales dans le modèle. L'emplacement des AD est mesuré par deux variables (coordonnées spatiales x et y) toutes les deux exprimées en mètres évoluant conjointement, au sens où les coordonnées x n'interagissent pas avec les coordonnées y , mais forment à elles deux un espace propre. Au-delà de la problématique de l'échelle des données, il est important de retenir que les *splines* d'interaction tendent à être davantage pénalisées que les *splines multivariées*.

La *spline* d'interaction représentée à la figure 11.15 indique que les AD avec la plus grande proportion de leur surface couverte par des îlots de chaleur sont situées à moins de 25 kilomètres du centre-ville, au-delà de cette distance, cette proportion chute en bas de 0,1, soit 10 % de la surface de l'AD. En revanche, à proximité du centre-ville (moins d'un kilomètre), même les AD disposant d'un fort pourcentage de surface végétalisée sont tout de même marquées par un fort pourcentage de surface couverte par des îlots de chaleur.

Les *splines bivariées* sont fréquemment utilisées pour capturer un potentiel patron spatial dans les données. En effet, si nous disposons des coordonnées spatiales de chaque observation (x,y), il est possible d'ajuster une *spline bivariée* sur ces coordonnées, contrôlant ainsi l'effet de l'espace.

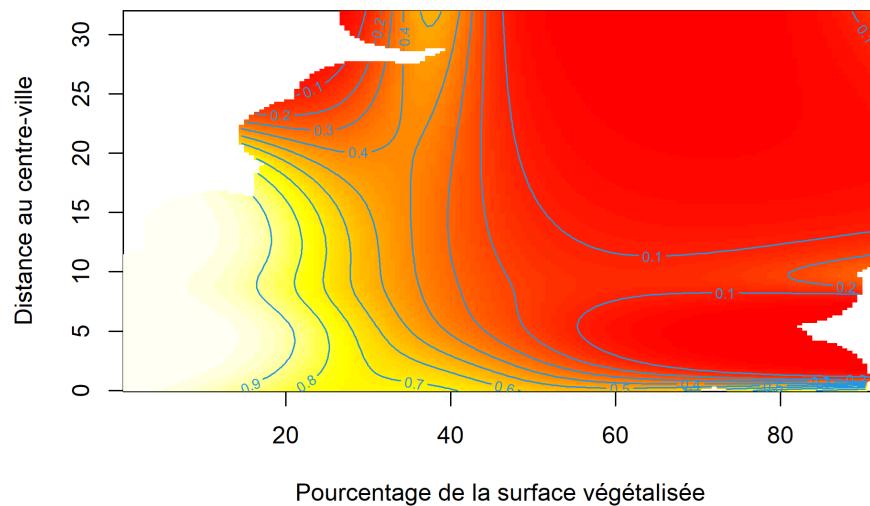


FIG. 11.15 : Spline d'interaction bivariée

Il n'y a pas de limite théorique au nombre de variables qui peuvent être ajoutées dans une *spline* d'interaction ou multivariée. Notez cependant que plus le nombre de dimensions augmente, plus la fonction à estimer est complexe et plus le volume de données nécessaire est grand et doit couvrir densément l'ensemble de l'espace d'échantillonnage multidimensionnel.

11.6 Mise en oeuvre dans R

Il est possible d'ajuster des *splines* de régression dans n'importe quel *package* permettant d'ajuster des coefficients pour un modèle de régression. Il suffit de construire les bases des *splines* en amont à l'aide du *package splines2* et de les ajouter directement dans l'équation de régression. En revanche, il est nécessaire

d'utiliser des *packages* spécialisés pour ajuster des *splines* de lissage. Parmi ceux-ci, `mgcv` est probablement le plus populaire du fait de sa (très) grande flexibilité, suivi des *packages* `gamLSS`, `gam` et `VGAM`. Nous comparons ici les deux approches, puis nous tentons d'améliorer le modèle que nous avons ajusté pour prédire le pourcentage de surface couverte par des îlots de chaleur dans les aires de diffusion de Montréal, dans une perspective d'équité environnementale. Pour rappel, la variable dépendante est exprimée en pourcentage et nous utilisons une distribution bêta pour la modéliser.

```
library(mgcv)
# Chargement des données
dataset <- read.csv("data/gam/data_chaleur.csv", fileEncoding = "utf8")
# Ajustement du modèle de base
refmodel <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  poly(prt_veg, degree = 2) + Arrond,
  data = dataset, family = betar(link = "logit"))
```

Dans notre première analyse de ces données, nous avons ajusté une polynomiale d'ordre 2 pour représenter un potentiel effet non linéaire de la végétation sur les îlots de chaleur. Nous remplaçons à présent ce terme par une *spline* de régression en sélectionnant quatre nœuds.

```
library(splines2)
# Création des bases de la spline
basis <- bSpline(x = dataset$prt_veg, df = 4, intercept = FALSE)
# Ajouter les bases au DataFrame
basisdf <- as.data.frame(basis)
names(basisdf) <- paste('spline', 1:ncol(basisdf), sep = '')
dataset <- cbind(dataset, basisdf)
# Ajuster le modèle
model0 <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  spline1 + spline2 + spline3 + spline4 + Arrond,
  data = dataset, family = betar(link = "logit"))
```

Nous pouvons à présent ajuster une *spline* de lissage et laisser `mgcv` déterminer son niveau de complexité.

```
# Ajustement du modèle avec une spline simple
model1 <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  s(prt_veg) + Arrond,
  data = dataset, family = betar(link = "logit"))
```

Notez ici que la syntaxe à employer est très simple, il suffit de spécifier `s(prt_veg)` pour indiquer à la fonction `gam` que vous souhaitez ajuster une *spline* pour la variable `prt_veg`. Nous pouvons à présent comparer l'ajustement des deux modèles en utilisant la mesure de l'AIC.

```
# Comparaison des AIC
AIC(refmodel, model0, model1)

##          df      AIC
## refmodel 40.00000 -6399.784
## model0   42.00000 -6419.630
## model1   44.61065 -6417.562
```

Nous constatons que la valeur de l'AIC du second modèle est plus réduite, indiquant un meilleur ajustement du modèle avec une *spline* de régression. Notons cependant que la différence avec la *spline* de lissage est anecdotique (deux points de l'AIC) et que nous connaissons a priori le bon nombre de noeuds à utiliser. Pour des relations plus complexes, les *splines* de lissage ont tendance à nettement mieux performer. Voyons à présent comment représenter ces trois termes non linéaires.

```
# Création d'un DataFrame de prédiction dans lequel seule
# la variable prt_veg varie.
dfpred <- data.frame(
  prt_veg = seq(min(dataset$prt_veg), max(dataset$prt_veg), 0.5),
  A65Pct = mean(dataset$A65Pct),
  A014Pct = mean(dataset$A014Pct),
  PopFRPct = mean(dataset$PopFRPct),
  PopMVPct = mean(dataset$PopMVPct),
  Arrond = "Verdun"
)

# Recréation des bases de la spline de régression
# pour les nouvelles observations
nvl_bases <- data.frame(predict(basis,newx = dfpred$prt_veg))
names(nvl_bases) <- paste('spline',1:ncol(basisdf),sep='')
dfpred <- cbind(dfpred, nvl_bases)

# Définition de la fonction inv.logit, soit l'inverse de la fonction
# de lien du modèle pour retrouver les prédictions dans l'échelle
# originales des données
inv.logit <- function(x){exp(x)/(1+exp(x))}

# Utilisation des deux modèles pour effectuer les prédictions
predref <- predict(refmodel, newdata = dfpred, type = 'link', se.fit = T)
predmod0 <- predict(model0, newdata = dfpred, type = 'link', se.fit = T)
predmod1 <- predict(model1, newdata = dfpred, type = 'link', se.fit = T)

# Calcul de la valeur prédite et construction des intervalles de confiance
dfpred$polypred <- inv.logit(predref$fit)
dfpred$poly025 <- inv.logit(predref$fit - 1.96 * predref$se.fit)
dfpred$poly975 <- inv.logit(predref$fit + 1.96 * predref$se.fit)

dfpred$regsplinepred <- inv.logit(predmod0$fit)
dfpred$regspline025 <- inv.logit(predmod0$fit - 1.96 * predmod0$se.fit)
dfpred$regspline975 <- inv.logit(predmod0$fit + 1.96 * predmod0$se.fit)

dfpred$splinepred <- inv.logit(predmod1$fit)
dfpred$spline025 <- inv.logit(predmod1$fit - 1.96 * predmod1$se.fit)
dfpred$spline975 <- inv.logit(predmod1$fit + 1.96 * predmod1$se.fit)

# Créer un graphique pour afficher les résultats
ggplot(dfpred) +
  geom_ribbon(aes(x = prt_veg, ymin = poly025, ymax = poly975),
              alpha = 0.4, color = 'grey') +
  geom_ribbon(aes(x = prt_veg, ymin = spline025, ymax = spline975),
              alpha = 0.4, color = 'grey') +
  geom_ribbon(aes(x = prt_veg, ymin = regspline025, ymax = regspline975),
              alpha = 0.4, color = 'grey') +
  geom_line(aes(y = polypred, x = prt_veg, color = 'polynomiale')),
```

```

    size = 1) +
geom_line(aes(y = regssplinepred, x = prt_veg, color = 'spline de régression'),
          size = 1) +
geom_line(aes(y = splinepred, x = prt_veg, color = 'spline de lissage'),
          size = 1)

```

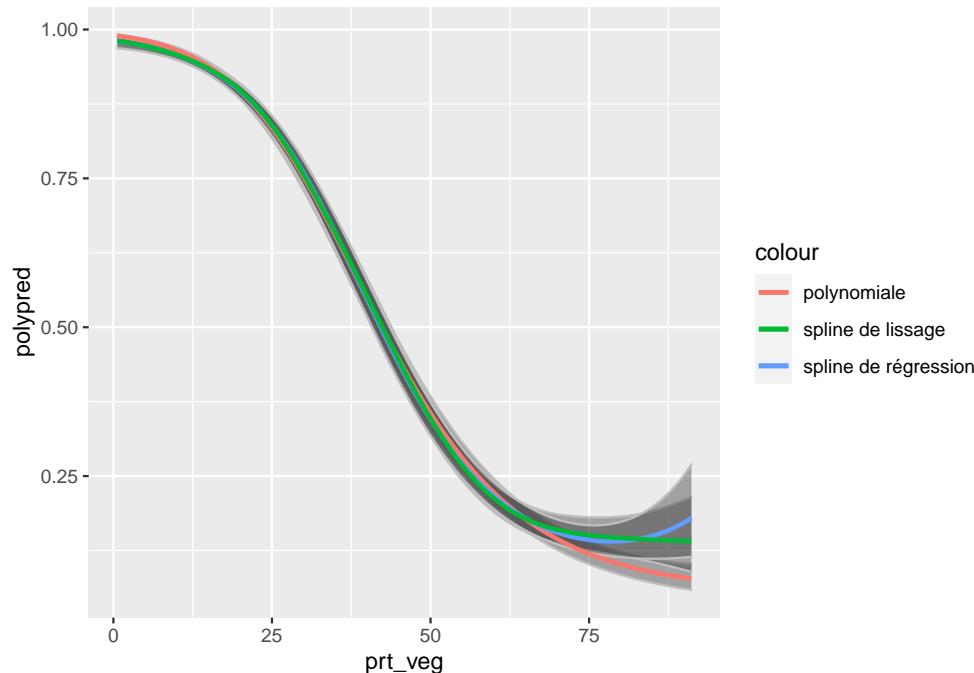


FIG. 11.16 : Comparaison d'une spline et d'une polynomiale

Nous constatons que les trois termes renvoient des prédictions très similaires et qu'une légère différence n'est observable que pour les secteurs avec les plus hauts niveaux de végétation (supérieurs à 75 %).

Jusqu'ici, nous utilisons l'arrondissement dans lequel est comprise chaque aire de diffusion comme une variable nominale afin de capturer la dimension spatiale du jeu de données. Puisque nous avons abordé la notion de *splines* bivariées, il serait certainement plus efficace d'en construire une à partir des coordonnées géographiques (x,y) des centroïdes des aires de diffusion. En effet, il est plus probable que la distribution des îlots de chaleur suive un patron spatial continu sur le territoire plutôt que les délimitations arbitraires des arrondissements.

```

# Ajustement du modèle avec une spline bivariée pour l'espace
model2 <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  s(prt_veg) + s(X,Y),
  data = dataset, family = betar(link = "logit"))

```

Notez ici que l'expression $s(X,Y)$ permet de créer une *spline* bivariée à partir des coordonnées (x,y), soit deux colonnes présentes dans le jeu de données. Ces coordonnées sont exprimées toutes deux en mètres et n'interagissent pas ensemble au sens strict, nous devons donc ajuster une *spline* bivariée. Si vous avez besoin d'ajuster une *spline* d'interaction (notamment quand les variables sont dans des unités différentes), il est nécessaire d'utiliser une autre syntaxe $te(x,y)$ ou $t2(x,y)$ faisant appel à une structure mathématique légèrement différente, soit des *tensor product smooths*.

Puisque notre modèle intègre deux *splines*, nous devons nous assurer que nous n'avons pas de problème de concurivité, ce que nous pouvons faire avec la fonction `concurvity` du package `mgcv`.

```
values <- concurvity(model2, full = FALSE)

# Worst, estimation pessimiste de la concurivité
round(values$worst,3)

##          para s(prt_veg) s(X,Y)
## para      1     0.000  0.000
## s(prt_veg) 0     1.000  0.458
## s(X,Y)     0     0.458  1.000

# Observed, estimation optimiste de la concurivité
round(values$observed,3)

##          para s(prt_veg) s(X,Y)
## para      1     0.000  0.000
## s(prt_veg) 0     1.000  0.154
## s(X,Y)     0     0.403  1.000

# Estimate, estimation entre deux de la concurivité
round(values$estimate,3)

##          para s(prt_veg) s(X,Y)
## para      1     0.000  0.000
## s(prt_veg) 0     1.000  0.142
## s(X,Y)     0     0.358  1.000
```

Nous pouvons ainsi constater des niveaux de concurivité tout à fait acceptables dans notre modèle. Des valeurs supérieures à 0,8 devraient être considérées comme alarmantes, surtout si elles sont reportées pour `observed` et `estimate`.

Voyons désormais, le résumé d'un modèle GAM tel que présenté dans R.

```
summary(model2)

##
## Family: Beta regression(15.469)
## Link function: logit
##
## Formula:
## hot ~ A65Pct + A014Pct + PopFRPct + PopMVPct + s(prt_veg) + s(X,
##       Y)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.6050031  0.0645191 -9.377  < 2e-16 ***
## A65Pct      0.0027671  0.0014072   1.966   0.0493 *
## A014Pct     -0.0019040  0.0027674  -0.688   0.4914
```

```

## PopFRPct      0.0095992  0.0014323   6.702 2.06e-11 ***
## PopMVPct      0.0010113  0.0008159   1.239   0.2152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df Chi.sq p-value
## s(prt_veg) 6.38    7.565   6731 <2e-16 ***
## s(X,Y)     27.10   28.764   1349 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.891   Deviance explained = 90.8%
## -REML = -3234.3  Scale est. = 1           n = 3157

```

La première partie du résumé comprend les résultats pour les effets fixes et linéaires du modèle. Ils s'interprètent comme pour ceux d'un GLM classique. La seconde partie présente les résultats pour les termes non linéaires. La valeur de p permet de déterminer si la *spline* a ou non un effet différent de 0. Une valeur non significative indique que la *spline* ne contribue pas au modèle. Les colonnes *edf* et *Ref.df* indiquent la complexité de la *spline* et peuvent être considérées comme une approximation du nombre de noeuds. Dans notre cas, la *spline* spatiale (*s(X,Y)*) est environ 5 fois plus complexe que la *spline* ajustée pour la végétation (*s(prt_veg)*). Cela n'est pas surprenant puisque la dimension spatiale (*spline* bivariée) du phénomène est certainement plus complexe que l'effet de la végétation. Notez ici que des valeurs *edf* et *Ref.df* proches de 1 signaleraient que l'effet d'un prédicteur est essentiellement linéaire et qu'il n'est pas nécessaire de recourir à une *spline* pour cette variable.

La dernière partie du résumé comprend deux indicateurs de qualité d'ajustement, soit le R^2 ajusté et la part de la déviance expliquée.

```
AIC(refmodel, model1, model2)
```

```

##             df      AIC
## refmodel 40.00000 -6399.784
## model1   44.61065 -6417.562
## model2   40.06053 -6596.884

```

Nous pouvons constater que le fait d'introduire la *spline* spatiale dans le modèle contribue à réduire encore la valeur de l'AIC, et donc à améliorer le modèle. À ce stade, nous pourrions tenter de forcer la *spline* à être plus complexe en augmentant le nombre de noeuds.

```

# Augmentation de la complexité de la spline spatiale
model3 <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  s(prt_veg) + s(X,Y,k = 40),
  data = dataset, family = betar(link = "logit"))

```

```
AIC(refmodel, model1, model2, model3)
```

```

##             df      AIC
## refmodel 40.00000 -6399.784
## model1   44.61065 -6417.562

```

```
## model2    40.06053 -6596.884
## model3    48.28633 -6639.955
```

Cela a pour effet d'améliorer de nouveau le modèle. Pour vérifier si l'augmentation du nombre noeuds est judicieuse, il est possible de représenter le résultat des deux *splines* précédentes. Pour ce faire, nous proposons de calculer les valeurs prédictes de la *spline* pour chaque localisation dans notre terrain d'étude, en le découplant préalablement en pixels de 100 de côté. Pour cette prédiction, nous maintenons toutes les autres variables à leur moyenne respective afin d'évaluer uniquement l'effet de la *spline* spatiale.

```
library(viridis)
library(metR) # pour placer des étiquettes sur les isolignes

# Création d'un DataFrame fictif pour les prédictions
dfpred <- expand.grid(
  prt_veg = mean(dataset$prt_veg),
  A65Pct = mean(dataset$A65Pct),
  A014Pct = mean(dataset$A014Pct),
  PopFRPct = mean(dataset$PopFRPct),
  PopMVPct = mean(dataset$PopMVPct),
  X = seq(min(dataset$X), max(dataset$X), 100),
  Y = seq(min(dataset$Y), max(dataset$Y), 100)
)

dfpred$predicted1 <- predict(model2, newdata = dfpred, type = 'response')
dfpred$predicted2 <- predict(model3, newdata = dfpred, type = 'response')

# Centrage des prédictions
dfpred$predicted1 <- dfpred$predicted1 - mean(dfpred$predicted1)
dfpred$predicted2 <- dfpred$predicted2 - mean(dfpred$predicted2)

# Représentation des splines
plot1 <- ggplot(dfpred) +
  geom_raster(aes(x = X, y = Y, fill = predicted1)) +
  geom_point(aes(x = X, y = Y),
             size = 0.2, alpha = 0.4,
             color = 'black', data = dataset) +
  geom_contour(aes(x = X, y = Y, z = predicted1), binwidth = 0.1,
               color = 'white', linetype = 'dashed') +
  geom_text_contour(aes(x = X, y = Y, z = predicted1),
                   color = 'white', binwidth = 0.1) +
  scale_fill_viridis() +
  coord_cartesian() +
  theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank()
      ) +
  labs(subtitle = 'spline de base', fill = "prédictions")

plot2 <- ggplot(dfpred) +
  geom_raster(aes(x = X, y = Y, fill = predicted2)) +
  geom_point(aes(x = X, y = Y),
             size = 0.2, alpha = 0.4,
             color = 'black', data = dataset) +
  geom_contour(aes(x = X, y = Y, z = predicted2),
               binwidth = 0.1, color = 'white', linetype = 'dashed') +
```

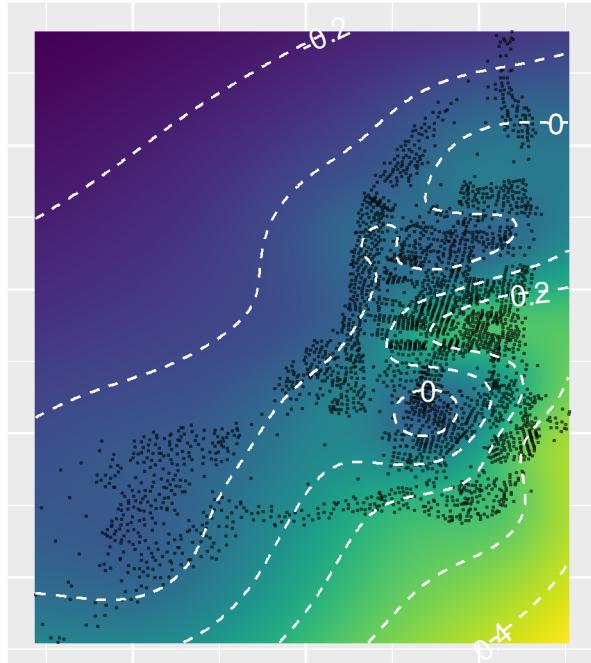
```

geom_text_contour(aes(x = X, y = Y, z = predicted2), color = 'white', binwidth = 0.1) +
scale_fill_viridis() +
coord_cartesian() +
theme(axis.title= element_blank(),
      axis.text = element_blank(),
      axis.ticks = element_blank()
) +
labs(subtitle = 'spline plus complexe', fill = "prédictions")

ggarrange(plot1, plot2, nrow = 1, ncol = 2, common.legend = TRUE, legend = 'bottom')

```

spline de base



spline plus complexe

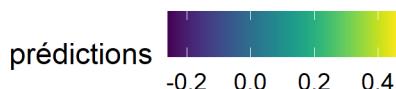
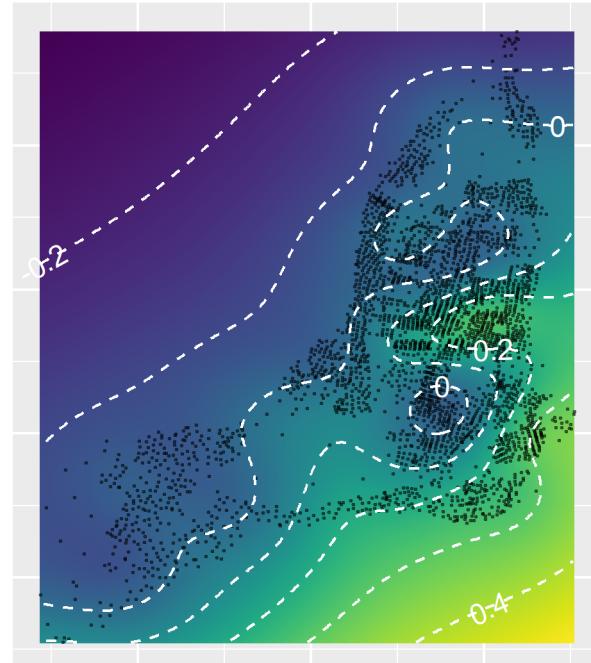


FIG. 11.17 : Comparaison de deux splines spatiales

Or, il s'avère que les deux *splines* spatiales sont très similaires (figures 11.17). Par conséquent, il est vraisemblablement plus pertinent de conserver la plus simple des deux. Notez que le Mont-Royal, compris dans le cercle central avec une isoligne à 0, est caractérisé par des valeurs plus faibles d'îlots de chaleur, alors que les quartiers centraux situés un peu plus au nord sont au contraire marqués par des pourcentages d'îlots de chaleur supérieurs de 20 points de pourcentage en moyenne.

11.7 GAMM

Bien entendu, il est possible de combiner les modèles généralisés additifs (GAM) avec les modèles à effet mixtes (GLMM) abordés dans les sections précédentes. Ces modèles généralisés additifs à effets mixtes (GAMM) peuvent facilement être mis en œuvre avec `mgcv`.

11.7.0.1 GAMM et interceptes aléatoires

Pour définir des constantes aléatoires, il suffit d'utiliser la notation `s(var, bs = 're')` avec `var` une variable nominale. Reprenons l'exemple précédent, mais avec cette fois-ci les arrondissements comme un intercepte aléatoire.

```
dataset$Arrond <- as.factor(dataset$Arrond)
model4 <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  s(prt_veg) + s(Arrond, bs = "re"),
  data = dataset, family = betar(link = "logit"))
```

L'enjeu est ensuite d'extraire la variance propre à cet effet aléatoire ainsi que les valeurs des interceptes pour chaque arrondissement.

```
gam.vcomp(model4)
```

```
##
## Standard deviations and 0.95 confidence intervals:
##
##           std.dev      lower      upper
## s(prt_veg) 0.007047166 0.003785275 0.01311993
## s(Arrond)   0.393539474 0.302707198 0.51162747
##
## Rank: 2/2
```

Nous constatons donc que l'écart-type de l'effet aléatoire des arrondissements est de 0,39, ce qui signifie que les effets de chaque arrondissement seront compris à 95 % entre -1,17 et 1,17 ($1.17 = 3 \times 0.39$) sur l'échelle du prédicteur linéaire. En effet, rappelons que les effets aléatoires sont modélisés comme des distributions normales et que 95 % de la densité d'une distribution normale se situe entre -3 et +3 écarts-types. Pour extraire les interceptes spécifiques de chaque arrondissement, nous pouvons utiliser la fonction `get_random` du package `itsadug`.

```
library(itsadug)
values <- get_random(model4)[[1]]
df <- data.frame(
  ri = as.numeric(values),
  arrond = names(values)
)

ggplot(df) +
  geom_point(aes(x = ri, y = reorder(arrond, ri))) +
  geom_vline(xintercept = 0, color = "red") +
  labs(y = "Arrondissement", x = "intercepte aléatoire")
```

Nous constatons ainsi, à la figure 11.18, que pour une partie des arrondissements, la densité d'îlot de chaleur est systématiquement supérieure à la moyenne régionale représentée ici par la ligne rouge (0 = effet moyen pour tous les arrondissements). Il convient alors d'améliorer ce graphique en ajoutant le niveau d'incertitude associé à chaque intercepte. Pour ce faire, nous utilisons la fonction `extract_random_effects` du package `mixedup`. Notez que ce package n'est actuellement pas disponible sur CRAN et doit être téléchargé sur `github` avec la commande suivante :

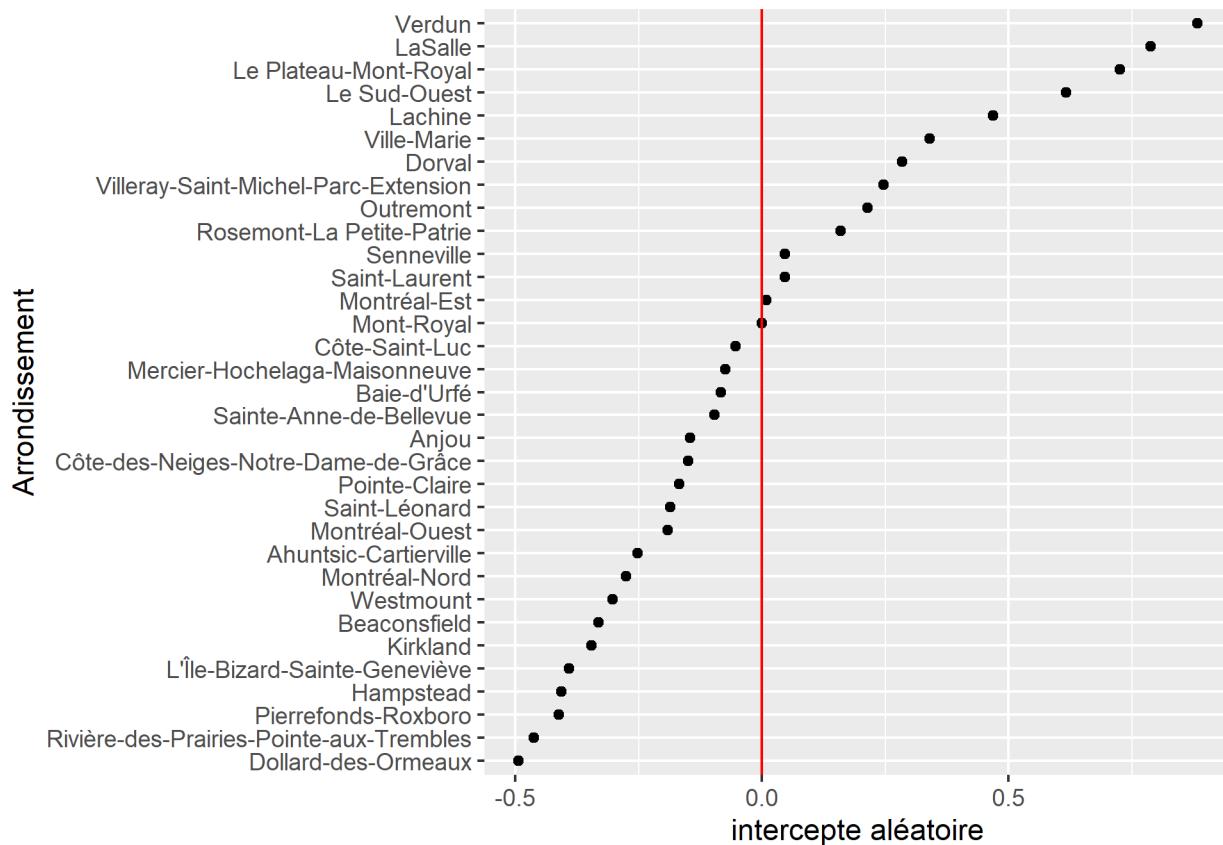


FIG. 11.18 : Constantes aléatoires pour les arrondissements

```
remotes:::install_github('m-clark/mixedup')
```

Avec la version 4.0.1 de R, nous avons rencontré des difficultés pour installer `mixedup`. Nous avons donc simplement récupéré le code source de la fonction et l'avons enregistré dans un fichier de code séparé que nous appelons ici.

```
source("code_complementaire/gam_functions.R")
```

Nous pouvons ensuite procéder à l'extraction des effets aléatoires et les représenter à nouveau (figure 11.19).

```
df_re <- extract_random_effects.gam(model4, re = "Arrond")

ggplot(df_re) +
  geom_errorbar(aes(xmin = lower_2.5, xmax = upper_97.5, y = reorder(group, value))) +
  geom_point(aes(x = value, y = reorder(group, value))) +
  geom_vline(xintercept = 0, color = "red") +
  labs(y = "Arrondissement", x = "Intercepte aléatoire")
```

Cela permet de distinguer quels écarts sont significativement différents de 0 au seuil de 95 %. À titre de rappel, pour être significatif à ce seuil, un intervalle représenté par une ligne noire horizontale ne doit pas intersecter la ligne rouge verticale. Puisque nous utilisons ici la distribution bêta et une fonction de

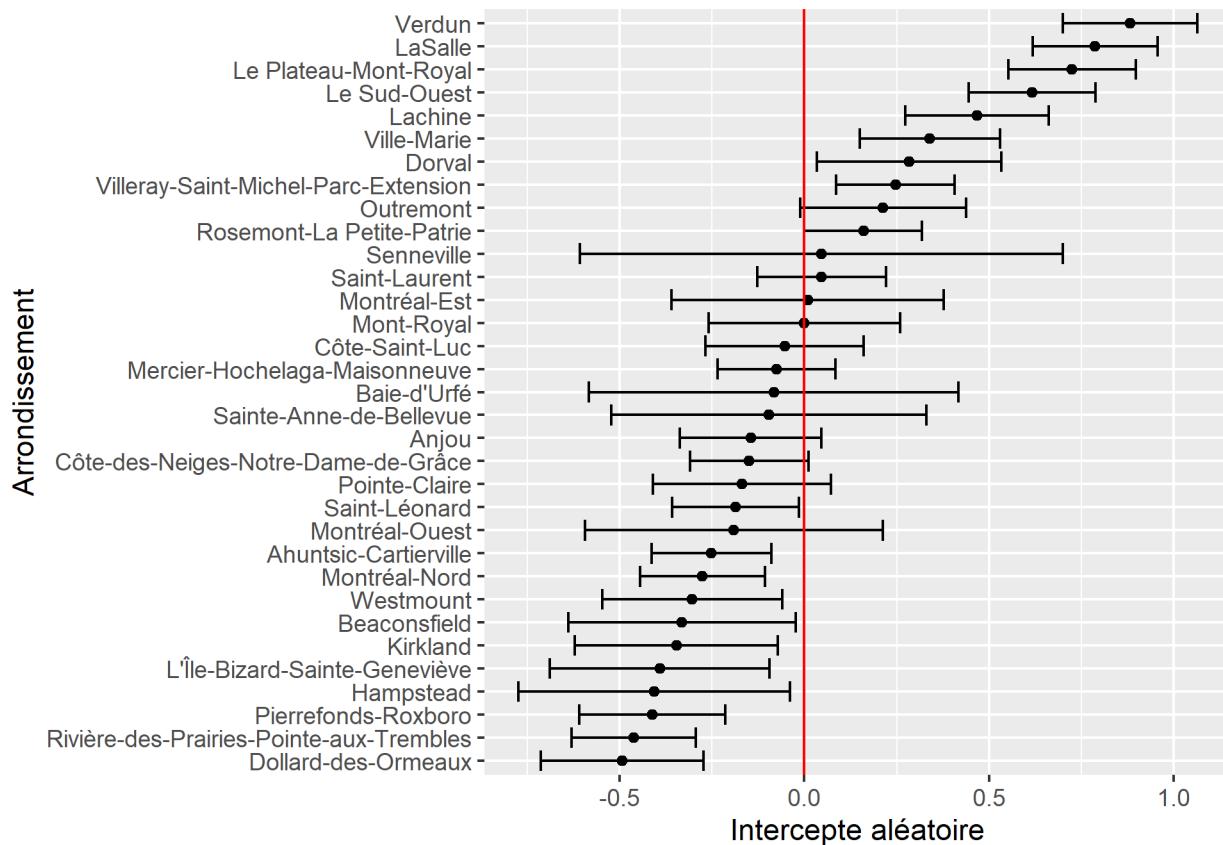


FIG. 11.19 : Constantes aléatoires pour les arrondissements avec intervalle de confiance

lien logistique, nous devons utiliser des prédictions pour simplifier l’interprétation des coefficients. Nous fixons ici toutes les variables à leur moyenne respective, sauf l’arrondissement, et calculons les prédictions dans l’échelle originale (0 à 1).

```

dfpred <- data.frame(
  A65Pct = mean(dataset$A65Pct),
  A014Pct = mean(dataset$A014Pct),
  PopFRPct = mean(dataset$PopFRPct),
  PopMVPct = mean(dataset$PopMVPct),
  prt_veg = mean(dataset$prt_veg),
  Arrond = as.character(unique(dataset$Arrond))
)

# Calculer les prédictions pour le prédicteur linéaire
dfpred$preds <- predict(model4, newdata = dfpred, type = "link")

# Calculer l'intervalle de confiance en utilisant les valeurs
# extraites avec extract_random_effects
dfpred <- dfpred[order(dfpred$Arrond),]
df_re <- df_re[order(df_re$group),]

dfpred$lower <- dfpred$preds - 1.96*df_re$se
dfpred$upper <- dfpred$preds + 1.96*df_re$se

```

```
# Il nous reste juste à reconvertir le tout dans l'unité d'origine
# en utilisant l'inverse de la fonction logistique
inv.logit <- function(x){exp(x)/(1+exp(x))}

dfpred$lower <- inv.logit(dfpred$lower)
dfpred$upper <- inv.logit(dfpred$upper)
dfpred$preds <- inv.logit(dfpred$preds)

ggplot(dfpred) +
  geom_errorbarh(aes(xmin = lower, xmax = upper, y = reorder(Arrond, preds))) +
  geom_point(aes(x = preds, y = reorder(Arrond, preds))) +
  geom_vline(xintercept = mean(dfpred$preds), color = "red") +
  labs(y = "Arrondissement", x = "intercepte aléatoire")
```

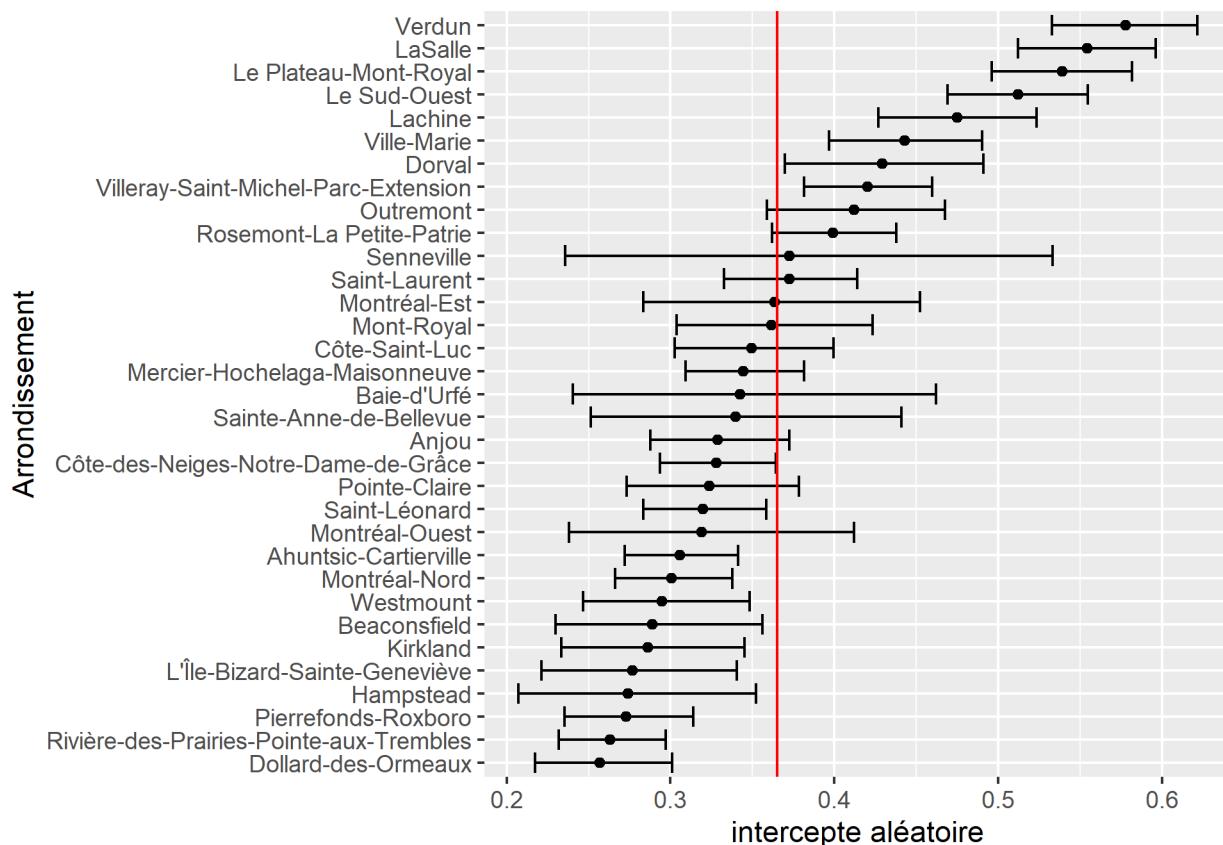


FIG. 11.20 : Prédictions pour les différents arrondissements pour une AD fictive moyenne

Nous constatons ainsi, à la figure 11.20, que pour une hypothétique aire de diffusion moyenne, la différence de densité d’ilot de chaleur peut être de 0,32 (32 % de la surface de l’AD) entre les arrondissements Verdun et Dollard-des-Ormeaux.

11.7.0.2 GAMM et coefficients aléatoires

En plus des interceptes aléatoires, il est aussi possible de définir des coefficients aléatoires. Reprenons notre exemple et tentons de faire varier l’effet de la variable PopFRPct en fonction de l’arrondissement.

```
model5 <- gam(hot ~
  A65Pct + A014Pct + PopFRPct + PopMVPct +
  s(prt_veg) + s(Arrond, bs = "re") +
  s(PopFRPct, Arrond, bs = "re"),
  data = dataset, family = betar(link = "logit"))
```

Notez ici une distinction importante ! Le modèle n'assume aucune corrélation entre les coefficients aléatoires pour la variable `PopFRPct` et pour les constantes aléatoires. Il est présumé que ces deux effets proviennent de deux distributions normales distinctes. En d'autres termes, le modèle ne dispose pas des paramètres nécessaires pour vérifier si les arrondissements avec les constantes les plus fortes (avec des densités supérieures d'îlot de chaleur) sont aussi des arrondissements dans lesquels l'effet de la variable `PopFRPct` est plus prononcé (et vice-versa). Pour plus d'informations sur cette distinction, référez-vous à la section 9.2.3.

```
AIC(model4, model5)
```

```
##           df      AIC
## model4 41.54635 -6421.791
## model5 56.84734 -6466.726
```

Ce dernier modèle présente une valeur de l'AIC plus faible et serait donc ainsi mieux ajusté que notre modèle avec seulement un intercepte aléatoire. Nous pouvons donc extraire les coefficients aléatoires et les représenter à la figure 11.21.

```
df_re <- extract_random_effects.gam(model5)
df_re <- subset(df_re, df_re$effect == 'PopFRPct')

ggplot(df_re) +
  geom_errorbarh(aes(xmin = lower_2.5, xmax = upper_97.5, y = reorder(group, value))) +
  geom_point(aes(x = value, y = reorder(group, value))) +
  geom_vline(xintercept = 0, color = "red") +
  labs(y = "Arrondissement", x = "coefficients aléatoires")
```

Nous constatons notamment que seuls trois arrondissements ont des coefficients aléatoires significativement différents de 0. Ainsi, pour les arrondissements Anjou et Plateau-Mont-Royal, les coefficients aléatoires sont respectivement de -0,013 et -0,015, et viennent donc se retrancher à la valeur moyenne régionale de 0,0154 qui atteint alors presque 0. Du point de vue de l'interprétation, nous pouvons en conclure que le groupe des personnes à faible revenu ne subit pas de surexposition aux îlots de chaleur à l'échelle des AD dans ces arrondissements.

En revanche, dans l'arrondissement Mercier-Hochelaga-Maisonneuve, la situation est à l'inverse plus systématiquement en défaveur des populations à faible revenu, avec une taille d'effet près de deux fois supérieure à la moyenne régionale. En effet, l'effet moyen régional (coefficient fixe) est de 0,0154, auquel vient s'ajouter l'effet spécifique (coefficient aléatoire) de Mercier-Hochelaga-Maisonneuve, soit 0,011, pour un effet total de 0,0264



Des effets aléatoires plus complexes dans les GAMM

Il est possible de spécifier des GAMM avec des effets aléatoires plus complexes autorisant, par exemple, des corrélations entre les différents effets / niveaux. Il faut pour cela utiliser la fonction `gamm` de `mgcv` ou la fonction `gamm4` du package `gamm4`. La première offre plus de flexibilité, mais la seconde est plus facile à utiliser et doit

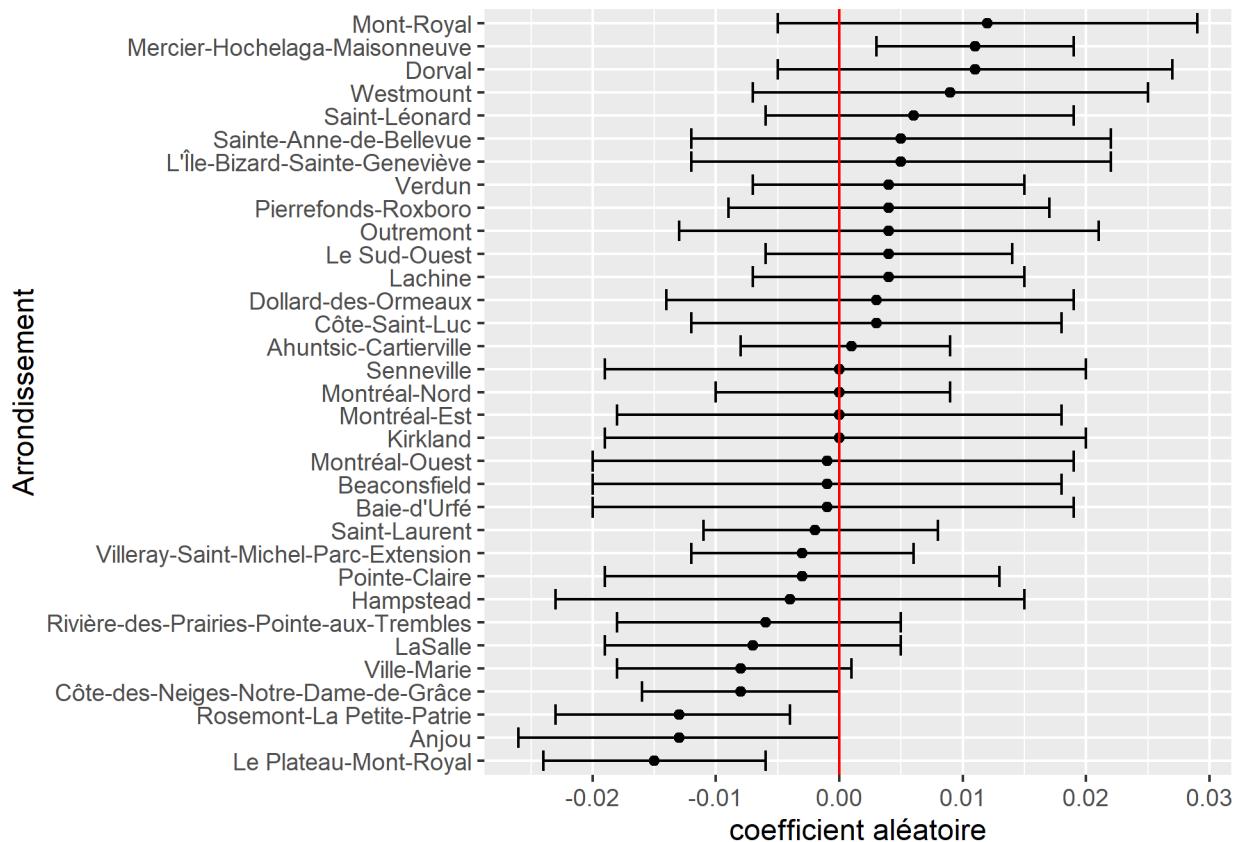


FIG. 11.21 : Pentes et constantes aléatoires pour les arrondissements

être privilégiée quand un modèle comporte un très grand nombre de groupes dans un effet aléatoire, ou lorsque la distribution du modèle n'est pas gaussienne. La fonction `gamm` permet d'ajuster des modèles non gaussiens, mais elle utilise une approche appelée PQL (*Penalized Quasi-Likelihood* en anglais) connue pour être moins stable et moins précise.

Cependant, dans l'exemple de cette section, nous utilisons un modèle GAMM avec une distribution bêta, ce qui n'est actuellement pas supporté par les fonctions `gamm` et `gamm4`. Pour un modèle GAMM plus complexe utilisant une distribution bêta, il est nécessaire d'utiliser le package `gamlss`, mais ce dernier utilise aussi une approche de type PQL. Nous montrons tout de même ici comment ajouter un modèle qui inclut une corrélation entre les deux effets aléatoires de l'exemple précédent. Notez ici que le terme `re` apparaissant dans la formule permet de spécifier un effet aléatoire en utilisant la syntaxe du package `nlme`. Plus spécifiquement, `gamlss` fait un pont avec `nlme` et utilise son algorithme d'ajustement au sein de ces propres routines. De même, le terme `pb` permet de spécifier une *spline* de lissage dans le même esprit que `mgcv`. Il est également possible d'utiliser le terme `ga` faisant le lien avec `mgcv` et de profiter de sa flexibilité dans `gamlss`.

```
library(gamlss)
library(gamlss.add)

model6 <- gammss(hot ~
  pb(prt_veg) +
  re(fixed = ~ A65Pct + A014Pct + PopFRPct + PopMVPct,
     random = ~(1 + PopFRPct)|Arrond),
  data = dataset, family = BE(mu.link = "logit"))
```

Nous pouvons ensuite accéder à la partie du modèle qui nous intéresse, soit celle concernant les effets aléatoires.

```
randomPart <- model6$mu.coefSmo[[2]]
print(randomPart)

## Linear mixed-effects model fit by maximum likelihood
##   Data: Data
##   Log-likelihood: -2964.494
##   Fixed: fix.formula
##   (Intercept) A65Pct     A014Pct     PopFRPct     PopMVPct
## -0.060862832 -0.001945204 -0.010139278  0.017259606 -0.002599745
##
## Random effects:
##   Formula: ~(1 + PopFRPct) | Arrond
##   Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
##   (Intercept) 0.47298363 (Intr)
##   PopFRPct    0.01078909 -0.646
##   Residual    0.99888012
##
## Variance function:
##   Structure: fixed weights
##   Formula: ~W.var
##   Number of Observations: 3157
##   Number of Groups: 33
```

À lecture de la partie du résumé consacrée aux résultats pour les effets aléatoires, nous constatons que la corrélation entre les interceptes aléatoires et les coefficients aléatoires est de -0,65. Cela signifie que pour les arrondissements avec des interceptes élevés (plus grande proportion d'îlots de chaleur), l'effet de la variable PopFRPct tend à être plus faible. Autrement dit, dans les arrondissements avec beaucoup d'îlots de chaleur, les personnes à faible revenu ont tendance à être moins exposées, tel qu'illustré à la figure 11.22.

```
df <- ranef(randomPart)
df$arrond <- rownames(df)
names(df) <- c('Intercept', 'PopFRPct', 'Arrondissement')

ggplot(df) +
  geom_hline(yintercept = 0, color = "red") +
  geom_vline(xintercept = 0, color = "red") +
  geom_point(aes(x = Intercept, y = PopFRPct))
```

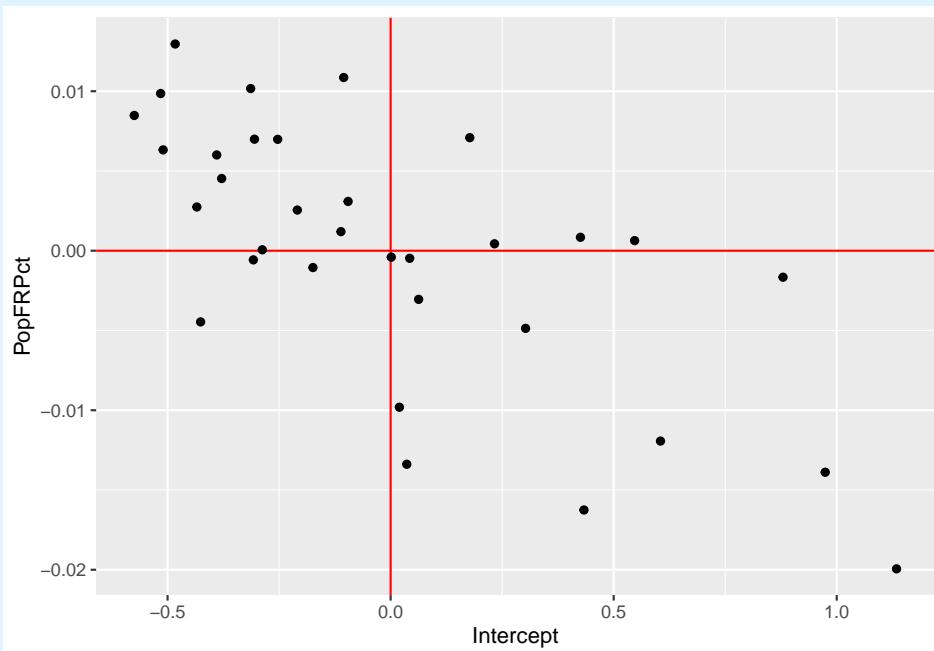


FIG. 11.22 : Relation entre les effets aléatoires des arrondissements et la variable population à faible revenu

11.8 Quiz de révision du chapitre

Questions

- Que signifie l'acronyme GAM?

- Gaussian Asymmetric Model
- Generalized Additive Model
- Gaussian Asynchronous Model
- Generalized Asymmetric Model

Relisez l'introduction du chapitre 11 au besoin.

- Quel est l'intérêt d'un modèle GAM comparativement à un modèle GLM?

- la possibilité d'ajuster n'importe quelle distribution dans le modèle
- la possibilité de tenir compte de structures de corrélation entre les observations
- l'ajout de termes non-linéaires pour les prédicteurs
- la transformation de la variable Y pour quelle se rapproche d'une distribution normale

Relisez au besoin la section 11.1.

- Une spline ajustée sur une variable X dans un GAM est construite comme :

- un ensemble de coefficients ajustés à différentes sections de la variable X identifiées par des points de ruptures
- la somme d'un ensemble de polynomiales de X multipliées par des coefficients ajustés par le modèle
- la somme d'un ensemble de fonctions de base appliquées à X et multipliées par des coefficients ajustés par le modèle

Relisez au besoin la section 11.1.4

- Le degré de complexité d'une spline est contrôlé par :

- le nombre de noeuds de la spline
- le nombre de variables X sur lesquelles la spline est ajustée
- le type de fonction de base de la spline
- la distribution du modèle

Relisez au besoin la section 11.1.4.

- Le nombre de noeuds d'une spline peut être :

- défini manuellement, nous parlons alors de spline de régression
- déterminé par une approche automatique appelée pénalisation de la vraisemblance, nous parlons alors de spline de lissage
- déterminé automatiquement en re-paramétrisant la spline comme un effet aléatoire, nous parlons alors de spline de lissage
- déterminé de façon itérative en ajoutant un noeud à chaque fois et en comparant le R2 de Nagelkerke

Relisez au besoin la section 11.3.

- Une spline bivariée est un autre nom pour une spline d'interaction.

- Vrai
- Faux

Relisez au besoin la section [11.5](#)

- **Une spline peut être ajustée sur plus que deux variables X simultanément.**

- Vrai
- Faux

Relisez au besoin la section [11.5](#).

- **Pour interpréter les résultats d'une spline, il est possible de :**

- observer le nombre de degrés de liberté (estimated degree of freedom) de cette dernière pour se faire une idée de son degré de complexité
- représenter graphiquement les fonctions de base utilisées par la spline
- représenter les prédictions du modèle “toutes choses égales par ailleurs” afin d'obtenir les effets marginaux des termes non linéaires
- extraire les coefficients de la spline et les interpréter de façon classique

Relisez au besoin la section [11.4](#).

Réponses

- Que signifie l'acronyme GAM ?
 - Generalized Additive Model
- Quel est l'intérêt d'un modèle GAM comparativement à un modèle GLM ?
 - l'ajout de termes non-linéaires pour les prédicteurs
- Une spline ajustée sur une variable X dans un GAM est construite comme :
 - la somme d'un ensemble de fonctions de base appliquées à X et multipliées par des coefficients ajustés par le modèle
- Le degré de complexité d'une spline est contrôlé par :
 - le nombre de noeuds de la spline
- Le nombre de noeuds d'une spline peut être :
 - défini manuellement, nous parlons alors de spline de régression
 - déterminé par une approche automatique appelée pénalisation de la vraisemblance, nous parlons alors de spline de lissage
 - déterminé automatiquement en re-paramétrisant la spline comme un effet aléatoire, nous parlons alors de spline de lissage
- Une spline bivariée est un autre nom pour une spline d'interaction.
 - Faux
- Une spline peut être ajustée sur plus que deux variables X simultanément.
 - Vrai
- Pour interpréter les résultats d'une spline, il est possible de :
 - observer le nombre de degrés de liberté (estimated degree of freedom) de cette dernière pour se faire une idée de son degré de complexité
 - représenter les prédictions du modèle “toutes choses égales par ailleurs” afin d'obtenir les effets marginaux des termes non linéaires

Cinquième partie

Analyses exploratoires multivariées

Chapitre 12

Méthodes factorielles

Dans le cadre de ce chapitre, nous présentons les trois méthodes factorielles les plus utilisées en sciences sociales : l'analyse en composantes principales (ACP, section 12.2), l'analyse factorielle des correspondances (AFC, section 12.3) et l'analyse factorielle des correspondances multiples (ACM, section 12.4). Ces méthodes, qui permettent d'explorer et de synthétiser l'information de différents tableaux de données, relèvent de la statistique exploratoire multidimensionnelle.



Dans ce chapitre, nous utilisons les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2`, le seul, l'unique!
 - * `ggpubr` pour combiner des graphiques.
- Pour les analyses factorielles :
 - * `FactoMineR` pour réaliser une ACP, une AFC et une ACM.
 - * `factoextra` pour réaliser des graphiques à partir des résultats d'une analyse factorielle.
 - * `explor` pour les résultats d'une ACP, d'une AFC ou d'une ACM avec une interface Web interactive.
- Autres *packages* :
 - * `geocmeans` pour un jeu de données utilisé pour calculer une ACP.
 - * `ggplot2`, `ggpubr`, `stringr` et `corrplot` pour réaliser des graphiques personnalisés sur les résultats d'une analyse factorielle.
 - * `tmap` et `RColorBrewer` pour cartographier les coordonnées factorielles.
 - * `Hmisc` pour l'obtention d'une matrice de corrélation.



Réduction de données et identification de variables latentes

Les méthodes factorielles sont souvent dénommées des **méthodes de réduction de données**, en raison de leur objectif principal : résumer l'information d'un tableau en quelques nouvelles variables synthétiques (figure 12.1). Ainsi, elles permettent de réduire l'information d'un tableau volumineux — comprenant par exemple 1000 observations et 100 variables — en p nouvelles variables (par exemple cinq avec toujours 1000 observations) résumant X % de l'information contenue dans le tableau initial. Formulée plus mathématiquement, Lebart et al. (1995, 13) en donnent une formulation plus mathématique : ils signalent qu'avec les méthodes factorielles, « on cherche à réduire les dimensions du tableau de données en représentant les associations entre individus et entre variables dans des espaces de faibles dimensions ».

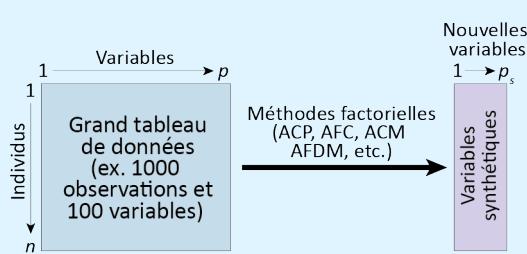


FIG. 12.1 : Principe de base des analyses factorielles

Ces nouvelles variables synthétiques peuvent être considérées comme des **variables latentes** puisqu'elles ne sont pas directement observées ; elles sont plutôt produites par la méthode factorielle utilisée afin de résumer les relations/associations entre plusieurs variables mesurées initialement.

12.1 Aperçu des méthodes factorielles

12.1.1 Méthodes factorielles et types de données

En analyse factorielle, la nature même des données du tableau à traiter détermine la méthode à employer : l'analyse en composantes principales (ACP) est adaptée aux tableaux avec des variables continues (idéalement normalement distribuées), l'analyse factorielle des correspondances (AFC) s'applique à des tableaux de contingence tandis que l'analyse des correspondances multiples (ACM) permet de résumer des tableaux avec des données qualitatives (issues d'un sondage par exemple) (tableau 12.1). Sachez toutefois qu'il existe d'autres méthodes factorielles qui ne sont pas abordées dans ce chapitre, notamment : l'analyse factorielle de données mixtes (AFDM) permettant d'explorer des tableaux avec à la fois des variables continues et des variables qualitatives et l'analyse factorielle multiple hiérarchique (AFMH) permettant de traiter des tableaux avec une structure hiérarchique. Pour s'initier à ces deux autres méthodes factorielles plus récentes, consultez notamment l'excellent ouvrage de Jérôme Pagès (2013).

TAB. 12.1 : Trois principales méthodes factorielles

Méthode factorielle	Abr.	Type de données	Type de distance
Analyse en composantes principales	ACP	Variables continues	Distance euclidienne
Analyse factorielle des correspondances	AFC	Tableau de contingence	Distance du khi-deux
Analyse factorielle des correspondances multiples	ACM	Variables qualitatives	Distance du khi-deux

12.1.2 Bref historique des méthodes factorielles

Il existe une longue tradition de l'utilisation des méthodes factorielles dans le monde universitaire francophone puisque plusieurs d'entre elles ont été proposées par des statisticiens et des statisticiennes francophones à partir des années 1960. L'analyse en composantes principales (ACP) a été proposée dès les années 1930 par le statisticien américain Harold Hotelling (1933). En revanche, l'analyse des correspondances (AFC) et son extension, l'analyse des correspondances multiples (ACM), ont été proposées par le statisticien français Jean-Paul Benzécri (1973), tandis que l'analyse factorielle de données mixtes (AFDM) a été proposée par Brigitte Escofier et Jérôme Pagès (Escofier 1979 ; Pagès 2002).

Ainsi, plusieurs ouvrages de statistique sur les méthodes factorielles, désormais classiques, ont été publiés en français (Benzécri 1973 ; Escofier et Pagès 1998 ; Lebart, Morineau et Piron 1995 ; Pagès 2013).

Ils méritent grandement d'être consultés, notamment pour mieux comprendre les formulations mathématiques (matricielles et géométriques) de ces méthodes. À cela s'ajoutent plusieurs ouvrages visant à « vulgariser ces méthodes » en sciences sociales ; c'est notamment le cas de l'excellent ouvrage de Léna Sanders (1989) en géographie.

12.2 Analyses en composantes principales (ACP)

D'emblée, notez qu'il existe deux types d'analyse en composantes principales (ACP) (*Principal Component Analysis, PCA* en anglais) :

- **l'ACP non normée** dans laquelle les variables quantitatives du tableau sont uniquement centrées (moyenne = 0).
- **l'ACP normée** dans laquelle les variables quantitatives du tableau sont préalablement centrées réduites (moyenne = 0 et variance = 1 ; section 2.5.5.2).

Puisque les variables d'un tableau sont souvent exprimées dans des unités de mesure différentes ou avec des ordres de grandeur différents (intervalles et écarts-types bien différents), l'utilisation de l'ACP normée est bien plus courante. Elle est d'ailleurs l'option par défaut dans les fonctions R permettant de calculer une ACP. Par conséquent, nous détaillons dans cette section uniquement l'ACP normée.

Autrement dit, le recours à une ACP non normée est plus rare et s'applique uniquement à la situation suivante : toutes les variables du tableau sont mesurées dans la même unité (par exemple, en pourcentage) ; il pourrait être ainsi judicieux de conserver leurs variances respectives.

12.2.1 Recherche d'une simplification

L'ACP permet d'explorer et de résumer un tableau constitué uniquement de variables quantitatives (figure 12.2), et ce, de trois façons : 1) en montrant les ressemblances entre les individus (observations), 2) en révélant les liaisons entre les variables quantitatives et 3) en résumant l'ensemble des variables du tableau par des variables synthétiques nommées composantes principales.

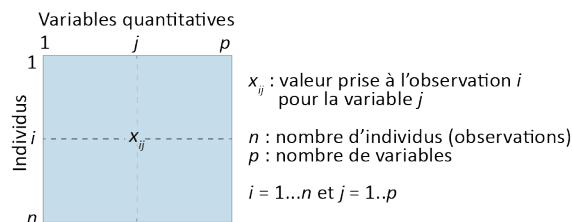


FIG. 12.2 : Tableau pour une ACP

Ressemblance entre les individus. Concrètement, deux individus se ressemblent si leurs valeurs respectives pour les p variables du tableau sont similaires. Cette proximité/ressemblance est évaluée à partir de la distance euclidienne (équation (12.1)). La notion de distance fait l'objet d'une section à part entière (section 13.2) que vous pouvez consulter dès à présent si elle ne vous est pas familière.

$$d^2(a, b) = \sum_{j=1}^p (x_{aj} - x_{bj})^2 \quad (12.1)$$

Prenons un exemple fictif avec trois individus (i , j et k) ayant des valeurs pour trois variables préalablement centrées réduites (V1 à V3) (tableau 12.2). La proximité entre les paires de points est évaluée comme suit :

$$d^2(i, j) = (-1,15 - 0,49)^2 + (-1,15 - 0,58)^2 + (0,83 + 1,11)^2 = 9,44$$

$$d^2(i, k) = (-1,15 + 0,66)^2 + (-1,15 - 0,58)^2 + (0,83 - 0,28)^2 = 5,98$$

$$d^2(j, k) = (0,49 + 0,66)^2 + (0,58 - 0,58)^2 + (-1,11 - 0,28)^2 = 1,97$$

Nous pouvons en conclure que i est plus proche de k que de j , mais aussi que la paire de points les plus proches est (i, k) . En d'autres termes, les deux observations i et k sont les plus similaires du jeu de données selon la distance euclidienne.

Liaisons entre les variables. Dans une ACP normée, les liaisons entre les variables deux à deux sont évaluées avec le coefficient de corrélation (section 4.3.1), soit la moyenne du produit des deux variables centrées réduites (équation (12.2)). Notez que dans une ACP non normée, plus rarement utilisée, les liaisons sont évaluées avec la covariance puisque les variables sont uniquement centrées (équation (12.3)).

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} = \sum_{i=1}^n \frac{Zx_i Zy_i}{n} \quad (12.2)$$

$$\text{cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n} \quad (12.3)$$

Composantes principales. Au chapitre 4, nous avons abordé deux méthodes pour identifier des relations linéaires entre des variables continues normalement distribuées :

- la corrélation de Pearson (section 4.3), qu'il est possible d'illustrer graphiquement à partir d'un nuage de points ;
- la régression linéaire simple (section 4.4), permettant de résumer la relation linéaire entre deux variables avec une droite de régression de type $Y = a + bX$.

Brièvement, plus deux variables sont corrélées (positivement ou négativement), plus le nuage de points qu'elles forment est allongé et plus les points sont proches de la droite de régression (figure 12.3, partie a). À l'inverse, plus la liaison entre les deux variables normalement distribuées est faible, plus le nuage prend la forme d'un cercle et plus les points du nuage sont éloignés de la droite de régression (figure 12.3, partie b). Puisqu'en ACP normée, les variables sont centrées réduites, le centre de gravité du nuage de points est $(x=0, y=0)$ et il est toujours traversé par la droite de régression. Finalement, nous avons vu que la méthode des moindres carrés ordinaires (MCO) permet de déterminer cette droite en minimisant les distances entre les valeurs observées et celles projetées orthogonalement sur cette droite (valeurs prédictes). Dans le cas de deux variables uniquement, l'axe factoriel principal/la composante principale est donc la droite qui résume le mieux la liaison entre les deux variables (en rouge). L'axe 2 représente la seconde plus importante composante (axe, dimension) et il est orthogonal (perpendiculaire) au premier axe (en bleu).

TAB. 12.2 : Données fictives

Individu	Variables centrées réduites		
	V1	V2	V3
i	-1,15	-1,15	0,83
j	0,49	0,58	-1,11
k	0,66	0,58	0,28

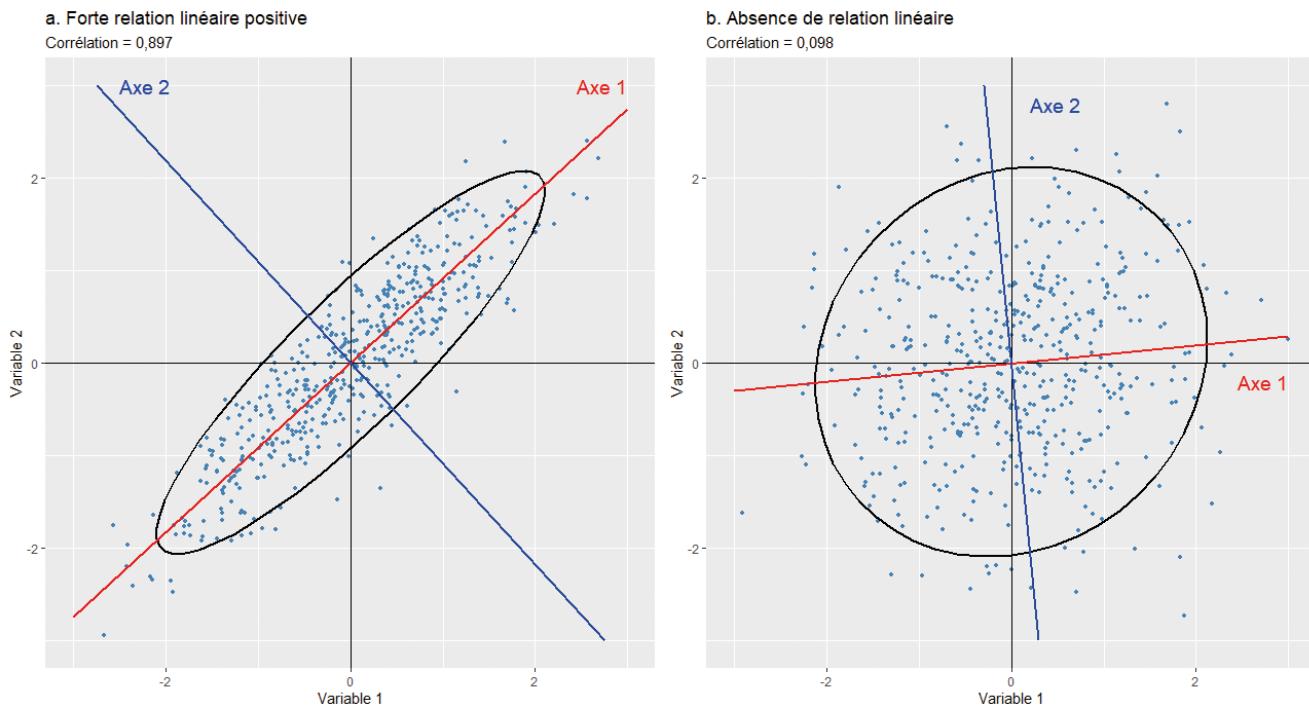


FIG. 12.3 : Corrélation, allongement du nuage de points et axes factoriels

Imaginez maintenant trois variables pour lesquelles vous désirez identifier un axe, une droite qui résume le mieux les liaisons entre elles. Visuellement, vous passez d'un nuage de points en deux dimensions (2D) à un nuage en dimensions (3D). Si les corrélations entre les trois variables sont très faibles, alors le nuage prend la forme d'un ballon de soccer (football en Europe). Par contre, plus ces liaisons sont fortes, plus la forme est allongée comme un ballon de rugby et plus les points sont proches de l'axe traversant le ballon.

Ajouter une autre variable revient alors à ajouter une quatrième dimension qu'il est impossible de visualiser, même pour les plus fervents adeptes de science-fiction. Pourtant, le problème reste le même : identifier, dans un plan en p dimensions (variables), les axes factoriels – les composantes principales – qui concourent le plus à résumer les liaisons entre les variables continues préalablement centrées réduites, et ce, en utilisation la méthode des moindres carrés ordinaires.



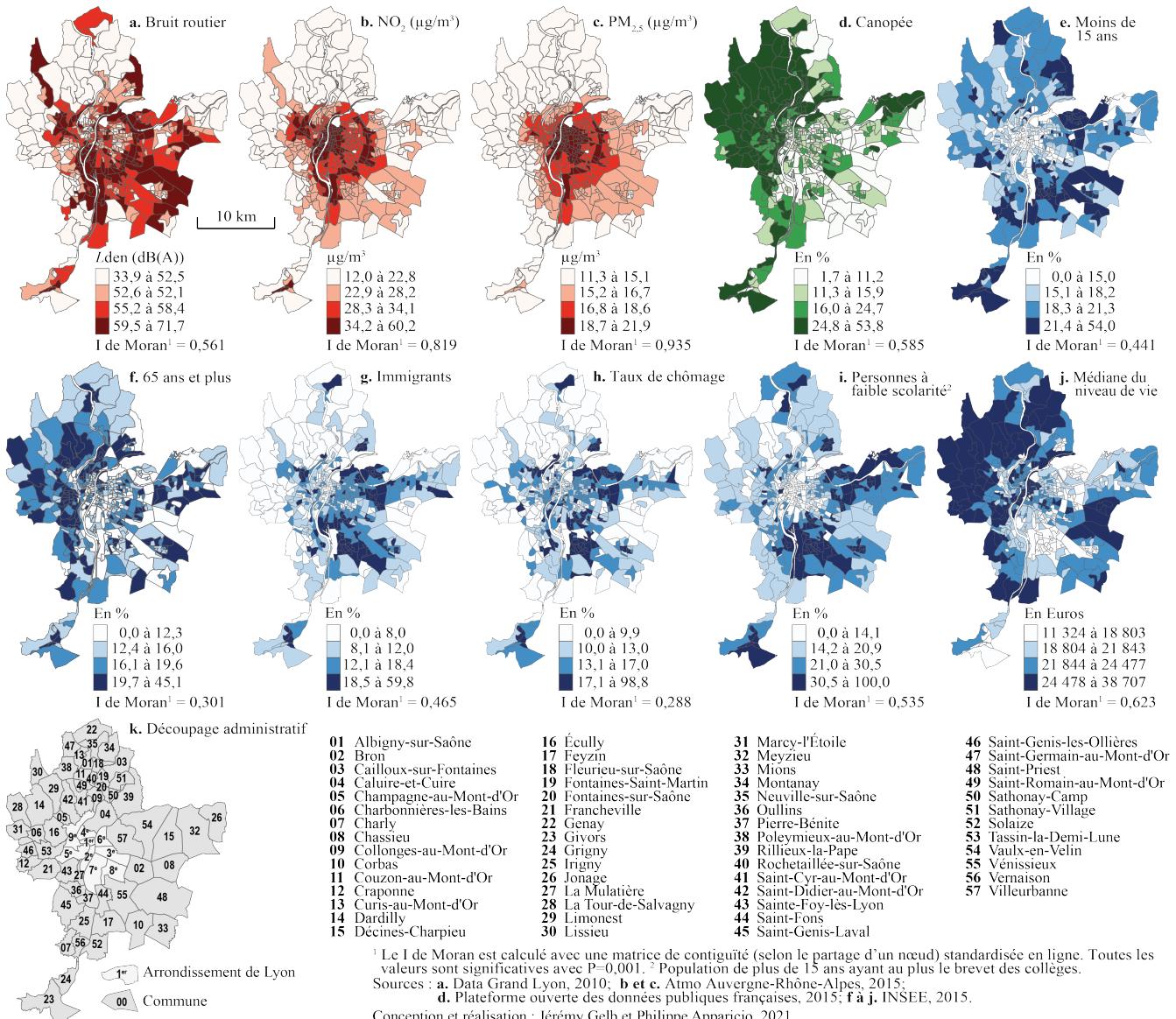
Les termes **composantes principales** et **axes factoriels** sont des synonymes employés pour référer aux nouvelles variables synthétiques produites par l'ACP et résumant l'information du tableau initial.

12.2.2 Aides à l'interprétation

Pour illustrer les aides à l'interprétation de l'ACP, nous utilisons un jeu de données spatiales tiré d'un article sur l'agglomération lyonnaise en France (Gelb et Apparicio 2021b). Ce jeu de données comprend dix variables, dont quatre environnementales (EN) et six socioéconomiques (SE), pour les îlots regroupés pour l'information statistique (IRIS) de l'agglomération lyonnaise (tableau 12.3 et figure 12.4). Sur ces dix variables, nous calculons une **ACP normée**.

TAB. 12.3 : Statistiques descriptives pour le jeu de données utilisé pour l'ACP

Nom	Intitulé	Type	Moy.	E.-T.	Min.	Max.
Lden	Bruit routier (Lden dB(A))	EN	55,6	4,9	33,9	71,7
NO2	Dioxyde d'azote ($\mu\text{g}/\text{m}^3$)	EN	28,7	7,9	12,0	60,2
PM25	Particules fines ($\text{PM}_{2,5}$)	EN	16,8	2,1	11,3	21,9
VegHautPrt	Canopée (%)	EN	18,7	10,1	1,7	53,8
Pct0_14	Moins de 15 ans (%)	SE	18,5	5,7	0,0	54,0
Pct_65	65 ans et plus (%)	SE	16,2	5,9	0,0	45,1
Pct_Img	Immigrants (%)	SE	14,5	9,1	0,0	59,8
TxChom1564	Taux de chômage	SE	14,8	8,1	0,0	98,8
Pct_brevet	Personnes à faible scolarité (%)	SE	23,5	12,6	0,0	100,0
NivVieMed	Médiane du niveau de vie (Euros)	SE	21 804,5	4 922,5	11 324,0	38 707,0

**FIG. 12.4 : Cartographie des dix variables utilisées pour l'ACP**



Trois étapes pour bien analyser une ACP et comprendre la signification des axes factoriels :

1. Interprétation des résultats des valeurs propres pour identifier le nombre d'axes (de composantes principales) à retenir. L'enjeu est de garder un nombre d'axes limité qui résume le mieux le tableau initial (réduction des données).
2. Analyse des résultats pour les variables (coordonnées factorielles, cosinus carrés et contributions sur les axes retenus).
3. Analyse des résultats pour les individus (coordonnées factorielles, cosinus carrés et contributions sur les axes retenus).

Les deux dernières étapes permettent de comprendre la signification des axes retenus et de les qualifier. Cette étape d'interprétation est essentielle en sciences sociales. En effet, nous avons vu dans l'introduction du chapitre que les méthodes factorielles permettent de résumer l'information d'un tableau en quelques nouvelles variables synthétiques, souvent considérées comme des variables latentes dans le jeu de données. Il convient alors de bien comprendre ces variables synthétiques (latentes), si nous souhaitons les utiliser dans une autre analyse subséquente (par exemple, les introduire dans une régression).

12.2.2.1 Résultats de l'ACP pour les valeurs propres

À titre de rappel, une ACP normée est réalisée sur des variables préalablement centrées réduites (équation (12.4)), ce qui signifie que pour chaque variable :

- Nous soustrayons à chaque valeur la moyenne de la variable correspondante (centrage); la moyenne est donc égale à 0.
- Nous divisons cette différence par l'écart-type de la variable correspondante (réduction); la variance est égale à 1.

$$z = \frac{x_i - \mu}{\sigma} \quad (12.4)$$

Par conséquent, la variance totale (ou inertie totale) d'un tableau sur lequel est calculée une ACP normée est égale au nombre de variables qu'il comprend. Puisque nous l'appliquons ici à dix variables, la variance totale du tableau à réduire – c'est-à-dire à résumer en K nouvelles variables synthétiques, composantes principales, axes factoriels – est donc égale à 10. Trois mesures reportées au tableau 12.4 permettent d'analyser les valeurs propres :

- VP_k , la valeur propre (*eigenvalue* en anglais) de l'axe k , c'est-à-dire la quantité de variance du tableau initial résumé par l'axe.
- VP_k/P avec P étant le nombre de variables que comprend le tableau initial. Cette mesure représente ainsi le pourcentage de la variance totale du tableau résumé par l'axe k , autrement dit la quantité d'informations du tableau initial résumée par l'axe, la composante principale k . Cela nous permet ainsi d'évaluer le pouvoir explicatif de l'axe.
- Le pourcentage cumulé pour les axes.

Avant d'analyser en détail le tableau 12.4, notez que la somme des valeurs propres de toutes les composantes de l'ACP est toujours égale au nombre de variables du tableau initial. Aussi, la quantité de variance expliquée (les valeurs propres) décroît de la composante 1 à la composante K .

Combien d'axes d'une ACP faut-il retenir? Pour répondre à cette question, deux approches sont possibles :

- **Approche statistique** (avec le critère de Kaiser (1960)). Nous retenons uniquement les composantes qui présentent une valeur propre supérieure à 1. Rappelez-vous qu'en ACP normée, les variables

TAB. 12.4 : Résultats de l'ACP pour les valeurs propres

Composante	Valeur propre	Pourcentage	Pourc. cumulé
1	3,543	35,425	35,425
2	2,760	27,596	63,021
3	1,042	10,422	73,443
4	0,751	7,511	80,954
5	0,606	6,059	87,013
6	0,388	3,880	90,893
7	0,379	3,788	94,681
8	0,244	2,441	97,122
9	0,217	2,167	99,289
10	0,071	0,711	100,000

sont préalablement centrées réduites, et donc que leur variance respective est égale à 1. Par conséquent, une composante ayant une valeur propre inférieure à 1 a un pouvoir explicatif inférieur à celui d'une variable. À la lecture du tableau, nous retenons les trois premières composantes si nous appliquons ce critère.

- **Approche empirique** basée sur la lecture des pourcentages et des pourcentages cumulés. Nous pouvons retenir uniquement les deux premières composantes. En effet, ces deux premiers facteurs résument près des deux tiers de la variance totale du tableau (63,02 %). Cela démontre bien que l'ACP, comme les autres méthodes factorielles, est bien une méthode de réduction de données puisque nous résumons dix variables avec deux nouvelles variables synthétiques (axes, composantes principales). Pour faciliter le choix du nombre d'axes, il est fortement conseillé de construire des histogrammes à partir des valeurs propres, des pourcentages et des pourcentages cumulés (figure 12.5). Or, à la lecture de ces graphiques, nous constatons que la variance expliquée chute drastiquement après les deux premières composantes. Par conséquent, nous pouvons retenir uniquement les deux premiers axes.



Lecture du diagramme des valeurs propres

Plus les variables incluses dans l'ACP sont corrélées entre elles, plus l'ACP est intéressante : plus les valeurs propres des premiers axes sont fortes et plus il y a des sauts importants dans le diagramme des valeurs propres. À l'inverse, lorsque les variables incluses dans l'ACP sont peu corrélées entre elles, il n'y a pas de sauts importants dans l'histogramme, autrement dit les valeurs propres sont uniformément décroissantes.

12.2.2.2 Résultats de l'ACP pour les variables

Pour qualifier les axes, quatre mesures sont disponibles pour les variables :

- **Les coordonnées factorielles des variables** sont simplement les coefficients de corrélation de Pearson des variables sur l'axe k et varient ainsi de -1 à 1 (relire au besoin la section 4.3). Pour qualifier un axe, il convient alors de repérer les variables les plus corrélées positivement et négativement sur l'axe, autrement dit, de repérer les variables situées aux extrémités l'axe.
- **Les cosinus carrés des variables** (Cos^2) (appelés aussi les qualités de représentation des variables sur un axe) permettent de repérer le ou les axes qui concourent le plus à donner un sens à la variable. Ils sont en fait les coordonnées des variables mises au carré. La somme des cosinus carrés d'une variable sur tous les axes de l'ACP est donc égale à 1 (sommation en ligne). **La qualité de représentation d'une variable sur les n premiers axes** est simplement la somme des cosinus carrés d'une variable sur les axes retenus. Par exemple, pour la variable $Lden$, la qualité de représentation de la variable sur le premier axe est égale : $0,42^2 = 0,17$. Pour cette même variable, la qualité de la $Lden$ sur les trois premiers axes est égale à : $0,17 + 0,32 + 0,26 = 0,75$.

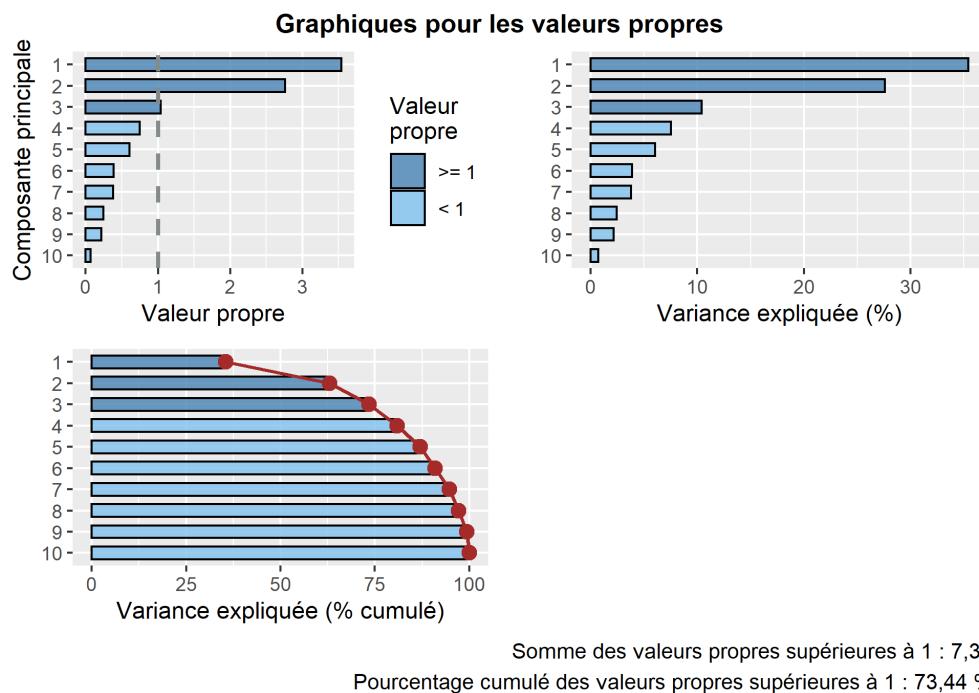


FIG. 12.5 : Graphiques personnalisés pour les valeurs propres pour l'ACP

- **Les contributions des variables** permettent de repérer celles qui participent le plus à la formation d'un axe. Elles s'obtiennent en divisant les cosinus carrés par la valeur propre de l'axe multiplié par 100. La somme des contributions des variables pour un axe donné est donc égale à 100 (sommaire en colonne). Par exemple, pour la variable Lden, la contribution sur le premier axe est égale : $0,174/3,543 \times 100 = 4,920$.

Les résultats de l'ACP pour les variables sont présentés au tableau 12.5.

TAB. 12.5 : Résultats de l'ACP pour les variables

Variable	Coordonnées			Cosinus carrés				Contributions		
	1	2	3	1	2	3	Qualité	1	2	3
Lden	0,42	0,57	0,51	0,17	0,32	0,26	0,75	4,92	11,64	24,80
NO2	0,15	0,93	0,19	0,02	0,86	0,04	0,92	0,66	31,07	3,54
PM25	0,19	0,92	0,03	0,04	0,84	0,00	0,87	1,01	30,36	0,12
VegHautPrt	-0,40	-0,42	0,40	0,16	0,18	0,16	0,50	4,63	6,35	15,46
Pct0_14	0,55	-0,53	0,08	0,30	0,28	0,01	0,59	8,61	10,28	0,55
Pct_65	-0,41	-0,27	0,72	0,17	0,07	0,51	0,75	4,73	2,66	49,26
Pct_Img	0,87	-0,09	0,11	0,76	0,01	0,01	0,78	21,56	0,29	1,08
TxChom1564	0,77	-0,09	-0,07	0,60	0,01	0,00	0,61	16,89	0,27	0,45
Pct_brevet	0,73	-0,43	0,22	0,53	0,19	0,05	0,77	14,94	6,81	4,61
NivVieMed	-0,88	0,09	0,04	0,78	0,01	0,00	0,79	22,06	0,28	0,14

Analyse de la première composante principale (valeur propre de 3,54, 35,43 %)

- À la lecture des contributions, il est clair que quatre variables contribuent grandement à la formation de l'axe 1 : NivVieMed (22,06 %), Pct_Img (21,56 %), TxChom1564 (16,89 %) et Pct_brevet (14,94 %). Il convient alors d'analyser en détail leurs coordonnées factorielles et leurs cosinus carrés.
- À la lecture des coordonnées factorielles, nous constatons que trois variables socioéconomiques sont

fortement corrélées positivement avec l'axe 1, soit le *pourcentage d'immigrants* (0,87), le *taux de chômage* (0,77) et le *pourcentage de personnes avec une faible scolarité* (0,73). À l'autre extrémité, la *médiane du niveau de vie* (en Euros) est négativement corrélée avec l'axe 1. Comment interpréter ce résultat? Premièrement, cela signifie que plus la valeur de l'axe 1 est positive et élevée, plus celles des trois variables (*Pct_Img*, *TxChom1564* et *Pct_brevet*) sont aussi élevées (corrélations positives) et plus la valeur de *NivVieMed* est faible (corrération négative). Inversement, plus la valeur de l'axe 1 est négative et faible, les valeurs de *Pct_Img*, *TxChom1564* et *Pct_brevet* sont faibles et plus celle de *NivVieMed* est forte. Deuxièmement, cela signifie que les trois variables (*Pct_Img*, *TxChom1564* et *Pct_brevet*) sont fortement corrélées positivement entre elles puisqu'elles se situent sur la même extrémité de l'axe et qu'elles sont toutes trois négativement corrélées avec la variable *NivVieMed*. Cela peut être rapidement confirmé avec la matrice de corrélation entre les dix variables (tableau 12.6).

- À la lecture des cosinus carrés de l'axe 1, nous constatons que plus des trois quarts de la dispersion/de l'information des variables *NivVieMed* (0,78) et *Pct_Img* (0,76) est concentrée sur l'axe 1.

Analyse de la deuxième composante principale (valeur propre de 2,76, 27,60 %)

- À la lecture des contributions, trois variables environnementales contribuent à la formation de l'axe 2 : principalement, celles sur la pollution de l'air (*NO2* = 31,07 % et *PM25* = 30,36 %) et secondairement, celle sur le bruit routier (*Lden* = 11,64 %).
- À la lecture des coordonnées factorielles, ces trois variables sont fortement corrélées positivement avec l'axe 2 : *NO2* (0,93), *PM25* (0,92) et *Lden* (0,57). À l'autre extrémité de l'axe, la variable *Pct0_14* est négativement, mais pas fortement, corrélée (-0,53). La lecture de la matrice de corrélation au tableau 12.6 confirme que ces trois variables environnementales sont fortement corrélées positivement entre elles (par exemple, un coefficient de corrélation de Pearson de 0,90 entre *NO2* et *PM25*).
- À la lecture des cosinus carrés de l'axe 2, nous constatons que près de 90 % de la dispersion/de l'information des variables *NO2* (0,86) et *PM25* (0,84) est concentrée sur l'axe 2.

Analyse de la troisième composante principale (valeur propre de 1,042, 10,42 %)

- Le *pourcentage de personnes âgées* (*Pct_65*) contribue principalement à la formation de l'axe 3 avec lequel il est corrélé positivement (contribution de 49,26 % et coordonnée factorielle de 0,72). S'en suit la variable *Lden*, qui joue un rôle beaucoup moins important (contribution de 24,80 % et coordonnée factorielle de 0,51).



Lien entre la valeur propre d'un axe et le nombre de variables contribuant à sa formation

Vous avez compris que plus la valeur propre d'un axe est forte, plus il y a potentiellement de variables qui concourent à sa formation. Cela explique que pour la troisième composante, qui a une faible valeur propre

TAB. 12.6 : Matrice de corrélation de Pearson entre les variables utilisées pour l'ACP

Variable	A	B	C	D	E	F	G	H	I	J
A. Lden		0,62	0,49	-0,23	0,04	-0,09	0,28	0,19	0,14	-0,26
B. NO2	0,62		0,90	-0,28	-0,34	-0,21	0,07	0,04	-0,25	-0,04
C. PM25	0,49	0,90		-0,39	-0,34	-0,26	0,12	0,07	-0,25	-0,10
D. VegHautPrt	-0,23	-0,28	-0,39		0,04	0,32	-0,22	-0,18	-0,14	0,32
E. Pct0_14	0,04	-0,34	-0,34	0,04		-0,12	0,46	0,36	0,54	-0,45
F. Pct_65	-0,09	-0,21	-0,26	0,32	-0,12		-0,24	-0,30	0,00	0,32
G. Pct_Img	0,28	0,07	0,12	-0,22	0,46	-0,24		0,66	0,64	-0,73
H. TxChom1564	0,19	0,04	0,07	-0,18	0,36	-0,30	0,66		0,47	-0,62
I. Pct_brevet	0,14	-0,25	-0,25	-0,14	0,54	0,00	0,64	0,47		-0,67
J. NivVieMed	-0,26	-0,04	-0,10	0,32	-0,45	0,32	-0,73	-0,62	-0,67	

(1,042), seule une variable contribue significativement à sa formation.

Analyse de la qualité de représentation des variables sur les premiers axes de l'ACP

À titre de rappel, la qualité est simplement la somme des cosinus carrés d'une variable sur les axes retenus. Si nous retenons trois axes, les six variables qui sont le mieux résumées – et qui ont donc le plus d'influence sur les résultats de l'ACP – sont : N02 (0,92), PM25 (0,87), NivVieMed (0,79), Pct_Img (0,78), Pct_brevet (0,77) et Lden (0,75).

Qualification, dénomination d'axes factoriels

L'analyse des coordonnées, des contributions et des cosinus carrés doit vous permettre de formuler un intitulé pour chacun des axes retenus. Nous vous proposons les intitulés suivants :

- *Niveau de défavorisation socioéconomique* (axe 1). Plus la valeur de l'axe est élevée, plus le niveau de défavorisation de l'entité spatiale (IRIS) est élevé.
- *Qualité environnementale* (axe 2). Plus la valeur de l'axe est forte, plus les niveaux de pollution atmosphérique (dioxyde d'azote et particules fines) et de bruit (Lden) sont élevés.

Recours à des graphiques pour analyser les résultats de l'ACP pour des variables

Plus le nombre de variables utilisées pour calculer l'ACP est important, plus l'analyse des coordonnées factorielles, des cosinus carrés et des contributions reportés dans un tableau devient fastidieuse. Puisque l'ACP a été calculée sur dix variables, l'analyse des valeurs du tableau 12.5 a été assez facile et rapide. Imaginez maintenant que nous réalisons une ACP sur une centaine de variables, la taille du tableau des résultats pour les variables sera considérable... Par conséquent, il est recommandé de construire plusieurs graphiques qui facilitent l'analyse des résultats pour les variables.

Par exemple, à la figure 12.6, nous avons construit des graphiques avec les coordonnées factorielles sur les trois premiers axes de l'ACP. En un coup d'œil, il est facile de repérer les variables les plus corrélées positivement ou négativement avec chacun d'entre eux. Aussi, il est fréquent de construire un nuage de points avec les coordonnées des variables sur les deux premiers axes factoriels, soit un graphique communément appelé **nuage de points des variables sur le premier plan factoriel** sur lequel est représenté le cercle des corrélations (figure 12.7). Bien entendu, cet exercice peut être fait avec d'autres axes factoriels (les axes 3 et 4 par exemple).

12.2.2.3 Résultats de l'ACP pour les individus

Comme pour les variables, nous retrouvons les mêmes mesures pour les individus : les coordonnées factorielles, les cosinus carrés et les contributions. Les coordonnées factorielles des individus sont les projections orthogonales des observations sur l'axe. Puisqu'en ACP normée, les variables utilisées pour l'ACP sont centrées réduites, la moyenne des coordonnées factorielles des individus pour un axe est toujours égale à zéro. En revanche, contrairement aux coordonnées factorielles pour les variables, les coordonnées pour les individus ne varient pas de -1 à 1! Les cosinus carrés quantifient à quel point chaque axe représente chaque individu. Enfin, les contributions quantifient l'apport de chaque individu à la formation d'un axe.

Si le jeu de données comprend peu d'observations, il est toujours possible de créer un **nuage de points des individus sur le premier plan factoriel** sur lequel vous pouvez ajouter les étiquettes permettant d'identifier les observations (figure 12.8). Ce graphique est rapidement illisible lorsque le nombre d'observations est important. Il peut rester utile si certaines des observations du jeu de données doivent faire l'objet d'une analyse spécifique.

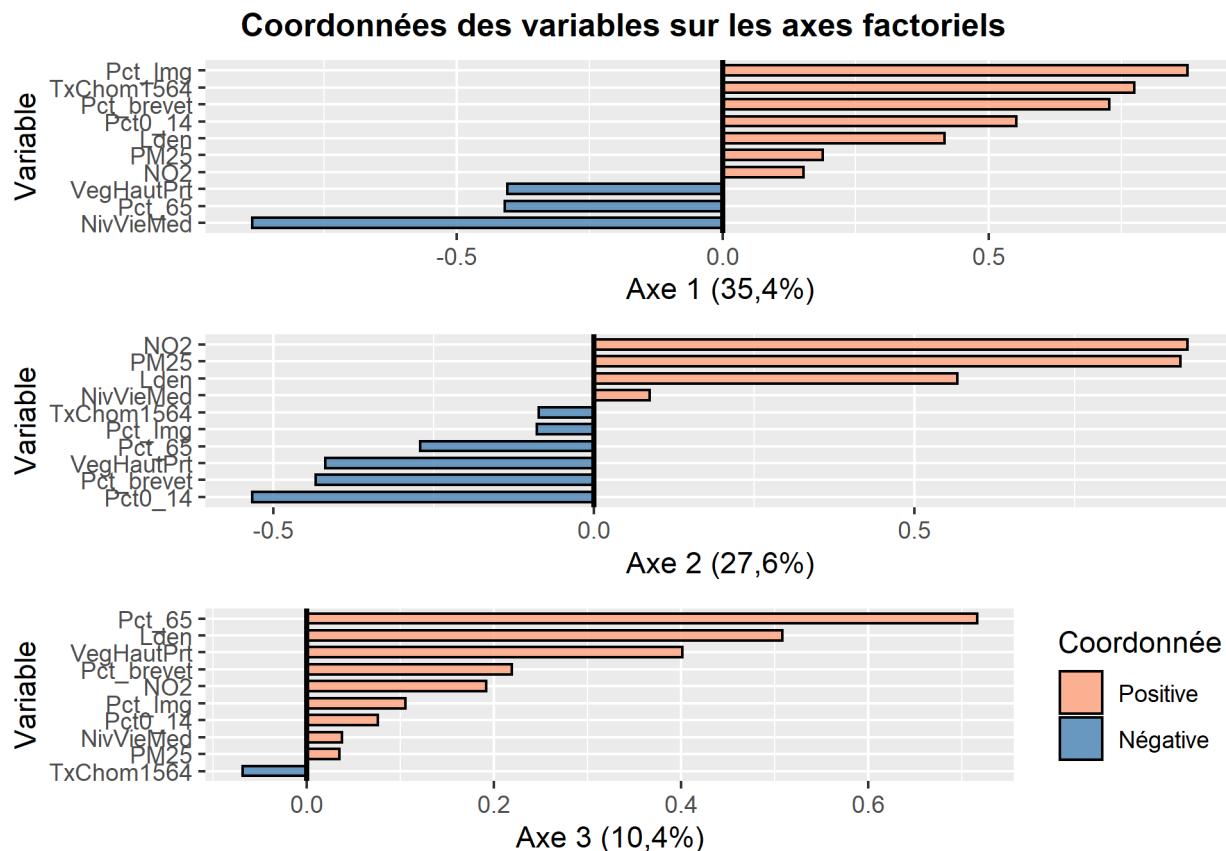


FIG. 12.6 : Coordonnées factorielles des variables

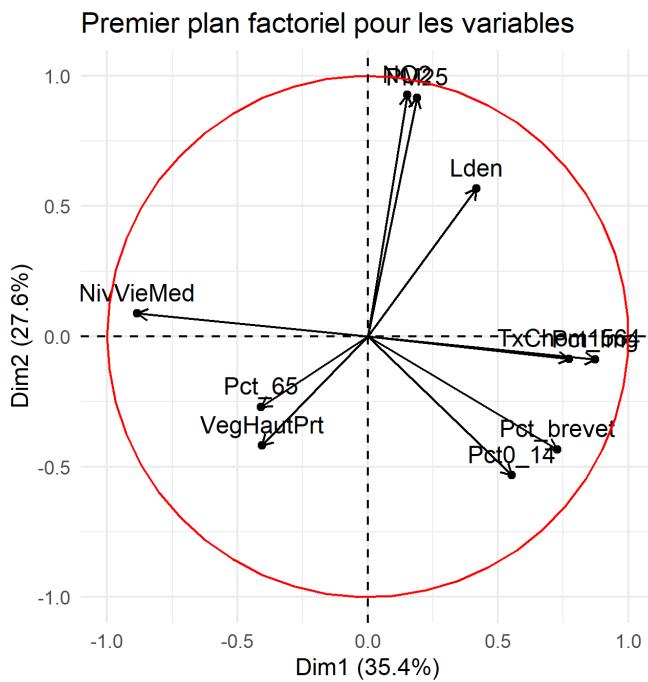


FIG. 12.7 : Premier plan factoriel de l'ACP pour les variables

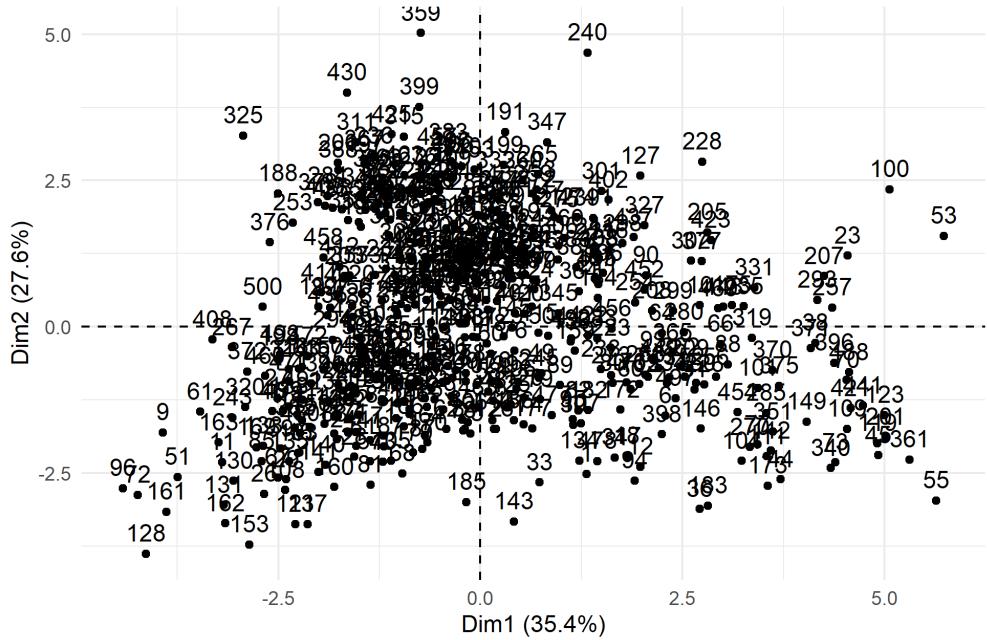


FIG. 12.8 : Premier plan factoriel pour les individus

Lorsque les observations sont des unités spatiales, il est très intéressant de cartographier les coordonnées factorielles des individus (figure 12.8). À la lecture de la carte choroplète de gauche (axe 1), nous pouvons constater que le niveau de défavorisation socioéconomique est élevé dans l'est (IRIS en vert), et inversement, très faible à l'ouest de l'agglomération (IRIS en rouge). À la lecture de la carte de droite (axe 2), sans surprise, la partie centrale de l'agglomération est caractérisée par des niveaux de pollution atmosphérique et de bruit routier bien plus élevés qu'en périphérie.



Nous abordons ici plusieurs autres éléments intéressants de l'ACP.

Ajout de variables ou d'individus supplémentaires

Premièrement, il est possible d'ajouter des variables continues ou des individus supplémentaires qui n'ont pas été pris en compte dans le calcul de l'ACP (figure 12.10). Concernant les variables continues supplémentaires, il s'agit simplement de calculer leurs corrélations avec les axes retenus de l'ACP. Concernant les individus, il s'agit de les projeter sur les axes factoriels. Pour plus d'informations sur le sujet, consultez les excellents ouvrages de Ludovic Lebart, Alain Morineau et Marie Piron (1995, 42-45) ou encore de Jérôme Pagès (2013, 22-24).

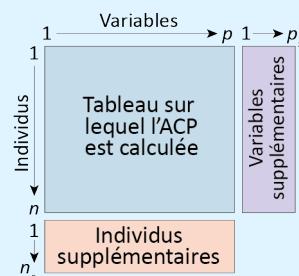
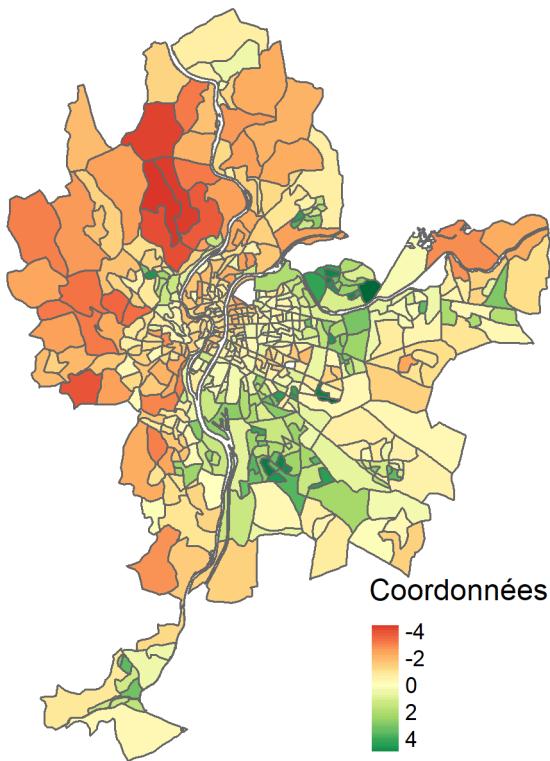
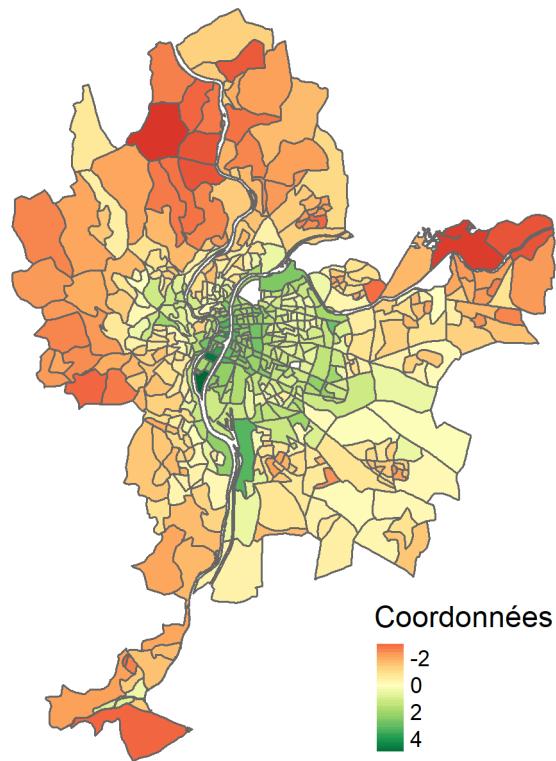


FIG. 12.10 : Variables et individus supplémentaires pour l'ACP

Axe 1 : Défavorisation socioéco.



Axe 2 : Qualité environnementale

**FIG. 12.9 :** Cartographie des coordonnées factorielles des individus

Pondération des individus et des variables

Deuxièmement, il est possible de pondérer à la fois les individus et, plus rarement, les variables lors du calcul de l'ACP.

Analyse en composantes principales non paramétrique

Troisièmement, il est possible de calculer une ACP sur des variables préalablement transformées en rang (section 2.5.5.2). Cela peut être justifié lorsque les variables sont très anormalement distribuées en raison de valeurs extrêmes. Les coordonnées factorielles pour les variables sont alors le coefficient de Spearman (section 4.3.3) et non de Pearson. Aussi, les variables sont centrées non pas sur leurs moyennes respectives, mais sur leurs médianes. Pour plus d'informations sur cette approche, consultez de nouveau Lebart et al. (1995, 51-52).

Analyse en composantes principales robuste

Finalement, d'autres méthodes plus avancées qu'une ACP non paramétrique peuvent être utilisées afin d'obtenir des composantes principales qui ne sont pas influencées par des valeurs extrêmes : les ACP robustes (Rivest et Plante 1988; Hubert, Rousseeuw et Vanden Branden 2005) qui peuvent être mises en œuvre, entre autres avec le package `roscpca`.

12.2.3 Mise en œuvre dans R

Plusieurs *packages* permettent de calculer une ACP dans R, notamment `psych` (fonction `principal`), `ade4` (fonction `dudi.pca`) et `FactoMineR` (fonction `PCA`). Ce dernier est certainement le plus abouti. De plus, il permet également de calculer une analyse des correspondances (AFC), une analyse des correspondances

multiples (ACM) et une analyse factorielle de données mixtes (AFDM). Nous utilisons donc `FactoMineR` pour mettre en œuvre les trois types de méthodes factorielles abordées dans ce chapitre (ACP, AFC et ACM). Pour l'ACP, nous exploitons un jeu de données issu du package `geocmeans` qu'il faut préalablement charger à l'aide des lignes de code suivantes.

```
library(geocmeans)
data(LyonIris)
Data <- LyonIris@data[c("CODE_IRIS","Lden","NO2","PM25","VegHautPrt",
                      "Pct0_14","Pct_65","Pct_Img",
                      "TxChom1564","Pct_brevet","NivVieMed")]
```

12.2.3.1 Calcul et exploration d'une ACP avec `FactoMineR`

Pour calculer l'ACP, il suffit d'utiliser la fonction `PCA` de `FactoMineR`, puis la fonction `summary`(`MonACP`) qui renvoie les résultats de l'ACP pour :

- Les valeurs propres (section `Eigenvalues`) pour les composantes principales (`Dim.1` à `Dim.n`) avec leur variance expliquée brute (`Variance`) en pourcentage (% of var.) et en pourcentage cumulé (Cumulative % of var.).
- Les dix premières observations (section `Individuals`) avec les coordonnées factorielles (`Dim.1` à `Dim.n`), les contributions (`ctr`) et les cosinus carrés (`cos2`). Pour accéder aux résultats pour toutes les observations, utilisez les fonctions `res.acp$ind` ou encore `res.acp$ind$coord` (uniquement les coordonnées factorielles), `res.acpindcontrib` (uniquement les contributions) et `res.acpindcos2` (uniquement les cosinus carrés).
- Les variables (section `Variables`) avec les coordonnées factorielles (`Dim.1` à `Dim.n`), les contributions (`ctr`) et les cosinus carrés (`cos2`).

```
library(FactoMineR)
# Version classique avec FactoMineR
# Construction d'une ACP sur les colonnes 2 à 11 du DataFrame Data
res.acp <- PCA(Data[,2:11], scale.unit=TRUE, graph=F)
# Affichage des résultats de la fonction PCA
print(res.acp)
```

```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 506 individuals, described by 10 variables
## *The results are available in the following objects:
##
##      name            description
## 1  "$eig"          "eigenvalues"
## 2  "$var"          "results for the variables"
## 3  "$var$coord"    "coord. for the variables"
## 4  "$var$cor"      "correlations variables - dimensions"
## 5  "$var$cos2"     "cos2 for the variables"
## 6  "$var$contrib"  "contributions of the variables"
## 7  "$ind"          "results for the individuals"
## 8  "$ind$coord"    "coord. for the individuals"
## 9  "$ind$cos2"     "cos2 for the individuals"
## 10 "$ind$contrib"  "contributions of the individuals"
## 11 "$call"         "summary statistics"
```

```

## 12 "$call$centre"      "mean of the variables"
## 13 "$call$ecart.type"  "standard error of the variables"
## 14 "$call$row.w"       "weights for the individuals"
## 15 "$call$col.w"       "weights for the variables"

# Résumé des résultats (valeurs propres, individus, variables)
summary(res.acp)

## 
## Call:
## PCA(X = Data[, 2:11], scale.unit = TRUE, graph = F)
##
## 
## Eigenvalues
##                               Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance                 3.543   2.760   1.042   0.751   0.606   0.388   0.379
## % of var.                35.425  27.596  10.422  7.511   6.059   3.880   3.788
## Cumulative % of var.    35.425  63.021  73.443  80.954  87.013  90.893  94.681
##                               Dim.8   Dim.9   Dim.10
## Variance                 0.244   0.217   0.071
## % of var.                2.441   2.167   0.711
## Cumulative % of var.   97.122  99.289 100.000
## 
## 
## Individuals (the 10 first)
##          Dist   Dim.1   ctr   cos2   Dim.2   ctr   cos2   Dim.3
## 1 | 3.054 | 1.315  0.096  0.185 | -2.515  0.453  0.678 | 0.221
## 2 | 1.882 | 0.193  0.002  0.011 | -1.744  0.218  0.859 | 0.082
## 3 | 2.820 | 2.338  0.305  0.687 | -0.860  0.053  0.093 | -0.765
## 4 | 2.816 | -0.740  0.031  0.069 |  2.265  0.367  0.647 | 1.293
## 5 | 3.210 | -2.208  0.272  0.473 | -1.597  0.183  0.248 | 1.471
## 6 | 3.016 | 2.287  0.292  0.575 | -1.515  0.164  0.252 | 0.390
## 7 | 3.022 | -1.540  0.132  0.260 | -1.803  0.233  0.356 | 0.465
## 8 | 3.122 | -1.536  0.132  0.242 | -2.038  0.298  0.426 | -0.120
## 9 | 4.743 | -3.930  0.862  0.687 | -1.806  0.234  0.145 | 0.993
## 10 | 3.055 | 2.713  0.411  0.789 |  0.368  0.010  0.014 | -0.391
##          ctr   cos2
## 1 | 0.009  0.005 |
## 2 | 0.001  0.002 |
## 3 | 0.111  0.074 |
## 4 | 0.317  0.211 |
## 5 | 0.411  0.210 |
## 6 | 0.029  0.017 |
## 7 | 0.041  0.024 |
## 8 | 0.003  0.001 |
## 9 | 0.187  0.044 |
## 10 | 0.029  0.016 |

## 
## Variables
##          Dim.1   ctr   cos2   Dim.2   ctr   cos2   Dim.3   ctr   cos2
## Lden | 0.417  4.920  0.174 | 0.567 11.640  0.321 | 0.508 24.799  0.258

```

```

## NO2      |  0.153  0.657  0.023 |  0.926 31.068  0.857 |  0.192  3.540  0.037
## PM25     |  0.189  1.007  0.036 |  0.915 30.355  0.838 |  0.035  0.117  0.001
## VegHautPrt | -0.405  4.630  0.164 | -0.419  6.353  0.175 |  0.401 15.459  0.161
## Pct0_14   |  0.552  8.605  0.305 | -0.533 10.281  0.284 |  0.076  0.553  0.006
## Pct_65    | -0.409  4.730  0.168 | -0.271  2.658  0.073 |  0.716 49.258  0.513
## Pct_Img   |  0.874 21.559  0.764 | -0.089  0.288  0.008 |  0.106  1.077  0.011
## TxChom1564 |  0.774 16.893  0.598 | -0.086  0.267  0.007 | -0.068  0.450  0.005
## Pct_brevet |  0.727 14.936  0.529 | -0.434  6.813  0.188 |  0.219  4.612  0.048
## NivVieMed | -0.884 22.062  0.782 |  0.088  0.278  0.008 |  0.038  0.136  0.001
##
## Lden      |
## NO2       |
## PM25      |
## VegHautPrt |
## Pct0_14   |
## Pct_65    |
## Pct_Img   |
## TxChom1564 |
## Pct_brevet |
## NivVieMed |

```

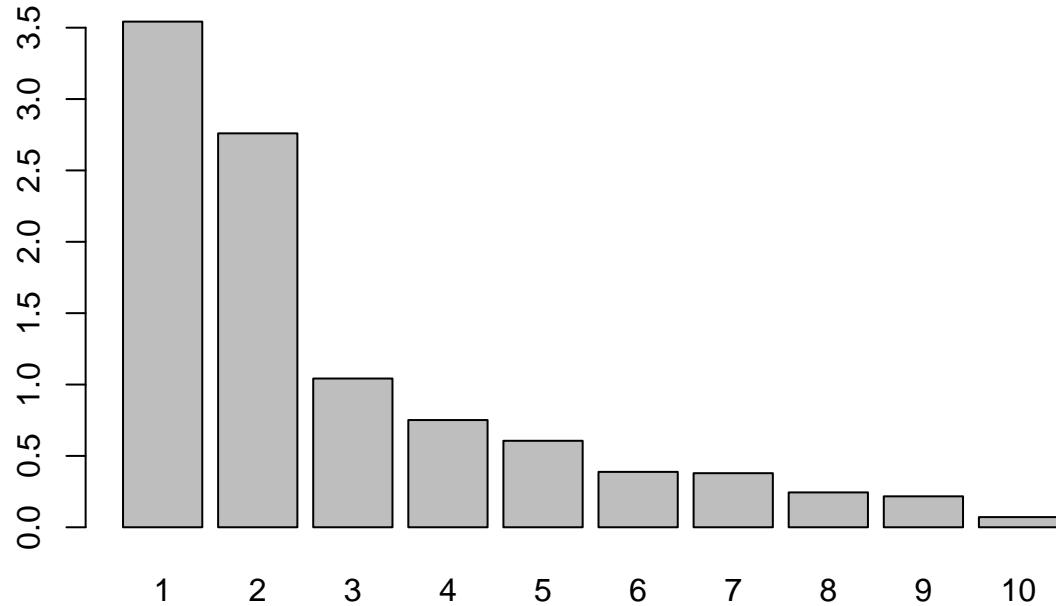
Avec les fonctions de base `barplot` et `plot`, il est possible de construire rapidement des graphiques pour explorer les résultats de l'ACP pour les valeurs propres, les variables et les individus.

```

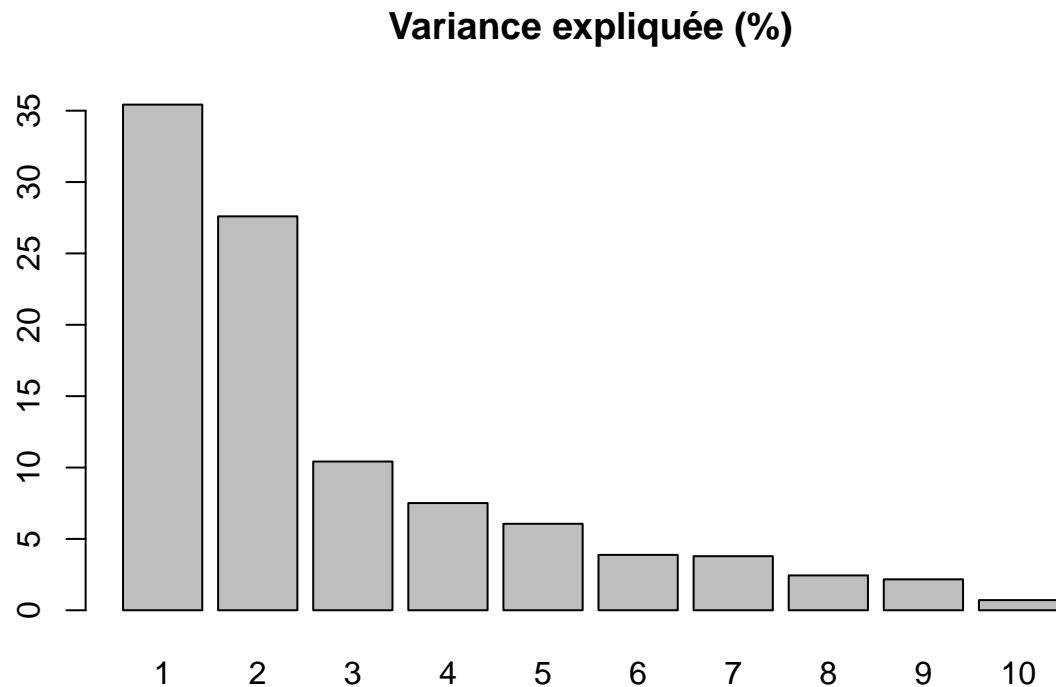
# Graphiques pour les valeurs propres
barplot(res.acp$eig[,1], main="Valeurs propres", names.arg=1:nrow(res.acp$eig))

```

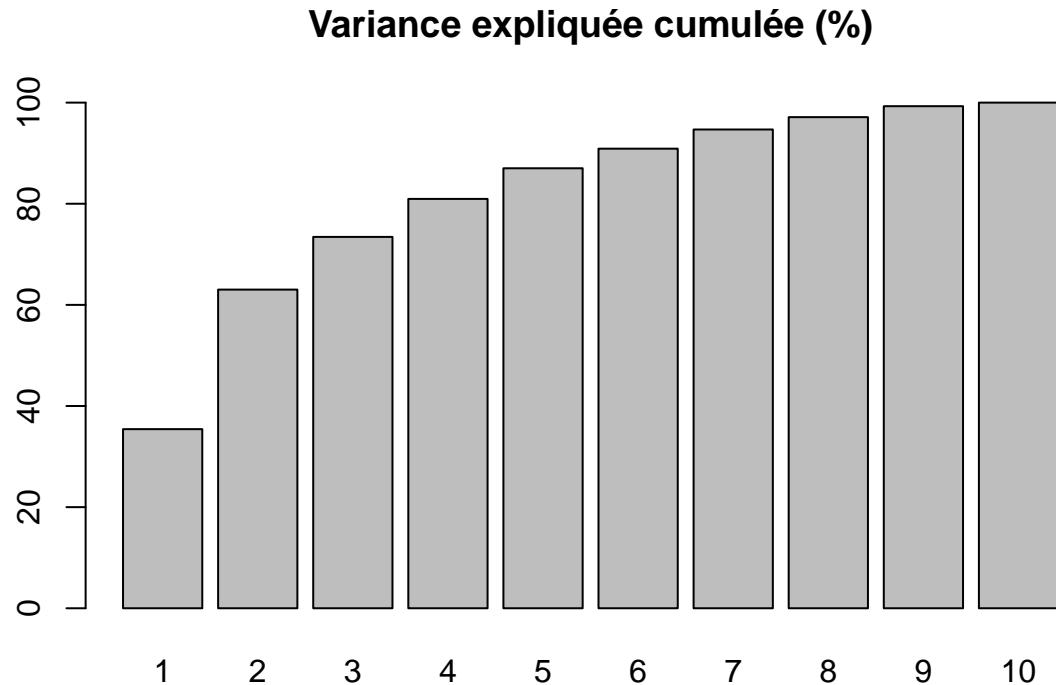
Valeurs propres



```
barplot(res.acp$eig[,2], main="Variance expliquée (%)", names.arg=1:nrow(res.acp$eig))
```

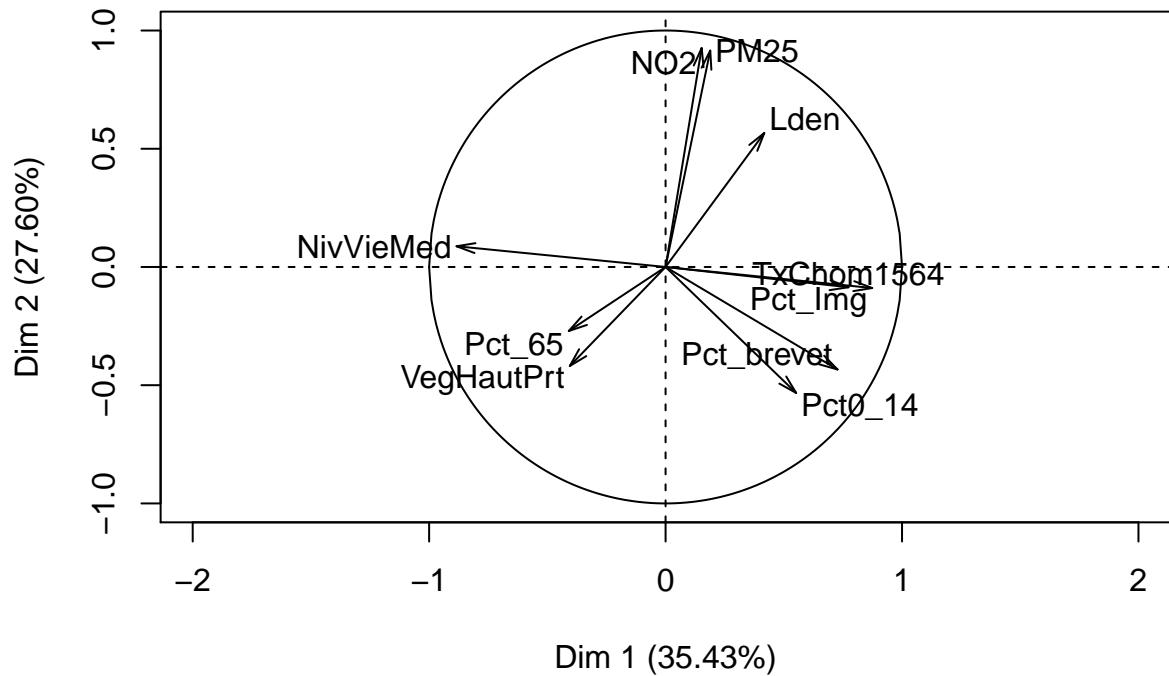


```
barplot(res.acp$eig[,3], main="Variance expliquée cumulée (%)",
        names.arg=1:nrow(res.acp$eig))
```

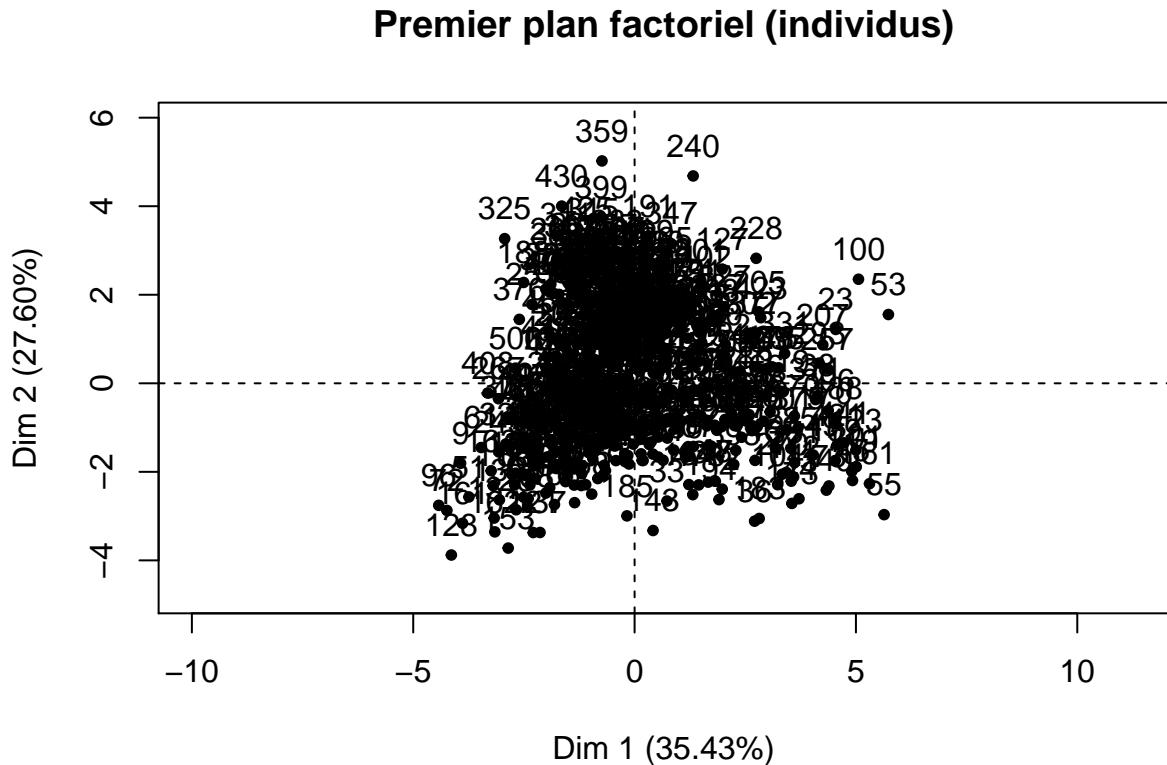


```
# Nuage du points du premier plan factoriel pour les variables et les individus
plot(res.acp, graph.type = "classic", choix="var", axes = 1:2,
     title = "Premier plan factoriel (variables)")
```

Premier plan factoriel (variables)



```
plot(res.acp, graph.type = "classic", choix="ind", axes = 1:2,
     title = "Premier plan factoriel (individus)")
```



Nous avons vu, dans un encadré ci-dessus, qu'il est possible d'ajouter des variables et des individus supplémentaires dans une ACP, ce que permet la fonction `PCA` de `FactoMineR` avec les paramètres `ind.sup` et `quanti.sup`. Aussi, pour ajouter des pondérations aux individus ou aux variables, utiliser les paramètres `row.w` et `col.w`. Pour plus d'informations sur ces paramètres, consulter l'aide de la fonction en tapant `?PCA` dans la console de RStudio.

12.2.3.2 Exploration graphique des résultats de l'ACP avec `factoextra`

Visuellement, vous avez pu constater que les graphiques ci-dessus (pour les valeurs propres et pour le premier plan factoriel pour les variables et les individus), réalisés avec les fonctions de base `barplot` et `plot`, sont peu attrayants. Avec le package `factoextra`, quelques lignes de code suffisent pour construire des graphiques bien plus esthétiques.

Premièrement, la syntaxe ci-dessous renvoie deux graphiques pour analyser les résultats des valeurs propres (figure 12.11).

```
library(factoextra)
library(ggplot2)
library(ggpubr)

# Graphiques pour les variables propres avec factoextra
G1 <- fviz_screenplot(res.acp, choice ="eigenvalue", addlabels = TRUE,
                      x="Composantes",
                      y="Valeur propre",
                      title="")
```

```
G2 <- fviz_screenplot(res.acp, choice ="variance", addlabels = TRUE,
                      x="Composantes",
                      y="Pourcentage de la variance expliquée",
                      title="")
ggarrange(G1, G2)
```

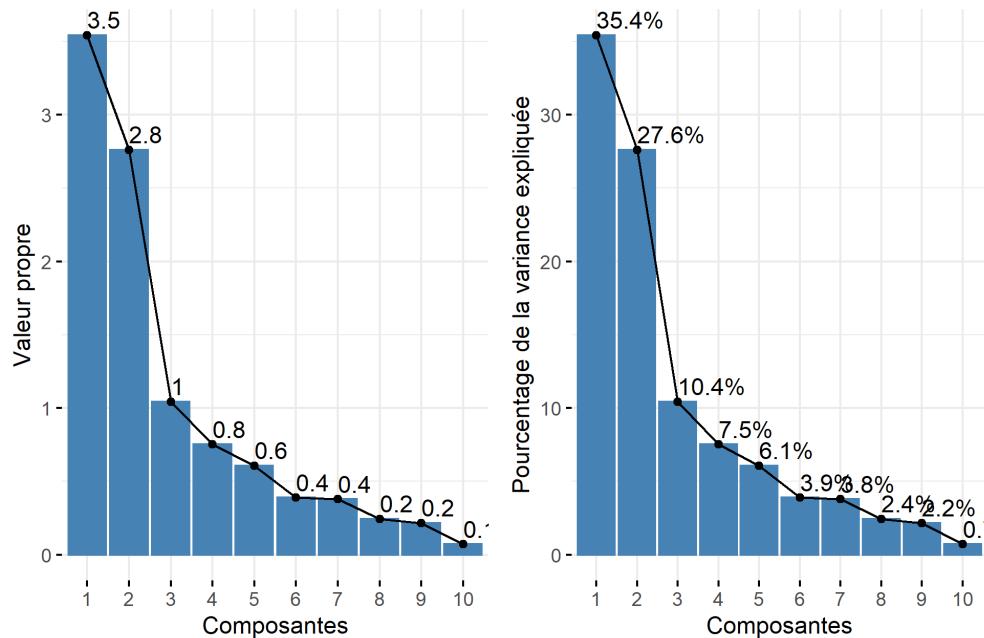


FIG. 12.11 : Graphiques pour les valeurs propres de l'ACP avec factoextra

Deuxièmement, la syntaxe ci-dessous renvoie trois graphiques pour analyser les contributions de chaque variable aux deux premiers axes de l'ACP (figures 12.12 et 12.13) et la qualité de représentation des variables sur les trois premiers axes (figure 12.14), c'est-à-dire la somme des cosinus carrés sur les trois axes retenus.

```
# Contributions des variables aux deux premières composantes avec factoextra
fviz_contrib(res.acp, choice = "var", axes = 1, top = 10,
            title = "Contributions des variables à la première composante")
fviz_contrib(res.acp, choice = "var", axes = 2, top = 10,
            title = "Contributions des variables à la deuxième composante")
fviz_cos2(res.acp, choice = "var", axes = 1:3)+
  labs(x="", y="Somme des cosinus carrés sur les 3 axes retenus",
       title = "Qualité de représentation des variables sur les axes retenus de l'ACP")
```

Troisièmement, le code ci-dessous renvoie un nuage de points pour le premier plan factoriel de l'ACP (axes 1 et 2) pour les variables (figure 12.15) et les individus (figure 12.16).

```
# Premier plan factoriel pour les variables avec factoextra
fviz_pca_var(res.acp, col.var="contrib",
             title = "Premier plan factoriel pour les variables")+
  scale_color_gradient2(low="#313695", mid="#ffffbf", high="#a50026",
                        midpoint=mean(res.acp$var$contrib[,1]))
```

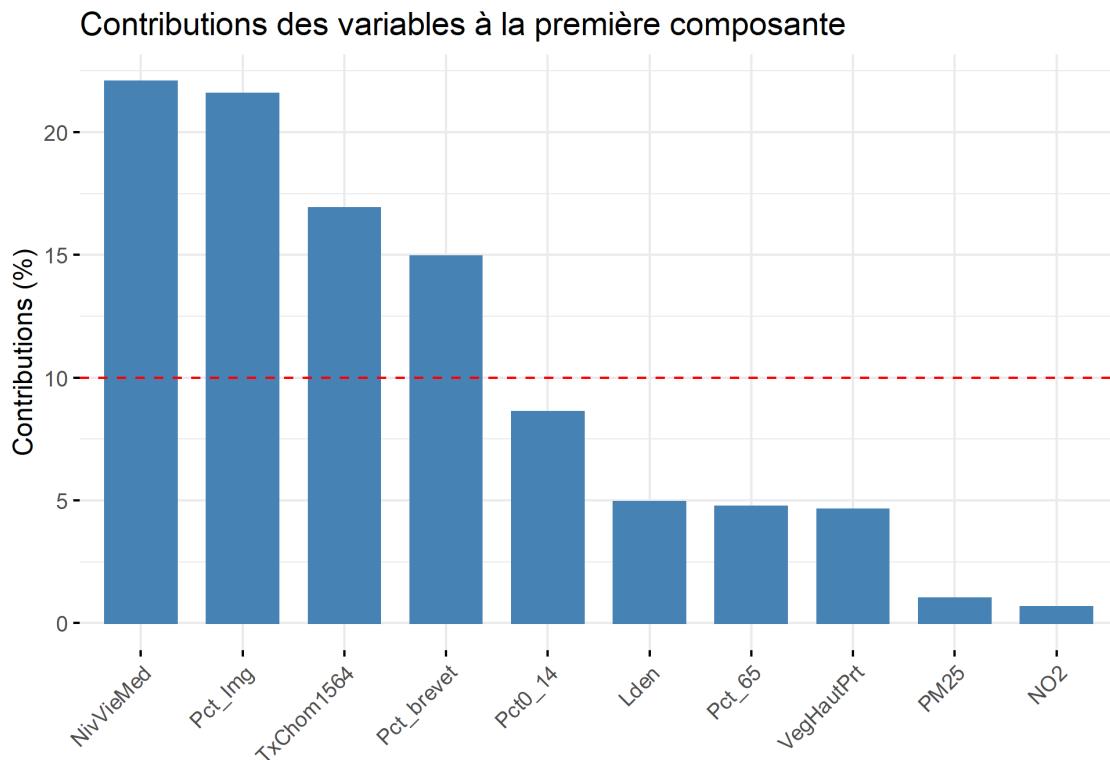


FIG. 12.12 : Contributions des variables à la première composante avec factoextra

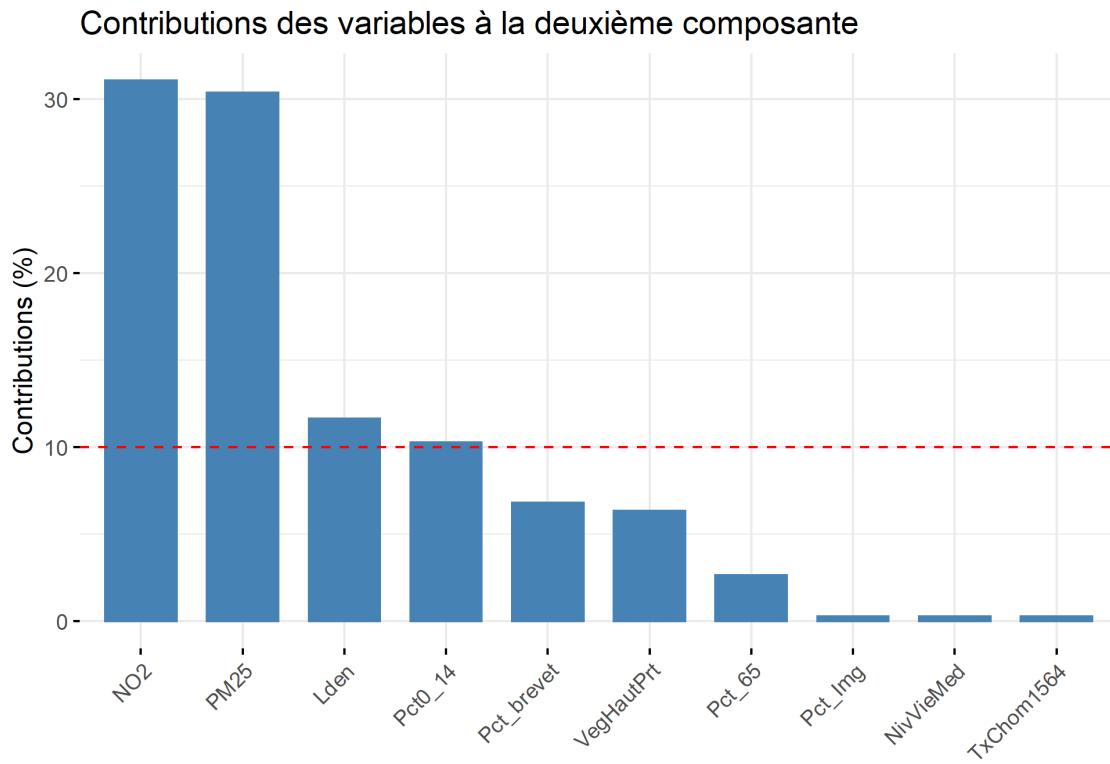


FIG. 12.13 : Contributions des variables à la deuxième composante avec factoextra

Qualité de représentation des variables sur les axes retenus de l'ACP

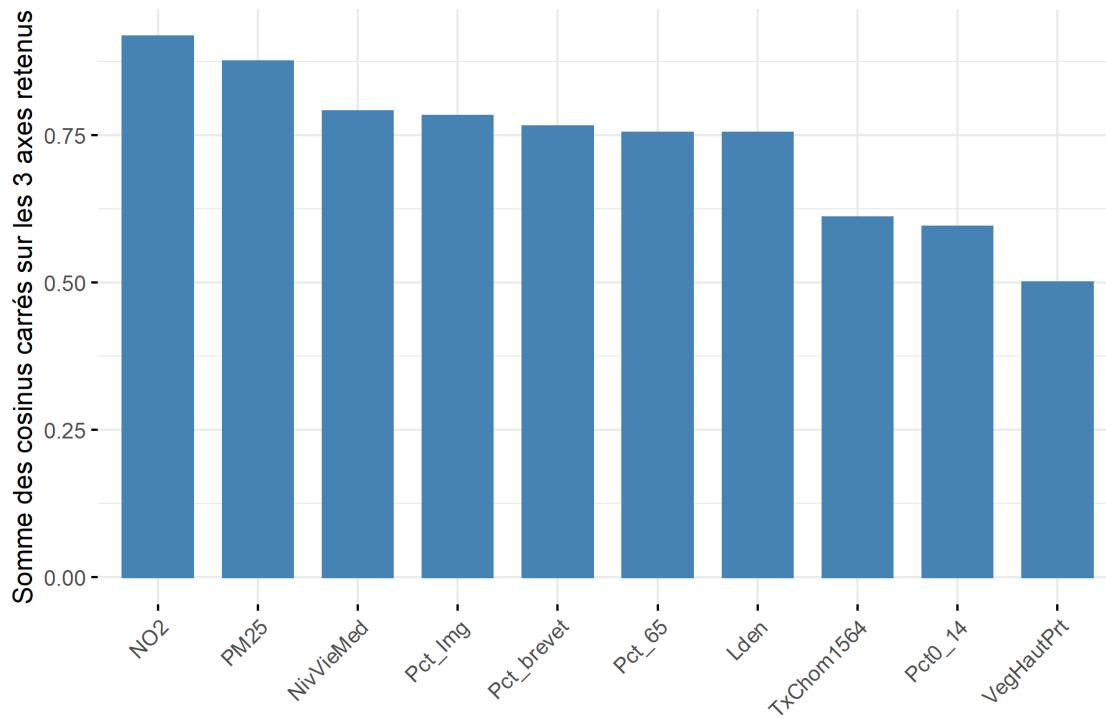


FIG. 12.14 : Qualité des variables sur les trois premières composantes avec factoextra

```
# Premier plan factoriel pour les individus avec factoextra
fviz_pca_ind(res.acp, label="none", title="ACP. Individus")
fviz_pca_ind(res.acp, col.ind="cos2", title="ACP. Individus") +
  scale_color_gradient2(low="blue", mid="white", high="red", midpoint=0.50)
```

12.2.3.3 Personnalisation des graphiques avec les résultats de l'ACP

Avec un peu plus de lignes de code et l'utilisation d'autres *packages* (*ggplot2*, *ggpubr*, *stringr*, *corrplot*), vous pouvez aussi construire des graphiques personnalisés.

Premièrement, la syntaxe ci-dessous permet de réaliser trois graphiques pour analyser les valeurs propres (figure 12.17). Notez que, d'un coup d'œil, nous pouvons identifier les composantes principales avec une valeur propre égale ou supérieure à 1.

```
library(ggplot2)
library(ggpubr)
library(stringr)
library(corrplot)

# Calcul de l'ACP
res.acp <- PCA(Data[,2:11], ncp=5, scale.unit=TRUE, graph=F)
print(res.acp)

# Construction d'un DataFrame pour les valeurs propres
dfACPvp <- data.frame(res.acp$eig)
```

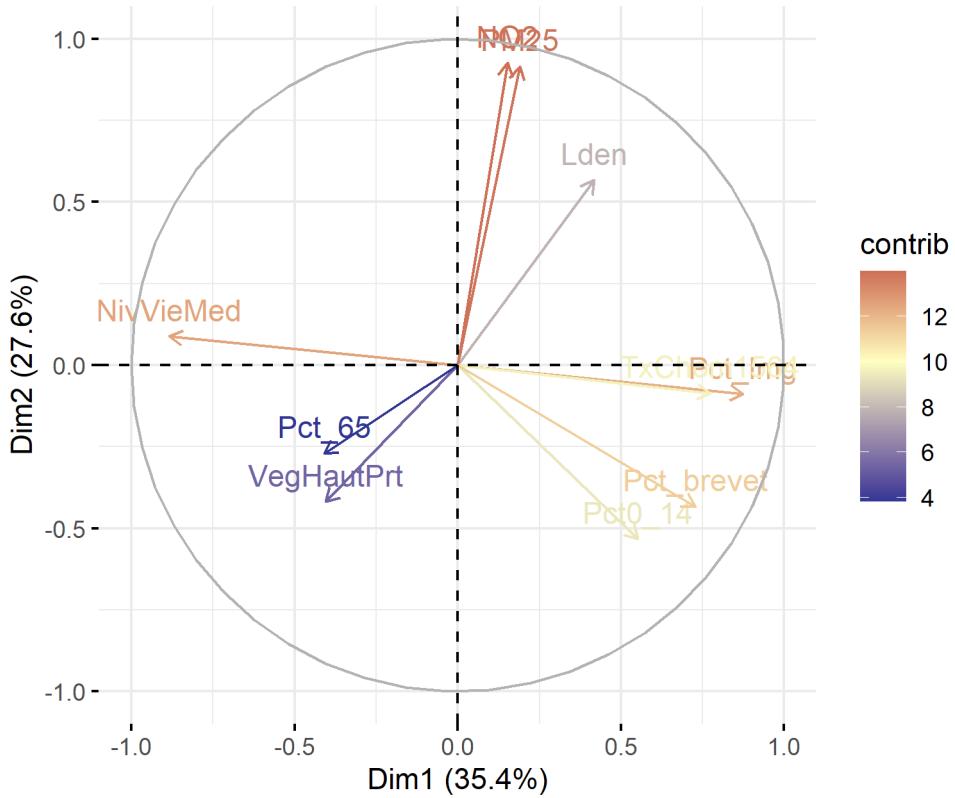


FIG. 12.15 : Premier plan factoriel de l'ACP pour les variables avec factoextra

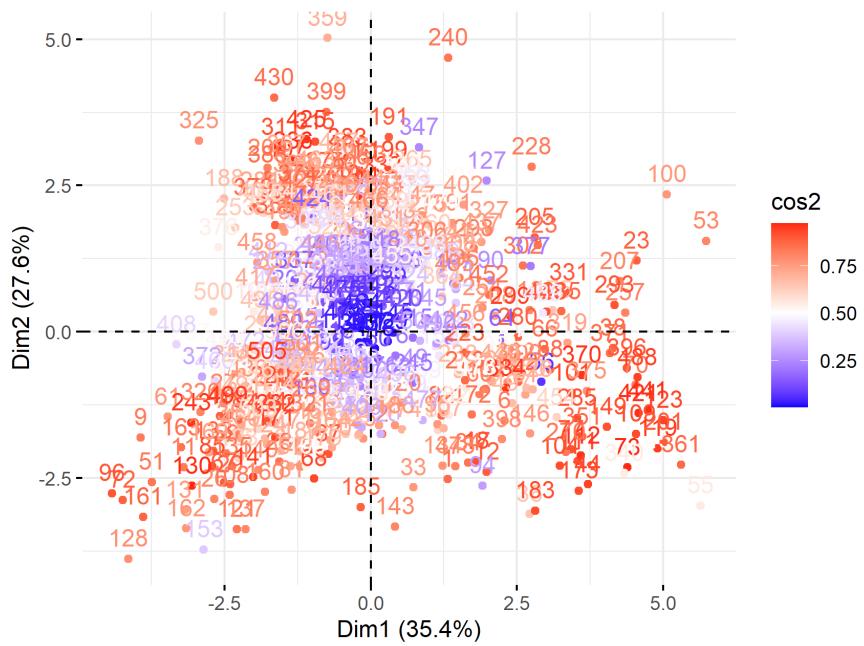


FIG. 12.16 : Premier plan factoriel de l'ACP pour les individus avec factoextra

```

names(dfACPvp) <- c("VP", "VP_pct", "VP_cumupct")
dfACPvp$Composante <- factor(1:nrow(dfACPvp), levels=rev(1:nrow(dfACPvp)))
couleursAxes <- c("steelblue", "skyblue2")
vpsup1 <- round(sum(subset(dfACPvp, VP >= 1)$VP), 2)
vpsup1cumul <- round(sum(subset(dfACPvp, VP >= 1)$VP_pct), 2)

plotVP1 <- ggplot(dfACPvp, aes(x=VP, y=Composante, fill=VP<1))+  

  geom_bar(stat="identity", width = .6, alpha=.8, color="black")+
  geom_vline(xintercept=1, linetype="dashed", color = "azure4", size=1)+  

  scale_fill_manual(name="Valeur\npropre", values=couleursAxes, labels = c(">= 1", "< 1"))+  

  labs(x="Valeur propre", y="Composante principale")
plotVP2 <- ggplot(dfACPvp, aes(x=VP_pct, y=Composante, fill=VP<1))+  

  geom_bar(stat="identity", width = .6, alpha=.8, color="black")+
  scale_fill_manual(name="Valeur\npropre", values=couleursAxes, labels = c(">= 1", "< 1"))+  

  theme(legend.position="none")+
  labs(x="Pourcentage de la variance expliquée", y="")
plotVP3 <- ggplot(dfACPvp, aes(x=VP_cumupct, y=Composante, fill=VP<1, group=1))+  

  geom_bar(stat="identity", width = .6, alpha=.8, color="black")+
  scale_fill_manual(name="Valeur\npropre", values=couleursAxes, labels = c(">= 1", "< 1"))+  

  geom_line(colour="brown", linetype="solid", size=.8) +
  geom_point(size=3, shape=21, color="brown", fill="brown")+
  theme(legend.position="none")+
  labs(x="Pourcentage cumulé de la variance expliquée", y="")

text1 <- paste0("Somme des valeurs propres supérieures à 1 : ",  

  vpsup1,  

  ".\nPourcentage cumulé des valeurs propres supérieures à 1 : ",  

  vpsup1cumul, "%.")

annotate_figure(ggarrange(plotVP1, plotVP2, plotVP3, ncol=2, nrow=2),
  text_grob("Analyse des valeurs propres",
            color = "black", face = "bold", size = 12),
  bottom = text_grob(text1,
                     color = "black", hjust = 1, x = 1, size = 10))

```

Deuxièmement, la syntaxe ci-dessous permet de :

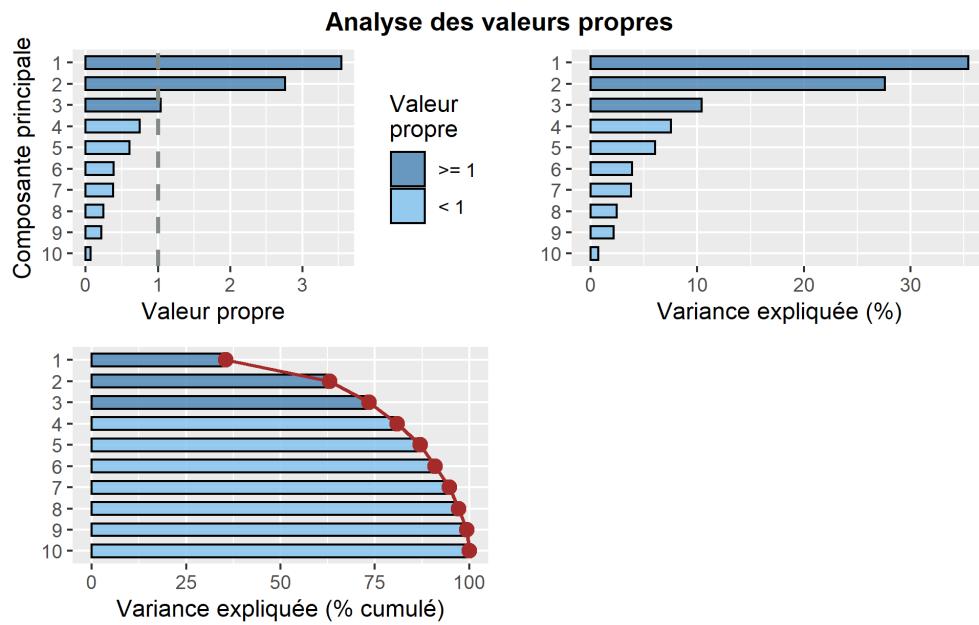
- Construire un *DataFrame* avec les résultats des variables.
- Construire des histogrammes avec les coordonnées des variables sur les axes factoriels (figure 12.18). Notez que les coordonnées négatives sont indiquées avec des barres bleues et celles positives, avec des barres de couleur saumon.
- Un graphique avec les contributions des variables sur les axes retenus (figure 12.19).
- Un graphique avec les cosinus carrés des variables sur les axes retenus (figure 12.20).
- Un histogramme avec la qualité des variables sur les axes retenus (figure 12.21), soit la sommation de leurs cosinus carrés sur les axes retenus.

```

# Analyse des résultats de L'ACP pour les variables
library(corrplot)
library(stringr)
library(ggplot2)
library(ggpubr)

# Indiquer le nombre d'axes à conserver suite à l'analyse des valeurs propres
nComp <- 3

```



Somme des valeurs propres supérieures à 1 : 7.34.
Pourcentage cumulé des valeurs propres supérieures à 1 : 73.44%.

FIG. 12.17 : Graphiques personnalisés pour les valeurs propres

```
# Variance expliquée par les axes retenus
vppct <- round(dfACPvp[1:nComp],"VP_pct"),1)
# Dataframe des résultats pour les variables
CoordsVar <- res.acp$var$coord[, 1:nComp]
Cos2Var   <- res.acp$var$cos2[, 1:nComp]
CtrVar    <- res.acp$var$contrib[, 1:nComp]
dfACPVars <- data.frame(Variable = row.names(res.acp$var$coord[, 1:nComp]),
                        Coord = CoordsVar,
                        Cos2 = Cos2Var,
                        Qualite = rowSums(Cos2Var),
                        Ctr = CtrVar)
row.names(dfACPVars) <- NULL
names(dfACPVars) <- str_replace(names(dfACPVars), ".Dim.", "Comp")
dfACPVars

# Histogrammes pour les coordonnées
couleursCoords <- c("lightsalmon","steelblue")
plotCoordF1 <- ggplot(dfACPVars,
                       aes(y = reorder(Variable, CoordComp1),
                           x = CoordComp1, fill=CoordComp1<0))+ 
  geom_bar(stat="identity", width = .6, alpha=.8, color="black")+
  geom_vline(xintercept=0, color = "black", size=1)+ 
  scale_fill_manual(name="Coordonnée",values=couleursCoords,
                    labels = c("Positive","Négative"))+
  labs(x=paste0("Axe 1 (", vppct[1],"%)"), y="Variable")+
  theme(legend.position="none")
plotCoordF2 <- ggplot(dfACPVars,
                       aes(y = reorder(Variable, CoordComp2),
                           x = CoordComp2, fill=CoordComp2<0))+ 
  geom_bar(stat="identity", width = .6, alpha=.8, color="black")+
  geom_vline(xintercept=0, color = "black", size=1)+ 
  scale_fill_manual(name="Coordonnée",values=couleursCoords,
                    labels = c("Positive","Négative"))+
  labs(x=paste0("Axe 2 (", vppct[2],"%)"), y="Variable")+
  theme(legend.position="none")
```

```

geom_vline(xintercept=0, color = "black", size=1)+
scale_fill_manual(name="Coordonnée",values=couleursCoords,
                  labels = c("Positive","Négative"))+
labs(x=paste0("Axe 2 (", vppct[2],"%)"), y="Variable")+
theme(legend.position="none")
plotCoordF3 <- ggplot(dfACPVars,
                      aes(y = reorder(Variable, CoordComp3),
                          x = CoordComp3, fill=CoordComp3<0))+ 
geom_bar(stat="identity", width = .6, alpha=.8, color="black")+
geom_vline(xintercept=0, color = "black", size=1)+
scale_fill_manual(name="Coordonnée", values=couleursCoords,
                  labels = c("Positive","Négative"))+
labs(x=paste0("Axe 3 (", vppct[3],"%)"), y="Variable")

annotate_figure(ggarrange(plotCoordF1, plotCoordF2, plotCoordF3, nrow=nComp),
               text_grob("Coordonnées des variables sur les axes factoriels",
                         color = "black", face = "bold", size = 12))

# Contributions des variables à la formation des axes
corrplot(CtrVar, is.corr=FALSE, method ="square",
          addCoef.col = 1, cl.pos = FALSE)

# La qualité des variables sur les composantes retenues : cosinus carrés
corrplot(Cos2Var, is.corr=FALSE, method ="square",
          addCoef.col = 1, cl.pos = FALSE)

ggplot(dfACPVars)+ 
  geom_bar(aes(y=reorder(Variable, Qualite), x=Qualite),
           stat="identity", width = .6, alpha=.8, fill="steelblue")+
  labs(x="", y="Somme des cosinus carrés sur les axes retenus",
       title ="Qualité de représentation des variables sur les axes retenus de l'ACP",
       subtitle = paste0("Variance expliquée par les ", nComp,
                        " composantes : ", sum(vppct), "%"))

```

Troisièmement, lorsque les observations sont des unités spatiales, il convient de cartographier les coordonnées factorielles des individus. Dans le jeu de données utilisé, les observations sont des polygones délimitant les îlots regroupés pour l'information statistique (IRIS) pour l'agglomération de Lyon (France). Nous utilisons les packages `tmap` et `RColorBrewer` pour réaliser des cartes choroplèthes avec les coordonnées des deux premières composantes (figure 12.22).

```

library("tmap")
library("RColorBrewer")
# Analyse des résultats de l'ACP pour les individus
# Dataframe des résultats pour les individus
CoordsInd <- res.acp$ind$coord[, 1:nComp]
Cos2Ind   <- res.acp$ind$cos2[, 1:nComp]
CtrInd    <- res.acp$ind$contrib[, 1:nComp]
dfACPInd <- data.frame(Coord = CoordsInd, Cos2 = Cos2Ind, Ctr = CtrInd)
names(dfACPInd) <- str_replace(names(dfACPInd), ".Dim.", "Comp")
# Fusion du tableau original avec les résultats de l'ACP pour les individus
CartoACP <- cbind(LyonIris, dfACPInd)
# Cartographie des coordonnées factorielles pour les individus pour les
# deux premières composantes
Carte1 <- tm_shape(CartoACP) +

```

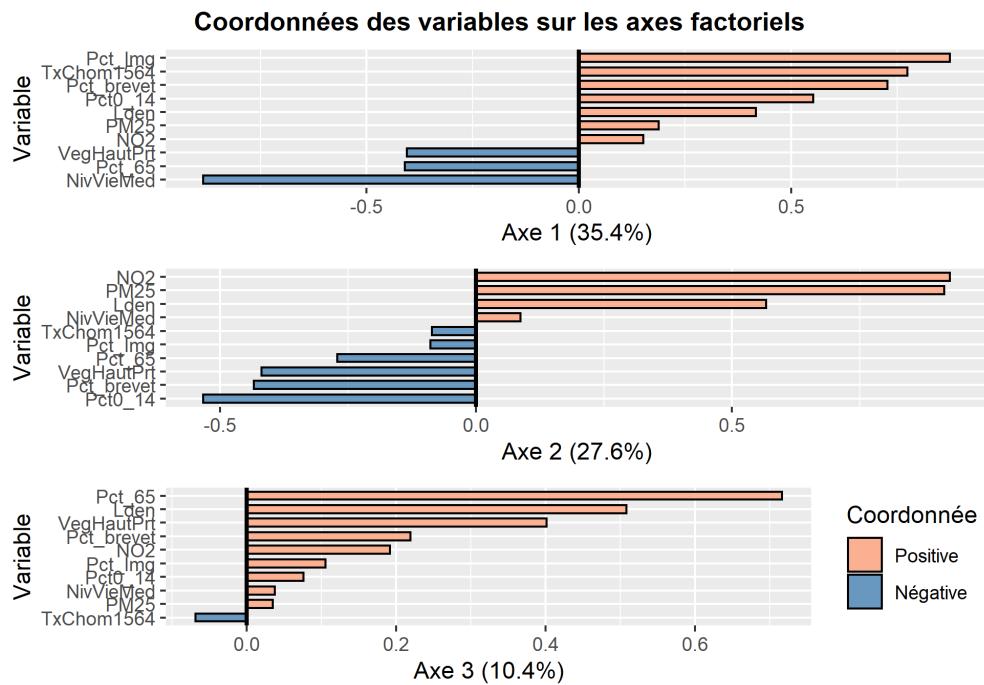


FIG. 12.18 : Histogrammes personnalisés avec les coordonnées factorielles pour les variables

```
tm_polygons(col = "CoordComp1", style = "cont",
            midpoint = 0, title = 'Coordinées')+
  tm_layout(main.title = paste0("Axe 1 (", vppct[1],"%)"),
            attr.outside = TRUE, frame = FALSE, main.title.size = 1)
Carte2 <- tm_shape(CartoACP) +
  tm_polygons(col = "CoordComp2", style = "cont",
              midpoint = 0, title = 'Coordinées')+
  tm_layout(main.title = paste0("Axe 2 (", vppct[2],"%)"),
            attr.outside = TRUE, frame = FALSE, main.title.size = 1)
tmap_arrange(Carte1, Carte2)
```



Exploration interactive des résultats d'une ACP avec le package *explor*.

Vous avez compris qu'il ne suffit pas de calculer une ACP, il faut retenir les n premiers axes de l'ACP qui nous semblent les plus pertinents, puis les interpréter à la lecture des coordonnées factorielles, des cosinus carrés et des contributions des variables et des individus sur les axes. Il faut donc bien explorer les résultats à l'aide de tableaux et de graphiques! Cela explique que nous avons mobilisé de nombreux graphiques dans les deux sections précédentes (12.2.3.2 et 12.2.3.3). L'exploration des données d'une ACP peut aussi être réalisée avec des graphiques interactifs. Or, un superbe package dénommé *explor* (<https://juba.github.io/explor/>), reposant sur *Shiny* (<https://shiny.rstudio.com/>), permet d'explorer de manière interactive les résultats de plusieurs méthodes factorielles calculés avec *FactorMinerR*. Pour cela, il vous suffit de lancer les deux lignes de code suivantes :

```
library(explor)
explor(res.acp)
```

Finalement, *explor* permet également d'explorer les résultats d'une analyse des correspondances (AFC) et d'une analyse des correspondances multiples (ACM).

	Dim.1	Dim.2	Dim.3
Lden	4.92	11.64	24.8
NO2	0.66	31.07	3.54
PM25	1.01	30.36	0.12
VegHautPrt	4.63	6.35	15.46
Pct0_14	8.61	10.28	0.55
Pct_65	4.73	2.66	49.26
Pct_Img	21.56	0.29	1.08
TxChom1564	16.89	0.27	0.45
Pct_brevet	14.94	6.81	4.61
NivVieMed	22.06	0.28	0.14

FIG. 12.19 : Graphiques personnalisés avec les contributions des variables

12.3 Analyse factorielle des correspondances (AFC)



Pour bien comprendre l'AFC, il est essentiel de bien maîtriser les notions de tableau de contingence (marges du tableau, fréquences observées et théoriques, pourcentages en ligne et en colonne, contributions au khi-deux) et de distance du khi-deux. Si ce n'est pas le cas, il est conseillé de (re)lire le chapitre 5.

Dans le chapitre 5, nous avons vu comment construire un tableau de contingence (figure 12.23) à partir de deux variables qualitatives comprenant plusieurs modalités, puis comment vérifier s'il y a dépendance entre les deux variables qualitatives avec le test du khi-deux. Or, s'il y a bien dépendance, il est peut-être judicieux de résumer l'information que contient le tableau de contingence en quelques nouvelles variables synthétiques, objectif auquel répond l'analyse factorielle des correspondances (AFC).

À titre de rappel (section 12.1.2), l'AFC a été développée par le statisticien français Jean-Paul Benzécri (1973). Cela explique qu'elle est souvent enseignée et utilisée en sciences sociales dans les universités francophones, mais plus rarement dans les universités anglophones. Pourtant, les applications de l'AFC sont nombreuses dans différentes disciplines des sciences sociales comme illustrées avec les exemples suivants :

- En géographie, les modalités de la première variable du tableau de contingence sont souvent des

	Dim.1	Dim.2	Dim.3
Lden	0.17	0.32	0.26
NO2	0.02	0.86	0.04
PM25	0.04	0.84	0
VegHautPrt	0.16	0.18	0.16
Pct0_14	0.3	0.28	0.01
Pct_65	0.17	0.07	0.51
Pct_Img	0.76	0.01	0.01
TxChom1564	0.6	0.01	0
Pct_brevet	0.53	0.19	0.05
NivVieMed	0.78	0.01	0

FIG. 12.20 : Graphiques personnalisés avec les cosinus carrés des variables

entités spatiales (régions, municipalités, quartiers, etc.) croisées avec les modalités d'une autre variable qualitative (catégories socioprofessionnelles, modes de transport, tranches de revenu des ménages ou des individus, etc.).

- En économie régionale, nous pourrions vouloir explorer un tableau de contingence croisant des entités spatiales (par exemple, MRC au Québec, départements en France) et les effectifs d'emplois pour différents secteurs d'activité.
- En sciences politiques, le recours à l'AFC peut être intéressant pour explorer les résultats d'une élection. Les deux variables qualitatives pourraient être les *circonscriptions électorales* et les *partis politiques*. Le croisement des lignes et des colonnes du tableau de contingence représenterait le nombre de votes obtenus par un parti politique j dans la circonscription électorale i . Appliquer une AFC sur un tel tableau de contingence permettrait de révéler les ressemblances entre les différents partis politiques et celles entre les circonscriptions électorales.



Application d'une ACP sur un tableau de contingence transformé en un tableau avec les pourcentages en ligne : un bien mauvais calcul...

Il pourrait être tentant de transformer le tableau de contingence initial (tableau 12.7) en un tableau avec les pourcentages en ligne (tableau 12.8) afin de lui appliquer une analyse en composantes principales. Une telle démarche a deux inconvénients majeurs : chacune des modalités de la première variable qualitative (I) aurait

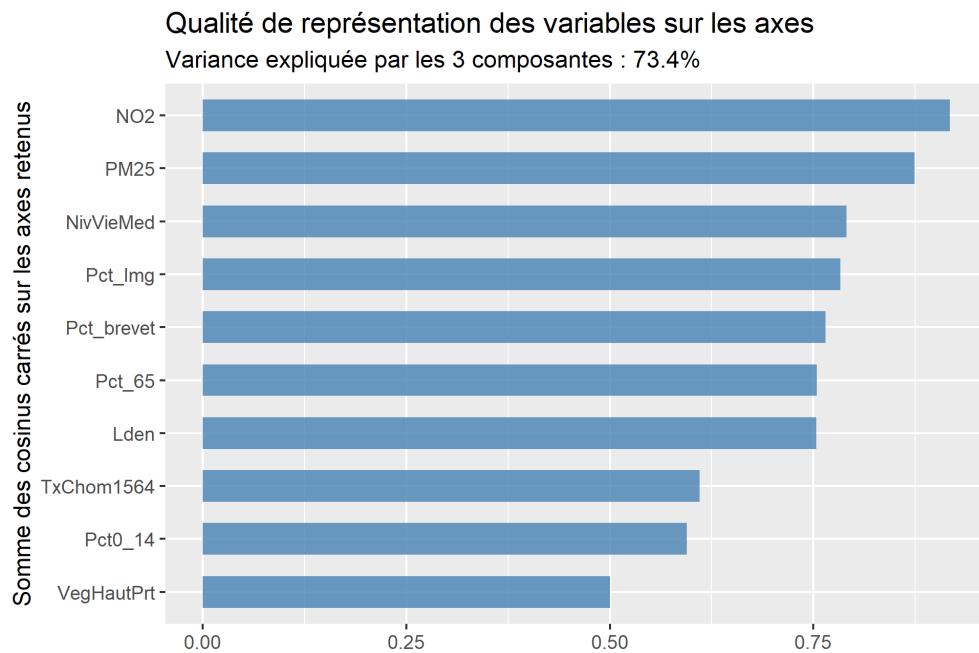


FIG. 12.21 : Graphique personnalisé avec la qualité des variables sur les axes retenus de l'ACP

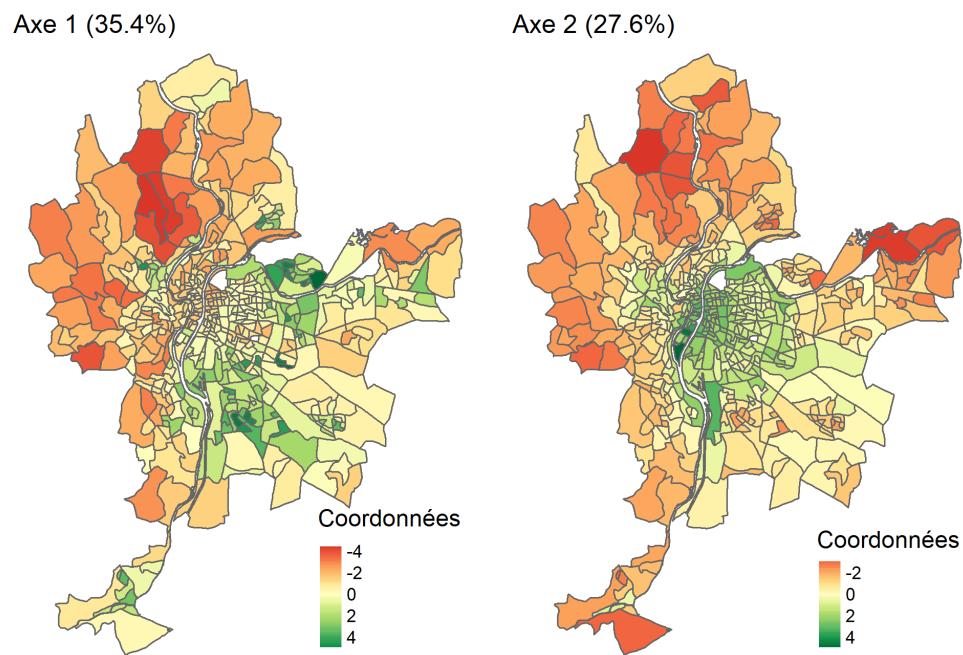


FIG. 12.22 : Cartographie des coordonnées factorielles des individus

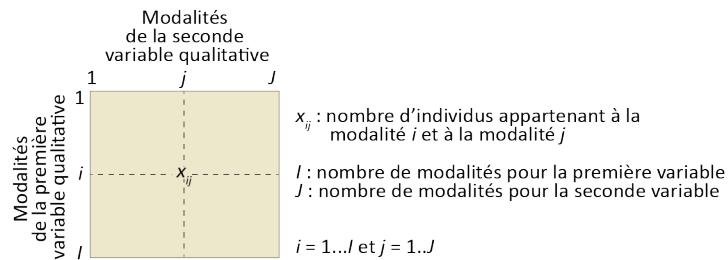


FIG. 12.23 : Tableau de contingence pour une AFC

alors le même poids ; chacune des modalités de la deuxième variable (J) aurait aussi le même poids. Or, à la lecture des marges en ligne et en colonne au tableau 12.7, il est clair que les modalités j_1 et i_1 comprennent bien plus d'individus que les autres modalités respectives.

Si nous reprenons le dernier exemple applicatif, cela signifierait que le même poids est accordé à chaque parti puisque les variables sont centrées réduites en ACP (moyenne = 0 et variance = 1). Autrement dit, les grands partis traditionnels seraient ainsi sur le pied d'égalité que les autres partis. Aussi, chaque circonscription électorale aurait le même poids bien que certaines comprennent bien plus d'électeurs et d'électrices que d'autres.

TAB. 12.7 : Exemple de tableau de contingence pour l'AFC

	j1	j2	j3	j4	j5	Marge (ligne)
i1	357 060	22 010	276 625	65 000	29 415	750 110
i2	427 530	26 400	295 860	69 410	30 645	849 845
i3	147 500	6 545	34 545	4 415	1 040	194 045
i4	128 520	6 405	42 925	6 565	2 670	187 085
Marge (colonne)	1 060 610	61 360	649 955	145 390	63 770	1 981 085

TAB. 12.8 : Exemple d'un tableau de contingence transformé (pourcentage en ligne) pour l'ACP

	V1	V2	V3	V4	V5
i1	47,6	2,9	36,9	8,7	3,9
i2	50,3	3,1	34,8	8,2	3,6
i3	76,0	3,4	17,8	2,3	0,5
i4	68,7	3,4	22,9	3,5	1,4

12.3.1 Recherche d'une simplification basée sur la distance du khi-deux

Sur le plan mathématique et des objectifs visés, l'AFC est similaire à l'ACP puisqu'elle permet d'explorer un tableau de trois façons : 1) en montrant les ressemblances entre un ensemble d'individus (I), 2) en révélant les liaisons entre les variables (J) et 3) en résumant le tout avec des variables synthétiques. Toutefois, avec l'AFC, les ensembles I et J sont les modalités de deux variables qualitatives (dont le croisement forme un tableau de contingence) et elle est basée sur la distance du khi-deux (et non sur la distance euclidienne comme en ACP).

Ainsi, avec la distance du khi-deux, la proximité (ressemblance) entre deux lignes (i et l) et deux colonnes (j et k) est mesurée comme suit :

$$d_{\chi^2_{il}} = \sum_j \frac{1}{f_{.j}} \left(\frac{f_{ij}}{f_{i.}} - \frac{f_{lj}}{f_{l.}} \right)^2 \quad (12.5)$$

$$d_{\chi^2_{jk}} = \sum_i \frac{1}{f_{i.}} \left(\frac{f_{ij}}{f_{.j}} - \frac{f_{ik}}{f_{.k}} \right)^2 \quad (12.6)$$

Prenons un exemple fictif pour calculer ces deux distances. Le tableau 12.9 comprend trois modalités en ligne (I) et trois autres en colonne (J). Le total des effectifs de ce tableau de contingence est égal à 1 665.

À partir des données brutes, il est facile de construire deux tableaux : le profil des lignes et le profil des colonnes (tableau 12.10), c'est-à-dire les proportions en ligne et en colonne.

Tab. 12.9 : Données brutes du tableau de contingence

	j1	j2	j3	Total (ligne)
i1	360	65	275	700
i2	420	70	290	780
i3	145	5	35	185
Total (colonnes)	925	140	600	1 665

Tab. 12.10 : Profils des lignes et des colonnes

	j1	j2	j3	Total
Profil des lignes				
i1	0,514	0,093	0,393	1
i2	0,538	0,090	0,372	1
i3	0,784	0,027	0,189	1
Profils des colonnes				
i1	0,389	0,464	0,458	
i2	0,454	0,500	0,483	
i3	0,157	0,036	0,058	
Total	1,000	1,000	1,000	

En divisant les valeurs du tableau 12.9 par le grand total (1 665), nous obtenons tous les termes utilisés dans les équations (12.5) et (12.6) au tableau 12.11 :

- Les fréquences relatives dénommées f_{ij} .
- La marge $f_{i.}$ est égale à la somme des fréquences relatives en ligne.
- La marge $f_{.j}$ est égale à la somme des fréquences relatives en colonne.
- La somme de toutes les fréquences relatives est donc égale à 1, soit $\sum f_{i.}$ ou $\sum f_{.j}$.

Tab. 12.11 : Données relatives du tableau de contingence (f_{ij})

	j1	j2	j3	Total ($f_{i.}$)
i1	0,216	0,039	0,165	0,420
i2	0,252	0,042	0,174	0,468
i3	0,087	0,003	0,021	0,111
Total ($f_{.j}$)	0,556	0,084	0,360	1,000

Il est possible de calculer les distances entre les différentes modalités de I en appliquant l'équation (12.5); par exemple, la distance entre les observations i1 et i2 est égale à :

$$d_{(i1,i2)} = \frac{1}{0,556} (0,216 - 0,252)^2 + \frac{1}{0,084} (0,039 - 0,042)^2 + \frac{1}{0,360} (0,165 - 0,174)^2 = 0,003$$

Avec l'équation (12.6), la distance entre les modalités j_1 et j_2 de J est égale à :

$$d_{(j1,j2)} = \frac{1}{0,420} (0,216 - 0,039)^2 + \frac{1}{0,468} (0,252 - 0,042)^2 + \frac{1}{0,111} (0,087 - 0,003)^2 = 0,233$$

À la lecture du tableau 12.12, les modalités les plus semblables sont i_1 et i_2 (0,003) pour I et j_1 et j_3 (0,058) pour J .

Finalement, l'approche pour déterminer les axes factoriels de l'AFC est similaire à celle de l'ACP : les axes factoriels sont les droites orthogonales qui minimisent les distances aux points du profil des lignes, excepté que la métrique pour mesurer ces distances est celle du khi-deux (et non celle la distance euclidienne comme pour l'ACP). Pour plus détails sur le calcul de ces axes (notamment les formulations matricielles), consultez notamment Benzécri (1973), Escofier et Pagès (1998) et Lebart et al. (1995).

12.3.2 Aides à l'interprétation

Pour illustrer les aides à l'interprétation de l'AFC, nous utilisons un jeu de données spatiales extrait du profil du recensement de 2016 de Statistique Canada¹ pour les secteurs de recensement de l'île de Montréal. La liste des modalités des variables qu'il comprend est reportée au tableau 12.13. L'AFC est calculée sur un tableau de contingence croisant les secteurs de recensement (lignes) et les modalités d'une variable relative au mode de transport utilisé pour les déplacements domicile-travail (colonnes). Ces modalités sont cartographiées à la figure 12.24.

TAB. 12.13 : Jeu de données utilisé pour l'analyse factorielle des correspondances

Nom	Intitulé	Somme
Modalités de la variable utilisée dans l'AFC (mode de transport)		
VehCond	Véhicule motorisé (conducteur-trice)	427 560
VehPass	Véhicule motorisé (passager-ère)	26 490
TranspC	Transport en commun	295 800
Apied	À pied	69 330
Velo	Bicyclette	30 615
AutreMoyen	Autre moyen	7 750
Modalités de la variable supplémentaire (durée du trajet)		
T15min	Moins de 15 minutes	130 435
T1529min	15 à 29 minutes	287 500
T3044min	30 à 44 minutes	244 425
T4559min	45 à 59 minutes	107 065
T60plus	60 minutes et plus	88 050

¹<https://www12.statcan.gc.ca/census-recensement/2016/dp-pd/prof/index.cfm?Lang=F>

TAB. 12.12 : Distances du khi-deux entre les modalités I et les modalités J

Ind.	i1	i2	i3	Col.	j1	j2	j3
i1	0,000	0,003	0,103	j1	0,000	0,233	0,058
i2	0,003	0,000	0,132	j2	0,233	0,000	0,078
i3	0,103	0,132	0,000	j3	0,058	0,078	0,000

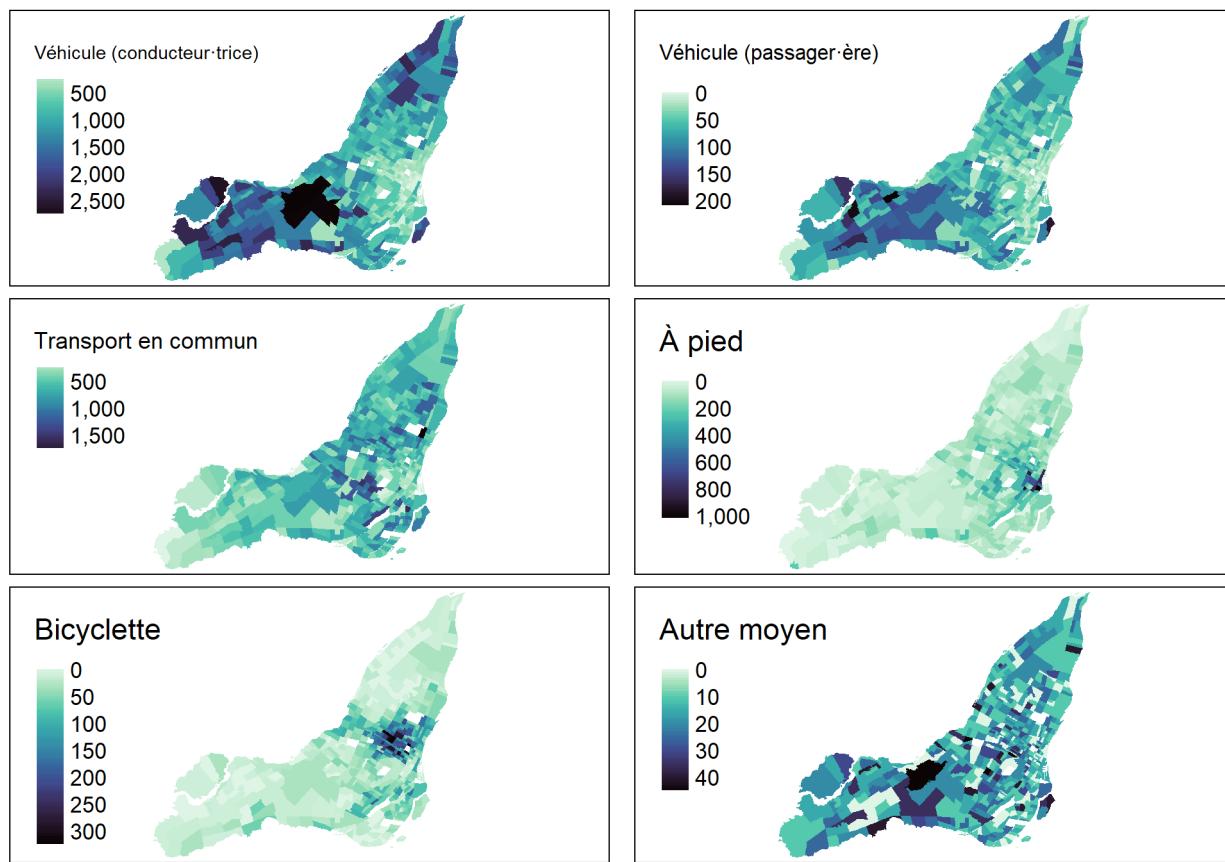


FIG. 12.24 : Cartographie des modalités de la variable mode de transport utilisée pour l’AFC

12.3.2.1 Résultats de l’AFC pour les valeurs propres

Avant de calculer l’AFC, il convient de vérifier s’il y a bien une dépendance entre les modalités des deux variables qualitatives. En effet, si les deux variables sont indépendantes, il n’est pas nécessaire de résumer le tableau de contingence avec une AFC. Pour ce faire, nous utilisons le test du khi-deux largement décrit à la section 5.2. Les résultats de ce test signalent qu’il existe des associations entre les modalités des deux variables ($\chi^2 = 203\,971$, $p < 0,001$, tableau 12.14). Nous pouvons donc appliquer une AFC sur ce tableau de contingence.

Nous avons vu qu’en ACP normée (section 12.2.2.1), la somme des valeurs propres est égale au nombre de variables puisqu’elles sont centrées réduites. Par contre, en AFC, cette somme est égale à l’inertie totale du tableau de contingence, c'est-à-dire à la valeur du khi-deux divisée par le nombre total des effectifs

TAB. 12.14 : Résultats du test du khi-deux sur le tableau de contingence

Mesure	Valeur
Modalités I (secteurs de recensement)	521,00
Modalités J (variable mode de transport)	6,00
Somme des données brutes (n_{ij})	857 545,00
Khi-deux (χ^2)	207 129,27
Degrés de liberté, soit $(c - 1) \times (l - 1)$	2 600,00
Valeur de p	0,00
Coefficient Phi au carré ($\phi^2 = \chi^2 / n_{ij}$)	0,24

bruts (soit le coefficient phi au carré, ϕ^2) (section 5.2). Le tableau 12.15 permet de vérifier que la somme des valeurs propres est bien égale au coefficient phi au carré :

$$0,156 + 0,046 + 0,031 + 0,004 + 0,004 = 0,24$$

$$\phi^2 = \chi^2/n_{ij} = 203\,971/849\,795 = 0,24$$

Tab. 12.15 : Résultats de l'AFC pour les valeurs propres

Axe factoriel	Valeur propre	Pourcentage	Pourc. cumulé
1	0,156	64,590	64,590
2	0,046	19,250	83,840
3	0,031	12,995	96,835
4	0,004	1,683	98,518
5	0,004	1,482	100,000

Combien d'axes d'une AFC faut-il retenir ?

- **Approche statistique.** Mike Bendixen (1995), cité dans l'excellent site STHDA², propose deux critères pour sélectionner les premiers axes d'une AFC : $c_1 = 1/(l-1) \times 100$ et $c_2 = 1/(c-1) \times 100$ avec l et c étant respectivement les nombres de modalités en ligne et en colonne. Autrement dit, lorsque les données sont distribuées aléatoirement, la valeur propre en pourcentage devrait être égale à c_1 et celle de l'axe factoriel moyen à c_2 . Par conséquent, nous pourrions retenir uniquement les axes dont les valeurs propres en pourcentage excèdent : $c_1 = 1/(521 - 1) \times 100 = 0,19$ et $c_2 = 1/(6 - 1) \times 100 = 20$. En appliquant ces deux critères, seul le premier axe factoriel qui résume 65,6 % mérite d'être retenu.
- **Approche empirique** basée sur la lecture des pourcentages et des pourcentages cumulés. Nous retenons uniquement les deux premières composantes qui résument 85 % de la variance totale. Pour faciliter le choix du nombre d'axes avec cette approche empirique, il est fortement conseillé de construire un histogramme à partir des valeurs propres, soit brutes, soit en pourcentage, soit en pourcentage cumulé (figure 12.25).

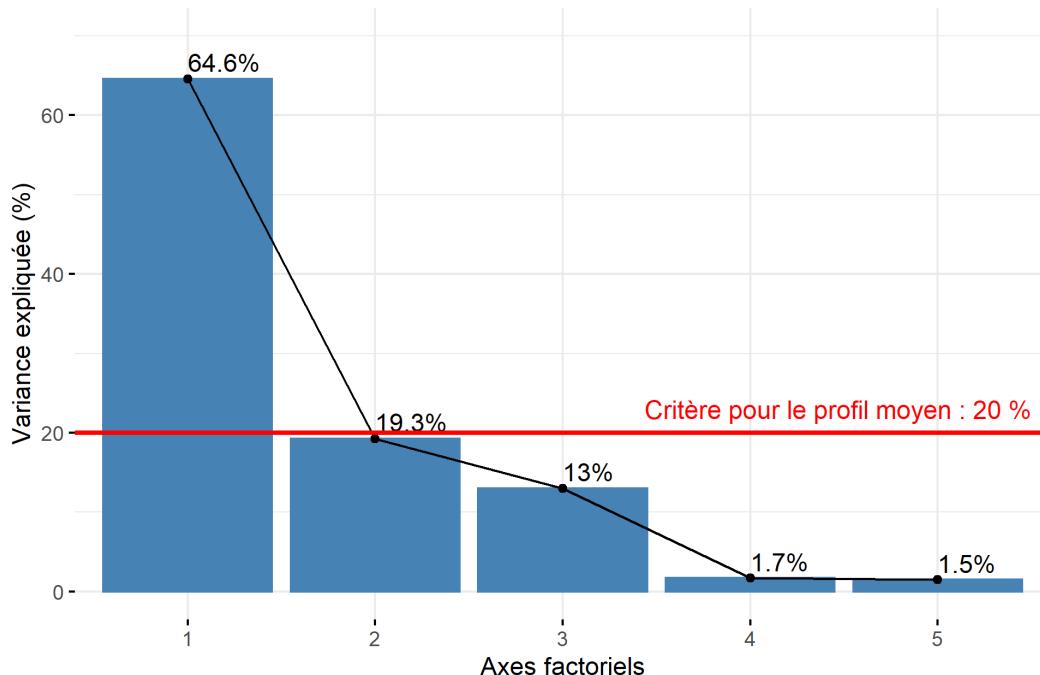


FIG. 12.25 : Histogramme des valeurs propres de l'AFC

12.3.2.2 Résultats de l'AFC pour les variables et les individus

Comme pour l'ACP, nous retrouvons les trois mêmes mesures pour les variables et les individus : 1) les coordonnées factorielles, 2) les contributions et 3) les cosinus carrés.



Compréhension des axes factoriels de l'AFC : une étape essentielle, incontournable...

Comme en ACP, l'analyse des trois mesures (coordonnées, contributions et cosinus carrés) pour les variables

²<http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/74-afc-analyse-factorielle-des-correspondances-avec-r-l-essentiel/#valeurs-propres-variances>

et les individus doit vous permettre de comprendre la signification des axes factoriels retenus de l'AFC. Cette étape d'interprétation est essentielle afin de qualifier les variables latentes (axes factoriels, variables synthétiques) produites par l'AFC.

- **Les coordonnées factorielles** sont simplement les projections des points-lignes et des points-colonnes sur les axes de l'AFC. Tant pour les lignes que pour les colonnes, ces coordonnées bénéficient de deux propriétés intéressantes. Premièrement, pour chaque axe factoriel k , la somme du produit des marges des variables ($f.j$, colonnes) ou des individus ($f.i.$, lignes) avec leurs coordonnées respectives (C_j^k et C_i^k) est égale à 0 (équation (12.7)). Deuxièmement, pour chaque axe factoriel k , la somme des produits entre les marges (en ligne et en colonne) et les coordonnées au carré (en ligne et en colonne) est égale à la valeur propre de l'axe (équation (12.8)).

$$\sum f.j(C_j^k) = 0 \text{ et } \sum f.i.(C_i^k) = 0 \quad (12.7)$$

$$\sum f.i.(C_i^k)^2 = \mu_k \text{ et } \sum f.j(C_j^k)^2 = \mu_k \quad (12.8)$$

En guise d'exemple, le tableau 12.16 permet de vérifier les deux propriétés des coordonnées pour les variables. Les sommes de $f.j(C_j^k)$ pour les axes 1 et 2 sont bien égales à 0; et les sommes de $f.j(C_j^k)^2$ pour les axes 1 et 2 sont bien égales aux valeurs propres de ces deux axes, soit 0,156 et 0,046 (comparez ces valeurs avec celles reportées au tableau 12.15 plus haut).

TAB. 12.16 : Vérification des deux propriétés des coordonnées factorielles pour les variables

Modalité	f.j	Coord.		f.j x Coord.		f.j x Coord2	
		1	2	1	2	1	2
VehCond	0,499	-0,329	0,077	-0,164	0,038	0,054	0,003
VehPass	0,031	-0,255	0,081	-0,008	0,003	0,002	0,000
TranspC	0,345	0,208	-0,229	0,072	-0,079	0,015	0,018
Apied	0,081	0,813	0,545	0,066	0,044	0,053	0,024
Velo	0,036	0,938	-0,187	0,033	-0,007	0,031	0,001
AutreMoyen	0,009	0,142	0,078	0,001	0,001	0,000	0,000
Somme	1,000			0,000	0,000	0,156	0,046



Contrairement à l'ACP, les coordonnées factorielles pour les variables en AFC ne sont pas les coefficients de corrélation de Pearson des variables sur les axes!

- **Les contributions** des colonnes ou des lignes en AFC permettent de repérer celles qui contribuent le plus à la formation des axes factoriels (de manière analogue à l'ACP). Pour un axe donné, leur sommation est égale à 100 %. Elles s'obtiennent en multipliant la coordonnée au carré avec la marge et en divisant le tout par la valeur propre de l'axe (équations (12.9) et (12.10)).

$$\text{Ctr}_j^k = \frac{f.j(C_j^k)^2}{\mu_k} \times 100 \quad (12.9)$$

$$\text{Ctr}_i^k = \frac{f.i.(C_i^k)^2}{\mu_k} \times 100 \quad (12.10)$$

- **Les cosinus carrés** (Cos^2) (appelés aussi les qualités de représentation sur un axe) permettent de repérer le ou les axes qui concourent le plus à donner un sens aux colonnes (variables) et aux lignes

(individus), de manière analogue à l'ACP. Pour une variable ou un individu, la sommation des Cos^2 pour tous les axes de l'AFC est aussi égale à 1.

Interprétation des résultats pour les colonnes (variables)

Maintenant, analysons ces trois statistiques pour les variables pour les deux premiers axes de l'AFC (tableau 12.17 et figure 12.26).

Pour l'axe 1, résumant 65 % de la variance, trois modalités concourent à sa formation : VehCond (34,69 %), Apied (34,25 %) et Velo (20,13 %). À la lecture des coordonnées factorielles sur cet axe, les modes de transport relatifs aux véhicules motorisés (VehCond = -0,33 et VehPass = -0,25) s'opposent clairement aux modes actifs (Apied = 0,81 et Velo = 0,94), constat qu'il est possible de confirmer visuellement avec la figure 12.26. La modalité VehCond a d'ailleurs la plus forte valeur de Cos^2 sur cet axe (0,92), ce qui signale, sans l'ombre d'un doute, que l'axe 1 est celui qui donne le plus de sens à cette modalité.

Puisque l'axe 2 résume une partie beaucoup plus limitée de la variance du tableau (19,25 %), il n'est pas étonnant qu'un nombre plus limité de modalités concourent à sa formation : seules les contributions de la modalité Apied (51,68 %) et secondairement de VehPass (38,81 %) sont importantes. Leurs coordonnées factorielles s'opposent d'ailleurs sur cet axe (respectivement 0,81 et 0,21).

TAB. 12.17 : Résultats de l'AFC pour les variables

Modalité	Coordonnées		Cosinus carrés		Contributions (%)	
	1	2	1	2	1	2
VehCond	-0,33	0,08	0,92	0,05	34,69	6,33
VehPass	-0,25	0,08	0,34	0,03	1,28	0,44
TranspC	0,21	-0,23	0,39	0,47	9,53	38,81
Apied	0,81	0,54	0,67	0,30	34,25	51,61
Velo	0,94	-0,19	0,56	0,02	20,13	2,69
AutreMoyen	0,14	0,08	0,05	0,01	0,12	0,12

Interprétation des résultats pour les individus



Premier plan factoriel pour les variables et les individus

Lorsque le jeu de données comprend à la fois peu de modalités en ligne et en colonne, il est judicieux de les représenter simultanément sur le premier plan factoriel (axes 1 et 2). Pour ce faire, vous pouvez utiliser la fonction `fviz_ca_biplot` du package `factoextra`.

Étant donné que notre jeu de données comprend 521 secteurs de recensement, nous proposons ici de cartographier les coordonnées factorielles des individus pour les deux premiers axes de l'AFC (figure 12.27). Pour l'axe 1, les secteurs de recensement à l'est et l'ouest de l'île de Montréal présentent les coordonnées les plus fortement négatives (en rouge) ; dans ces zones, l'usage des véhicules motorisés pour des déplacements domicile-travail est certainement surreprésenté, comparativement aux modes actifs. À l'inverse, dans les secteurs de recensement du centre de l'île présentant de fortes valeurs positives (en rouge), le recours aux modes de transports actifs (marche et vélo) est bien plus important, toutes proportions gardées. Quant à la cartographie des coordonnées pour l'axe 2, elle permet surtout de repérer quelques secteurs de recensement autour du centre-ville (très fortes valeurs positives en vert foncé) où les déplacements domicile-travail à pied sont plus fréquents, toutes proportions gardées.

En résumé, suite à l'analyse des coordonnées factorielles des variables et des individus, nous pouvons conclure que le premier axe est certainement le plus intéressant puisqu'il permet d'opposer l'usage des

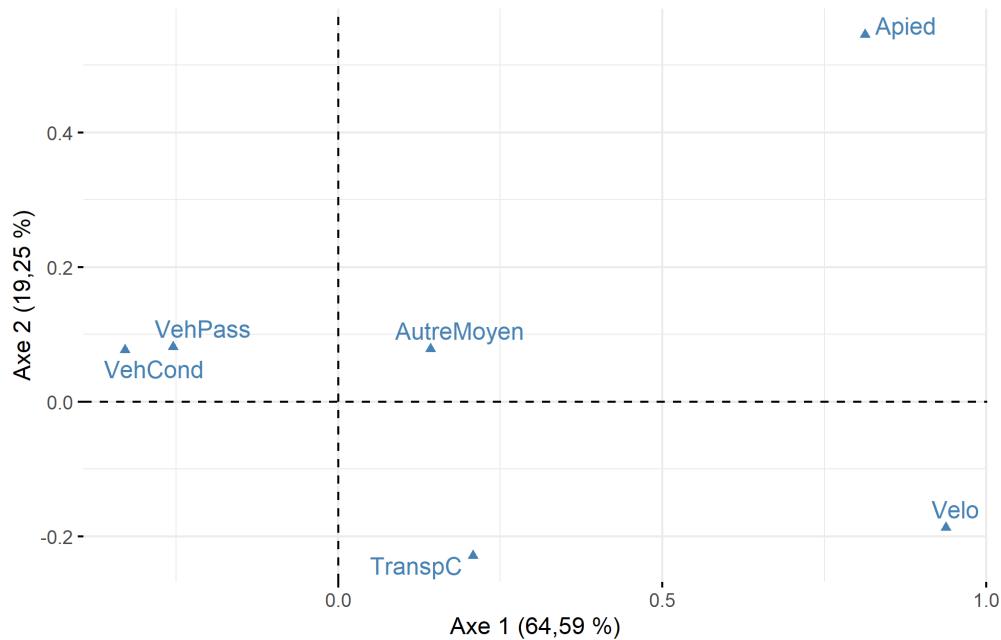


FIG. 12.26 : Premier plan factoriel de l'AFC pour les variables

modes de transports motorisés versus les modes de transports actifs pour les déplacements domicile-travail sur l'île de Montréal. Cette nouvelle variable synthétique (variable latente) pourrait ainsi être introduite dans des analyses subséquentes (par exemple, dans un modèle de régression). Cela démontre qu'au même titre que l'ACP, l'AFC est une méthode de réduction de données puisque nous sommes passés d'un tableau comprenant 512 secteurs de recensement et six modalités à un tableau comprenant une seule variable synthétique (axe 1).



Ajout de modalités supplémentaires dans une analyse des correspondances (AFC)

Comme pour l'ACP, il est possible d'ajouter des variables et des individus supplémentaires une fois l'AFC calculée. En guise d'illustration, nous avons ajouté à l'AFC précédemment analysée des modalités relatives à la durée des temps de déplacements : moins de 15 minutes, 15 à 29, 30 à 44, 45 à 59, 60 minutes et plus. Sans surprise, sur le premier plan factoriel à la figure 12.27, cette dernière modalité, représentant les trajets les plus longs, est la plus proche des modalités relatives à l'usage des véhicules motorisés (VehCond et VehPass).

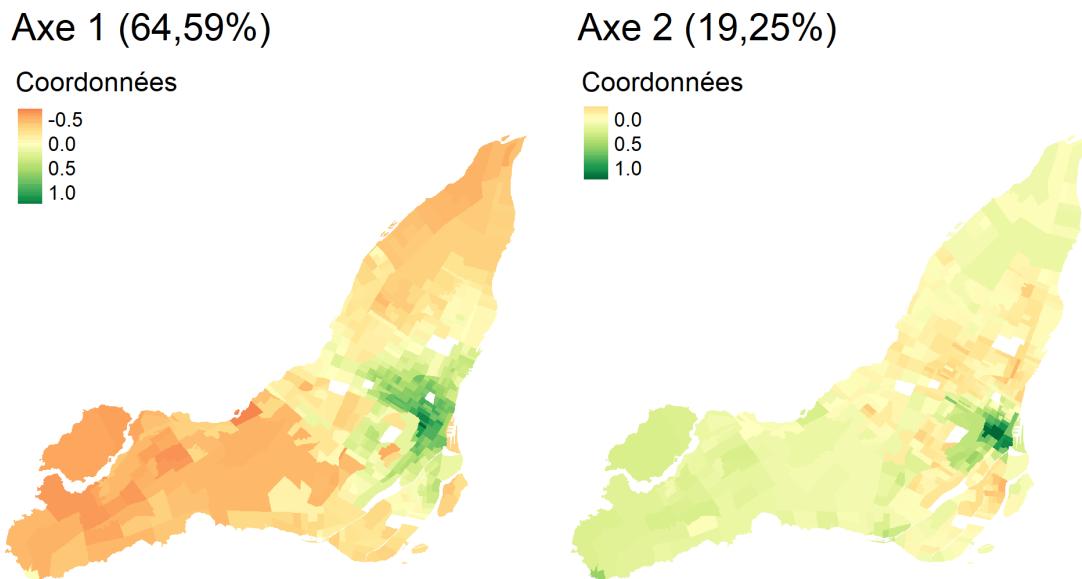


FIG. 12.27 : Cartographie de coordonnées factorielles des individus pour l'AFC

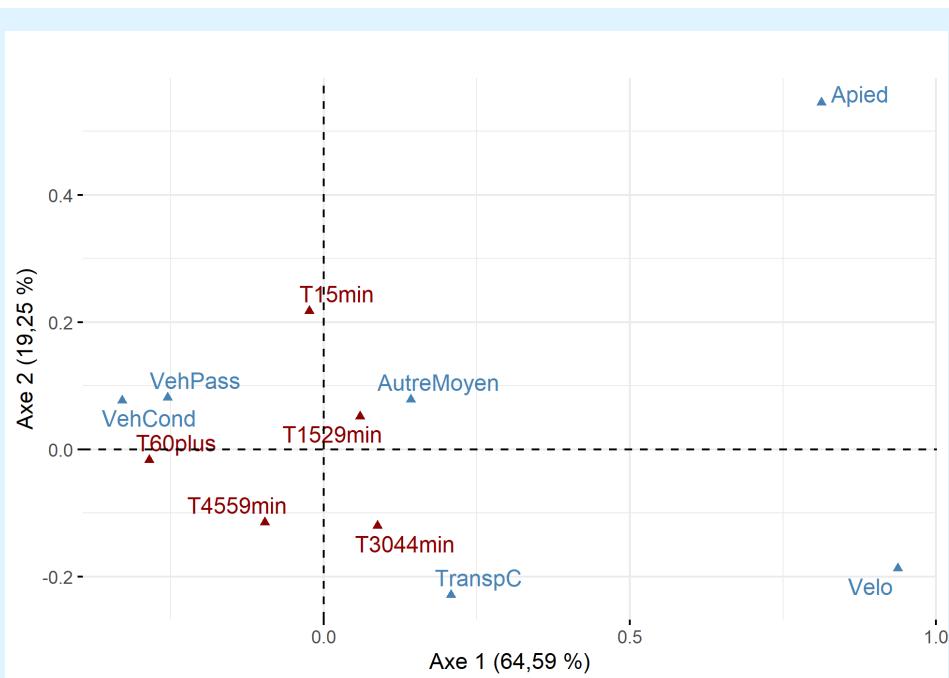


FIG. 12.28 : Ajout de modalités supplémentaires sur le premier plan factoriel de l'AFC

12.3.3 Mise en œuvre dans R

12.3.3.1 Calcul d'une AFC avec FactoMineR

Plusieurs *packages* permettent de calculer une AFC dans R, notamment `ca` (fonction `ca`), `MASS` (fonction `corresp`), `ade4` (fonction `dudi.coa`) et `FactoMineR` (fonction `CA`). De nouveau, nous utilisons `FactoMineR` couplé au *package* `factoextra` pour réaliser rapidement des graphiques avec les résultats pour les variables et les coordonnées.

Pour calculer l'AFC, il suffit d'utiliser la fonction `CA` de `FactoMineR`, puis la fonction `summary(res.afc)`, qui renvoie les résultats de l'AFC pour :

- Les valeurs propres (section `Eigenvalues`) pour les axes factoriels (`Dim.1` à `Dim.n`) avec leur variance expliquée brute (`Variance`), en pourcentage (`% of var.`) et en pourcentage cumulé (`Cumulative % of var.`).
- Les dix premières observations (section `Rows`) avec les coordonnées factorielles (`Dim.1` à `Dim.n`), les contributions (`ctr`) et les cosinus carrés (`cos2`). Pour accéder aux résultats pour toutes les observations, utilisez les fonctions `res.afc$row` ou encore `res.afc$row$coord` (uniquement les coordonnées factorielles), `res.afcrowcontrib` (uniquement les contributions) et `res.afcrowcos2` (uniquement les cosinus carrés).
- Les colonnes (section `Columns`) avec les coordonnées factorielles (`Dim.1` à `Dim.n`), les contributions (`ctr`) et les cosinus carrés (`cos2`).

```
# Chargement des packages
library(FactoMineR)
library(factoextra)

# Chargement des données
load("data/analysesfactorielles/DonneesAFC.Rdata")
# Avant de calculer l'AFC, il convient de vérifier si les deux variables
# qualitatives sont dépendantes avec le test du khi-deux
khideux <- chisq.test(dfDonneesAFC[,1:6])
print(khideux)

## 
## Pearson's Chi-squared test
##
## data: dfDonneesAFC[, 1:6]
## X-squared = 207129, df = 2600, p-value < 2.2e-16

if(khideux$p.value <=0.05){
  cat("La valeur de p < 0,05. Les variables sont dépendantes. Calculer l'AFC.")
}else {
  cat("La valeur de p > 0,05. Les variables sont indépendantes. Inutile de calculer l'AFC")
}

## La valeur de p < 0,05. Les variables sont dépendantes. Calculer l'AFC.

# Calcul de l'analyse des correspondances sur les six premières variables
res.afc <- CA(dfDonneesAFC[,1:6], graph=F)
# Affichage des résultats de la fonction CA
print(res.afc)

## **Results of the Correspondence Analysis (CA)**
```

```

## The row variable has 521 categories; the column variable has 6 categories
## The chi square of independence between the two variables is equal to 207129.3 (p-value = 0).
## *The results are available in the following objects:
##
##      name          description
## 1  "$eig"        "eigenvalues"
## 2  "$col"         "results for the columns"
## 3  "$col$coord"  "coord. for the columns"
## 4  "$col$cos2"   "cos2 for the columns"
## 5  "$col$contrib" "contributions of the columns"
## 6  "$row"         "results for the rows"
## 7  "$row$coord"  "coord. for the rows"
## 8  "$row$cos2"   "cos2 for the rows"
## 9  "$row$contrib" "contributions of the rows"
## 10 "$call"        "summary called parameters"
## 11 "$call$marge.col" "weights of the columns"
## 12 "$call$marge.row" "weights of the rows"

```

```

# Visualisation des marges en colonne
round(res.afc$call$marge.col,4)

```

```

##      VehCond    VehPass    TranspC     Apied     Velo AutreMoyen
##      0.4986    0.0309    0.3449    0.0808    0.0357    0.0090

```

```

# Visualisation des marges en ligne. Étant donnée que nous avons 521 individus,
# la ligne ci-dessous est en commentaire
# round(res.afc$call$marge.row,4)

```

```

# Sommaire des résultats de l'AFC
# Remarquez que la première ligne de ce sommaire est le résultat du khi-deux
summary(res.afc)

```

```

##
## Call:
## CA(X = dfDonneesAFC[, 1:6], graph = F)
##
## The chi square of independence between the two variables is equal to 207129.3 (p-value = 0).
##
## Eigenvalues
##
##           Dim.1    Dim.2    Dim.3    Dim.4    Dim.5
## Variance     0.156    0.046    0.031    0.004    0.004
## % of var.  64.590  19.250  12.995   1.683   1.482
## Cumulative % of var. 64.590  83.840  96.835  98.518 100.000
##
## Rows (the 10 first)
##
##           Iner*1000    Dim.1     ctr    cos2    Dim.2     ctr    cos2    Dim.3
## 1          | 0.155 | -0.304  0.095  0.961 | -0.023  0.002  0.006 |  0.048
## 2          | 0.123 | -0.232  0.067  0.850 |  0.028  0.003  0.012 | -0.021
## 3          | 0.268 | -0.246  0.127  0.737 | -0.002  0.000  0.000 | -0.046
## 4          | 0.102 | -0.168  0.034  0.513 | -0.111  0.049  0.222 | -0.117

```

```

## 5 | 0.118 | -0.251 0.067 0.883 | 0.004 0.000 0.000 | -0.007
## 6 | 0.120 | -0.130 0.024 0.313 | -0.103 0.051 0.196 | -0.144
## 7 | 0.124 | -0.029 0.002 0.022 | -0.158 0.167 0.626 | -0.079
## 8 | 0.073 | -0.157 0.028 0.598 | -0.090 0.031 0.195 | -0.006
## 9 | 0.014 | -0.060 0.005 0.506 | -0.033 0.005 0.150 | -0.018
## 10 | 0.040 | 0.004 0.000 0.000 | -0.204 0.073 0.838 | 0.053
##          ctr   cos2
## 1      0.012 0.024 |
## 2      0.003 0.007 |
## 3      0.022 0.026 |
## 4      0.080 0.246 |
## 5      0.000 0.001 |
## 6      0.146 0.380 |
## 7      0.061 0.155 |
## 8      0.000 0.001 |
## 9      0.002 0.048 |
## 10     0.007 0.056 |
##
## Columns
##           Iner*1000    Dim.1     ctr   cos2    Dim.2     ctr   cos2    Dim.3
## VehCond | 58.559 | -0.329 34.687 0.924 | 0.077 6.331 0.050 | 0.051
## VehPass | 5.923 | -0.255 1.283 0.338 | 0.081 0.440 0.035 | 0.001
## TranspC | 38.261 | 0.208 9.534 0.389 | -0.229 38.812 0.472 | -0.124
## Apied   | 79.193 | 0.813 34.252 0.675 | 0.545 51.610 0.303 | -0.147
## Velo    | 55.633 | 0.938 20.126 0.564 | -0.187 2.688 0.022 | 0.802
## AutreMoyen | 3.969 | 0.142 0.117 0.046 | 0.078 0.119 0.014 | 0.070
##          ctr   cos2
## VehCond 4.141 0.022 |
## VehPass 0.000 0.000 |
## TranspC 16.995 0.139 |
## Apied   5.543 0.022 |
## Velo    73.180 0.413 |
## AutreMoyen 0.140 0.011 |

```

12.3.3.2 Exploration graphique des résultats de l'AFC avec factoextra

Comme pour l'ACP, `factoextra` dispose de plusieurs fonctions très intéressantes pour construire rapidement des graphiques avec les résultats de l'AFC. Premièrement, la syntaxe ci-dessous (avec la fonction `fviz_screenplot`) renvoie deux graphiques pour analyser les résultats des valeurs propres de l'AFC (figure 12.29).

```

library(factoextra)
library(ggplot2)
library(ggpubr)

# Nombre de modalités en ligne et en colonne
ModalitesLig <- nrow(dfDonneesAFC)
ModalitesCol <- ncol(dfDonneesAFC[,1:6])
# Critère statistique du profil moyen
critere2 <- round(1/(ModalitesCol-1)*100,2)

```

```

texte <- paste0("Critère pour le profil moyen : ", as.character(critere2), " %")
# Graphique avec les valeurs propres
G1 <- fviz_screenplot(res.afc, choice="eigenvalue",
                      ylab="Valeurs propres",
                      xlab="Axes factoriels",
                      main="Valeurs propres")
G2 <- fviz_screenplot(res.afc, choice="variance", addlabels = TRUE, ylim = c(0, 70),
                      ylab="Variance expliquée (%)",
                      xlab="Axes factoriels",
                      main="Valeurs propres (%)")+
  geom_hline(yintercept = c2, linetype = 1, color = "red", size=1)+
  annotate(geom="text", x = ModalitesCol-.5,
           y= critere2+3, label=texte,
           color="red", hjust = 1, size = 4)
ggarrange(G1, G2)

```

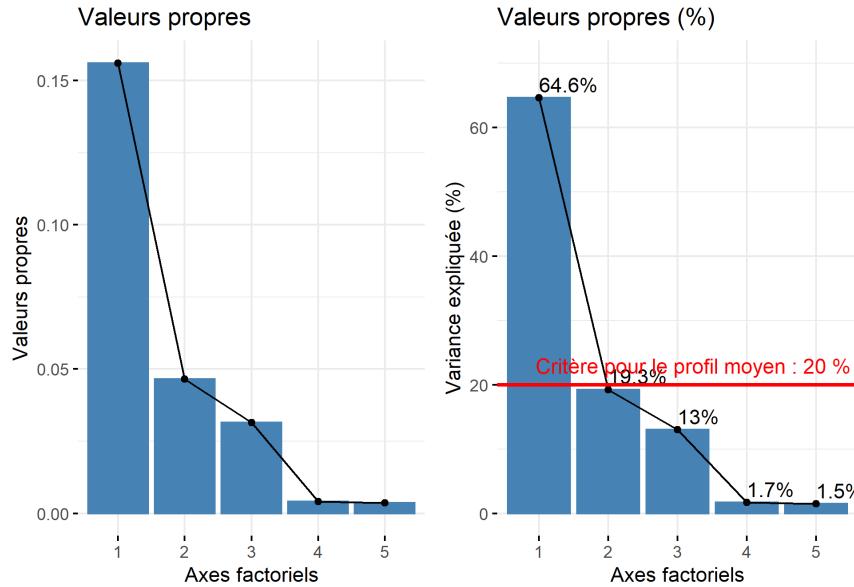


FIG. 12.29 : Graphiques pour les valeurs propres de l’AFC avec factoextra

Avec les fonctions `fviz_contrib` et `fviz_cos2`, il est très facile de réaliser des histogrammes pour les contributions et les cosinus carrés pour les variables (colonnes) ou les individus (lignes), et ce, avec le paramètre `choice = c("row", "col")` (figure 12.30).

```

library(factoextra)
library(ggplot2)
library(ggpubr)
VP1pct <- round(res.afc$eig[1,2],2)
VP2pct <- round(res.afc$eig[2,2],2)
G1 <- fviz_contrib (res.afc, choice = "col", axes = 1, title="Axe 1")
G2 <- fviz_contrib (res.afc, choice = "col", axes = 2, title="Axe 2")
ggarrange(G1, G2, ncol = 2, nrow = 1)

```

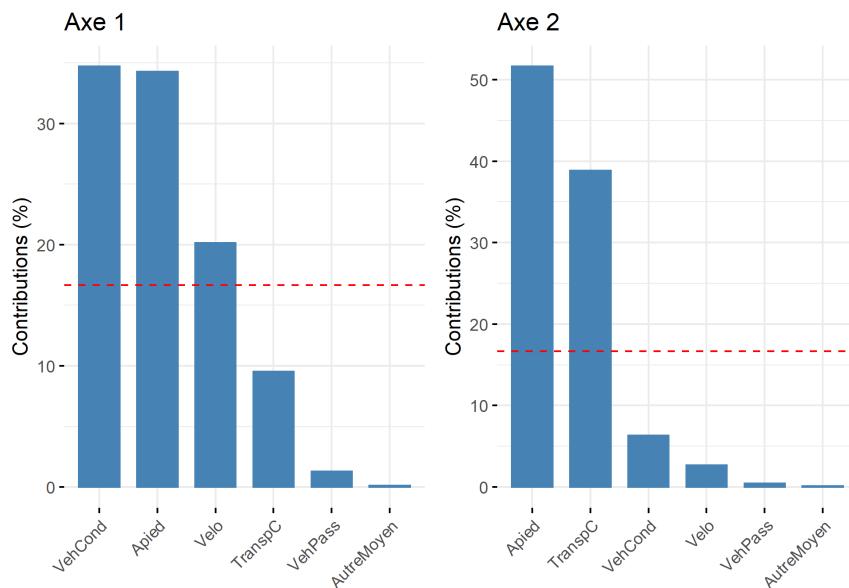


FIG. 12.30 : Contributions des variables avec factoextra

Quant aux fonctions `fviz_ca_col` et `fviz_ca_row`, elles permettent rapidement de construire le premier plan factoriel pour les colonnes (variables) et les lignes (individus) (figure 12.31). Aussi, la fonction `fviz_ca_biplot` permet de construire un plan factoriel, mais avec les lignes et les colonnes simultanément.

```
G3 <- fviz_ca_col(res.afc,
  repel = TRUE,
  geom= c("text","point"),
  col.col = "steelblue",
  title = "Mode de transport",
  xlab=paste0("Axe 1 (", VP1pct, " %)"),
  ylab=paste0("Axe 2 (", VP2pct, " %)"))

G4 <- fviz_ca_row(res.afc,
  repel = TRUE,
  geom= c("point"),
  col.row = "steelblue",
  title = "Secteurs de recensement",
  xlab=paste0("Axe 1 (", VP1pct, " %)"),
  ylab=paste0("Axe 2 (", VP2pct, " %)"))

ggarrange(G3, G4, ncol = 2, nrow = 1)
```

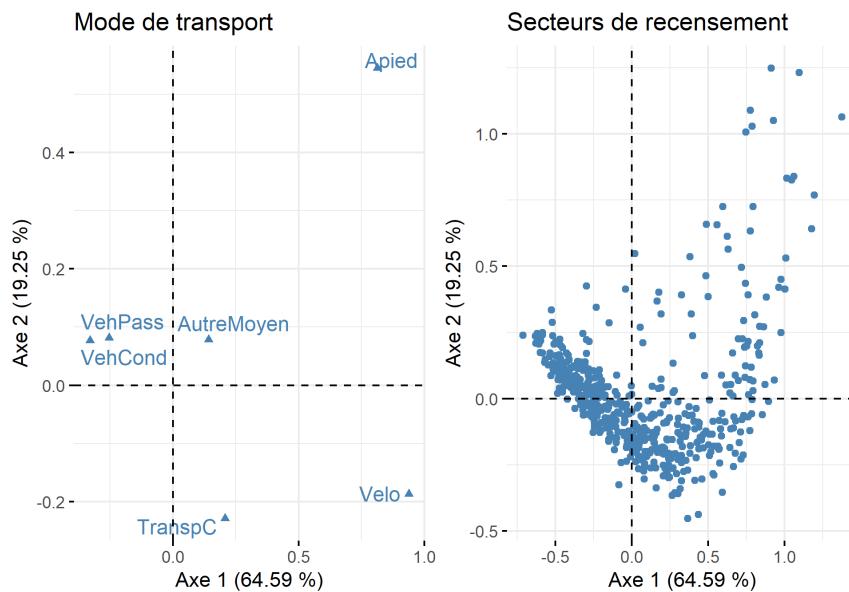


FIG. 12.31 : Premier plan factoriel de l'AFC pour les variables et les individus avec factoextra

La syntaxe ci-dessous permet d'ajouter des modalités supplémentaires dans l'AFC et de constituer le graphique du premier plan factoriel (figure 12.32).

```
# Les colonnes 7 à 11 sont mises comme des variables supplémentaires dans l'AFC
res.afc2 <- CA(dfDonneesAFC, col.sup = 7:11, graph = FALSE)
VP1pct <- round(res.afc2$eig[1,2],2)
VP2pct <- round(res.afc2$eig[2,2],2)
fviz_ca_col(res.afc2,
             repel = TRUE,
             geom= c("text","point"),
             col.col = "steelblue",
             title = "",
             xlab=paste0("Axe 1 (", VP1pct, " %)"),
             ylab=paste0("Axe 2 (", VP2pct, " %)"))
```

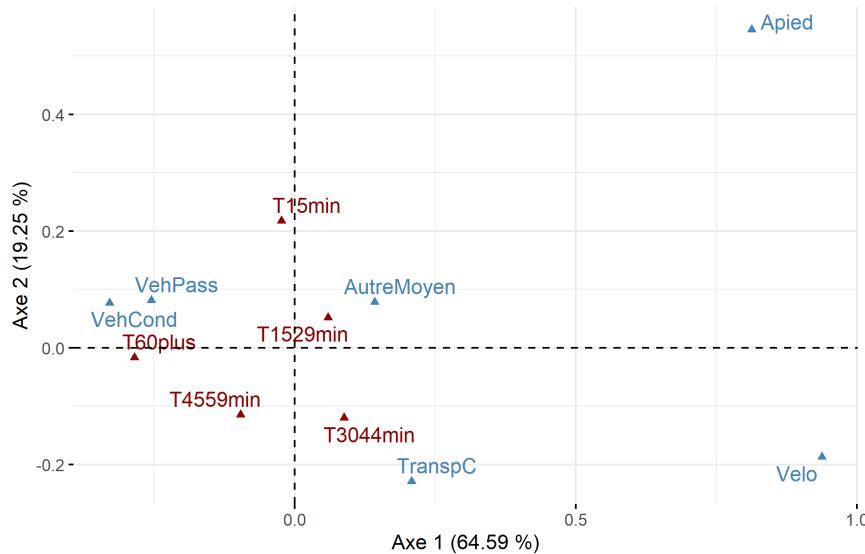


FIG. 12.32 : Ajout de modalités supplémentaires sur le premier plan factoriel l'AFC avec factoextra

Finalement, la syntaxe ci-dessous permet de cartographier les coordonnées factorielles des individus de l'AFC avec le package `tmap` (figure 12.33).

```
library(tmap)
library(stringr)
dfAFCInd <- data.frame(Coord = res.afc$row$coord,
                         Cos2 = res.afc$row$cos2,
                         Ctr = res.afc$row$contrib)
names(dfAFCInd) <- str_replace(names(dfAFCInd), ".Dim.", "Comp")
CartoAFC <- cbind(sfDonneesAFC, dfAFCInd)
VP1pct <- tofr(round(res.afc$eig[1,2],2))
VP2pct <- tofr(round(res.afc$eig[2,2],2))
Carte1 <- tm_shape(CartoAFC) +
  tm_fill(col = "CoordComp1", style = "cont", midpoint = 0, title = 'Coordonnées')+
  tm_layout(title = paste0("Axe 1 (", VP1pct,"%)"), attr.outside = TRUE, frame = FALSE)
Carte2 <- tm_shape(CartoAFC) +
  tm_fill(col = "CoordComp2", style = "cont", midpoint = 0, title = 'Coordonnées')+
  tm_layout(title = paste0("Axe 2 (", VP2pct,"%)"), attr.outside = TRUE, frame = FALSE)
tmap_arrange(Carte1, Carte2, nrow = 1)
```

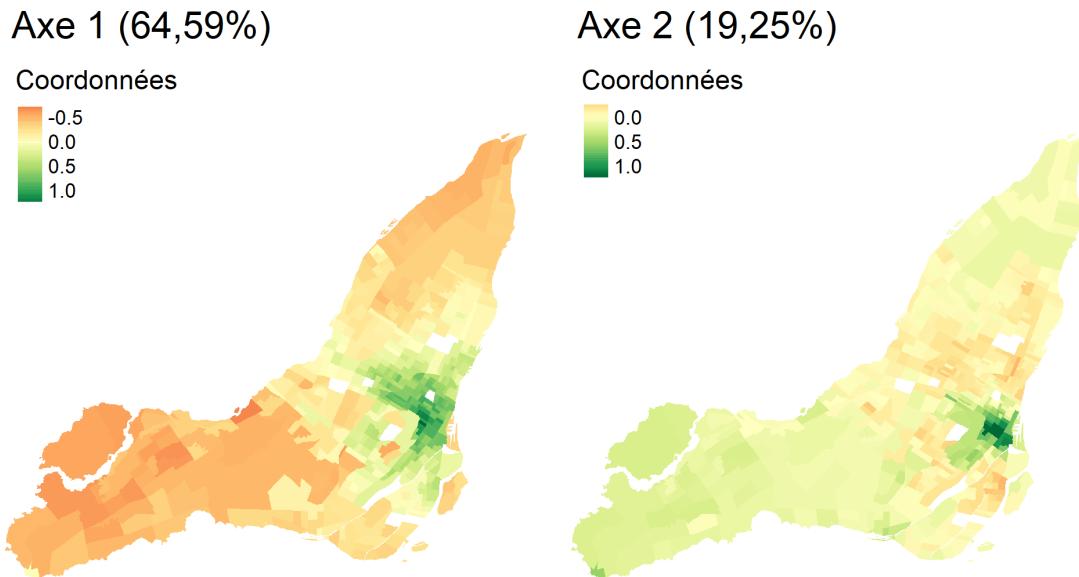


FIG. 12.33 : Cartographie de coordonnées factorielles des individus pour l’AFC

12.4 Analyse de correspondances multiples (ACM)

L’analyse des correspondances multiples (ACM) est particulièrement adaptée à l’exploration de données issues d’une enquête par sondage, puisqu’elle permet de résumer/synthétiser l’information d’un tableau comprenant uniquement des variables qualitatives (figure 12.34).

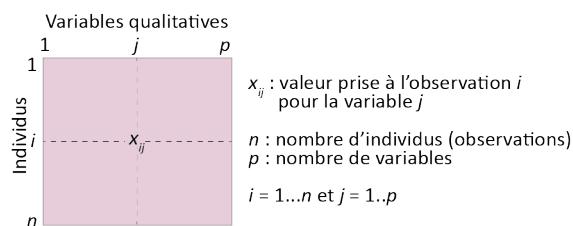


FIG. 12.34 : Tableau pour une ACM

Par exemple, une enquête sur la mobilité d’une population donnée pourrait comprendre plusieurs variables qualitatives dont celles reportées au tableau 12.18.

Pour analyser de telles données, il suffit de transformer le tableau condensé (de données brutes) en un tableau disjonctif complet dans lequel chaque modalité des variables qualitatives devient une variable binaire prenant les valeurs de 0 ou 1 (tableaux 12.19 et 12.20). Notez que la somme de chaque ligne est alors égale au nombre de variables qualitatives.

TAB. 12.18 : Exemple de variables qualitatives issues d'une enquête

Modalités des variables	Codage
Sexe	
Homme	1
Femme	2
Groupe d'âge	
Moins de 20 ans	1
20 à 39 ans	2
40 à 59 ans	3
60 ans et plus	4
Mode de transport	
Automobile	1
Transport en commun	2
Marche	3
Vélo	4

TAB. 12.19 : Tableau condensé (données brutes)

	Sexe	Groupe d'âge	Mode de transport
Ind. 1	1	1	2
Ind. 2	1	2	3
Ind. 3	2	3	1
Ind. 4	1	2	1
Ind. 5	2	4	2
Ind. 6	1	4	4

TAB. 12.20 : Tableau disjonctif complet

Individu	Sexe		Groupe d'âge				Mode de transport			
	Homme	Femme	Moins de 20 ans	20 à 39 ans	40 à 59 ans	60 ans et plus	Auto.	T.C.	Marche	Vélo
Ind. 1	1	0	1	0	0	0	0	1	0	0
Ind. 2	1	0	0	1	0	0	0	0	1	0
Ind. 3	0	1	0	0	1	0	1	0	0	0
Ind. 4	1	0	0	1	0	0	1	0	0	0
Ind. 5	0	1	0	0	0	1	0	1	0	0
Ind. 6	1	0	0	0	0	1	0	0	0	1



ACM versus AFC

Nous avons vu que l'AFC permet d'analyser un tableau de contingence avec deux variables qualitatives. En ACM, les colonnes sont les différentes modalités des variables qualitatives et les lignes sont les observations (par exemple, les individus ayant répondu à une enquête par sondage). En résumé, l'analyse des correspondances multiples (ACM) est simplement **une analyse des correspondances (AFC) appliquée sur un tableau disjonctif complet**.

L'ACM permet ainsi de révéler les ressemblances entre les différentes modalités des variables qualitatives et les ressemblances entre les différents individus. Par conséquent, elle produit également des variables synthétiques (axes factoriels) résumant l'information contenue dans le tableau initial. L'évaluation de ces ressemblances et la détermination des axes factoriels sont aussi basées sur la **distance du khi-deux**.

12.4.1 Aides à l'interprétation

Puisque l'ACM est une extension de l'AFC, nous retrouvons les mêmes aides à l'interprétation : les valeurs propres pour les axes, les coordonnées factorielles, les contributions et les cosinus carrés pour les variables et les individus.

Pour présenter l'ACM, nous utilisons des données ouvertes de la Ville de Montréal et, plus particulièrement, celles d'un sondage auprès de la population de l'île de Montréal sur l'agriculture urbaine³. Pour ce faire, nous avons conservé uniquement les personnes pratiquant l'agriculture urbaine ($n = 352$). Les variables qualitatives extraites pour l'ACM sont reportées au tableau 12.21 avec la description des questions, leurs modalités respectives avec les effectifs bruts et en pourcentage. Au final, l'ACM est calculée de la manière suivante :

- Neuf variables qualitatives relatives à la pratique de l'agriculture urbaine sont retenues (q3, q4, q5, q8, q9, q10, q11, q12 et q13).
- Quatre variables relatives au profil socioéconomique des personnes répondantes sont introduites comme variables supplémentaires (q15, q16, q17 et q21).
- Chaque ligne est pondérée avec la variable pond.

L'objectif de cette ACM est double :

1. Montrer les ressemblances entre les différentes modalités relatives à la pratique de l'agriculture urbaine. L'analyse des axes factoriels devrait nous permettre d'identifier différents profils des personnes pratiquant l'agriculture urbaine.
2. Projeter les modalités des variables socioéconomiques afin de vérifier si elles sont ou non associées aux axes factoriels, c'est-à-dire aux différents profils révélés par les axes.



L'analyse du sondage sur l'agriculture urbaine réalisée ici est purement exploratoire : elle vise uniquement à démontrer que l'ACM est un outil particulièrement intéressant pour analyser les données d'un sondage. Par contre, cette analyse n'a aucune prétention scientifique puisque nous ne sommes pas des spécialistes de l'agriculture urbaine. Dans ce champ de recherche très fertile qu'est l'agriculture urbaine (surement pas la meilleure blague du livre...), vous pourrez consulter plusieurs études montréalaises (McClintock 2018; Audate, Cloutier et Lebel 2021; Bhatt et Farah 2016).

12.4.1.1 Résultats de l'ACM pour les valeurs propres

Les résultats pour les valeurs propres sont reportés au tableau 12.22 et à la figure 12.35. En ACM, l'inertie totale du tableau des variables qualitatives est égale au nombre moyen de modalités par variable moins un, soit $\frac{K}{J} - 1$ avec K et J étant respectivement les nombres de modalités et de variables. Aussi, le nombre d'axes produits par l'ACM est égal à $K - J$. Pour notre tableau, l'inertie est donc égale à $25/9 = 1,77$ avec $25 - 9 = 16$ axes. Le nombre d'axes à retenir est souvent plus difficile à déterminer puisque, tel que signalé judicieusement par Jérôme Pagès (2002, 53) : « en pratique, comparée à l'ACP, l'ACM conduit, dans l'ensemble à : des pourcentages d'inertie plus petits; une décroissance de ces pourcentages plus douce ».

L'histogramme des valeurs propres (figure 12.35) révèle plusieurs sauts importants dans les valeurs propres qui pourraient justifier le choix du nombre d'axes factoriels, soit aux axes 1, 2, 3 et 6. Pour l'exercice, nous retenons les trois premiers axes qui résument 30 % de l'inertie du tableau initial.

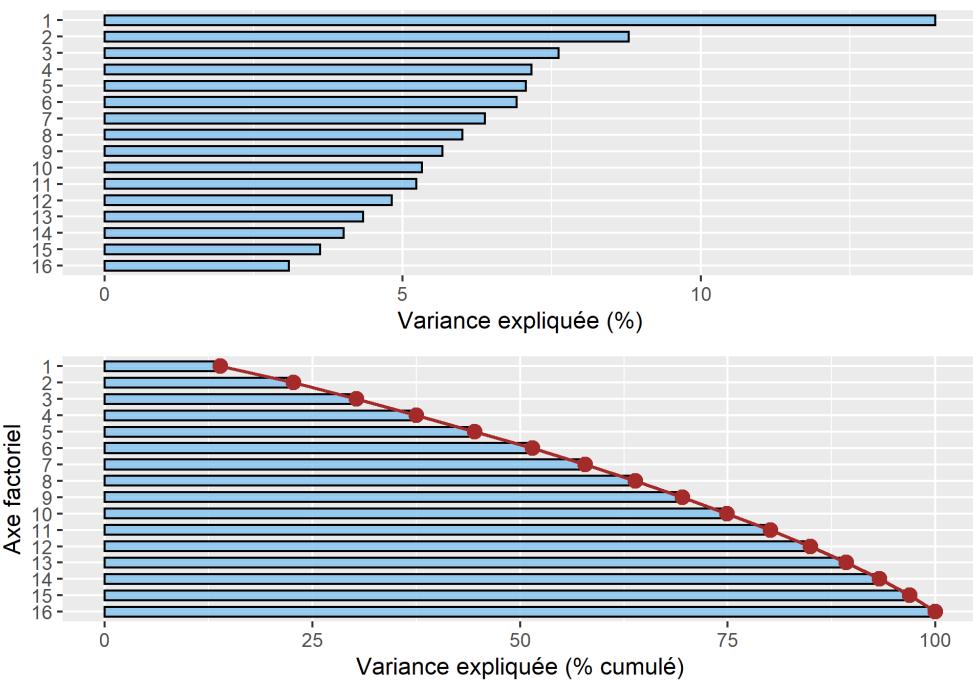
³<https://www.donneesquebec.ca/recherche/dataset/vmtl-agriculture-urbaine-sondage>

TAB. 12.21 : Variables qualitatives extraites du sondage sur l'agriculture urbaine de la Ville de Montréal

Modalité	N	%
Q3. Depuis combien de temps cultivez-vous des fruits, des fines herbes ou des légumes?		
Q3. Moins de 1 an	35	9,9
Q3. De 1 à 4 ans	101	28,7
Q3. De 5 à 9 ans	66	18,8
Q3. 10 ans ou plus	150	42,6
Q4. Selon vous, quelle proportion des fruits, des fines herbes et des légumes que vous consommez durant l'été provient de votre propre production?		
Q4. Moins de 10%	192	54,5
Q4. 10 à 25%	70	19,9
Q4. 26 à 50%	47	13,4
Q4. Plus de 50%	43	12,2
Q5. Utilisez-vous du compost provenant de vos déchets verts ou alimentaires pour faire pousser des fruits, des fines herbes ou des légumes?		
Q5. Oui	90	25,6
Q5. Non	262	74,4
Q8. Récupérez-vous l'eau de pluie pour irriguer vos cultures de fruits, de fines herbes ou des légumes ou encore votre jardin?		
Q8. Oui	72	20,5
Q8. Non	280	79,5
Q9. Combien de sortes de fruits, de fines herbes ou de légumes cultivez-vous?		
Q9. Moins de 5 sortes	170	48,3
Q9. 5 à 9 sortes	124	35,2
Q9. 10 à 14 sortes	42	11,9
Q9. 15 sortes ou plus	16	4,5
Q10. Cultivez-vous suffisamment de fruits, de fines herbes ou de légumes pour partager avec d'autres personnes?		
Q10. Oui	143	40,6
Q10. Non	209	59,4
Q11. Échangez-vous vos semis ou vos récoltes de fruits, de fines herbes ou de légumes avec d'autres personnes?		
Q11. Oui	90	25,6
Q11. Non	262	74,4
Q12. Selon vous, l'agriculture urbaine contribue-t-elle à améliorer les rapports entre les gens?		
Q12. Oui	283	80,4
Q12. Non	46	13,1
Q12. NSP/NRP	23	6,5
Q13. Saviez-vous que la Ville de Montréal encourage et soutient l'agriculture urbaine sur l'île de Montréal?		
Q13. Oui	203	57,7
Q13. Non	149	42,3
Q15. À quel groupe d'âge appartenez-vous?		
Q15. 18 à 34 ans	54	15,3
Q15. 35 à 49 ans	110	31,2
Q15. 50 à 64 ans	101	28,7
Q15. 65 ans et plus	87	24,7
Q16. Quelle est votre occupation principale?		
Q16. Travail temps plein	177	50,3
Q16. Travail temps partiel	26	7,4
Q16. Étudiant	14	4,0
Q16. Retraité	101	28,7
Q16. Sans emploi	10	2,8
Q16. À la maison	24	6,8
Q17. Quel est le plus haut niveau de scolarité que vous avez complété?		
Q17. Aucun certificat ou dipl.	25	7,1
Q17. Dipl. secondaires	80	22,7
Q17. Dipl. collégiales	75	21,3
Q17. Études universitaires	172	48,9
Q21. Êtes-vous propriétaire ou locataire de votre résidence?		
Q21. Propriétaire	250	71,0
Q22. Locataire	102	29,0

TAB. 12.22 : Résultats de l'ACM pour les valeurs propres

Axe factoriel	Valeur propre	Pourcentage	Pourc. cumulé
1	0,248	13,940	13,940
2	0,156	8,792	22,732
3	0,135	7,620	30,352
4	0,127	7,161	37,513
5	0,126	7,065	44,579
6	0,123	6,916	51,494
7	0,114	6,385	57,879
8	0,107	6,003	63,882
9	0,101	5,671	69,553
10	0,095	5,327	74,880
11	0,093	5,234	80,115
12	0,086	4,822	84,937
13	0,077	4,340	89,277
14	0,071	4,011	93,288
15	0,064	3,619	96,906
16	0,055	3,094	100,000

**FIG. 12.35 :** Graphiques pour les valeurs propres pour l'ACM

12.4.1.2 Résultats de l'ACM pour les modalités des variables

À titre de rappel, comme pour l'ACP et l'AFC, nous retrouvons les trois mêmes mesures pour les variables et les individus (coordonnées factorielles, contributions et cosinus carrés). Plus les variables qualitatives du jeu donnée comprennent de modalités, plus la taille du tableau des résultats des modalités est importante et plus il est fastidieux de l'analyser. Il est donc recommandé de construire des histogrammes avec les coordonnées factorielles et les contributions des modalités, mais aussi un nuage de points avec les coordonnées des modalités des variables qualitatives sur le premier, voire le deuxième plan factoriel.



Compréhension des axes factoriels de l'ACM : une étape essentielle, incontournable...

Comme en ACP et en AFC, l'analyse des trois mesures (coordonnées, contributions et cosinus carrés) pour les variables et les individus doit vous permettre de comprendre la signification des axes factoriels retenus de l'ACM. Prenez le temps de bien réaliser cette étape d'interprétation souvent plus fastidieuse qu'en ACP et ACM, en raison du nombre élevé de modalités. Cette étape est en effet essentielle afin de qualifier les variables latentes (axes factoriels, variables synthétiques) produites par l'ACM.

Les résultats pour les variables sont reportés 1) au tableau 12.23, 2) aux figures 12.36, 12.37 et 12.39 pour les coordonnées et les contributions et à la figure 12.38 pour le premier plan factoriel.

Interprétation des résultats de l'axe 1 pour les variables

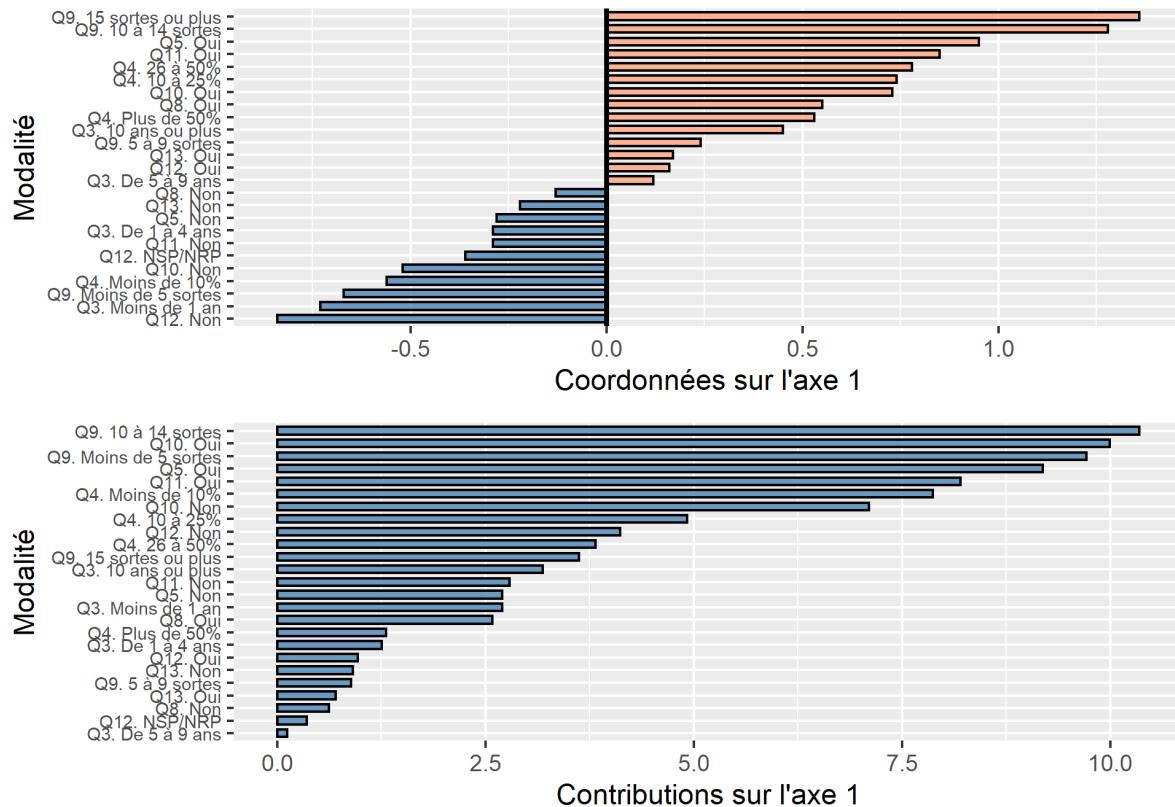
Sept modalités concourent le plus à la formation de l'axe 1 résumant 13,9 % de la variance : Q9. 10 à 14 sortes (10,35 %), Q10. Oui (9,99 %), Q9. Moins de 5 sortes (9,71 %), Q5. Oui (9,19 %), Q11. Oui (8,20 %), Q4. Moins de 10% (7,87 %) et Q10. Non (7,10 %). Aussi, les modalités suivantes sont aux deux extrémités de cet axe :

- **Coordonnées négatives** : Q12. Non (-0,84), Q3. Moins de 1 an (-0,73), Q9. Moins de 5 sortes (-0,67), Q4. Moins de 10% (-0,56), Q10. Non (-0,521). Cela signifie que lorsque les coordonnées des individus sont fortement négatives sur cet axe, les personnes pratiquant l'agriculture urbaine :
 - ne pensent pas que l'agriculture urbaine contribue à améliorer les rapports entre les gens (Q12);
 - cultivent des fruits, des fines herbes ou de légumes depuis moins d'un an (Q3);
 - cultivent moins de cinq sortes de fruits, de fines herbes ou de légumes (Q9);
 - moins de 10 % de la proportion des fruits, des fines herbes et des légumes consommés durant l'été provient de leur propre production (Q4);
 - ne cultivent pas suffisamment pour partager avec d'autres personnes (Q10).
- **Coordonnées positives** : Q9. 15 sortes ou plus (1,36), Q9. 10 à 14 sortes (1,28), Q5. Oui (0,95) et Q11. Oui (0,85). Cela signifie que lorsque les coordonnées des individus sont fortement positives sur cet axe, les personnes pratiquant l'agriculture urbaine :
 - cultivent plus de dix sortes de fruits, de fines herbes ou de légumes (Q9);
 - utilisent du compost provenant de leurs déchets verts ou de leurs déchets alimentaires pour faire pousser des fruits, des fines herbes ou de légumes (Q5);
 - échangent leurs semis ou leurs récoltes de fruits, de fines herbes ou des légumes avec d'autres personnes (Q11).

En résumé, l'axe 1 oppose clairement les **néophytes en agriculture** versus les **personnes expérimentées** cultivant des fruits et de légumes variés avec leur propre compost et échangeant leurs semis ou leurs récoltes.

TAB. 12.23 : Résultats de l'ACM pour les modalités des variables

Modalité	Coordonnées			Cosinus carrés			Contributions (%)		
	1	2	3	1	2	3	1	2	3
Q3. Moins de 1 an	-0,73	0,68	0,63	2,70	3,68	3,64	0,07	0,06	0,05
Q3. De 1 à 4 ans	-0,29	-0,79	0,39	1,26	15,30	4,30	0,04	0,33	0,08
Q3. De 5 à 9 ans	0,12	0,38	-1,11	0,12	2,02	19,43	0,00	0,04	0,29
Q3. 10 ans ou plus	0,45	0,35	0,02	3,19	3,01	0,02	0,11	0,07	0,00
Q4. Moins de 10%	-0,56	0,03	0,07	7,87	0,04	0,22	0,40	0,00	0,01
Q4. 10 à 25%	0,74	-0,76	-0,11	4,92	8,22	0,19	0,14	0,14	0,00
Q4. 26 à 50%	0,78	0,64	0,15	3,82	4,18	0,28	0,10	0,07	0,00
Q4. Plus de 50%	0,53	0,40	-0,37	1,31	1,17	1,19	0,03	0,02	0,02
Q5. Oui	0,95	0,56	-0,13	9,19	5,01	0,29	0,27	0,09	0,00
Q5. Non	-0,28	-0,16	0,04	2,70	1,47	0,09	0,27	0,09	0,00
Q8. Oui	0,55	0,37	1,21	2,58	1,86	23,31	0,07	0,03	0,35
Q8. Non	-0,13	-0,09	-0,29	0,62	0,45	5,60	0,07	0,03	0,35
Q9. Moins de 5 sortes	-0,67	0,01	0,47	9,71	0,00	8,83	0,42	0,00	0,21
Q9. 5 à 9 sortes	0,24	0,11	-0,79	0,89	0,30	17,31	0,03	0,01	0,32
Q9. 10 à 14 sortes	1,28	-0,97	0,07	10,35	9,42	0,06	0,27	0,15	0,00
Q9. 15 sortes ou plus	1,36	2,15	0,65	3,62	14,42	1,50	0,08	0,21	0,02
Q10. Oui	0,73	-0,15	0,07	9,99	0,63	0,18	0,38	0,02	0,00
Q10. Non	-0,52	0,10	-0,05	7,10	0,45	0,13	0,38	0,02	0,00
Q11. Oui	0,85	-0,83	0,32	8,20	12,45	2,11	0,25	0,23	0,03
Q11. Non	-0,29	0,28	-0,11	2,79	4,24	0,72	0,25	0,23	0,03
Q12. Oui	0,16	0,02	0,01	0,97	0,02	0,01	0,11	0,00	0,00
Q12. Non	-0,84	-0,38	0,09	4,12	1,32	0,08	0,11	0,02	0,00
Q12. NSP/NRP	-0,36	0,56	-0,31	0,36	1,39	0,48	0,01	0,02	0,01
Q13. Oui	0,17	0,31	0,31	0,70	3,91	4,37	0,04	0,13	0,12
Q13. Non	-0,22	-0,40	-0,40	0,91	5,05	5,66	0,04	0,13	0,12

**FIG. 12.36 : Graphiques pour les résultats des modalités de l'axe 1 de l'ACM**

Interprétation des résultats de l'axe 2 pour les variables

Quatre modalités concourent le plus à la formation de l'axe 2 résumant 8,8 % de la variance : Q3. De 1 à 4 ans (15,30 %), Q9. 15 sortes ou plus (14,42 %), Q11. Oui (12,45 %) et Q9. 10 à 14 sortes (9,42 %). Les modalités suivantes sont présentes aux deux extrémités de l'axe 2 :

- **Coordonnées négatives** : Q9. 10 à 14 sortes (-0,97), Q11. Oui (-0,83), Q3. De 1 à 4 ans (-0,79), Q4. 10 à 25% (-0,76). Cela signifie que lorsque les coordonnées des individus sont fortement négatives sur cet axe, les personnes pratiquant l'agriculture urbaine :
 - cultivent de 10 à 14 sortes de fruits, de fines herbes ou de légumes (Q9);
 - échangent leurs semis ou leurs récoltes de fruits, de fines herbes ou de légumes avec d'autres personnes (Q11);
 - cultivent des fruits, des fines herbes ou des légumes depuis 1 à 4 ans (Q3);
 - de 10 à 25 % de la proportion des fruits, des fines herbes et des légumes consommés durant l'été provient de leur propre production (Q4).
- **Coordonnées positives** : seule la modalité Q9. 15 sortes ou plus (2,15) présente une forte coordonnée positive.

En résumé, l'axe 2 permet surtout d'identifier des personnes pratiquant l'agriculture urbaine depuis quelques années (de 1 à 4 ans), mais cultivant déjà de nombreuses sortes de fruits et légumes et partageant aussi leurs semis ou récoltes.

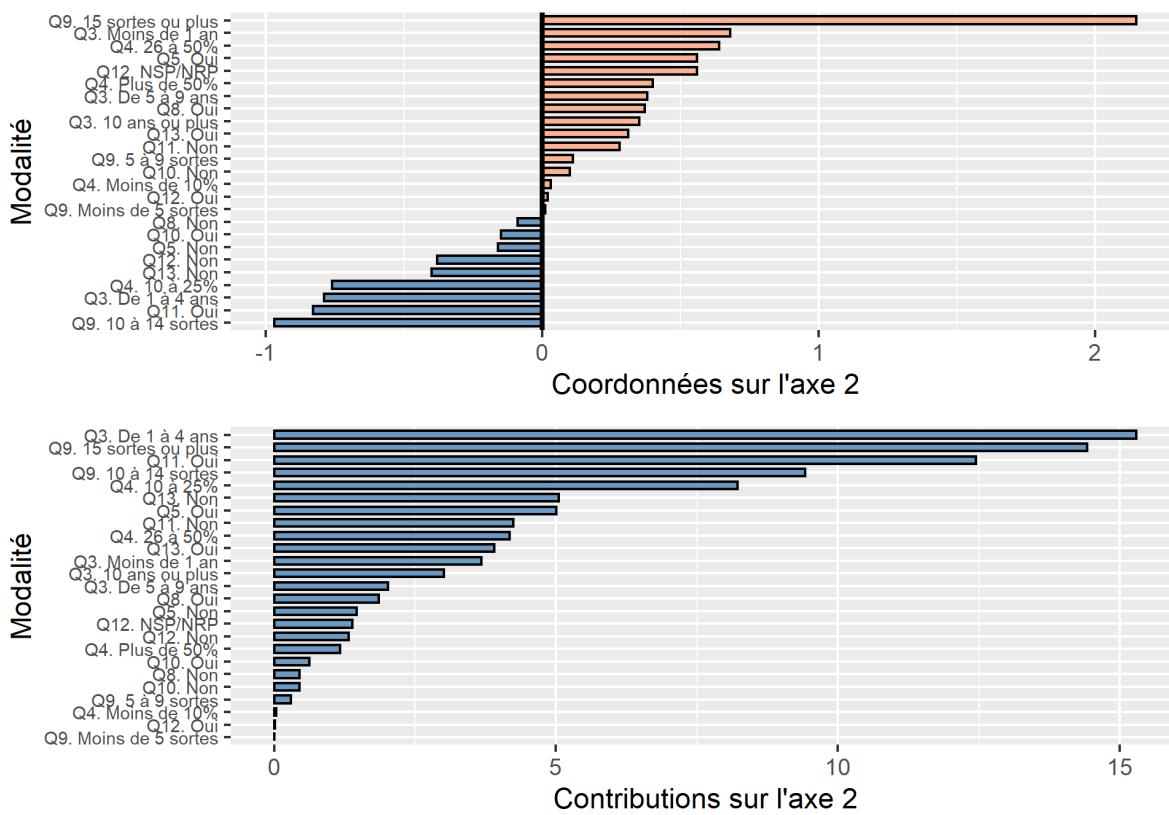


FIG. 12.37 : Graphiques pour les résultats des modalités de l'axe 2 de l'ACM

Interprétation des résultats de l'axe 3 pour les variables

Trois modalités concourent le plus à la formation de l'axe 3 résumant 7,6 % de la variance : Q8. Oui (23,31), Q3. De 5 à 9 ans (19,43) et Q9. 5 à 9 sortes (17,31). Les modalités suivantes sont présentes aux deux extrémités de l'axe 3 :

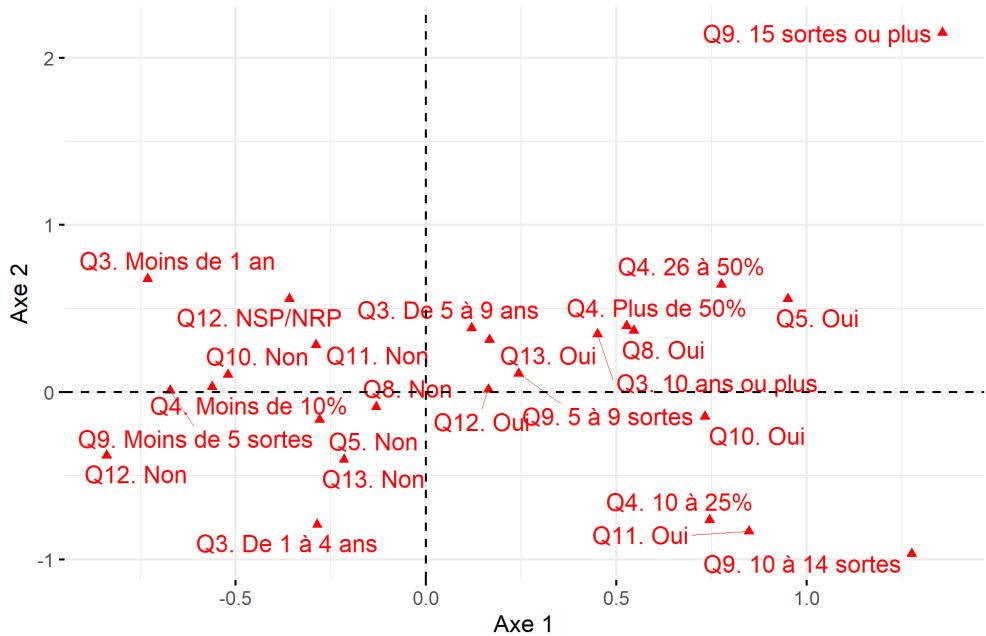


FIG. 12.38 : Premier plan factoriel de l'ACM pour les modalités

- **Coordonnées négatives** : Q3. De 5 à 9 ans (-1,11), Q9. 5 à 9 sortes (-0,79).
- **Coordonnées positives** : seule la modalité Q8. Oui présente une coordonnée fortement positive (1,21).

Par conséquent, cet axe semble plus complexe à analyser et surtout moins intéressant que les deux premiers.

Analyse des variables supplémentaires dans l'ACM

Il est ensuite possible de projeter les modalités supplémentaires sur les axes de l'ACM retenus (tableau 12.24 et figure 12.40). Les faibles valeurs des coordonnées factorielles des modalités supplémentaires sur les deux axes semblent indiquer que le profil socioéconomique des personnes pratiquant l'agriculture urbaine ne semble pas (ou peu) relié aux profils identifiés par les axes factoriels.



Visualisation de variables qualitatives ordinaires sur un plan factoriel

Lorsque les variables qualitatives sont ordinaires et non nominales, il peut être intéressant de relier les différentes modalités avec une ligne. Cela permet de comprendre en un coup d'œil la trajectoire que suivent les modalités sur les deux axes factoriels. En guise d'exemple, nous réalisons cet exercice pour les variables Q3 et Q9 (figure 12.41).

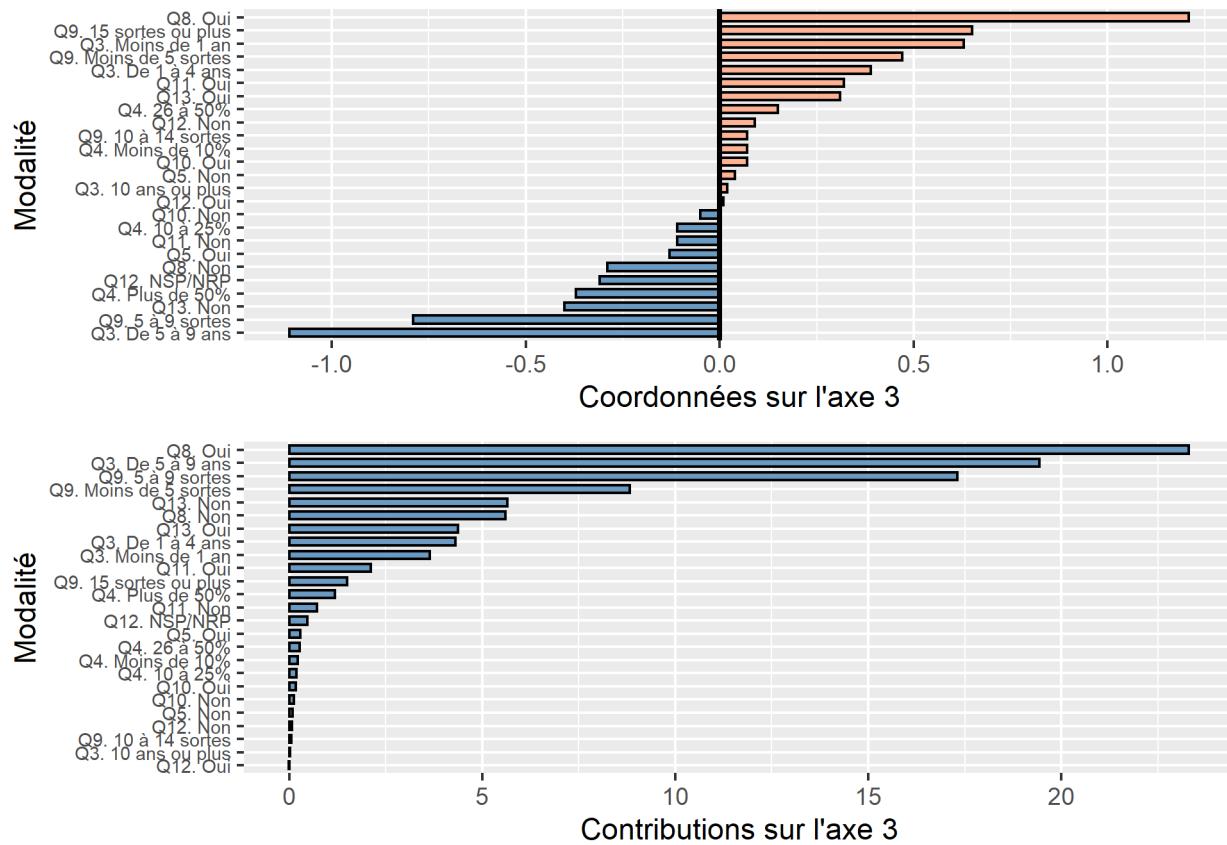


FIG. 12.39 : Graphiques pour les résultats des modalités de l'axe 3 de l'ACM

TAB. 12.24 : Résultats de l'ACM pour les modalités des variables supplémentaires

Modalité	Coordonnées		Cosinus carrés	
	1	2	1	2
Q15. 18 à 34 ans	-0,09	-0,27	0,00	0,04
Q15. 35 à 49 ans	-0,06	-0,01	0,00	0,00
Q15. 50 à 64 ans	0,14	0,25	0,01	0,02
Q15. 65 ans et plus	0,11	0,25	0,00	0,01
Q16. Travail temps plein	-0,10	-0,06	0,01	0,01
Q16. Travail. temps partiel	0,36	-0,15	0,01	0,00
Q16. Étudiant	-0,14	-0,11	0,00	0,00
Q16. Retraité	0,17	0,25	0,01	0,02
Q16. Sans emploi	0,44	-0,14	0,01	0,00
Q16. À la maison	-0,06	0,18	0,00	0,00
Q17. Aucun certificat ou dipl.	0,28	0,16	0,00	0,00
Q17. Dipl. secondaires	-0,09	0,01	0,00	0,00
Q17. Dipl. collégiales	0,01	0,19	0,00	0,01
Q17. Études universitaires	0,00	-0,10	0,00	0,01
Q21. Propriétaire	0,03	0,02	0,00	0,00
Q22. Locataire	-0,06	-0,03	0,00	0,00

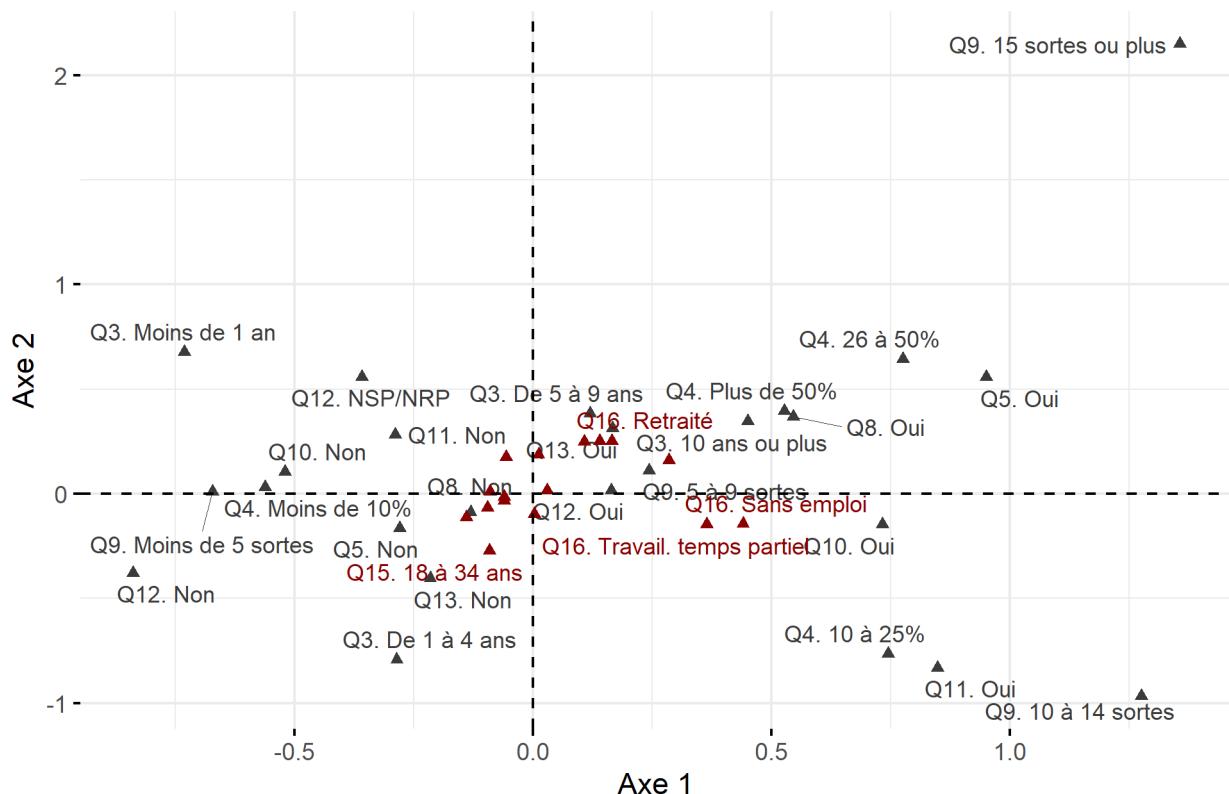


FIG. 12.40 : Premier plan factoriel de l'ACM avec toutes les modalités incluant celles supplémentaires

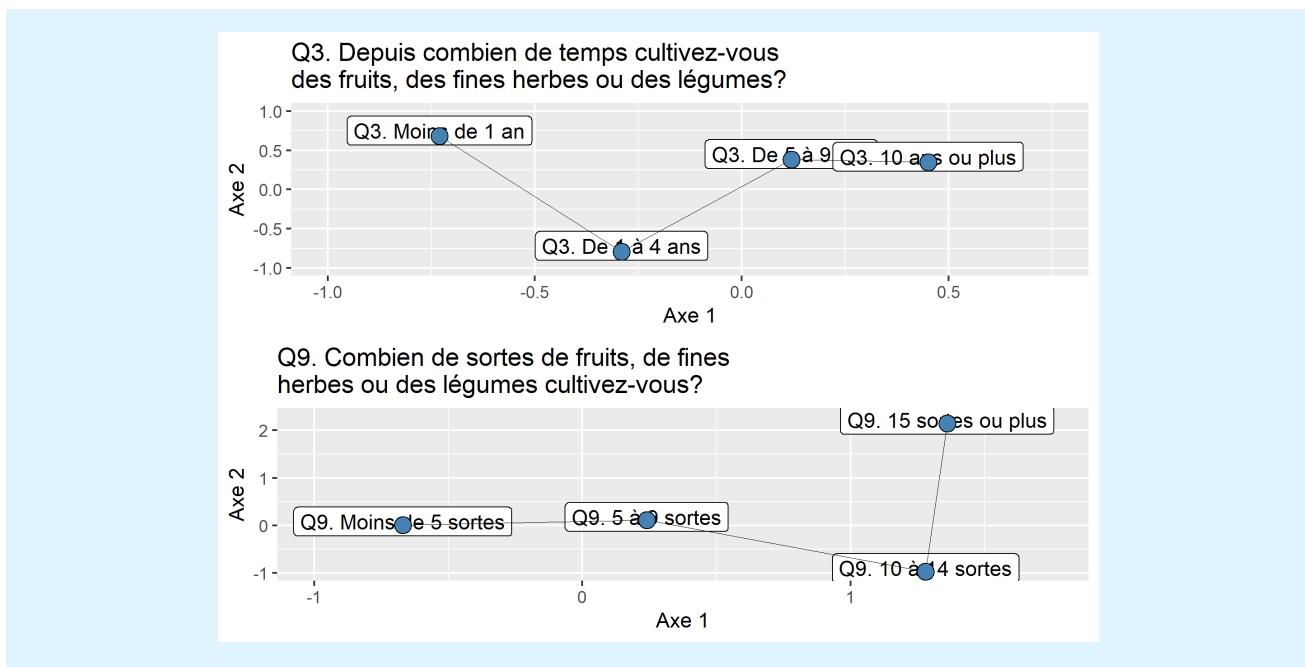


FIG. 12.41 : Trajectoires des variables ordinaires sur le premier plan factoriel de l'ACM

12.4.1.3 Résultats de l'ACM pour les individus

Comme toute méthode factorielle, les coordonnées factorielles, les cosinus carrés et les contributions sont aussi disponibles pour les individus en ACM. Nous proposons ici simplement de réaliser le premier plan factoriel pour les individus en attribuant un dégradé de couleurs avec les cosinus carrés (figure 12.42). Il est aussi possible d'attribuer des couleurs aux différentes modalités d'une variable. Par exemple, sur le premier plan factoriel, nous avons utilisé la variable Q12. Selon vous, l'agriculture urbaine contribue-t-elle à améliorer les rapports entre les gens?. Cela permet de repérer visuellement que les personnes ayant répondu négativement à cette question ont surtout des coordonnées négatives sur l'axe 1.

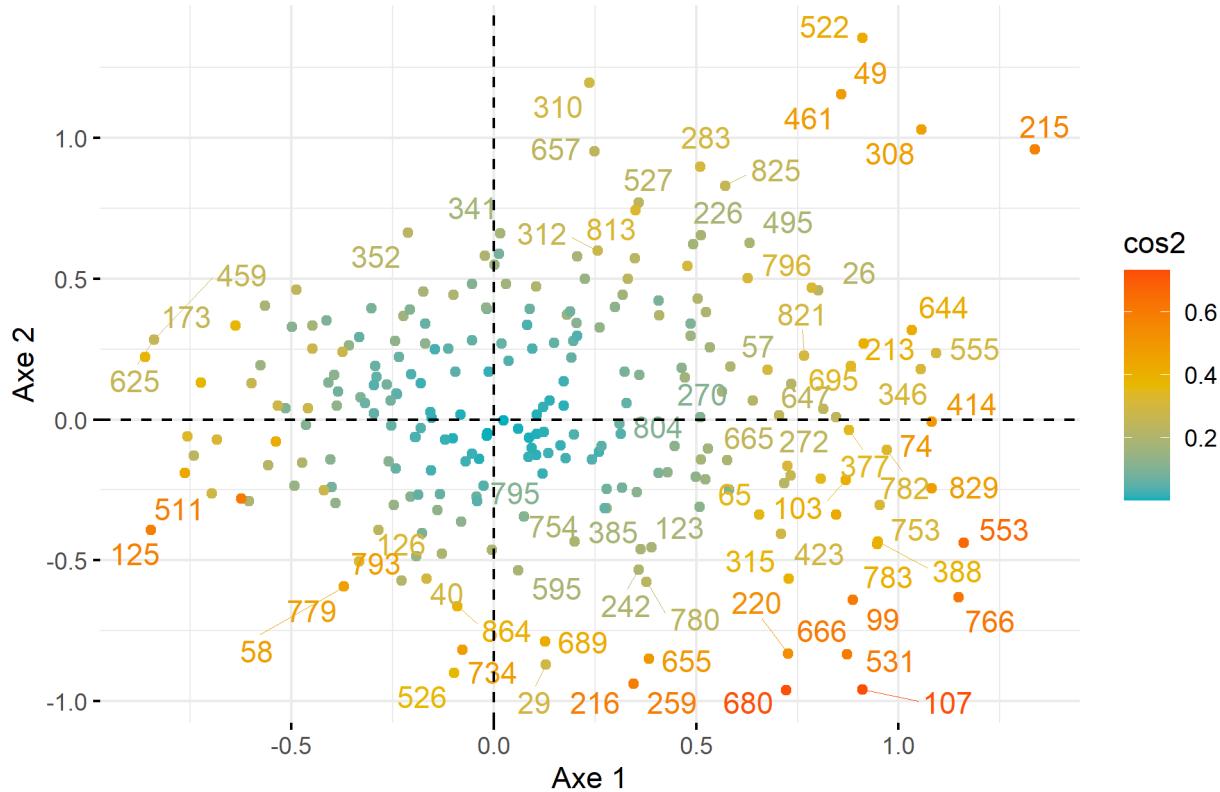


FIG. 12.42 : Premier plan factoriel de l'ACM pour les individus

12.4.2 Mise en œuvre dans R

12.4.2.1 Calcul d'une ACM avec FactoMineR

Plusieurs *packages* permettent de calculer une ACM dans R, notamment `ExPosition` (fonction `epMCA`), `ade4` (fonction `dudi.mca`) et `FactoMineR` (fonction `MCA`). De nouveau, nous utilisons `FactoMineR` couplé au *package* `factoextra` pour réaliser rapidement des graphiques.

Pour calculer l'ACM, il suffit d'utiliser la fonction `MCA` de `FactoMineR`, puis la fonction `summary(res.acm)` qui renvoie les résultats de l'ACM pour :

- Les valeurs propres (section `Eigenvalues`) pour les axes factoriels (`Dim.1` à `Dim.n`) avec leur variance expliquée brute (`Variance`), en pourcentage (`% of var.`) et en pourcentage cumulé (`Cumulative % of var.`).

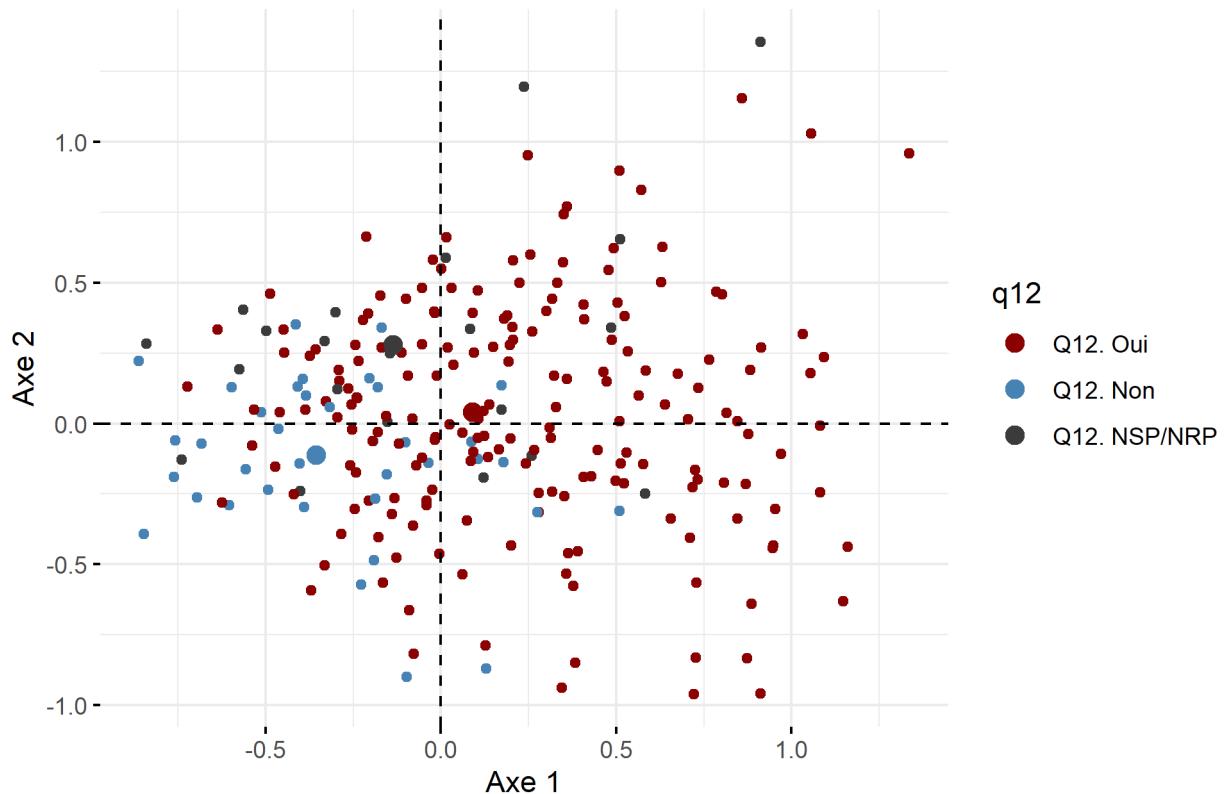


FIG. 12.43 : Premier plan factoriel de l'ACM pour les individus avec coloration d'une variable

- Les dix premières observations (section `Individuals`) avec les coordonnées factorielles (`Dim.1` à `Dim.n`), les contributions (`ctr`) et les cosinus carrés (`cos2`). Pour accéder aux résultats pour toutes les observations, utilisez les fonctions `res.acm$ind` ou encore `res.acm$ind$coord` (uniquement les coordonnées factorielles), `res.acmindcontrib` (uniquement les contributions) et `res.acmindcos2` (uniquement les cosinus carrés).
- Les dix premières modalités des variables (section `Categories`) avec les coordonnées factorielles (`Dim.1` à `Dim.n`), les contributions (`ctr`) et les cosinus carrés (`cos2`).

La syntaxe ci-dessous permet, dans un premier temps, de calculer l'ACM, puis de créer un *DataFrame* pour les résultats des valeurs propres.

```
library(FactoMineR)
# Calcul de l'AFC
res.acm <- MCA(dfACM,           # Nom du DataFrame
                 ncp = 3,          # Nombre d'axes retenus
                 quali.sup=10:13,  # Variables supplémentaires
                 graph = FALSE,
                 row.w = dfenquete$pond) # Variables pour la pondération des lignes
# Affichage des résultats
print(res.acm)
```

```
## **Results of the Multiple Correspondence Analysis (MCA)**
## The analysis was performed on 352 individuals, described by 13 variables
## *The results are available in the following objects:
```

```

## 
##      name           description
## 1  "$eig"          "eigenvalues"
## 2  "$var"           "results for the variables"
## 3  "$var$coord"    "coord. of the categories"
## 4  "$var$cos2"     "cos2 for the categories"
## 5  "$var$contrib"  "contributions of the categories"
## 6  "$var$v.test"   "v-test for the categories"
## 7  "$ind"           "results for the individuals"
## 8  "$ind$coord"    "coord. for the individuals"
## 9  "$ind$cos2"     "cos2 for the individuals"
## 10 "$ind$contrib"  "contributions of the individuals"
## 11 "$quali.sup"    "results for the supplementary categorical variables"
## 12 "$quali.sup$coord" "coord. for the supplementary categories"
## 13 "$quali.sup$cos2" "cos2 for the supplementary categories"
## 14 "$quali.sup$v.test" "v-test for the supplementary categories"
## 15 "$call"          "intermediate results"
## 16 "$call$marge.col" "weights of columns"
## 17 "$call$marge.li"  "weights of rows"

```

```
summary(res.acm)
```

```

## 
## Call:
## MCA(X = dfACM, ncp = 3, quali.sup = 10:13, graph = FALSE, row.w = dfenquete$pond)
## 
## 
## Eigenvalues
##                  Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance          0.248   0.156   0.135   0.127   0.126   0.123   0.114
## % of var.       13.940   8.792   7.620   7.161   7.065   6.916   6.385
## Cumulative % of var. 13.940  22.732  30.352  37.513  44.579  51.494  57.879
## 
##                  Dim.8   Dim.9   Dim.10  Dim.11  Dim.12  Dim.13  Dim.14
## Variance          0.107   0.101   0.095   0.093   0.086   0.077   0.071
## % of var.        6.003   5.671   5.327   5.234   4.822   4.340   4.011
## Cumulative % of var. 63.882  69.553  74.880  80.115  84.937  89.277  93.288
## 
##                  Dim.15  Dim.16
## Variance          0.064   0.055
## % of var.        3.619   3.094
## Cumulative % of var. 96.906 100.000
## 
## 
## Individuals (the 10 first)
##                  Dim.1     ctr    cos2   Dim.2     ctr    cos2
## 4                 | 0.261  0.063  0.052 | 0.327  0.155  0.081 |
## 10                | -0.533 0.688  0.278 | 0.050  0.010  0.002 |
## 11                | 0.135  0.020  0.014 | -0.120  0.025  0.011 |
## 15                | 0.020  0.000  0.000 | 0.271  0.073  0.061 |
## 17                | -0.133 0.014  0.012 | -0.264  0.088  0.049 |
## 18                | 0.196  0.024  0.018 | 0.279  0.078  0.037 |
## 19                | -0.193 0.041  0.014 | -0.063  0.007  0.002 |

```

```

## 21 |  0.845  0.731  0.369 | -0.337  0.184  0.059 |
## 23 | -0.253  0.155  0.058 | -0.020  0.002  0.000 |
## 26 |  0.802  0.552  0.170 |  0.460  0.288  0.056 |
##          Dim.3    ctr    cos2
## 4          0.251  0.105  0.048 |
## 10         -0.413  0.757  0.167 |
## 11         -0.354  0.252  0.097 |
## 15         -0.503  0.291  0.209 |
## 17          0.835  1.019  0.486 |
## 18          0.104  0.012  0.005 |
## 19          0.159  0.051  0.010 |
## 21         -0.390  0.285  0.079 |
## 23         -0.375  0.624  0.128 |
## 26          0.098  0.015  0.003 |
##          Categories (the 10 first)
##          Dim.1    ctr    cos2 v.test   Dim.2    ctr
## Q3. Moins de 1 an | -0.731  2.704  0.068 -4.940 |  0.677  3.681
## Q3. De 1 à 4 ans | -0.286  1.259  0.043 -3.921 | -0.791 15.299
## Q3. De 5 à 9 ans |  0.119  0.122  0.003  1.102 |  0.385  2.023
## Q3. 10 ans ou plus |  0.450  3.189  0.110  6.271 |  0.347  3.006
## Q4. Moins de 10% | -0.562  7.874  0.395 -11.913 |  0.033  0.044
## Q4. 10 à 25%    |  0.745  4.918  0.137  7.006 | -0.765  8.220
## Q4. 26 à 50%    |  0.775  3.815  0.099  5.966 |  0.644  4.176
## Q4. Plus de 50% |  0.527  1.310  0.033  3.424 |  0.395  1.167
## Q5. Oui          |  0.950  9.194  0.265  9.760 |  0.557  5.008
## Q5. Non          | -0.279  2.701  0.265 -9.760 | -0.164  1.471
##          cos2 v.test   Dim.3    ctr    cos2 v.test
## Q3. Moins de 1 an |  0.058  4.578 |  0.627  3.636  0.050  4.236 |
## Q3. De 1 à 4 ans |  0.328 -10.855 |  0.390  4.304  0.080  5.360 |
## Q3. De 5 à 9 ans |  0.035  3.556 | -1.110 19.427  0.293 -10.261 |
## Q3. 10 ans ou plus |  0.065  4.835 |  0.023  0.016  0.000  0.327 |
## Q4. Moins de 10% |  0.001  0.704 |  0.070  0.221  0.006  1.477 |
## Q4. 10 à 25%    |  0.144 -7.194 | -0.109  0.191  0.003 -1.021 |
## Q4. 26 à 50%    |  0.068  4.957 |  0.154  0.276  0.004  1.187 |
## Q4. Plus de 50% |  0.018  2.566 | -0.372  1.195  0.016 -2.417 |
## Q5. Oui          |  0.091  5.720 | -0.126  0.294  0.005 -1.290 |
## Q5. Non          |  0.091 -5.720 |  0.037  0.086  0.005  1.290 |
##          Categorical variables (eta2)
##          Dim.1 Dim.2 Dim.3
## q3          0.162 0.338 0.334 |
## q4          0.400 0.191 0.023 |
## q5          0.265 0.091 0.005 |
## q8          0.071 0.032 0.352 |
## q9          0.548 0.340 0.338 |
## q10         0.381 0.015 0.004 |
## q11         0.245 0.235 0.034 |
## q12         0.121 0.038 0.007 |
## q13         0.036 0.126 0.122 |

```

```

## 
## Supplementary categories (the 10 first)
## 
## Q15. 18 à 34 ans | -0.091 0.004 -1.221 | -0.271 0.037 -3.630 |
## Q15. 35 à 49 ans | -0.060 0.001 -0.731 | -0.014 0.000 -0.175 |
## Q15. 50 à 64 ans | 0.140 0.006 1.419 | 0.251 0.018 2.543 |
## Q15. 65 ans et plus | 0.107 0.002 0.874 | 0.248 0.011 2.020 |
## Q16. Travail temps plein | -0.096 0.012 -2.084 | -0.065 0.005 -1.404 |
## Q16. Travail. temps partiel | 0.364 0.010 1.875 | -0.147 0.002 -0.757 |
## Q16. Étudiant | -0.139 0.002 -0.774 | -0.111 0.001 -0.617 |
## Q16. Retraité | 0.166 0.006 1.521 | 0.254 0.015 2.331 |
## Q16. Sans emploi | 0.440 0.006 1.433 | -0.142 0.001 -0.464 |
## Q16. À la maison | -0.056 0.000 -0.279 | 0.175 0.002 0.878 |
## 
## Dim.3 cos2 v.test
## Q15. 18 à 34 ans | 0.092 0.004 1.232 |
## Q15. 35 à 49 ans | -0.112 0.005 -1.360 |
## Q15. 50 à 64 ans | -0.002 0.000 -0.020 |
## Q15. 65 ans et plus | 0.015 0.000 0.122 |
## Q16. Travail temps plein | -0.025 0.001 -0.540 |
## Q16. Travail. temps partiel | 0.291 0.006 1.497 |
## Q16. Étudiant | -0.088 0.001 -0.488 |
## Q16. Retraité | 0.031 0.000 0.284 |
## Q16. Sans emploi | -0.495 0.007 -1.611 |
## Q16. À la maison | 0.144 0.001 0.720 |
## 
## Supplementary categorical variables (eta2)
## 
## q15 | 0.010 0.048 0.007 |
## q16 | 0.027 0.020 0.015 |
## q17 | 0.006 0.014 0.014 |
## q21 | 0.002 0.001 0.007 |

```

```

# Construction d'un DataFrame pour les valeurs propres
dfACMvp <- data.frame(res.acm$eig)
names(dfACMvp) <- c("VP","VP_pct","VP_pctCumul")
dfACMvp$Axe <- factor(1:nrow(dfACMvp), levels=rev(1:nrow(dfACMvp)))
dfACMvp <- dfACMvp[,c(4,1:3)]

```

12.4.2.2 Exploration graphique des résultats de l'ACM pour les valeurs propres

Pour créer un histogramme des valeurs propres de l'ACM, vous pouvez utiliser la fonction `fviz_screeplot` de `factoextra`.

```

library(factoextra)
library(ggplot2)

fviz_screeplot(res.acm, addlabels = TRUE,
               x="Composantes", y="Valeur propre", title="")

```

Avec un peu plus de lignes de code, il est relativement facile d'exploiter le *DataFrame* des valeurs propres

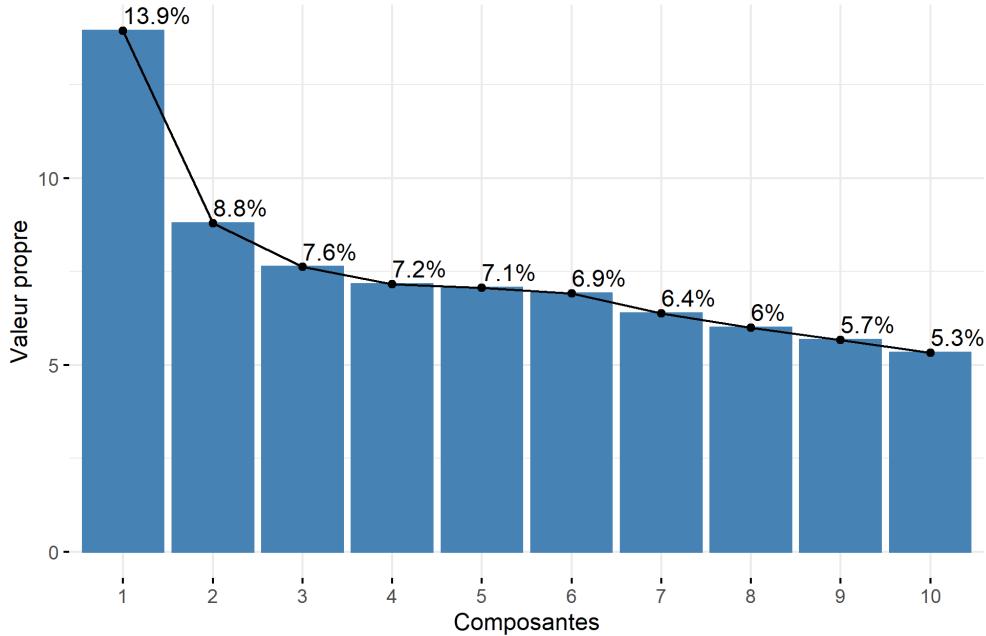


FIG. 12.44 : Graphique pour les valeurs propres de l'ACM avec factoextra

créé précédemment (dfACMvp) pour construire des graphiques plus personnalisés.

```
library(factoextra)
library(ggplot2)

couleursAxes <- c("steelblue","skyblue2")
g1 <- ggplot(dfACMvp,aes(x=VP, y=Axe))+
  geom_bar(stat="identity", width = .6, alpha=.8, color="black", fill="skyblue2")+
  labs(x="Valeur propre", y="Axe factoriel")
g2 <- ggplot(dfACMvp, aes(x=VP_pct, y=Axe))+ 
  geom_bar(stat="identity", width = .6, alpha=.8, color="black", fill="skyblue2")+
  theme(legend.position="none")+
  labs(x="Variance expliquée (%)", y="Axe factoriel")
g3 <- ggplot(dfACMvp, aes(x=VP_pctCumul, y=Axe, group=1))+ 
  geom_bar(stat="identity", width = .6, alpha=.8, color="black", fill="skyblue2")+
  geom_line(colour="brown", linetype="solid", size=.8) +
  geom_point(size=3, shape=21, color="brown", fill="brown")+
  theme(legend.position="none")+
  labs(x="Variance expliquée (% cumulé)", y="Axe factoriel")
ggarrange(g2, g3, nrow = 2)
```

La syntaxe ci-dessous permet de construire un tableau avec les coordonnées factorielles, les cosinus carrés et les contributions pour les modalités des variables qualitatives.

```
library(stringr)
nAxes <- 3
dfmodalites <- data.frame(Modalite = rownames(res.acm$var$coord),
                           Coord = round(res.acm$var$coord[, 1:nAxes],3),
                           Cos2 = round(res.acm$var$cos2[, 1:nAxes],3),
```

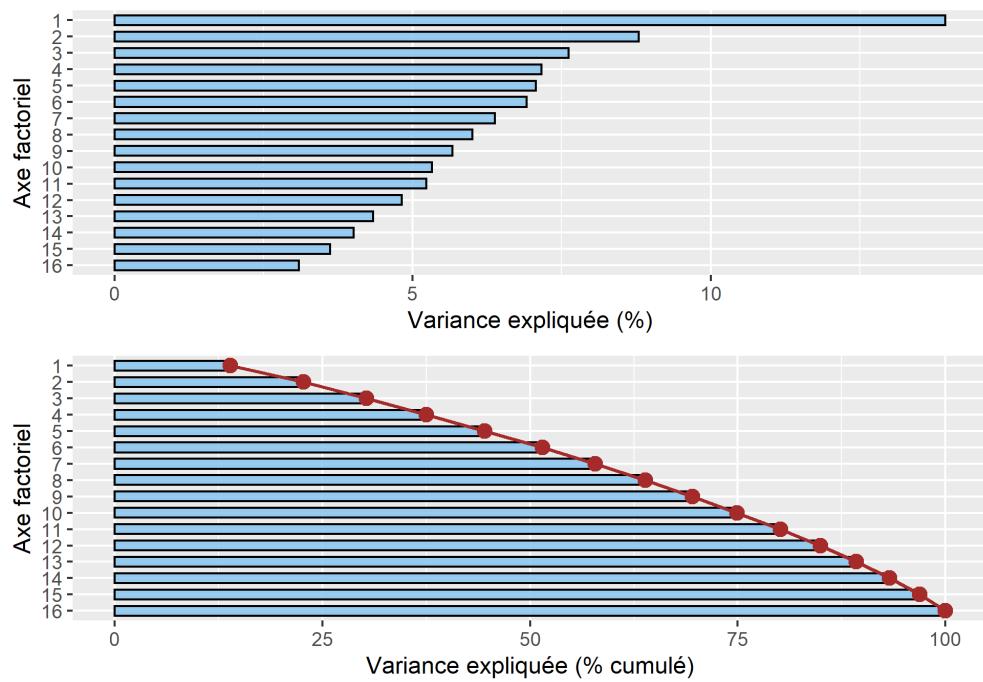


FIG. 12.45 : Graphiques pour les valeurs propres de l'ACM avec factoextra

```
ctr = round(res.acm$var$contrib[, 1:nAxes], 3)
rownames(dfmodalites) <- 1:nrow(dfmodalites)
names(dfmodalites) <- str_replace(names(dfmodalites), ".Dim.", "F")
```

12.4.2.3 Exploration graphique des résultats de l'ACM pour les modalités

Avant d'explorer graphiquement les résultats pour les modalités, il est judicieux de construire un *DataFrame* avec les coordonnées factorielles, les contributions et les cosinus carrés des modalités (voir la syntaxe ci-dessous).

```
library(kableExtra)
library(stringr)
nAxes <- 3
dfmodalites <- data.frame(Modalite = rownames(res.acm$var$coord),
                           Coord = round(res.acm$var$coord[, 1:nAxes], 2),
                           ctr = round(res.acm$var$contrib[, 1:nAxes], 2),
                           Cos2 = round(res.acm$var$cos2[, 1:nAxes], 2))
rownames(dfmodalites) <- 1:nrow(dfmodalites)
names(dfmodalites) <- str_replace(names(dfmodalites), ".Dim.", "F")
```

Plusieurs fonctions très faciles à utiliser de `factoextra` permettent de construire rapidement des graphiques : `fviz_mca_var` pour un nuage de points d'un plan factoriel, `fviz_cos2` et `fviz_contrib` (en utilisant le paramètre `choice=var.cat`) pour des histogrammes avec les cosinus carrés et les contributions des modalités. N'hésitez pas à consulter l'aide de ces fonctions ou encore cette section du site de STHDA⁴.

⁴de%20<http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/75-acm-analyse-des-correspondances-multiples-avec-r-l-essentiel/#graphique-des-variables>

Il est aussi possible de créer vos propres graphiques avec `ggplot2` en utilisant le `DataFrame` créé précédemment avec les modalités. Par exemple, la syntaxe ci-dessous renvoie deux histogrammes pour l'axe 1 : l'un avec les coordonnées, l'autre avec les contributions. Dans la syntaxe, repérez le terme `CoordF1`. Dupliquez la syntaxe et changez ce terme pour `CoordF2` et `CoordF3` pour réaliser les graphiques des axes 2 et 3.

```
# Histogrammes pour les coordonnées des modalités
couleursCoords <- c("lightsalmon","steelblue")
plotCoordF1 <- ggplot(dfmodalites,
  aes(y = reorder(Modalite, CoordF1),
      x = CoordF1, fill=CoordF1<0))+ 
  geom_bar(stat="identity", width = .6, alpha=.8, color="black")+
  geom_vline(xintercept=0, color = "black", size=1)+ 
  scale_fill_manual(name="Coordonnée",values=couleursCoords,
    labels = c("Positive","Négative"))+
  labs(x="Coordonnées sur l'axe 1", y="Modalité")+
  theme(legend.position="none", axis.text.y = element_text(size = 7))

plotCtrF1 <- ggplot(dfmodalites, aes(y = reorder(Modalite, ctrF1), x = ctrF1))+ 
  geom_bar(stat="identity", width = .6, alpha=.8, color="black", fill="steelblue")+
  labs(x="Contributions sur l'axe 1", y="Modalité")+
  theme(legend.position="none", axis.text.y = element_text(size = 7))

ggarrange(plotCoordF1, plotCtrF1, ncol = 1, nrow = 2)
```

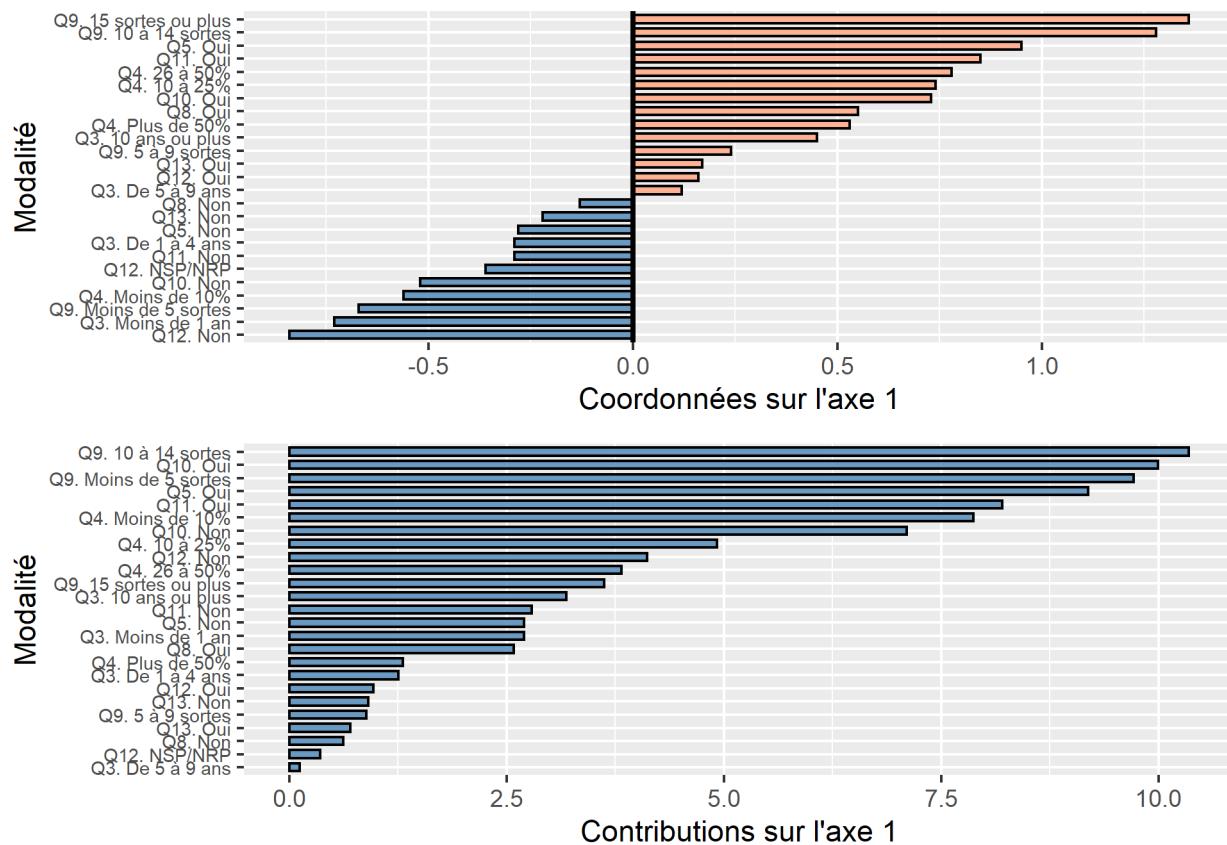


FIG. 12.46 : Exemple de graphiques pour les résultats des modalités

La syntaxe suivante permet de construire le premier plan factoriel pour les modalités avec la fonction

`fviz_mca_var` de `factoextra` (figure 12.47).

```
res.acm2 <- MCA(dfACM[1:9], ncp = 3, graph = FALSE, row.w = dfenquete$pond)
fviz_mca_var(res.acm2, repel = TRUE,
             choice="var.cat",
             axes = c(1, 2),
             # col.var = "black",
             title="", xlab="Axe 1", ylab="Axe 2",
             ggtheme = theme_minimal ())
```

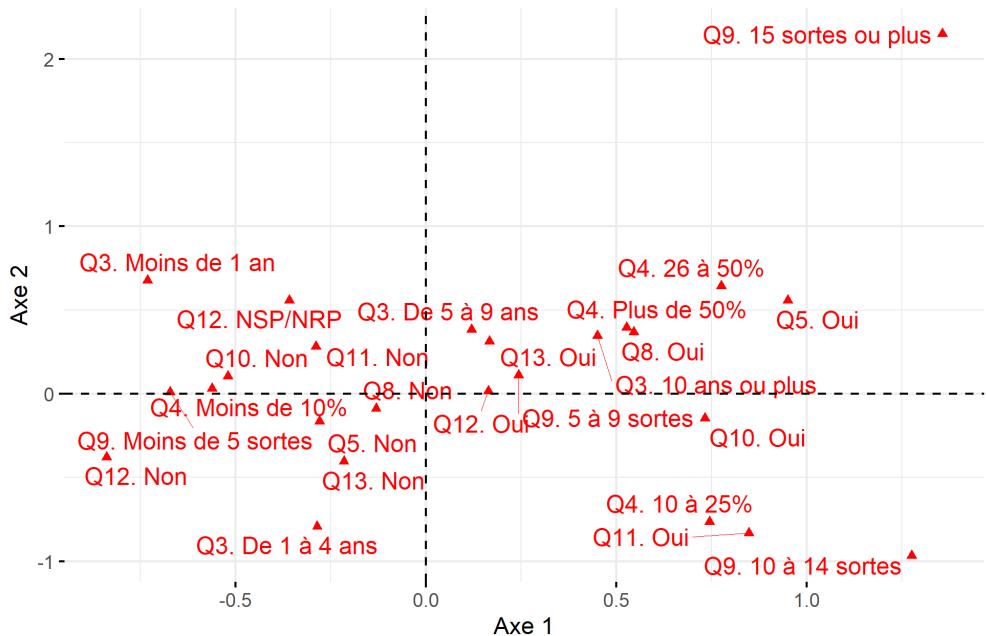


FIG. 12.47 : Premier plan factoriel de l'ACM pour les modalités

La syntaxe suivante permet de construire le premier plan factoriel pour les modalités supplémentaires avec la fonction `fviz_mca_var` de `factoextra` (figure 12.48).

```
fviz_mca_var(res.acm, repel = TRUE,
             choice="var.cat",
             axes = c(1, 2),
             col.var = "gray23",
             col.quali.sup = "darkred",
             labelsize = 3,
             title="", xlab="Axe 1", ylab="Axe 2",
             ggtheme = theme_minimal ())
```

Finalement, la syntaxe ci-dessous renvoie un graphique avec la trajectoire de la variable q3 (figure 12.49).

```
library(ggpubr)
Q3 <- dfmodalites[1:4, 1:3]
ggplot(Q3, aes(x=CoordF1, y=CoordF2, label=Modalite))+
  xlim(-1, .75)+ylim(-1, 1)+
  labs(title = "Q3. Depuis combien de temps cultivez-vous \n")
```

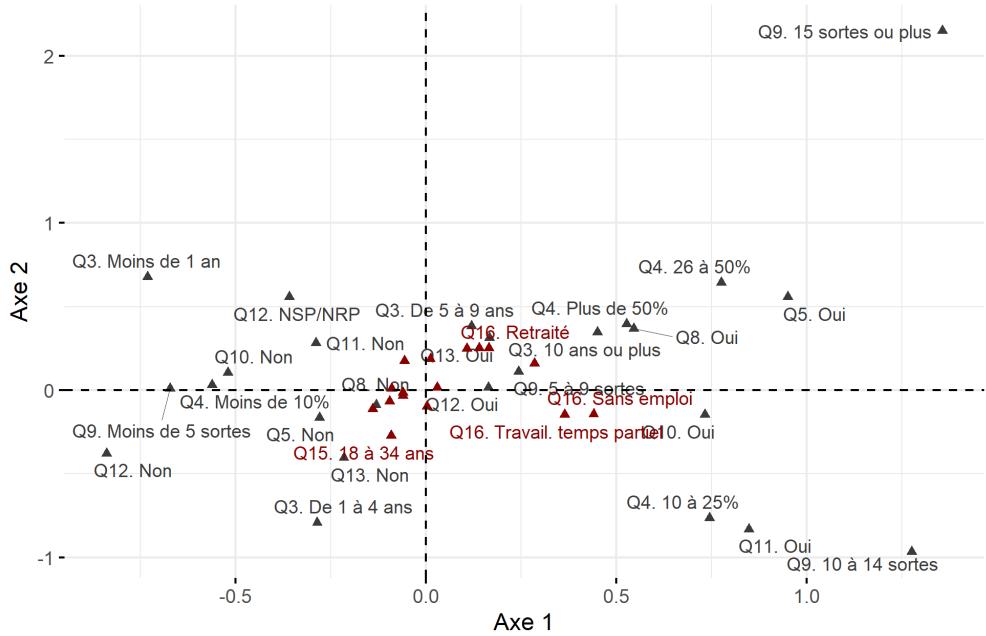


FIG. 12.48 : Premier plan factoriel de l'ACM pour les modalités supplémentaires

```

des fruits, des fines herbes ou des légumes?",  

x="Axe 1", y="Axe 2") +  

geom_label(nudge_x=0, nudge_y=0.07) +  

geom_line( color="black", size=.2) +  

geom_point(shape=21, color="black", fill="steelblue", size=4)

```

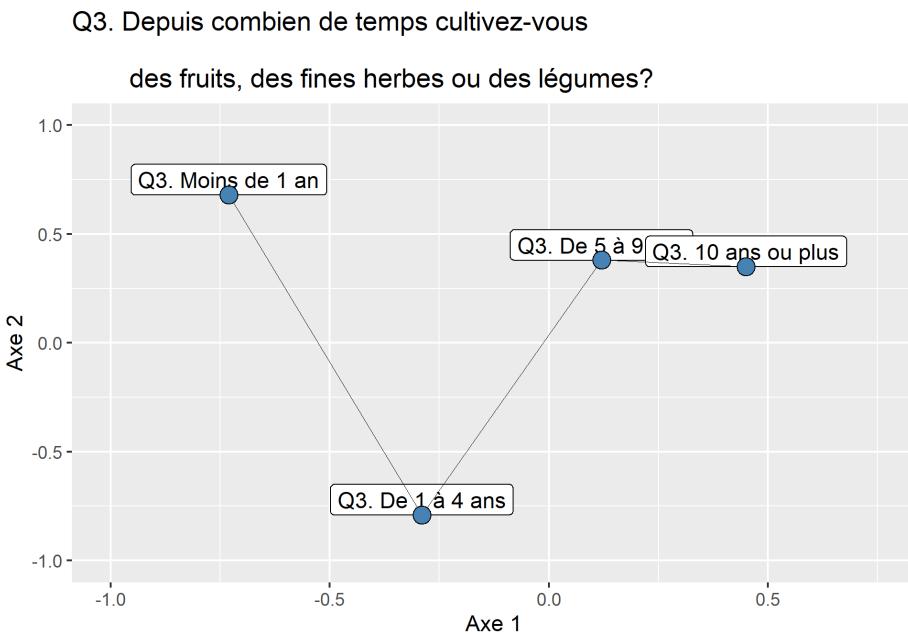


FIG. 12.49 : Trajectoires des variables ordinales sur le premier plan factoriel de l'ACM

12.4.2.4 Exploration graphique des résultats de l'ACM pour les individus

D'autres fonctions de `factoextra` produisent rapidement des graphiques pour les individus :

- `fviz_cos2` et `fviz_contrib` (en utilisant le paramètre `choice=ind`) pour construire des histogrammes pour les cosinus carrés et les contributions des individus.
- `fviz_mca_ind` pour un nuage de points d'un plan factoriel (axes 1 et 2 habituellement).

La syntaxe ci-dessous produit le premier axe factoriel pour les individus (figure 12.50).

```
fviz_mca_ind(res.acm, col.ind = "cos2",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
              repel = TRUE,
              xlab="Axe 1", ylab="Axe 2", title="",
              ggtheme = theme_minimal())
```

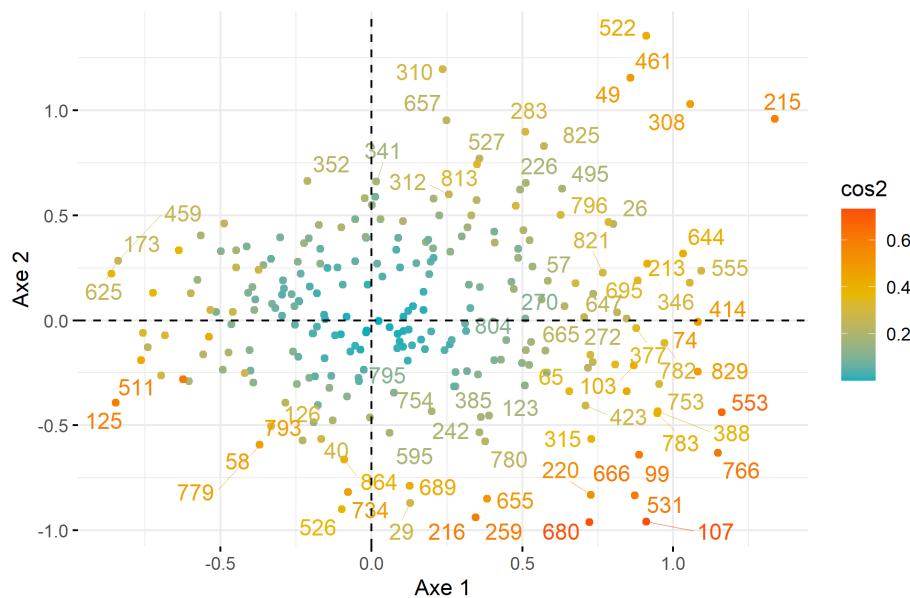


FIG. 12.50 : Premier plan factoriel de l'ACM pour les individus avec factoextra

La syntaxe ci-dessous produit aussi le premier plan factoriel pour les individus, mais en attribuant une couleur différente aux modalités de la variable `q12` (figure 12.51).

```
fviz_mca_ind (res.acm,
               label = "none",
               habillage = "q12", # colorer par groupes
               xlab="Axe 1", ylab="Axe 2", title="",
               palette = c ("darkred", "steelblue", "gray23"),
               ggtheme = theme_minimal ())
```

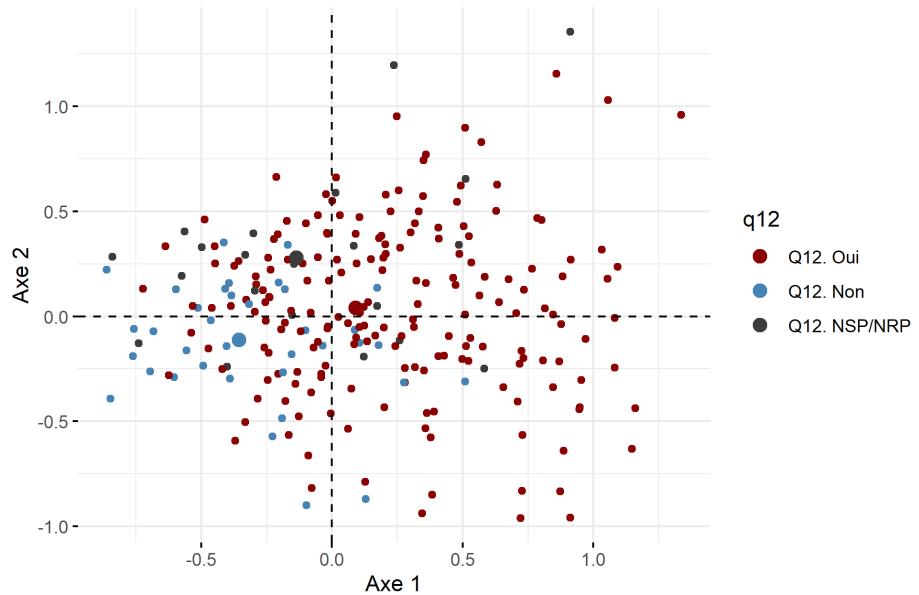


FIG. 12.51 : Premier plan factoriel de l'ACM pour les individus avec coloration d'une variable avec factoextra

12.5 Quiz de révision du chapitre

Questions

- Des variables latentes ne sont pas directement observées, mais plutôt produites par la méthode factorielle afin de résumer les relations/associations entre plusieurs variables mesurées initialement.

- Vrai
- Faux

Relisez au besoin l'introduction du chapitre [12](#).

- Quels sont les métriques utilisées pour les trois principales méthodes factorielles ?

- ACP (distance euclidienne), AFC (distance du khi-deux), ACM (distance du khi-deux)
- ACP (distance du khi-deux), AFC (distance du khi-deux), ACM (distance euclidienne)
- ACP (distance euclidienne), AFC (distance euclidienne), ACM (distance du khi-deux)

Relisez au besoin la section [12.1.1](#).

- En ACP normée, la somme des valeurs propres (inertie totale) est égale au :

- nombre de variables quantitatives du tableau initial
- nombre d'observations moins le nombre de variables

Relisez au besoin le début de la section [12.2.1](#).

- En ACP normée, la coordonnée factorielle d'une variable sur un axe est :

- la covariance de la variable avec la composante principale (axe factoriel)
- le coefficient de corrélation de la variable avec la composante principale (axe factoriel)
- la variance de la variable

Relisez au besoin la section [12.2.2](#).

- Quelles affirmations sont exactes pour toutes les méthodes factorielles ?

- La somme des cosinus carrés en ligne est toujours égale à 1
- La somme des valeurs propres est toujours égale à l'inertie totale du tableau
- La somme des contributions en colonne pour un axe est égale à 100 %
- Les coordonnées factorielles sont toujours le coefficient de corrélation de la variable avec l'axe

Relisez au besoin chacune des sections intitulées aides à l'interprétation pour les trois méthodes factorielles.

- L'analyse des correspondances multiples (ACM) est simplement une analyse des correspondances (AFC) sur un tableau disjonctif complet ?

- Vrai
- Faux

Relisez au besoin la section [12.4](#).

- Dans une ACM, les variables du tableau disjonctif complet sont :

- les fréquences des modalités des variables qualitatives
- les modalités des variables qualitatives transformées en variables binaires

Relisez au besoin la section [12.4](#).

- Quels sont les étapes essentielles pour bien interpréter une analyse factorielle (ACP, AFC ou ACM) ?
 - Interprétation des résultats des valeurs propres pour identifier le nombre d'axes à retenir
 - Analyse des résultats pour les variables (coordonnées, cosinus carrés et contributions)
 - Analyse des résultats pour les individus (coordonnées, cosinus carrés et contributions)
 - Dénommer, qualifier chacun des axes suite à l'analyse des résultats pour les variables et les individus

Relisez le deuxième encadré à la section 12.2.1.

Réponses

- Des variables latentes ne sont pas directement observées, mais plutôt produites par la méthode factorielle afin de résumer les relations/associations entre plusieurs variables mesurées initialement.
 - Vrai
- Quels sont les métriques utilisées pour les trois principales méthodes factorielles ?
 - ACP (distance euclidienne), AFC (distance du khi-deux), ACM (distance du khi-deux)
- En ACP normée, la somme des valeurs propres (inertie totale) est égale au :
 - nombre de variables quantitatives du tableau initial
- En ACP normée, la coordonnée factorielle d'une variable sur un axe est :
 - le coefficient de corrélation de la variable avec la composante principale (axe factoriel)
- Quelles affirmations sont exactes pour toutes les méthodes factorielles ?
 - La somme des cosinus carrés en ligne est toujours égale à 1
 - La somme des valeurs propres est toujours égale à l'inertie totale du tableau
 - La somme des contributions en colonne pour un axe est égale à 100 %
- L'analyse des correspondances multiples (ACM) est simplement une analyse des correspondances (AFC) sur un tableau disjonctif complet ?
 - Vrai
- Dans une ACM, les variables du tableau disjonctif complet sont :
 - les modalités des variables qualitatives transformées en variables binaires
- Quels sont les étapes essentielles pour bien interpréter une analyse factorielle (ACP, AFC ou ACM) ?
 - Interprétation des résultats des valeurs propres pour identifier le nombre d'axes à retenir
 - Analyse des résultats pour les variables (coordonnées, cosinus carrés et contributions)
 - Analyse des résultats pour les individus (coordonnées, cosinus carrés et contributions)
 - Dénommer, qualifier chacun des axes suite à l'analyse des résultats pour les variables et les individus

Chapitre 13

Méthodes de classification non supervisée

Dans le cadre de ce chapitre, nous présentons les méthodes les plus utilisées en sciences sociales pour explorer la présence de groupes homogènes au sein d'un jeu de données, soit les méthodes de classification non supervisée. Le qualificatif *non supervisé* signifie que ces classes/groupes ne sont pas connus a priori et doivent être identifiés à partir des données. Autrement dit, nous cherchons à regrouper les observations partageant des caractéristiques similaires sur la base de plusieurs variables. Ces méthodes descriptives et exploratoires multivariées peuvent être vues comme une façon de réduire le nombre d'observations d'un jeu de données à un ensemble d'observations synthétiques, représentant le mieux possible la population à l'étude.



Dans ce chapitre, nous utilisons les *packages* suivants :

- Pour créer des graphiques :
 - * `ggplot2` le seul, l'unique!
 - * `ggpubr` pour combiner des graphiques et réaliser des diagrammes.
- Outils généraux pour faciliter les classifications :
 - * `clusterCrit` pour calculer des indicateurs de qualité de classification.
 - * `NbClust` pour trouver le bon nombre de groupe dans une classification.
 - * `cluster` pour appliquer la méthode GAP.
 - * `proxy` pour calculer plusieurs types de distances.
 - * `Gmedian` pour calculer le k-médianes.
 - * `geocmeans` pour explorer les résultats de classifications floues.



Pourquoi recourir à des méthodes de classification non supervisée en sciences sociales ?

Les méthodes de classification sont très utilisées en sciences sociales. Elles visent à identifier des groupes cohérents au sein d'un ensemble d'observations sur la base de plusieurs variables (figure 13.1). Ces groupes peuvent ensuite être analysés et nous renseigner sur les caractéristiques communes partagées par les individus qui les composent.

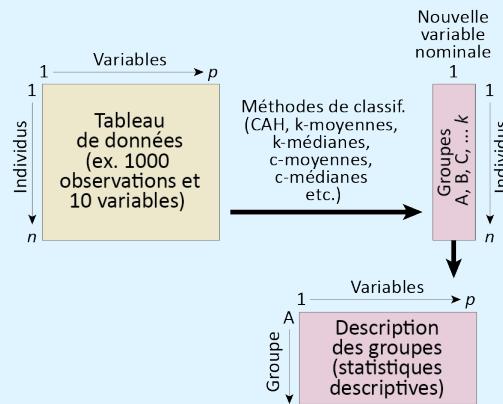


FIG. 13.1 : Principe de base des méthodes de classification non supervisée

Un exemple classique est l'identification de profils d'individus ayant répondu à un sondage, en fonction de plusieurs caractéristiques (par exemple, l'âge, le sexe, la situation de famille, le revenu, etc.). En identifiant ces groupes homogènes, il est ensuite possible d'explorer les associations entre ces profils et d'autres variables.

Un second exemple serait de regrouper les secteurs d'une ville selon leurs caractéristiques environnementales (végétation, niveau de bruit, pollution atmosphérique, etc.) et socioéconomiques (revenu médian des ménages, pourcentage d'immigrants, pourcentage de personnes à faible scolarité, taux de chômage, etc.).

13.1 Méthodes de classification : un aperçu

Il existe une multitude de méthodes de classification généralement regroupées dans plusieurs familles imbriquées à partir de deux distinctions importantes.

La première distinction vise à séparer les méthodes **supervisées** des **non supervisées**. Pour les premières, les catégories/groupes/classes des observations sont connues à l'avance. L'enjeu n'est pas de trouver les catégories puisqu'elles sont connues, mais de **déterminer des règles ou un modèle permettant d'attribuer des observations à ces catégories**. Parmi les méthodes de classification supervisée, les plus connues sont les forêts d'arbres décisionnels, les réseaux de neurones artificiels ou encore l'analyse factorielle discriminante. Nous n'abordons pas ces méthodes dans ce chapitre dédié uniquement aux méthodes de classification non supervisée. Pour ces dernières, les catégories ne sont pas connues à l'avance et l'enjeu est de **faire ressortir les structures des groupes propres aux données**. Ainsi, les méthodes de classification non supervisée « relèvent de la statistique exploratoire multidimensionnelle et permettent de classifier automatiquement les observations sans connaissance a priori sur la nature des classes présentes dans le jeu de données ; les plus connues sont sans conteste les algorithmes de classification ascendante hiérarchique (CAH) et du *k-means* (*k-moyennes*) » (Gelb et Apparicio 2021b, 1). Notez également qu'à la frontière entre ces deux familles, se situent les méthodes de classification semi-supervisée. Il s'agit de cas spécifiques où des informations partielles sont connues sur les groupes à détecter : seulement le groupe final de certaines observations est connu, certaines observations sont supposées appartenir à un même groupe même s'il est indéfini en lui-même (Bair 2013).

La seconde distinction vise à séparer les méthodes **strictes** des **floues**. Les premières ont pour objectif d'assigner chaque observation à une et une seule catégorie, alors que les secondes décrivent le degré d'appartenance de chaque observation à chaque catégorie. Autrement dit, « dans une classification stricte, chaque observation appartient à une seule classe. Mathématiquement parlant, l'appartenance à une classe donnée est binaire (0 ou 1) tandis que dans une classification floue, chaque observation a une probabilité d'appartenance variant de 0 à 1 à chacune des classes » (Gelb et Apparicio 2021b, 1). Bien entendu, pour chaque observation, la somme des degrés d'appartenance à chacune des classes est égale à 1 (figure 13.2). En termes de données, cela signifie que pour les méthodes strictes, le groupe d'appartenance d'une observation est contenu dans une seule variable nominale (une colonne d'un *DataFrame*). Pour les méthodes floues, il est nécessaire de disposer d'autant de variables continues (plusieurs colonnes numériques d'un *DataFrame*), soit une par groupe, dans lesquelles est enregistré le degré d'appartenance de chaque observation à chacun des groupes. Parmi les méthodes de classification supervisée floue, notez que nous avons déjà abordé la régression logistique multinomiale dans le chapitre sur les GLM (section 8.2.4).

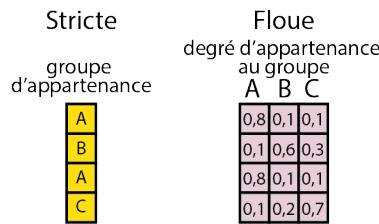


FIG. 13.2 : Classifications stricte et floue

En résumé, le croisement de ces deux distinctions permet ainsi de différencier les méthodes **supervisées strictes**, **supervisées floues**, **non supervisées strictes** et **non supervisées floues** (figure 13.3), auxquelles s'ajoutent les méthodes semi-supervisées discutées brièvement.

	Classifications non supervisées	Classifications supervisées
Classifications strictes	k-moyennes (<i>k-means</i>) k-médianes (<i>k-medians</i>) k-médoïdes (<i>k-medoids</i>) Isodata CAH ¹ Classification mixte BIRCH ² DBSCAN ³ OPTICS ⁴ Partitionnement spectral	Forêts d'arbres décisionnels Méthode des <i>k</i> plus proches voisins Machines à vecteurs de support Analyse factorielle discriminante Réseaux de neurones artificiels
Classifications floues	Classification k-moyennes floue (<i>Fuzzy c-means -FCM</i>) Classification k-medianes floue (<i>Fuzzy c-Least Medians clustering</i>) Modèles à mélanges finis	Régression multinomiale Modèle de Markov caché Classification naïve bayésienne

¹ Classification ascendante hiérarchique.

² *Balanced Iterative Reducing and Clustering using Hierarchies*.

³ *Density-Based Spatial Clustering of Applications with Noise*.

⁴ *Ordering Points To Identify the Clustering Structure*.

Conception et réalisation : Jérémie Gelb et Philippe Apparicio, 2021.

FIG. 13.3 : Synthèse des principales méthodes de classification (Gelb et Apparicio 2021)

Dans ce chapitre, nous décrivons les trois méthodes de classification non supervisée les plus utilisées et faciles à mettre en œuvre : la classification ascendante hiérarchique, les nuées dynamiques strictes (*k-means* et *k-medians*) et nuées dynamiques floues (*c-means* et *c-medians*).

13.2 Notions essentielles en classification

Avant de décrire différentes méthodes de classification non supervisée, il convient de définir deux notions centrales, soit la **distance** et l'**inertie**.

13.2.1 Distance

La distance en analyse de données est définie comme une fonction (d) permettant de déterminer à quel point deux observations sont semblables ou différentes l'une de l'autre. Elle doit respecter les conditions suivantes :

- **la non-négativité** : la distance minimale entre deux objets est égale à 0; $d(x, y) \geq 0$.
- **le principe d'identité des indiscernables** : la distance entre deux objets x et y est égale à 0, si $x = y$; $d(x, y) = 0$ si et seulement si $x = y$.
- **la symétrie** : la distance entre x et y est la même qu'entre y et x ; $d(x, y) = d(y, x)$.
- **le triangle d'inégalité** : passer d'un point x à un point z est toujours plus court ou égal que de passer par y entre x et z ; $d(x, z) \leq d(x, y) + d(y, z)$.

Il existe un grand nombre de types de distance qui peuvent être utilisés pour déterminer le degré de similitude entre les observations. Nous présentons ici les six types les plus fréquemment utilisés en sciences sociales, mais retenez qu'il en existe bien d'autres.

13.2.1.1 Distance euclidienne

Il s'agit vraisemblablement de la distance la plus couramment utilisée, soit la longueur de la ligne droite la plus courte entre les deux objets considérés. Pour la représenter, admettons que nous nous intéressons à trois classes d'étudiants et d'étudiantes A, B et C pour lesquelles nous avons calculé la moyenne de leurs notes dans les cours de méthodes quantitatives et qualitatives. Ces deux variables sont mesurées dans la même unité et varient de 0 à 100. Le nuage de points à la figure 13.4 illustre cette situation avec des données fictives.

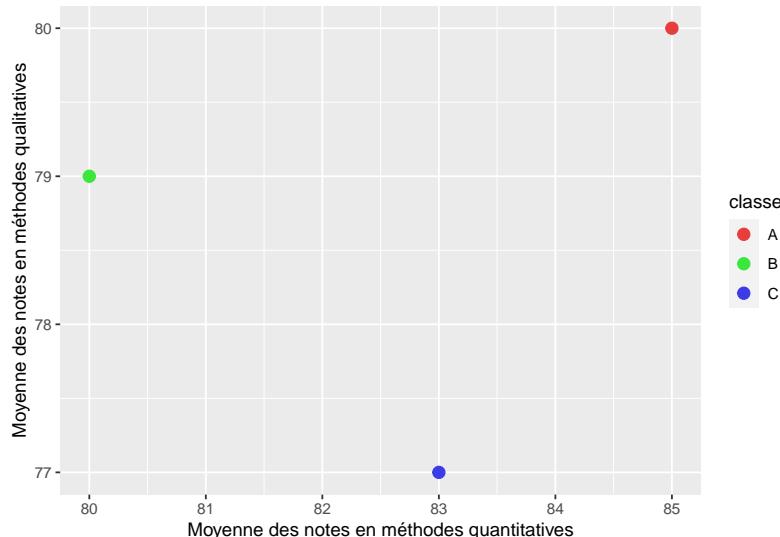


FIG. 13.4 : Situation de base pour le calcul de distance

Les distances euclidiennes entre les classes B et C et les classes C et A sont représentées par les lignes noires à la figure 13.5. Nous pouvons constater que la distance entre les classes C et B est plus petite que celle entre les classes A et C, ce qui signale que les deux premières se ressemblent davantage.

La formule de la distance euclidienne (équation (13.1)) est simplement la racine carrée de la somme des écarts au carré pour chacune des variables décrivant les observations a et b .

$$d(a, b) = \sqrt{\sum_{i=1}^v (a_i - b_i)^2} \quad (13.1)$$

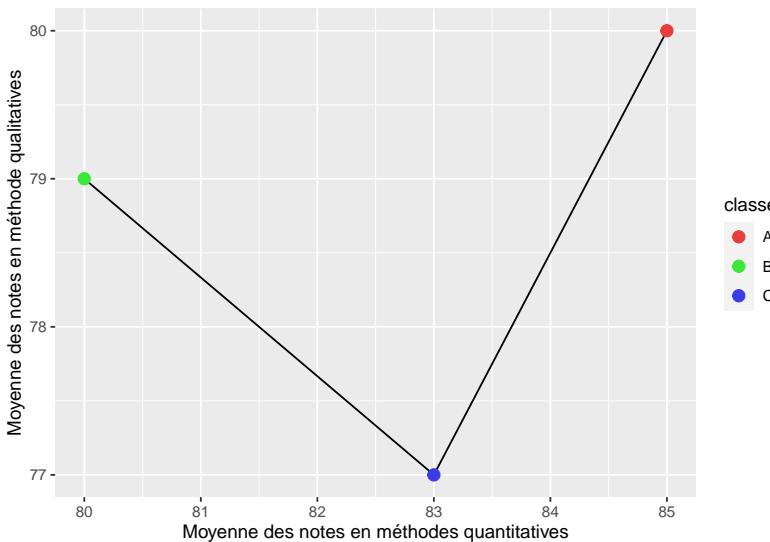


FIG. 13.5 : Représentation de la distance euclidienne

avec v le nombre de variables décrivant les observations a et b .

Nous pouvons facilement calculer la distance euclidienne pour notre jeu de données :

- $d(A, B) = \sqrt{(85 - 80)^2 + (80 - 77)^2} = 5,83$
- $d(B, C) = \sqrt{(80 - 83)^2 + (79 - 77)^2} = 3,60$

⚠️ Distance et unité de mesure

Il est très important de garder à l'esprit que la distance entre deux observations dépend directement des unités de mesure utilisées. Cela est très souvent problématique, car il est rare que toutes les variables utilisées pour décrire des observations soient mesurées dans la même unité. Ainsi, une variable dont les valeurs numériques sont plus grandes risque de déséquilibrer les calculs de distance. À titre d'exemple, une variable mesurée en mètres plutôt qu'en kilomètres produit des distances euclidiennes 1000 fois plus grandes.

Il est donc nécessaire de standardiser les variables utilisées avant de calculer des distances. Cette opération permet de transformer les variables originales vers une échelle commune. Plusieurs types de transformations peuvent être utilisés tels que décrits à la section 2.5.5.2 :

- **Le centrage et la réduction** qui consistent à soustraire de chaque valeur sa moyenne, puis à la diviser par son écart-type. La nouvelle variable obtenue s'exprime alors en écart-type (appelé aussi score-z). La formule de la transformation est $f(x) = \frac{x-\bar{x}}{\sigma_x}$, avec \bar{x} la moyenne de x et σ_x l'écart-type de x .
- **La transformation sur une mise à l'échelle de 0 à 1** qui permet de modifier l'étendue d'une variable afin que sa valeur maximale soit de 1 et sa valeur minimale soit de 0. La formule de cette transformation est $f(x) = \frac{x-\min(x)}{\max(x)-\min(x)}$.
- **La transformation en rang** qui consiste à remplacer les valeurs d'une variable par leur rang. La valeur la plus faible est remplacée par 1, et la plus forte par n (nombre d'observations). Notez que cette transformation modifie la distribution de la variable originale contrairement aux deux transformations précédentes. Cette propriété peut être désirable si les écarts absolus entre les valeurs ont peu d'importance, si la variable n'a pas été mesurée avec précision ou encore si des valeurs extrêmes sont présentes.
- **La transformation en percentile** qui consiste à remplacer les valeurs d'une variable par leur percentile correspondant. Elle peut être vue comme une standardisation de la transformation en rang, car elle ne dépend pas du nombre d'observations.

La figure 13.6 montre l'effet de ces transformations sur l'histogramme d'une variable.

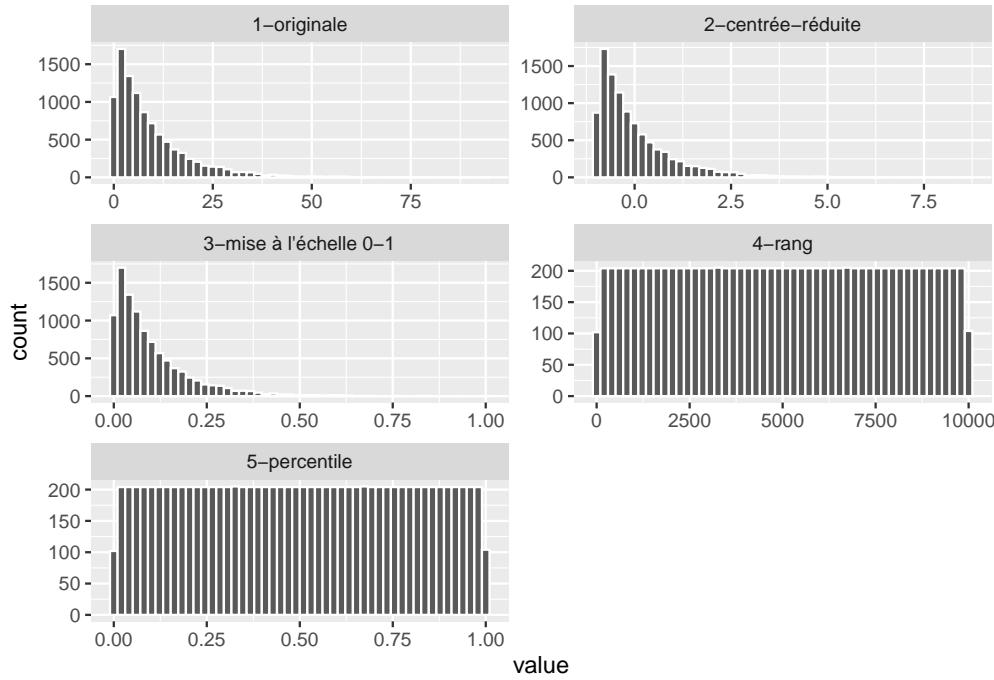


FIG. 13.6 : Effets de différentes transformations sur la distribution d'une variable

13.2.1.2 Distance de Manhattan

Cette seconde distance est également couramment utilisée. Elle doit son nom au réseau de rue de l'île de Manhattan qui suit un plan quadrillé. La distance de Manhattan correspond à la somme des écarts absolus entre les valeurs des différentes variables décrivant les observations (équation (13.2)). La figure 13.7 illustre que la distance Manhattan (lignes noires) représente les deux côtés opposés de l'hypoténuse d'un triangle rectangle; l'hypoténuse représentant quant à elle la distance euclidienne.

$$d(a, b) = \sum_{i=1}^v (|a_i - b_i|) \quad (13.2)$$

La distance de Manhattan doit être privilégiée à la distance euclidienne lorsque les données considérées ont un très grand nombre de dimensions (variables). En effet, lorsque le nombre de variables est important (supérieur à 30), la distance euclidienne tend à être grande pour toutes les paires d'observations et à moins bien discriminer les observations proches et lointaines les unes des autres. Du fait de sa nature additive, la distance de Manhattan est moins sujette à ce problème (Aggarwal, Hinneburg et Keim 2001).

Calculons la distance de Manhattan pour nos deux paires d'observations :

- $d(A, B) = |85 - 80| + |80 - 77| = 8$
- $d(B, C) = |80 - 83| + |79 - 77| = 5$

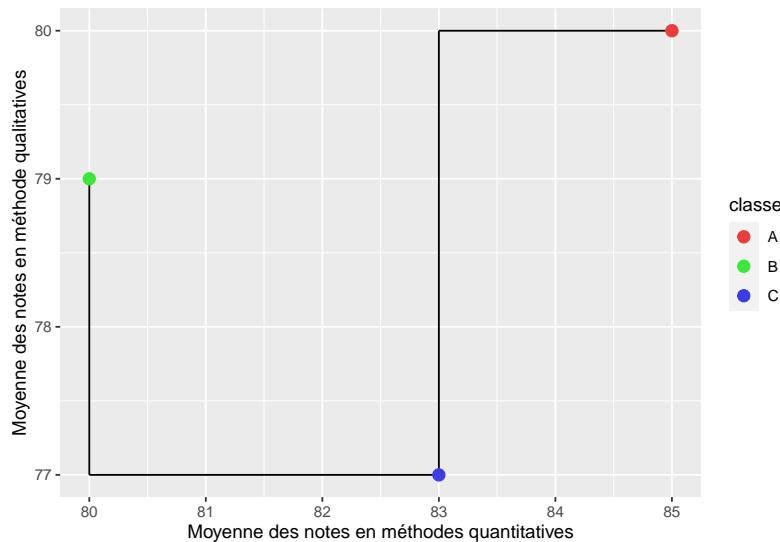


FIG. 13.7 : Représentation de la distance de Manhattan

13.2.1.3 Distance du khi-deux

La distance du khi-deux est basée sur le test du khi-deux (chapitre 5) et est généralement utilisée pour calculer la distance entre deux histogrammes, deux images ou deux ensembles de mots. Plus précisément, elle permet de mesurer la distance entre deux observations A et B, pour lesquelles nous disposons d'un ensemble de variables étant toutes des variables de comptage.

Prenons un exemple concret en générant trois histogrammes A, B et C sur l'intervalle [0,50] à partir des distributions normale, log-normale et Gamma, puis comptons le nombre de valeurs de chaque unité (1, 2, 3, 4, etc.). Ces histogrammes sont représentés à la figure 13.8.

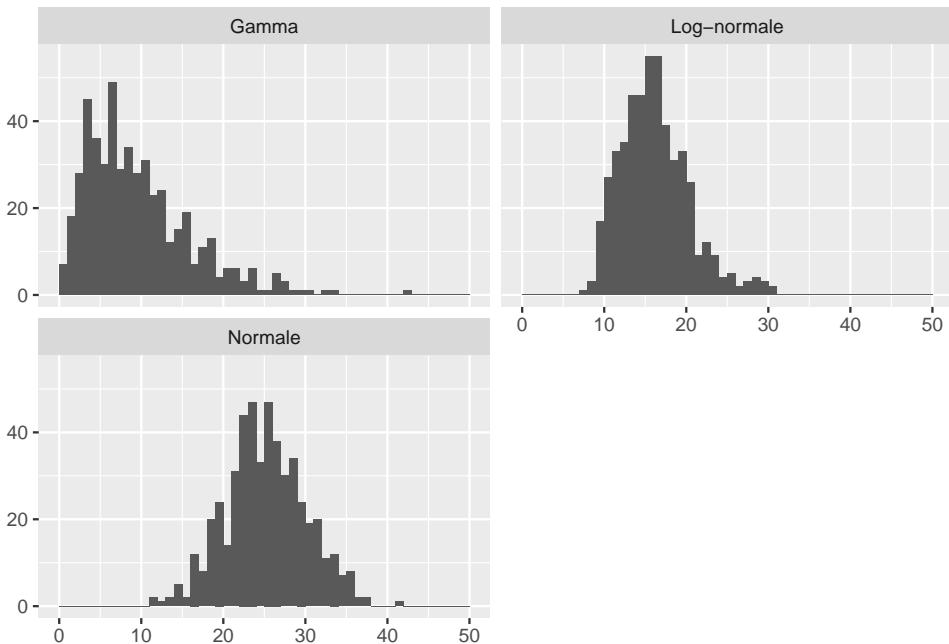


FIG. 13.8 : Trois histogrammes pour illustrer le calcul de la distance du khi-deux

Nous pouvons calculer les distances du khi-deux entre les paires d'histogrammes (tableau 13.1). Nous constatons ainsi que les histogrammes B et C sont les plus semblables.

La formule de cette distance est la suivante :

$$d_{\chi^2}(a, b) = \frac{1}{2} \sum_{i=1}^n \frac{(a_i - b_i)^2}{(a_i + b_i)} \quad (13.3)$$

avec a_i et b_i les comptages pour les histogrammes. Notez que si a_i et b_i valent tous les deux 0, il faut retirer ces valeurs avant le calcul, car cela provoquerait une division par 0.

À première vue, cette distance peut paraître moins utile que les deux précédentes. Pourtant, de nombreuses données sont collectées comme des histogrammes. Un premier exemple serait des images que nous pouvons représenter sous forme de trois histogrammes, un pour chaque canal de couleur (rouge, vert et bleu). Un second exemple serait des données sonores, souvent synthétisées sous forme d'histogrammes des fréquences sonores enregistrées (octaves ou tiers d'octaves). Un dernier exemple pourrait être le nombre d'accidents de la route enregistré à diverses intersections d'une ville chaque heure. Dans ce contexte, un histogramme serait formé par l'intersection avec les heures de la journée comme limites des bandes et le nombre d'accidents comme hauteur des bandes.

13.2.1.4 Distance de Mahalanobis

Proposée dans les années 1930 par le statisticien indien Prasanta Chandra Mahalanobis (1936), cette distance se base sur la matrice de covariance des variables analysées. Plus spécifiquement, elle est utilisée pour calculer la distance entre un point et une distribution normale multivariée. Elle permet notamment de tenir compte du fait que certaines variables sont corrélées et ainsi d'éviter de surestimer les distances entre des observations dans des jeux de données comprenant des variables corrélées entre elles.

La formule permettant de calculer cette distance est la suivante :

$$d(a, b) = \sqrt{(a - b)^T S^{-1} (a - b)} \quad (13.4)$$

avec S étant la matrice de covariance.

13.2.1.5 Distance de Hamming

Cette distance est utilisée quand les écarts entre les variables de deux observations sont uniquement binaires. Un bon exemple serait un jeu de données ne comprenant que des variables qualitatives pouvant avoir une valeur identique pour deux observations (distance = 0) ou différente (distance = 1). La distance de Hamming est la simple addition de ces écarts.

Prenons un exemple très simple en prenant trois maisons pour lesquelles nous connaissons cinq caractéristiques 13.2).

TAB. 13.1 : Distance du khi-deux entre trois histogrammes

Histogrammes	Distance du khi-deux
A-B	284,8375
A-C	376,7862
B-C	219,5133

TAB. 13.2 : Exemple de données pour la distance de Hamming

couleur	jardin	garage	cheminée	sous-sol
blanc	non	oui	oui	non
blanc	non	non	oui	non
rouge	oui	oui	non	oui

Nous pouvons utiliser la distance de Hamming pour estimer le niveau de dissimilarité entre ces différentes maisons et l'organiser dans une matrice de distances. À la lecture du tableau 13.3), les maisons 2 et 3 sont les plus dissimilaires (distance de Hamming = 5), et les maisons 1 et 2 les plus similaires (distance de Hamming = 1).

TAB. 13.3 : Distance de Hamming entre les maisons

	maison 1	maison 2	maison 3
maison 1	0	1	4
maison 2	1	0	5
maison 3	4	5	0

13.2.1.6 Distance de Gower

La distance de Gower (1971) peut être utilisée pour mesurer la distance entre deux observations lorsque les données sont à la fois qualitatives et quantitatives. Cette distance est comprise dans un intervalle de 0 à 1, 0 signifiant que les deux observations sont identiques et 1, que les observations sont radicalement différentes.

Elle se calcule de la façon suivante :

$$d(a, b) = 1 - \frac{1}{p} \sum_{j=1}^p s_{12j}$$

$$\begin{cases} s_{xyj} = 1 \text{ si } x_j = y_j, 0 \text{ autrement pour une variable qualitative} \\ s_{xyj} = 1 - \frac{|x_j - y_j|}{\max(j) - \min(j)} \text{ pour une variable quantitative} \end{cases} \quad (13.5)$$

avec p le nombre de variables, x et y deux observations et j une variable.

Autrement dit, si la valeur d'une variable qualitative diffère entre deux observations, la distance entre ces deux observations augmente de $1/p$. Pour une variable quantitative, la distance augmente selon la différence absolue entre les valeurs de la variable divisée par l'étendue totale de la variable, le tout à nouveau divisé par p .

Si cette mesure semble intéressante puisqu'elle permet de combiner des variables quantitatives et qualitatives, elle souffre de deux limites importantes :

- Elle ne prend pas en compte le fait que certaines modalités des variables qualitatives sont moins fréquentes ni que certaines combinaisons sont également moins fréquentes.
- Les variables qualitatives tendent à affecter bien plus la distance que les variables quantitatives. En effet, pour obtenir un écart de 1 sur une variable quantitative, il faut que les deux valeurs soient respectivement le maximum et le minimum de cette variable.



D'autres distances pour des données mixtes

Il existe bien d'autres distances qui peuvent être utilisées dans le cas de données mixtes. Le package `kmed` en implémente cinq (auxquelles s'ajoute la distance de Gower) dans sa fonction `distmix` : les distances de Wishart, de Podani, d'Huang, d'Harikumar et d'Ahamad. Ces différentes distances ont toutes leurs avantages et leurs défauts respectifs ; pour plus d'information, référez-vous à la documentation de la fonction `distmix`.

13.2.1.7 Distance du Phi²

La distance du Φ^2 (Phi²) est une variante de la distance du χ^2 . Il s'agit donc d'une distance à utiliser lorsque les données à analyser sont uniquement qualitatives. Elle calcule la distance entre deux observations en additionnant les différences entre les valeurs de chaque variable (1 si différentes, 0 si identiques, pour chaque variable), divisées respectivement par la fréquence totale d'occurrences de chaque modalité dans le jeu de données. En d'autres termes, cette distance tient compte du fait que certaines valeurs pour des variables qualitatives peuvent être observées plus fréquemment que d'autres et qu'une distance plus grande devrait être obtenue entre deux observations si l'une des deux présente des modalités rares comparativement au reste du jeu de données.

Elle peut être calculée de la façon suivante :

$$d_{\Phi^2}(i, j) = \frac{1}{Q} \sum_k \frac{(\delta_{ik} - \delta_{jk})^2}{f_k} \quad (13.6)$$

avec i et j deux observations, k une modalité d'une variable qualitative, Q le nombre total de modalités des variables qualitatives, $\delta_{ik} = 1$ si l'observation i a la modalité k , 0 sinon et f_k la fréquence de la modalité k dans le jeu de données.

La distance du Φ^2 est très utile pour analyser les résultats de questionnaires.

13.2.2 Inertie

Une notion importante à saisir dans le cadre des méthodes de classification non supervisée est celui celle l'**inertie** d'un jeu de données. Elle est proche de la notion de variance qui a été présentée dans le chapitre sur la statistique univariée (section 2.5.3).

L'inertie est une quantité permettant de décrire la dispersion des observations d'un jeu de données. Cette mesure dépend à la fois des données (nombres d'observations et de variables, échelle des variables) et de la mesure de distance retenue entre deux observations. Plus spécifiquement, l'inertie correspond à la somme des distances entre chaque observation et le centre du jeu de données.

$$\text{inertie} = \sum_{i=1}^n d(c, x_i) \quad (13.7)$$

avec c le centre du jeu de données, n le nombre d'observations, x une observation et d la fonction calculant la distance entre deux observations.

L'enjeu est de définir c dans un contexte où la distance euclidienne est utilisée. Il s'agit simplement d'une observation fictive dont les coordonnées sont les moyennes des différentes variables du jeu de données. Dans le cas d'autres distances, il peut s'agir de l'observation minimisant la distance à toutes les autres observations.

Pour bien visualiser la notion d'inertie, prenons une fois encore le jeu de données IRIS comme exemple. Admettons que nous ne nous intéressons qu'à deux variables de ce jeu de données : `sepal.Length` et `sepal.Width`. Nous pouvons représenter l'inertie totale du jeu de données à la figure 13.9.

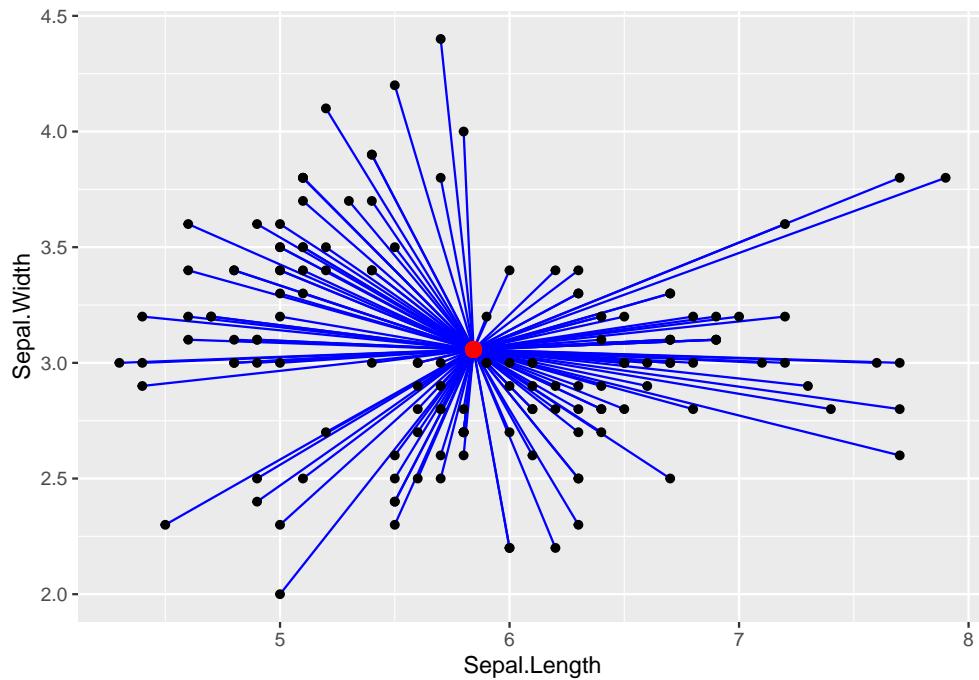


FIG. 13.9 : Représentation de l'inertie du jeu de données IRIS

Chaque ligne bleue représente la contribution de chaque point à l'inertie totale du jeu de données. Pour chaque iris, nous connaissons son espèce (Setosa, Versicolor ou Virginica). Nous pouvons donc attribuer chaque point de ce jeu de données à un groupe (une espèce dans notre cas). Il devient alors possible de calculer l'inertie de chacun des sous-groupes de notre jeu de données. Pour cela, nous devons calculer le centre de chaque groupe (généralement les moyennes des variables des observations au sein d'un groupe) et ensuite calculer l'inertie entre chaque observation et le centre de son groupe. Nous représentons cette situation à la figure 13.10.

Cette inertie propre aux groupes est toujours inférieure ou égale à l'inertie totale du jeu de données. Il s'agit en réalité de l'inertie que la structure de groupe n'est pas en mesure d'expliquer. En utilisant ces concepts, il est possible de calculer la part de l'inertie totale expliquée par les groupes (équation (13.8)) :

$$\text{inertie expliquée} = 1 - \frac{\text{inertie totale}}{\text{inertie restante}} \quad (13.8)$$

Cette valeur nous renseigne sur la capacité d'une classification à bien réduire l'inertie totale d'un jeu de données. Elle est comprise entre 0 et 1. Si l'inertie expliquée est à 0, c'est que la classification n'explique absolument aucune part de l'inertie totale. Si l'inertie expliquée est à 1, la classification utilisée explique l'intégralité de l'inertie, ce qui en pratique n'est atteignable que si le nombre de groupes de la classification est égal au nombre d'observations. En d'autres termes, chaque observation est attribuée à un groupe dont elle est la seule représentante. Un telle situation n'a aucun intérêt puisque l'objectif d'une classification est bien de réduire la complexité d'un jeu de données en regroupant les observations.

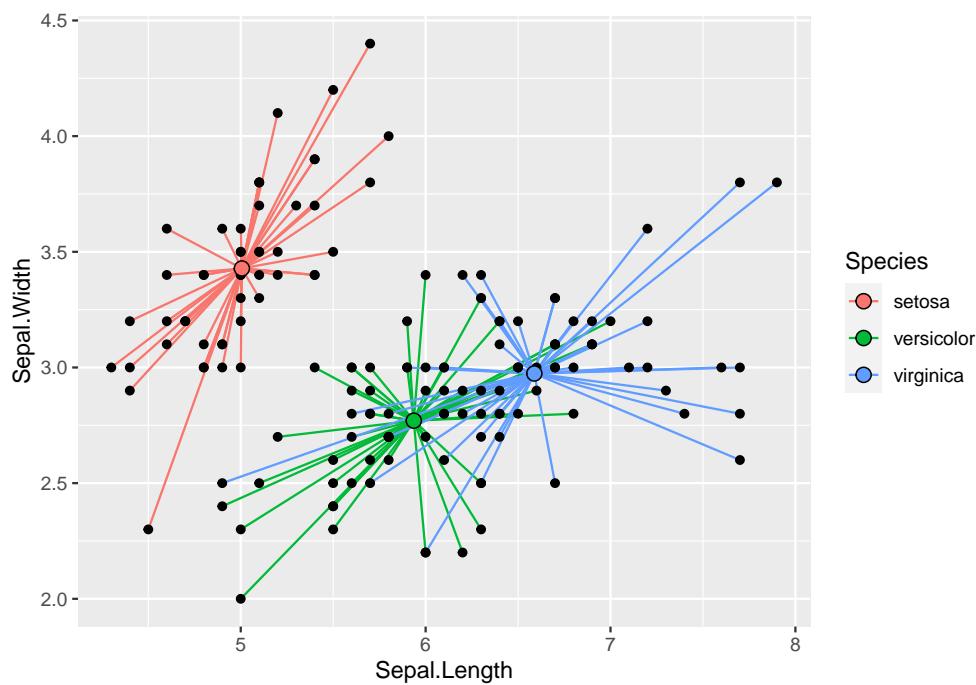


FIG. 13.10 : Représentation de l'inertie par groupe pour le jeu de données IRIS

13.3 Classification ascendante hiérarchique

La classification ascendante hiérarchique (CAH) est un algorithme de classification non supervisée dont l'objectif est de créer un arbre de classification des observations. Cet arbre est ensuite utilisé pour déterminer le nombre de groupes à former et à quel groupe appartient chaque observation.

13.3.1 Fonctionnement de l'algorithme

La classification ascendante hiérarchique est un algorithme permettant de regrouper les observations d'un jeu de données de façon itérative. À chaque itération, deux observations similaires sont agrégées en un groupe représenté par le point central entre les deux observations. Le processus est ensuite répété en considérant le nouveau point comme une observation jusqu'à ce que toutes les observations soient fusionnées en un seul groupe.

Ces regroupements successifs créent un arbre de classification appelé dendrogramme. La racine de cet arbre est le groupe unique fusionnant toutes les observations, et ses branches correspondent aux différentes agrégations effectuées jusqu'aux observations individuelles. Cet arbre peut être vu comme une hiérarchie de classification. Chaque niveau de l'arbre est un regroupement de plus en plus généraliste au fur et à mesure que nous nous approchons de sa racine.

Pour appliquer cette méthode, il est nécessaire de sélectionner une **fonction de distance** pour mesurer la dissimilarité ou la ressemblance entre deux observations. L'algorithme fonctionne avec n'importe quelle fonction de distance, ce qui permet de l'appliquer aussi bien à des données qualitatives que quantitatives. En effet, l'opération de regroupement des observations se base sur une matrice de distance, soit un tableau de taille $n \times n$ indiquant pour chaque paire d'observations leur degré de dissimilarité. La figure 13.11 illustre cette transformation en appliquant la distance du Φ^2 à un jeu de données comprenant cinq observations et 5 variables qualitatives.

En plus de la fonction de distance, il est également nécessaire de sélectionner un **critère d'agrégation**,

	couleur	jardin	garage	cheminee	cave
<i>maison 1</i>	blanc	non	oui	oui	non
<i>maison 2</i>	blanc	non	non	oui	non
<i>maison 3</i>	rouge	non	oui	non	non
<i>maison 4</i>	bleu	oui	oui	non	oui
<i>maison 5</i>	rouge	non	oui	non	non

	<i>maison 1</i>	<i>maison 2</i>	<i>maison 3</i>	<i>maison 4</i>	<i>maison 5</i>
<i>maison 1</i>	0	2.5	1.58	3.58	1.58
<i>maison 2</i>	2.5	0	1.58	3.58	1.58
<i>maison 3</i>	1.58	1.58	0	3.58	0
<i>maison 4</i>	3.58	3.58	3.58	0	1.58
<i>maison 5</i>	1.58	1.58	0	1.58	0

FIG. 13.11 : Du tableau de données à la matrice de distance

soit la règle permettant de décider à chaque itération quelles observations doivent être regroupées. Les méthodes les plus courantes sont :

- Le critère de Ward (1963) : cette méthode consiste à agréger à chaque itération les deux observations permettant de minimiser la variance (ou l'inertie) intra-groupe, ce qui revient à maximiser l'inertie inter-groupe (autrement dit, à rendre les groupes les plus homogènes possible et les plus dissemblables entre eux). Ainsi, l'enjeu est de fusionner les deux observations permettant d'avoir les groupes les plus dissimilaires possible après fusion.
- Le lien complet : à chaque itération, les deux groupes d'observations associés sont ceux pour lesquels la distance maximale entre les observations les composant est la plus petite parmi tous les groupes.
- Le lien simple : à chaque itération, les deux groupes d'observations associés sont ceux pour lesquels la distance minimum entre les observations les composant est la plus petite parmi tous les groupes.

La plus utilisée est de loin la méthode de Ward. La méthode du lien complet produit généralement des résultats similaires. En revanche, la méthode du lien simple peut produire des groupes non sphériques (non centrés sur leur moyenne) plus difficile à interpréter.

Prenons un instant pour visualiser cet algorithme (figure 13.12). Cette animation a été réalisée par David Sheehan et est également accessible sur son blog¹. Elle présente bien le processus d'agglomération de la classification ascendante hiérarchique et la construction progressive du dendrogramme.

¹<https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/>

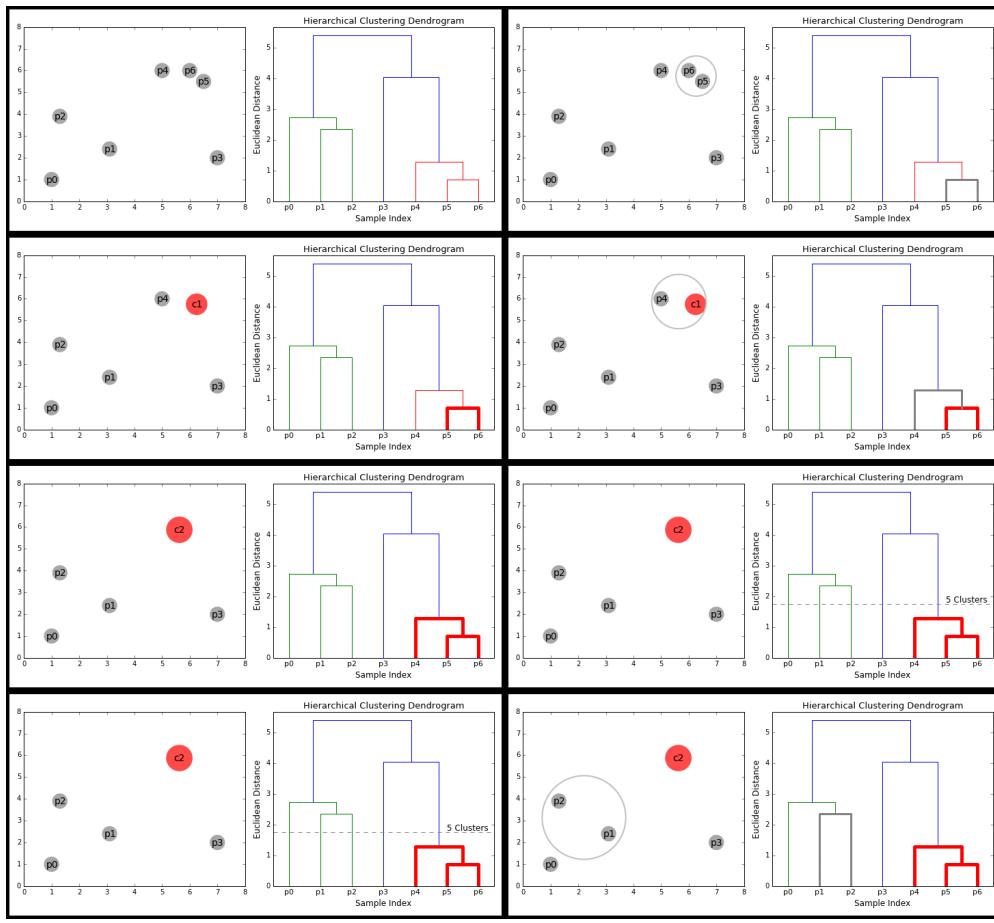


FIG. 13.12 : Principe de fonctionnement de la classification ascendante hiérarchique (auteur : David Sheehan)

13.3.2 Choisir le bon nombre de groupes

Une fois que l'algorithme a été appliqué aux données et le dendrogramme obtenu, il faut encore choisir le nombre optimal de groupes pour la classification finale. Chaque embranchement du dendrogramme constitue une classification possible, allant de la plus complexe (chaque observation appartient à un groupe formé d'elle seule) à la plus simple (toutes les observations appartiennent au même groupe). Si le nombre de groupes n'est pas connu à l'avance et qu'aucune forte justification théorique n'existe, il est possible d'utiliser plusieurs techniques pour déterminer un nombre de groupes judicieux à partir des données. Nous en présentons ici trois, mais il convient de ne pas s'en tenir uniquement à ses critères arbitraires. Il est important d'explorer les résultats de la classification obtenue pour plusieurs valeurs de k candidates et de tenir compte de la qualité des informations qu'elles fournissent. Au final, il est pertinent de retenir la classification dont les résultats offrent l'interprétation la plus claire avec un nombre de groupes réduit (principe de parcimonie).

13.3.2.1 Méthode du coude

Cette première approche est la plus simple à mettre en oeuvre. Il s'agit simplement de produire plusieurs classifications à partir du dendrogramme avec différentes valeurs de k (nombre de groupes) et de calculer à chaque fois la part de l'inertie expliquée. Chaque groupe supplémentaire ne peut qu'améliorer l'inertie expliquée, car pour rappel, si $k = n$, alors nous expliquons 100 % de l'inertie totale. L'objectif est de déterminer à quel moment l'ajout d'un groupe supplémentaire ne contribue que de façon marginale à

améliorer l'inertie expliquée. Si nous représentons les valeurs d'inertie expliquée pour les différentes valeurs de k dans un graphique, une rupture (un coude) indiquerait le point au-delà duquel les groupes supplémentaires ne captent finalement que du bruit et non plus de l'information.

Si nous reprenons l'exemple du jeu de données IRIS, nous pouvons créer ce graphique avec k allant de 2 à 8 (figure 13.13). Un premier coude très net est observable pour $k = 3$ et un second plus faible, mais tout de même marqué pour $k = 4$.

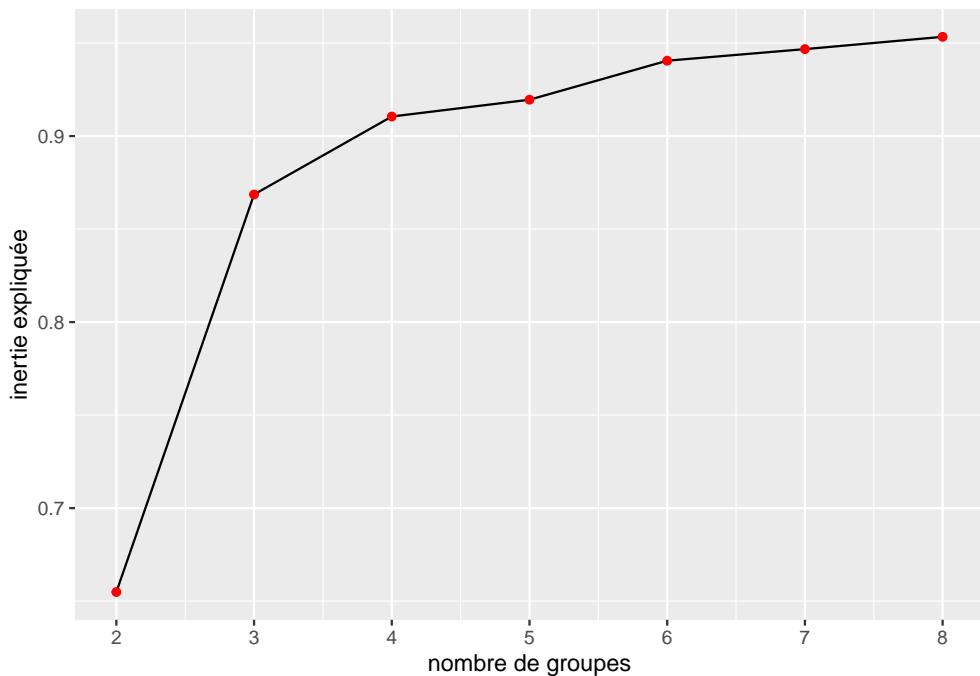


FIG. 13.13 : Méthode du coude



Inertie expliquée et centre de groupe

Pour calculer l'inertie expliquée, il est nécessaire de pouvoir déterminer pour le centre de gravité (ou centroïde) chaque groupe. Lorsque la distance euclidienne est utilisée, il s'agit simplement de calculer pour chaque groupe la valeur moyenne des différentes colonnes des observations. Cependant, lorsque d'autres distances sont utilisées, il peut être plus difficile de déterminer le centre d'un groupe. Avec la distance de Manhattan, il est par exemple recommandé d'utiliser la médiane des colonnes plutôt que la moyenne. Pour la distance de Hamming, la moyenne peut aussi être utilisée, car elle représente pour cette distance la fréquence d'occurrence des différentes modalités des variables qualitatives. Pour d'autres distances plus complexes, il est préférable de définir le centre d'un groupe comme le point de ce groupe minimisant les distances à tous les autres points du groupe. Il s'agit du médoïde du groupe.

13.3.2.2 Indicateur de silhouette

Si un coude net ne s'observe pas pour la méthode précédente, il est possible d'utiliser l'indicateur de silhouette. Il permet de mesurer pour une classification à quel point une observation est similaire à celles dans son propre groupe (cohésion) comparativement aux observations des autres groupes. Elle se calcule de la façon suivante :

$$\begin{aligned}
 s(i) &= \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \\
 a(i) &= \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j) \\
 b(i) &= \min_{i \neq j} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j)
 \end{aligned} \tag{13.9}$$

avec $s(i)$ la valeur de l'indice de silhouette pour l'observation i , $a(i)$ la distance moyenne entre l'observation i et son groupe C_i et $b(i)$ la distance minimale entre l'observation i et le centre de chaque autre groupe C_j .

La valeur totale de l'indice est simplement la moyenne des valeurs moyennes des indices de silhouette au sein de chaque groupe. Une valeur plus élevée indique une meilleure classification. Il est nécessaire de déterminer le centre des groupes pour calculer cet indicateur ce qui peut être un exercice difficile quand une distance autre que la distance euclidienne est utilisée. Référez-vous à la note de la section précédente pour plus d'informations. L'indice de silhouette semble indiquer que seulement trois groupes serait un choix optimal, soit la valeur la plus haute (figure 13.14).

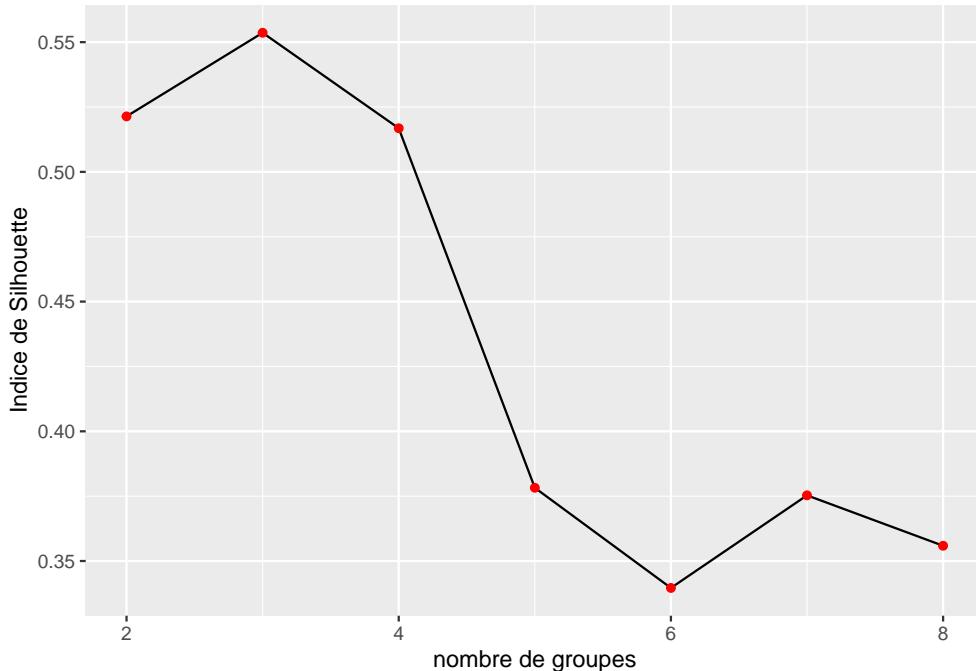


FIG. 13.14 : Méthode de l'indice de silhouette

13.3.2.3 Méthode GAP

Cette méthode, proposée par Tibshirani, Walther et Hastie (2001), consiste à comparer l'inertie intra-groupe (inexpliquée) avec l'inertie observée pour un jeu de données généré aléatoirement (distribution uniforme des valeurs entre le minimum et le maximum de chaque variable) pour différentes valeurs successives de k . Une fois ces calculs effectués, l'objectif est de trouver la valeur de k telle que la valeur de GAP à $k + 1$ n'est pas plus grande qu'un écart type pour GAP à $k + 1$.

La statistique GAP est calculée ainsi :

$$GAP(k) = \frac{1}{nsim} \sum_{sim=1}^{nsim} \log(W_{k sim}) - \log(W_k) \quad (13.10)$$

avec W_k l'inertie non expliquée (intra-groupe), $W_{k sim}$ l'inertie non expliquée (intra-groupe) obtenue pour un jeu de données simulé et k le nombre de groupes.

L'idée est qu'une bonne classification doit produire des résultats plus structurés que ce que nous pourrions attendre du hasard. Chaque groupe supplémentaire permet de réduire l'inertie, mais lorsque l'ajout d'un groupe ne permet pas un gain significatif comparativement au hasard, alors l'ajout de ce groupe ne se justifie pas. À nouveau, il est possible de visualiser la situation avec un simple graphique (figure 13.15). Selon cette méthode, il faudrait sélectionner quatre groupes, car il s'agit de la première valeur de k validant le critère de cette méthode. La seconde valeur retenue par cette méthode est 6.

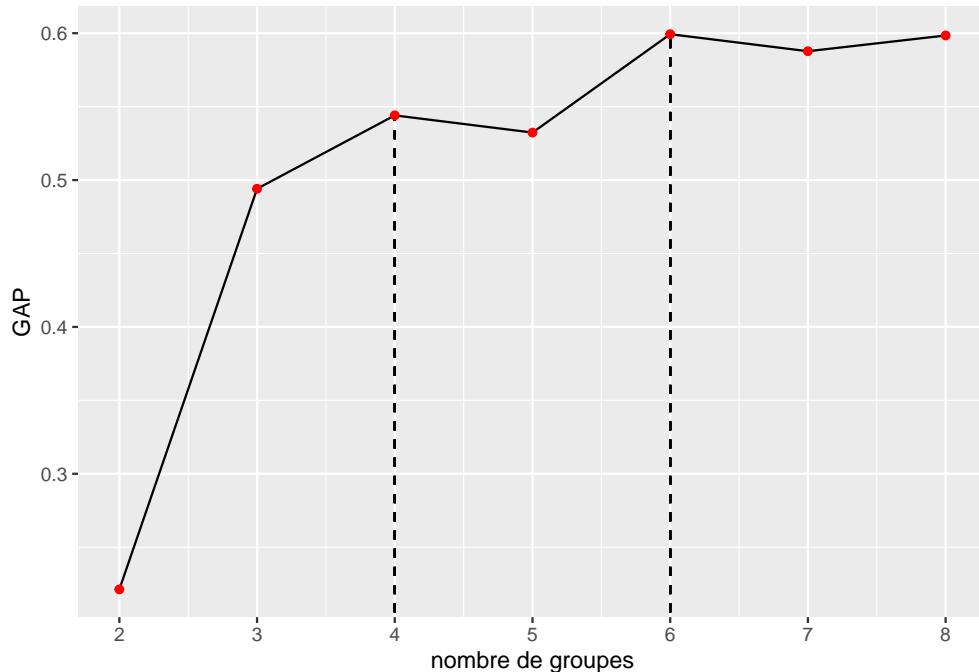


FIG. 13.15 : Méthode GAP

13.3.3 Limites de la classification ascendante hiérarchique

Bien que très flexible (choix de la fonction de distance et du critère d'agrégation), la CAH fait face à un enjeu majeur : la vitesse d'exécution et la consommation de mémoire lorsque des grands jeux de données sont utilisés. En effet, il est nécessaire de calculer à chaque étape une matrice de distance entre les groupes. Si un jeu de données comprend 1000 observations, cette matrice comprend donc 1000×1000 cases, soit un million de distances. Même en divisant ce nombre par deux (les éléments de la matrice sont symétriques, donc $d(ij) = d(ji)$), ce nombre augmente avec le carré du nombre d'observations. Pour de grands jeux de données, la CAH peut donc échouer à cause des limites de l'ordinateur utilisé. Il existe des versions plus performantes de l'algorithme réduisant cette limite, mais il convient de la garder en mémoire. Quand un très grand jeu de données doit être analysé, les méthodes des nuées dynamiques sont une solution à considérer.

13.3.4 Mise en oeuvre dans R

Nous proposons ici un exemple issu d'un article portant sur les parcs urbains de Montréal (Apparicio et al. 2010), dont l'objectif était notamment de classifier ces parcs en fonction de leur superficie et des équipements qu'ils comprennent, et ce, en utilisant la classification ascendante hiérarchique. Nous proposons ici de reproduire l'étape de classification effectuée dans cet article. La base de données comporte 653 parcs pour lesquels la présence de 18 équipements est codée comme un ensemble de variables binaires (0 signifiant absence et 1 présence). Nous disposons également de la taille de ces parcs, recodée en cinq catégories : moins d'un hectare, de 1 à 5 hectares, de 5 à 10 hectares, de 10 à 20 hectares et 20 hectares et plus. Le tableau 13.4 indique le nombre d'équipements recensés dans les parcs.

TAB. 13.4 : Équipements recensés dans les différents parcs de Montréal

Équipements	N
Équipements pour les 0 à 4 ans	
Aire de jeux	601
Pataugeoire	161
Jeux d'eau	28
Terrains de sport	
Baseball	188
Soccer (football)	169
Basketball	144
Tennis	125
Football	36
Volleyball	24
Athlétisme	20
Équipements d'hiver	
Patinoire extérieure	241
Glissade	30
Piste de ski de fond	14
Piste de raquette	9
Équipements spécialisés	
Parc de planches à roulettes	18
Patins à roues alignées	8
Autres équipements	
Piscine intérieure	92
Chemin de randonnée	15

Puisque notre jeu de données ne comporte que des variables qualitatives, nous utilisons la distance du Φ^2 pour construire notre matrice de distance entre les parcs. Notons que, dans l'article original, la distance euclidienne au carré avait été utilisée, alors nous n'obtiendrons probablement pas les mêmes résultats, car la distance du Φ^2 tient compte des fréquences d'occurrence des modalités des variables qualitatives.

13.3.4.1 Calcul de la matrice de distance

La première étape consiste donc à charger notre jeu de données et à calculer la matrice de distance.

```
# chargement du jeu de données et sélection des colonnes pour l'analyse
parcs <- read.csv("data/classification/Parcs.txt", header = TRUE, stringsAsFactors = FALSE)
X <- parcs[c(5:22, 27)]
```

Pour calculer la distance du Φ^2 , nous utilisons la fonction `dist` du package `proxy` avec le paramètre `method = "Phi-squared"`. Elle requiert que l'ensemble des variables catégorielles soient converties en variables binaires. Pour cela, nous pouvons utiliser la fonction `dummy_cols` du package `fastDummies`.

```
library(fastDummies)
library(proxy)

X <- dummy_cols(X,select_columns = "HaTypo",remove_selected_columns = TRUE)
parc_distances <- dist(as.matrix(X), method = "Phi-squared")
```

13.3.4.2 Application de l'algorithme de classification ascendante hiérarchique

Une fois la matrice obtenue, il ne reste plus qu'à appliquer la fonction `hclust` disponible de base dans R pour obtenir le dendrogramme. Comme dans l'article, nous utilisons le critère d'agrégation de Ward pour la création des groupes.

```
dendrogramme_parcs <- hclust(parc_distances, method = "ward.D")
```

Puisque nous n'utilisons pas la distance euclidienne, nous optons ici pour l'indice de silhouette pour déterminer le nombre adéquat de groupes à former. Nous testons toutes les valeurs comprises entre 2 et 10.

```
library(cluster)
ks <- 2:10

# Calcul des indices de silhouette pour les différentes valeurs de k
values <- sapply(ks, function(k){
  # découpage du dendrogramme
  groupes <- cutree(dendrogramme_parcs, k = k)
  # calcul des valeurs de silhouette
  sil <- silhouette(groupes, dist = parc_distances)
  # extraction de l'indice global (moyenne des moyennes)
  idx <- mean(summary(sil)$clus.avg.widths)
  return(idx)
})

# Création d'un graphique avec les résultats
df <- data.frame(k = ks, silhouette = values)
ggplot(df) +
  geom_line(aes(x = k, y = silhouette)) +
  geom_point(aes(x = k, y = silhouette), color = "red") +
  labs(x = "nombre de groupes", y="indice global de silhouette")
```

Si nous écartons d'emblée les résultats pour $k = 2$ et $k = 3$ (trop peu de groupes pour l'interprétation), nous constatons que la solution optimale selon ce critère est $k = 5$. Dans l'article original, la solution $k = 6$ avait été retenue en examinant le dendrogramme. Comparons les résultats pour $k = 5$ et $k = 6$.

```
resk5 <- cutree(dendrogramme_parcs, k = 5)
resk6 <- cutree(dendrogramme_parcs, k = 6)
sil5 <- silhouette(resk5, dist = parc_distances)
sil6 <- silhouette(resk6, dist = parc_distances)

# résumé pour l'indice de silhouette pour k = 5
summary(sil5)
```

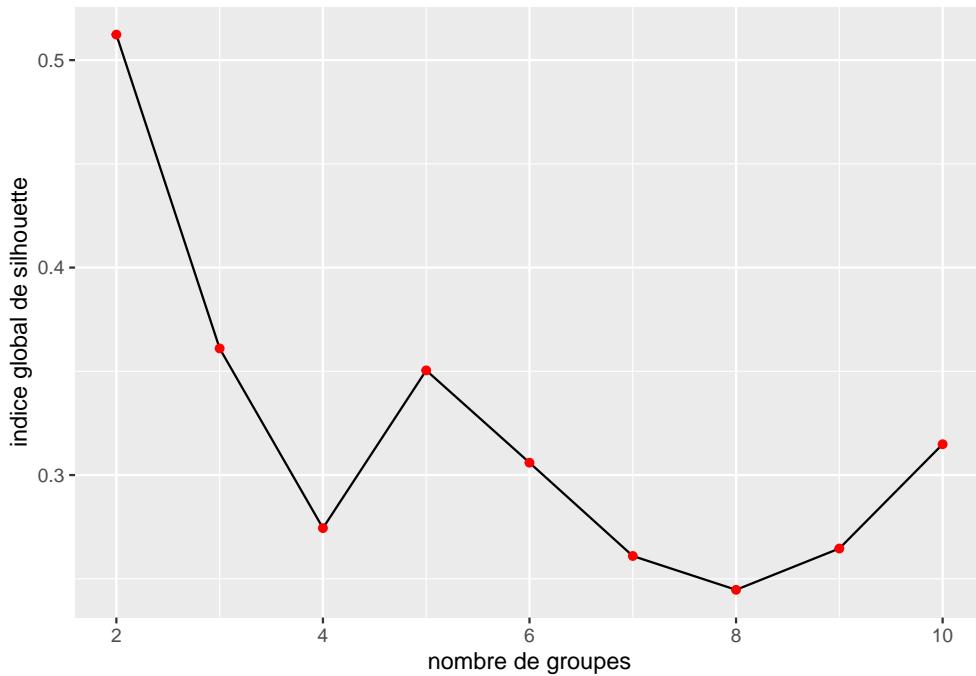


FIG. 13.16 : Valeur de l'indice de silhouette pour différents nombres de groupes

```
## Silhouette of 693 units in 5 clusters from silhouette.default(x = resk5, dist = parc_distances) :
## Cluster sizes and average silhouette widths:
##      116       212       246        84       35
##  0.07029553  1.00000000 -0.11827930 -0.19969707  1.00000000
## Individual silhouette widths:
##   Min. 1st Qu. Median Mean 3rd Qu. Max.
## -0.62041 -0.08502  0.09814  0.30200  1.00000  1.00000
```

```
# résumé pour l'indice de silhouette pour k = 6
summary(sil6)
```

```
## Silhouette of 693 units in 6 clusters from silhouette.default(x = resk6, dist = parc_distances) :
## Cluster sizes and average silhouette widths:
##      116       212       197        49       84       35
##  0.05906553  1.00000000 -0.10289391  0.07935325 -0.19969707  1.00000000
## Individual silhouette widths:
##   Min. 1st Qu. Median Mean 3rd Qu. Max.
## -0.62041 -0.06414  0.10998  0.31846  1.00000  1.00000
```

Nous constatons que le groupe supplémentaire vient séparer le groupe trois comprenant 246 parcs dans la solution avec $k = 5$. Ce dernier ne comprend plus que 197 parcs pour la solution $k = 6$ et le nouveau groupe en compte 49. Ce nouveau groupe à un indice de silhouette moyen relativement faible (0,079), et le fait de découper le groupe trois améliore très peu sa propre valeur (passant de -0,12 à -0,10). Nous retenons cependant ici la solution avec $k = 6$ afin de tenter de reproduire les résultats de l'article.

13.3.4.3 Interprétation des résultats

La dernière étape consiste à identifier les groupes obtenus et leur attribuer un intitulé en fonction de leurs caractéristiques. Dans notre cas, la classification ne comporte que des variables binaires, nous pouvons donc calculer le pourcentage de valeurs à 1 (présence d'un équipement) dans chacun des groupes.

```
# calcul du nombre de fois où chaque modalité est observée dans un groupe
X$groupe <- resk6
df_groupes <- X %>%
  group_by(groupe) %>% summarise_all(.funs = sum)

# calcul du nombre d'observations par groupe
nb_gp <- table(resk6)
groupe_ratios <- round(100 * as.matrix(df_groupes)[,2:ncol(df_groupes)] / as.vector(nb_gp),1)
groupe_ratios <- as.data.frame(t(groupe_ratios))
names(groupe_ratios) <- paste0("groupe ", 1:ncol(groupe_ratios))

# calcul du nombre moyen d'équipements par catégorie par parc
equip_class <- list(
  c("AIRE_JEUX", "JEUX_EAU", "PATAUGEOIRE"),
  c("ATHLETISME", "BASEBALL_S", "BASKETBALL", "FOOTBALL", "SOCCER", "TENNIS", "VOLLEY_BALL"),
  c("TOBOGAN_G", "PATINOIRE_E", "RAQUETTES", "SKI_FOND"),
  c("PATIN_ROUE", "ROULI_ROUL"),
  c("PISC_EXT", "RANDONNEE")
)

class_compte <- data.frame(sapply(equip_class, function(equip){
  rowSums(X[equip])
}))
names(class_compte) <- c("enfants", "terrain_sport", "hiver", "specialise", "autre")
class_compte$groupe <- resk6
df_class_equip <- class_compte %>%
  group_by(groupe) %>%
  summarise_all(mean)

df_class_equip <- t(df_class_equip[2:ncol(df_class_equip)])
colnames(df_class_equip) <- paste0("groupe ", 1:ncol(df_class_equip))

# comptage du nombre moyen total d'équipements
df_equip_tot <- data.frame(
  nb = rowSums(X[1:18]),
  groupe = resk6
)
df_equip_tot_mean <- df_equip_tot %>%
  group_by(groupe) %>%
  summarise_all(mean)

# mise dans l'ordre de la première partie du tableau
all_types <- do.call(c,equip_class)
idxs <- match(all_types, row.names(groupe_ratios[1:length(all_types),]))
groupe_ratios <- rbind(groupe_ratios[idxs,],
                        groupe_ratios[(length(all_types)+1):nrow(groupe_ratios),])

# combinaison des deux tableaux
groupe_ratios <- rbind(groupe_ratios, df_class_equip, df_equip_tot_mean$nb, as.integer(nb_gp))
```

Il est ensuite possible d'afficher le tableau obtenu pour l'interpréter. Les résultats sont ici rapportés au tableau 13.5.

- Le premier groupe correspond à de grands parcs (superficie généralement comprise entre 5 et plus de 20 hectares), il comporte 116 observations. Ces grands parcs sont en moyenne équipés de deux terrains de sport et d'un équipement d'hiver. Il s'agit vraisemblablement des grands parcs identifiés dans l'article original, dans lesquels se retrouvent également les parcs à vocation métropolitaine.
- Le second groupe (212 parcs) correspond à de très petits parcs (moins d'un hectare) comportant uniquement une aire de jeu.
- Le troisième groupe (197 parcs) correspond à de petits parcs (entre 1 et 5 hectares), souvent équipés d'une piscine extérieure (27,4 % des cas), et en moyenne de deux terrains de sports (essentiellement des terrains de tennis et de soccer). Ces parcs comprennent en moyenne plus de 4 équipements et doivent donc correspondre à la classe D dans l'article original (Petit parc (1 à 5 ha) avec en moyenne six équipements, dont une patinoire et une piscine).

TAB. 13.5 : Caractéristiques des groupes obtenus lors de la CAH

	groupe 1	groupe 2	groupe 3	groupe 4	groupe 5	groupe 6
Équipements pour les 0 à 4 ans (%)						
Aire de jeux	69,8	100	83,2	71,4	88,1	100
Jeux d'eau	7,8	0	2,5	18,4	6,0	0
Pataugeoire	36,2	0	47,2	2,0	29,8	0
Terrains de sport (%)						
Athlétisme	12,1	0	2,0	2,0	1,2	0
Baseball	62,1	0	50,8	0,0	19,0	0
Basketball	37,9	0	36,0	16,3	21,4	0
Football américain	15,5	0	7,1	8,2	0,0	0
Soccer (football)	52,6	0	29,9	87,8	7,1	0
Tennis	38,8	0	32,5	8,2	14,3	0
Volleyball	7,8	0	4,6	12,2	0,0	0
Équipements d'hiver (%)						
Glissade	19,8	0	3,0	2,0	0,0	0
Patinoire	58,6	0	59,4	34,7	46,4	0
Piste de ski de fond	7,8	0	0,0	0,0	0,0	0
Raquettes	12,1	0	0,0	0,0	0,0	0
Équipements spécialisés (%)						
Parc de planches à roulettes	6,0	0	0,5	0,0	0,0	0
Patins à roues alignées	8,6	0	4,1	0,0	0,0	0
Autres équipements (%)						
Piscine extérieure	27,6	0	27,4	4,1	4,8	0
Chemin de randonnée	12,9	0	0,0	0,0	0,0	0
Superficie (%)						
Moins d'un hectare	0,0	100	0,0	0,0	100,0	0
1 à 5 hectares	5,2	0	98,5	100,0	0,0	100
5 à 10 hectares	61,2	0	0,0	0,0	0,0	0
10 à 20 hectares	17,2	0	1,0	0,0	0,0	0
20 hectares et plus	16,4	0	0,5	0,0	0,0	0
Nombre moyen d'équipement du type						
Équipements pour les 0 à 4 ans	1,1	1	1,3	0,9	1,2	1
Terrains de sport	2,3	0	1,6	1,3	0,6	0
Équipements d'hiver	1,0	0	0,6	0,4	0,5	0
Équipements spécialisés	0,1	0	0,0	0,0	0,0	0
Autres équipements	0,4	0	0,3	0,0	0,0	0
Tous les équipements	4,9	1	3,9	2,7	2,4	1
Nombre d'observations par groupe						
	116,0	212	197,0	49,0	84,0	35

- Le quatrième groupe (49 parcs) comprend de petits parcs (entre 1 et 5 hectares) qui ressemblent aux parcs du groupe 2 mais tendent à disposer en plus d'un terrain de sport (baseball ou basketball).
- Le quatrième groupe (84 parcs) correspond à de petits parcs, il est caractérisé par une présence plus marquée de pataugeoires (39 %).
- Le cinquième groupe (35 parcs) est très similaire au second groupe (uniquement une aire de jeux), excepté sont les parcs qui s'y trouvent sont de taille supérieure (de 1 à 5 hectares).

Considérant les différences minimes entre certains des groupes que nous avons obtenus, il est clair que retenir seulement trois ou cinq groupes serait préférable. Notez également l'importance du choix de la distance, car nous obtenons des résultats sensiblement différents de ceux de l'article original en ayant opté pour la distance du Φ^2 plutôt que la distance euclidienne au carré.

13.3.4.4 Utilisation de la matrice de distance euclidienne au carré

Pour obtenir des résultats plus proches de ceux de l'article original, nous pouvons reprendre notre analyse et utiliser cette fois-ci une distance euclidienne au carré.

```
X$groupe <- NULL
# calcule de la matrice de distance
parc_distances_euc <- dist(as.matrix(X), method = "Euclidean")**2

# Application de la CAH
dendrogramme_parcs_euc <- hclust(parc_distances_euc, method = "ward.D")

# calcul de l'indice de silhouette
ks <- 2:10
values <- sapply(ks, function(k){
  # découpage du dendrogramme
  groupes <- cutree(dendrogramme_parcs_euc, k = k)
  # calcul des valeurs de silhouette
  sil <- silhouette(groupes, dist = parc_distances_euc)
  # extraction de l'indice global (moyenne des moyennes)
  idx <- mean(summary(sil)$clus.avg.widths)
  return(idx)
})

# création d'un graphique avec les résultats

df <- data.frame(
  k = ks,
  silhouette = values
)

ggplot(df) +
  geom_line(aes(x = k, y = silhouette)) +
  geom_point(aes(x = k, y = silhouette), color = "red") +
  labs(x = "nombre de groupes", y="indice global de silhouette")
```

Nous constatons cette fois-ci, que quatre groupes serait probablement le meilleur choix et qu'au-delà de ce nombre, l'indice global de silhouette ne fait que diminuer. Tentons cependant de reproduire les résultats de l'article avec $k = 6$.

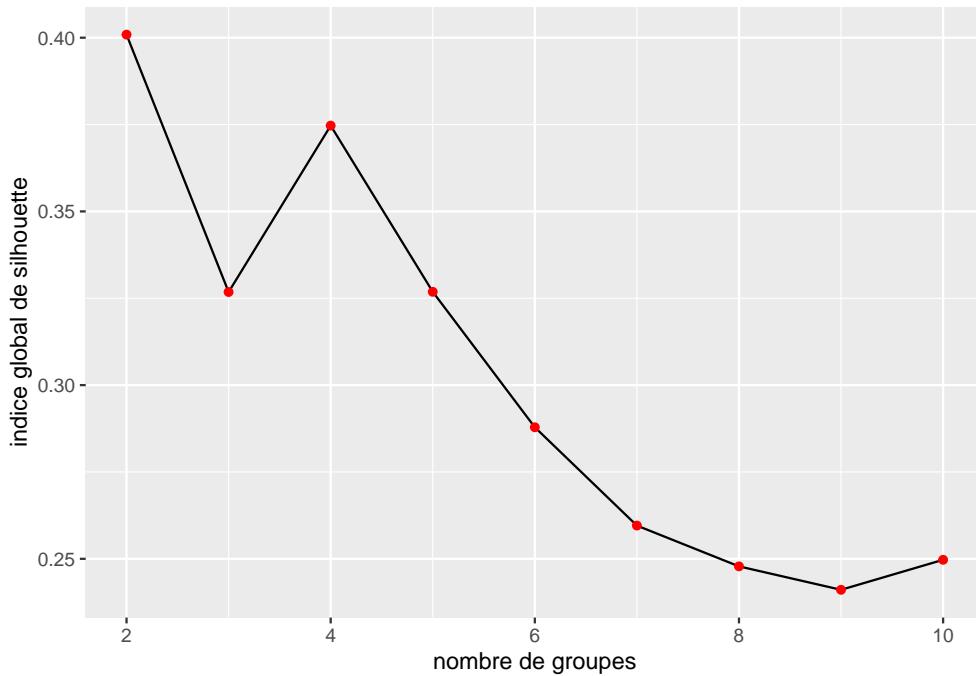


FIG. 13.17 : Valeur de l'indice de silhouette pour différents nombres de groupes (distance euclidienne au carré)

```

resk6 <- cutree(dendrogramme_parcs_euc, k = 6)

# calcul du nombre de fois où chaque modalité est observée dans un groupe
X$groupe <- resk6
df_groupes <- X %>%
  group_by(groupe) %>% summarise_all(.funs = sum)

# calcul du nombre d'observations par groupe
nb_gp <- table(resk6)

groupe_ratios <- round(100 * as.matrix(df_groupes)[,2:ncol(df_groupes)] / as.vector(nb_gp),1)
groupe_ratios <- as.data.frame(t(groupe_ratios))
names(groupe_ratios) <- paste0("groupe ", 1:ncol(groupe_ratios))

# calcul du nombre moyen d'équipements par catégorie par parc
equip_class <- list(
  c("AIRE_JEUX", "JEUX_EAU", "PATAUGEOIRE"),
  c("ATHLETISME", "BASEBALL_S", "BASKETBALL", "FOOTBALL", "SOCCER", "TENNIS", "VOLLEY_BALL"),
  c("TOBOBOGAN_G", "PATINOIRE_E", "RAQUETTES", "SKI_FOND"),
  c("PATIN_ROUE", "ROULI_ROUL"),
  c("PISC_EXT", "Randonnee")
)

class_compte <- data.frame(sapply(equip_class, function(equip){
  rowSums(X[equip])
}()))
names(class_compte) <- c("enfants", "terrain_sport", "hiver", "specialise", "autre")
class_compte$groupe <- resk6

```

```

df_class_equip <- class_compte %>%
  group_by(groupe) %>%
  summarise_all(mean)

df_class_equip <- t(df_class_equip[2:ncol(df_class_equip)])
colnames(df_class_equip) <- paste0("groupe ", 1:ncol(df_class_equip))

# comptage du nombre moyen d'équipements
df_equip_tot <- data.frame(
  nb = rowSums(X[1:18]),
  groupe = resk6
)
df_equip_tot_mean <- df_equip_tot %>%
  group_by(groupe) %>%
  summarize_all(mean)

# mise dans l'ordre de la première partie du tableau
all_types <- do.call(c, equip_class)
idxs <- match(all_types, row.names(groupe_ratios[1:length(all_types),]))
groupe_ratios <- rbind(groupe_ratios[idxs,],
                        groupe_ratios[(length(all_types)+1):nrow(groupe_ratios),])

# combinaison des deux tableaux
groupe_ratios <- rbind(groupe_ratios, df_class_equip, df_equip_tot_mean$nb, as.integer(nb_gp))

```

Recréons le tableau final des résultats au tableau 13.6. Si vous comparez ce tableau avec celui de l'article original, vous verrez que notre groupe 3 correspond exactement à la classe A et que notre groupe 5 correspond exactement à la classe F. Pour les autres groupes, nous pouvons observer de légères variations, ce qui correspond vraisemblablement à des divergences d'implémentation des algorithmes entre le logiciel utilisé pour l'article (SAS) et R.

TAB. 13.6 : Caractéristiques des groupes obtenus lors de la CAH (distance euclidienne au carré)

	groupe 1	groupe 2	groupe 3	groupe 4	groupe 5	groupe 6
Équipements pour les 0 à 4 ans (%)						
Aire de jeux	79,6	74,2	96,6	100,0	20,0	79,3
Jeux d'eau	11,1	3,0	1,7	5,1	0,0	5,9
Pataugeoire	59,3	42,4	8,4	61,0	0,0	19,7
Terrains de sport (%)						
Athlétisme	13,0	15,2	0,3	0,0	0,0	1,0
Baseball	88,9	63,6	5,4	89,8	6,7	13,8
Basketball	83,3	30,3	6,1	35,6	0,0	18,2
Football	31,5	12,1	0,0	10,2	0,0	2,5
Soccer (football)	75,9	57,6	2,0	27,1	0,0	33,5
Tennis	90,7	19,7	4,1	35,6	0,0	14,8
Volleyball	20,4	3,0	0,0	1,7	0,0	4,9
Équipements d'hiver (%)						
Glissade	14,8	16,7	0,0	1,7	33,3	2,5
Patinoire	87,0	57,6	13,2	86,4	26,7	30,5
Piste de ski de fond	1,9	1,5	0,0	0,0	46,7	0,0
Raquettes	0,0	1,5	0,0	0,0	86,7	0,0
Équipements spécialisés (%)						
Parc de planches à roulettes	0,0	0,0	0,0	1,7	46,7	0,0
Patins à roues alignées	16,7	7,6	0,0	5,1	0,0	0,5
Autres équipements (%)						
Piscine extérieure	75,9	16,7	1,4	11,9	6,7	13,8
Chemin de andonnée	1,9	0,0	0,0	0,0	93,3	0,0
Superficie (%)						
Moins d'un hectare	0,0	0,0	100,0	0,0	0,0	0,0
1 à 5 hectares	42,6	1,5	0,0	98,3	0,0	99,5
5 à 10 hectares	46,3	69,7	0,0	0,0	0,0	0,0
10 à 20 hectares	1,9	27,3	0,0	0,0	13,3	0,5
20 hectares et plus	9,3	1,5	0,0	1,7	86,7	0,0
Nombre moyen d'équipement du type						
Équipements pour les 0 à 4 ans	1,5	1,2	1,1	1,7	0,2	1,0
Terrains de sport	4,0	2,0	0,2	2,0	0,1	0,9
Équipements d'hiver	1,0	0,8	0,1	0,9	1,9	0,3
Équipements spécialisés	0,2	0,1	0,0	0,1	0,5	0,0
Autres équipements	0,8	0,2	0,0	0,1	1,0	0,1
Tous les équipements	7,5	4,2	1,4	4,7	3,7	2,4
Nombre d'observations par groupe						
	54,0	66,0	296,0	59,0	15,0	203,0

13.4 Nuées dynamiques

Les méthodes des nuées dynamiques regroupent plusieurs algorithmes, tous plus ou moins liés avec l'algorithme le plus connu : *k-means*, originellement proposé par James MacQueen (1967). Nous présentons également ici plusieurs variantes du *k-means*, soit le *k-medians*, le *k-medoids*, le *c-means* et le *c-medians*.

13.4.1 *K-means*

13.4.1.1 Fonctionnement de l'algorithme

Nous commençons ici par détailler le fonctionnement de cet algorithme afin de mieux le cerner. D'emblée, cet algorithme nécessite que certains éléments soient définis d'avance :

- Une matrice de données X comportant n lignes (nombre d'observations) et p colonnes (nombre de variables). Chaque variable de cette matrice doit être quantitative et continue et de préférence dans une échelle standardisée (par exemple des variables centrées réduites).
- Le nombre de groupes à identifier k doit être choisi par l'utilisateur ou l'utilisatrice.

- La distance d à utiliser entre les observations.

Le fonctionnement classique du *k-means* est le suivant :

1. Définir k centres de groupes de façon aléatoire.
2. Déterminer pour chaque observation le centre de son groupe le plus proche en utilisant la fonction de distance.
3. Pour chacun des groupes ainsi formés, recalculer le centre du groupe en calculant le centroïde (moyennes le plus souvent) des observations appartenant à ce groupe.
4. Répéter l'opération 2 avec les nouveaux centres.
5. Calculer l'inertie expliquée par la nouvelle classification.
6. Comparer cette inertie expliquée avec celle obtenue lors de l'itération précédente.
7. Si la variation entre les deux valeurs est supérieure à une certaine limite, reprendre à l'étape 2, sinon, l'algorithme prend fin.

Ainsi, l'algorithme *k-means* part d'une première classification obtenue aléatoirement et la raffine jusqu'au point où l'amélioration de la classification devient négligeable. Du fait de ce point de départ aléatoire, cet algorithme est dit heuristique, car deux exécutions risquent de ne pas donner exactement le même résultat. Par conséquent, en relâchant l'algorithme, vous pourriez obtenir des résultats légèrement différents, avec par exemple des groupes similaires, mais obtenus dans un autre ordre, le groupe 1 étant devenu le groupe 3 et vice-versa. Il est aussi possible d'obtenir des résultats radicalement différents d'une tentative à l'autre, ce qui signifie que les groupes formés sont très instables et ne sont pas représentatifs de la population étudiée.



Réplicabilité des résultats dans R

Lorsqu'une méthode heuristique ou faisant appel au hasard est utilisée dans R, il est nécessaire de s'assurer que les résultats sont reproductibles. Cela permet notamment de relancer le même code et de réobtenir exactement les mêmes résultats : l'idée étant de figer le hasard.

Ultimement, un programme informatique est incapable de générer un résultat véritablement aléatoire, car il ne fait que suivre une suite d'opérations prédéterminées. Pour générer des résultats qui ressemblent au hasard, des algorithmes ont été proposés, partant d'une configuration initiale et appliquant une série d'opérations complexes permettant de générer des nombres semblant se distribuer aléatoirement. Si nous connaissons le point de départ de la suite d'opérations et que nous réappliquons ces dernières, alors nous sommes certains d'obtenir le même résultat. Il est possible, dans R, de définir un *état initial de hasard* à l'aide de la fonction `set.seed`. Avec ce point de départ défini, nous sommes certains d'obtenir les mêmes résultats en relâchant les mêmes opérations.

Prenons un exemple concret en sélectionnant aléatoirement 3 chiffres dans un vecteur allant de 1 à 10.

```
vec <- 1:10

# prenons un premier échantillon
sample(vec, size = 3)

## [1] 4 9 3

# et un second échantillon
sample(vec, size = 3)

## [1] 6 10 8
```

Nous obtenons bien deux échantillons différents. Recommençons en utilisant la fonction `set.seed` pour obtenir cette fois-ci des résultats identiques.

```

vec <- 1:10

# prenons un premier échantillon
set.seed(123)
sample(vec, size = 3)

## [1] 3 10 2

# et un second échantillon
set.seed(123)
sample(vec, size = 3)

## [1] 3 10 2

# prenons un troisième échantillon
set.seed(4568997)
sample(vec, size = 3)

## [1] 5 6 7

# et un quatrième échantillon
set.seed(4568997)
sample(vec, size = 3)

## [1] 5 6 7

```

Vous constatez que nous utilisons cette fonction plusieurs fois au cours de cette section. Elle nous permet de nous assurer que les résultats obtenus ne changent pas entre le moment où nous écrivons le livre et le moment où nous le formatons. Sinon, le texte pourrait ne plus être en phase avec les images ou les tableaux.

Pour mieux comprendre le fonctionnement du *k-means*, nous proposons ici une visualisation de ses différentes itérations.

Nous pouvons constater que, pour ce jeu de données relativement simple, l'algorithme converge très rapidement et que sa solution varie peu au-delà de la troisième itération. Si vous utilisez la version HTML du livre, la figure 13.18 devrait être animée et illustrer pourquoi le *k-means* est aussi appelé algorithme de nuées dynamiques.

Centre de groupe et *k-means*

À nouveau, puisque chaque itération du *k-means* nécessite de recalculer les centres des groupes formés, des problèmes peuvent être rencontrés avec certains types de distance. C'est pourquoi il est recommandé d'utiliser la distance euclidienne avec le *k-means* original. Si des distances plus complexes doivent être utilisées, il est préférable d'utiliser la classification ascendante hiérarchique.

13.4.1.2 Choix du nombre optimal de groupes

Comme pour la CAH, le principal enjeu avec le *k-means* est de déterminer le nombre idéal de groupes pour effectuer la classification. Si ce nombre n'est pas connu à l'avance et qu'aucune forte justification théorique n'existe, il est possible d'utiliser les mêmes techniques que pour la CAH, soit la méthode du coude, l'indicateur de silhouette ou la méthode GAP.

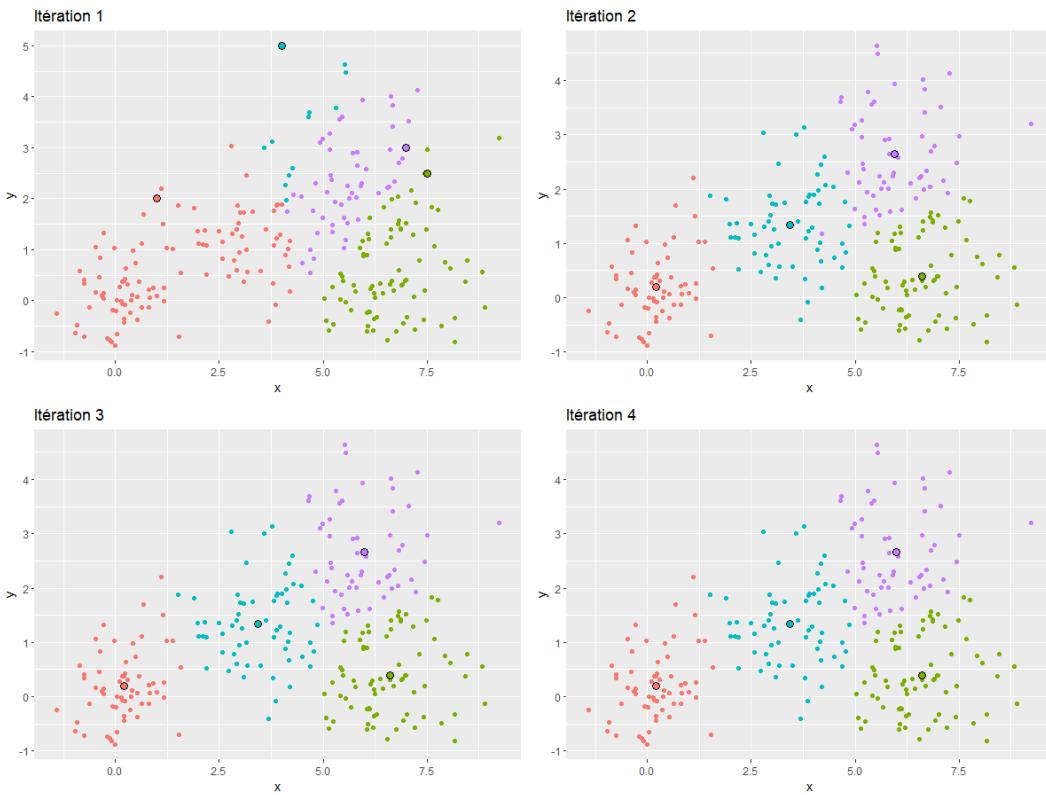


FIG. 13.18 : Classifications stricte et floue

13.4.2 K-médianes

Le *k-medians* est une variante du *k-means*. Contrairement au *k-means* privilégiant la distance euclidienne, le *k-medians* est à utiliser en priorité avec une distance de Manhattan. En effet, le centre d'un groupe n'est pas déterminé comme la moyenne des variables des observations appartenant à ce groupe (*k-means*), mais comme la médiane pour chaque variable (*k-medians*). En dehors de ces deux spécificités, il reprend le fonctionnement décrit plus haut pour le *k-means*. Il est particulièrement pertinent de l'utiliser quand un jeu de données comprend un très grand nombre de colonnes, car dans ce contexte, la distance euclidienne peine à représenter les différences entre les observations. De plus, l'utilisation de la médiane le rend moins sensible aux valeurs extrêmes.

13.4.3 K-médoïds

Le *k-médoïds* est également une variante du *k-means*. Le *k-means* crée des groupes en cherchant les centres de ces groupes dans l'espace multidimensionnel des données. Ces centres de groupes peuvent très bien ne pas correspondre à un point du jeu de données, au même titre que la moyenne d'une variable ne coïncide que rarement avec une observation réelle de cette variable. Pour le *k-médoïds*, les groupes sont formés en cherchant les centres de ces groupes **parmi** les observations du jeu de données. Ainsi, chaque groupe est centré sur une observation réelle, la plus similaire à l'ensemble des observations du groupe.

L'algorithme effectue les opérations suivantes :

1. Sélectionner aléatoirement k observations du jeu de données, elles constituent les centres des groupes initiaux.
2. Attribuer chaque observation au centre du groupe le plus proche.
3. Tant que la nouvelle solution est plus efficace, effectuer les opérations suivantes :

- pour chaque centre m et pour chaque observation o ,
 - considérer l'inversion de m et o
 - si cette permutation est meilleure que les précédentes, la conserver en mémoire
- effectuer la meilleure permutation retenue si elle améliore la classification, sinon l'algorithme prend fin.

Le *k-médoïds* est moins utilisé que le *k-means*, mais il est plus performant quand des distances autres que la distance euclidienne sont utilisées ou encore que des valeurs aberrantes/extrêmes sont présentes dans les données.

13.4.4 Mise en oeuvre dans R

Pour cet exemple, nous proposons d'utiliser le jeu de données spatiales `LyonIris` du package `geocmeans`. Ce jeu de données spatiales pour l'agglomération lyonnaise (France) comprend dix variables, dont quatre environnementales (EN) et six socioéconomiques (SE), pour les îlots regroupés pour l'information statistique (IRIS) de l'agglomération lyonnaise (tableau 13.7 et figure 12.4). Nous proposons de réaliser une analyse similaire à celle de l'article de Gelb et Apparicio (2021b), soit de classer les IRIS de Lyon selon ces caractéristiques pour déterminer si certains groupes d'IRIS combinent des situations désavantageuses sur les plans sociaux et environnementaux, dans une perspective d'équité environnementale.

Tab. 13.7 : Statistiques descriptives du jeu de données `LyonIris`

Nom	Intitulé	Type	Moy.	E.-T.	Min.	Max.
Lden	Bruit routier (Lden dB(A))	EN	55,6	4,9	33,9	71,7
NO2	Dioxyde d'azote (ug/m ³)	EN	28,7	7,9	12,0	60,2
PM25	Particules fines (PM _{2,5})	EN	16,8	2,1	11,3	21,9
VegHautPrt	Canopée (%)	EN	18,7	10,1	1,7	53,8
Pct_14	Moins de 15 ans (%)	SE	18,5	5,7	0,0	54,0
Pct_65	65 ans et plus (%)	SE	16,2	5,9	0,0	45,1
Pct_Img	Immigrants (%)	SE	14,5	9,1	0,0	59,8
TxChom1564	Taux de chômage	SE	14,8	8,1	0,0	98,8
Pct_brevet	Personnes à faible scolarité (%)	SE	23,5	12,6	0,0	100,0
NivVieMed	Médiane du niveau de vie (Euros)	SE	21 804,5	4 922,5	11 324,0	38 707,0

13.4.4.1 Préparation des données

La première étape consiste donc à charger les données et à les préparer pour l'analyse. Toutes les variables que nous utilisons sont des variables continues. Cependant, elles ne sont pas exprimées dans la même échelle, nous proposons donc de les standardiser ici en les centrant (moyenne = 0) et en les réduisant (écart-type = 1). Cette opération peut être effectuée simplement dans R en utilisant la fonction `scale`.

```
# Chargement des données
library(geocmeans)
data(LyonIris)

# NB : LyonIris est un objet spatial, il faut donc extraire uniquement son DataFrame
X <- LyonIris@data[c("Lden", "NO2", "PM25", "VegHautPrt", "Pct0_14", "Pct_65", "Pct_Img",
                     "TxChom1564", "Pct_brevet", "NivVieMed")]

# Centrage et réduction de chaque colonne du DataFrame
for (col in names(X)){
  X[[col]] <- scale(X[[col]], center = TRUE, scale = TRUE)
}
```

13.4.4.2 Choix du nombre de groupes optimal

La seconde étape consiste à déterminer le nombre de groupes optimal. Pour cela, nous comparons les résultats des trois méthodes proposées : la méthode du coude, l'indice de silhouette et la méthode GAP. Pour chaque méthode, nous testons les nombres de groupes de 2 à 10.

13.4.4.2.1 Méthode du coude

Commençons par appliquer la méthode du coude. Nous calculons donc l'inertie expliquée par la classification pour différentes valeurs de k (nombre de groupes) avant de construire la figure 13.19.

```
ks <- 2:10

## ---- Méthode du coude ---- ##
inertie_exps <- sapply(ks, function(k){
  # calcul du kmeans avec k
  resultat <- kmeans(X, centers = k)
  # calcul de l'inertie expliquée (1 - inertie intragroupe / inertie totale)
  inertie_exp <- 1-(sum(resultat$withinss) / resultat$totss)
  return(inertie_exp)
})

df <- data.frame(
  k = ks,
  inertie_exp = inertie_exps
)

ggplot(df) +
  geom_line(aes(x = k, y = inertie_exp)) +
  geom_point(aes(x = k, y = inertie_exp), color = "red") +
  labs(x = "nombre de groupes", y = "inertie expliquée (%)")
```

Dans l'article original, quatre groupes avaient été retenus. Nous pouvons constater ici qu'un coude fort se situe à $k = 3$ et qu'au-delà de cette limite, l'ajout d'un groupe supplémentaire contribue à expliquer une plus petite partie de l'inertie supplémentaire comparativement au précédent.

13.4.4.2.2 Indice de silhouette

Poursuivons avec l'indice de silhouette calculé de nouveau avec des valeurs de k allant de 2 à 10. Notez que nous devons au préalable créer une matrice de distances entre les observations du jeu de données pour construire notre indice de silhouette. Puisque nous utilisons l'algorithme *k-means*, nous utilisons la distance euclidienne.

```
ks <- 2:10

# calcul d'une matrice de distance euclidienne entre les observations
dist_mat <- dist(X, method = "euclidean")

## ---- indice de silhouette ---- ##
values <- sapply(ks, function(k){
  resultat <- kmeans(X, centers = k)
  groupes <- resultat$cluster
  # calcul des valeurs de silhouette
```

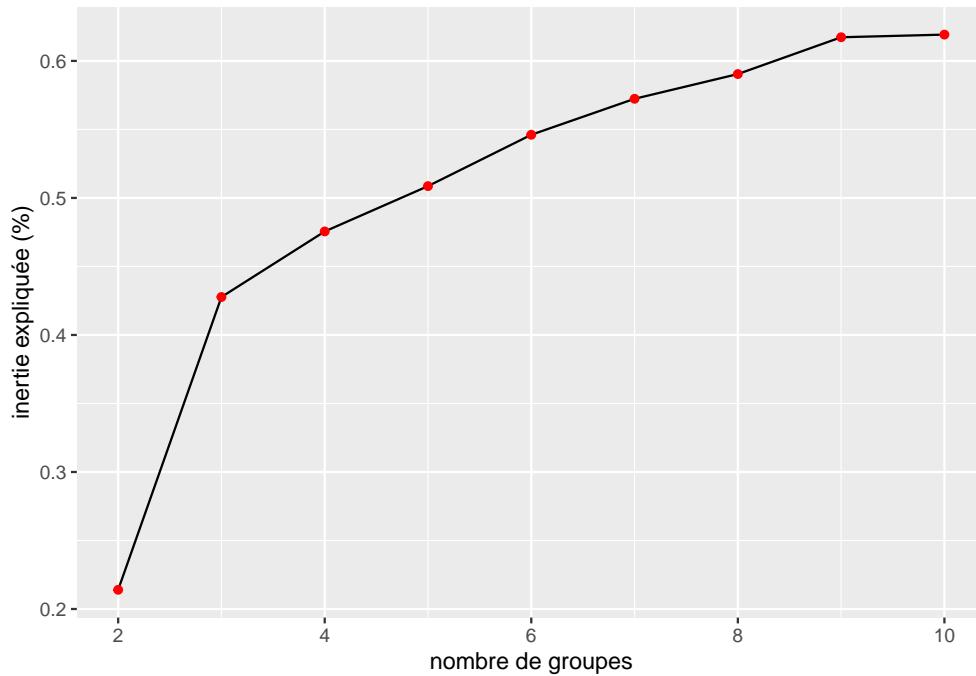


FIG. 13.19 : Inertie expliquée pour différents nombres de groupes pour le k-means

```

sil <- silhouette(groupes, dist = dist_mat)
# extraction de l'indice global (moyenne des moyennes)
idx <- mean(summary(sil)$clus.avg.widths)
return(idx)

})

df <- data.frame(
  k = ks,
  silhouette = values
)

ggplot(df) +
  geom_line(aes(x = k, y = silhouette)) +
  geom_point(aes(x = k, y = silhouette), color = "red") +
  labs(x = "nombre de groupes", y = "Indice de silhouette")

```

À nouveau, la figure 13.20 indique que le nombre de groupes optimal est trois selon l'indice de silhouette.

13.4.4.2.3 Méthode GAP

Pour appliquer la méthode GAP, nous proposons d'utiliser la fonction `clusGap` du package `NbClust`. Pour l'utiliser, il est nécessaire de définir une fonction renvoyant pour le nombre de groupes k et le jeu de données x une liste comprenant un vecteur attribuant chaque observation à chaque groupe. Il est possible de considérer ce type de fonction comme un « adaptateur ».

```

library(NbClust)

# définition de la fonction adaptateur

```

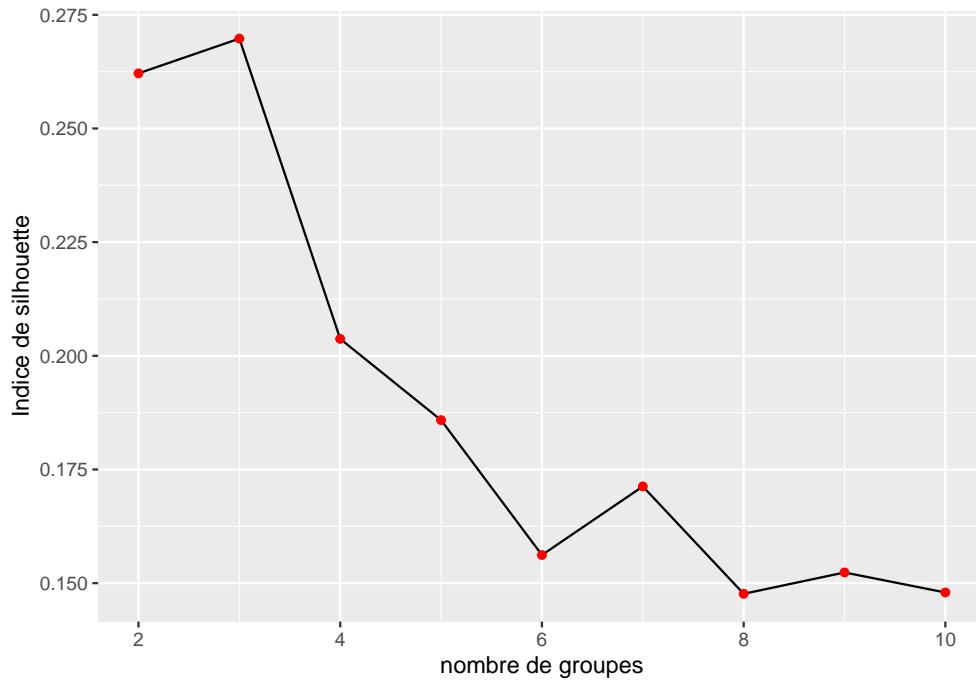


FIG. 13.20 : Indice de silhouette pour différents nombres de groupes pour le k-means

```

adaptor <- function(x,k){
  clust <- kmeans(x,k)
  return(list(
    "cluster" = clust$cluster
  ))
}

# calcul de la méthode GAP
vals <- clusGap(X, adaptor, K.max = 10, verbose = FALSE)
tab <- data.frame(vals$Tab)
tab$k <- 1:nrow(tab)

# détermination des valeurs de k retenues par la méthode (1ere et 2e)
is_valid <- sapply(2:nrow(tab), function(i){
  tab[i-1,"gap"] >= (tab[i,"gap"] - tab[i,"SE.sim"])
})
valids <- subset(tab,is_valid)[1,]
valids2 <- subset(tab,is_valid)[2,]

# réalisation du graphique
ggplot(tab) +
  geom_line(aes(x = k, y = gap)) +
  geom_segment(x = valids$k, xend = valids$k, y = min(tab$gap), yend = valids$gap,
               linetype = "dashed") +
  geom_segment(x = valids2$k, xend = valids2$k, y = min(tab$gap), yend = valids2$gap,
               linetype = "dashed") +
  geom_point(aes(x = k, y = gap), color = 'red') +
  scale_x_continuous(breaks = 1:10) +
  labs(x = "nombre de groupes", y = "GAP")

```

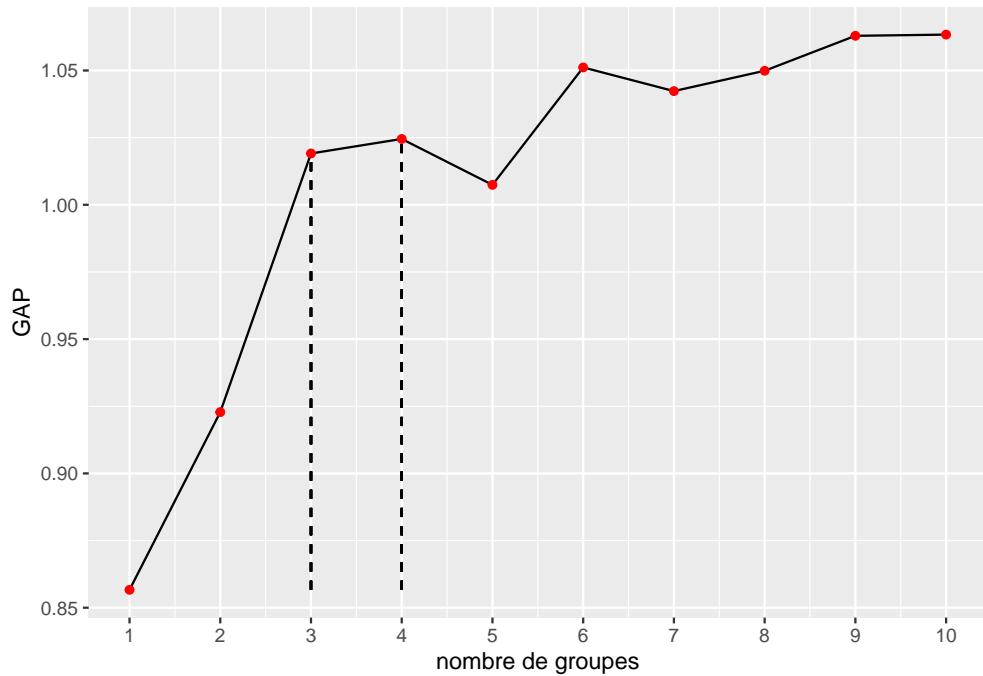


FIG. 13.21 : Méthode GAP pour différents nombres de groupes pour le k-means

La figure 13.21 indique également que le nombre de groupes à retenir est trois. Nous retenons cependant quatre groupes pour pouvoir plus facilement comparer nos résultats avec ceux de l'article original.

13.4.4.3 Application l'algorithme du *k-means*

Maintenant que nous avons choisi le nombre de groupes à former, nous pouvons simplement appliquer la fonction `kmeans` présente de base dans R.

```
set.seed(145)
resultats <- kmeans(X, centers = 4)
```

13.4.4.4 Interprétation des résultats

Une fois les groupes obtenus, l'étape la plus importante est de parvenir à interpréter ces groupes. Pour cela, il est nécessaire de les explorer en profondeur au travers des variables utilisées pour les constituer. Dans notre cas, le jeu de données `LyonIris` est spatialisé, nous pouvons donc commencer par cartographier les groupes.

```
library(tmap)
LyonIris$groupes <- paste("groupe", resultats$cluster, sep = " ")
tm_shape(LyonIris) +
  tm_polygons(col = "groupes", palette =
    c("#EFBE89", "#4A6A9F", "#7DB47C", "#FAF29C"), lty = 1, lwd = 0.1)
```

Il est ainsi possible de constater que le groupe 3 forme un ensemble assez compact d'IRIS au centre de Lyon. Le groupe 4 correspond quant à lui à des IRIS situés en périphérie plutôt éloignée, essentiellement

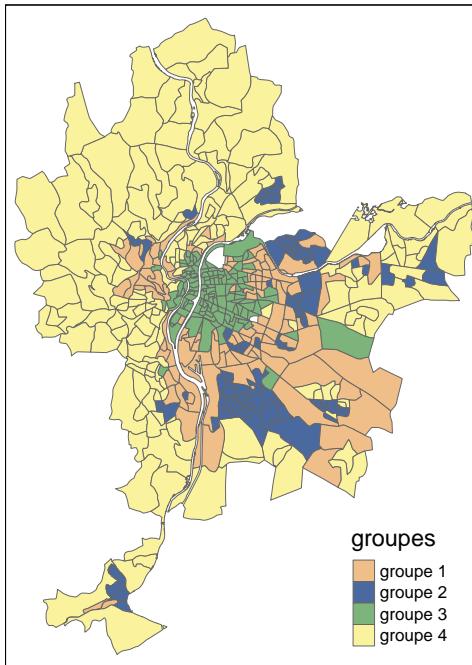


FIG. 13.22 : Cartographie des groupes obtenus avec la méthode du k-means

à l'ouest. Le groupe 1 correspond à une périphérie proche du groupe 2 et apparaît comme un ensemble d'enclaves dispersées.

Pour distinguer rapidement les profils des différents groupes, il est possible d'utiliser un graphique en radar. La construction d'un tel graphique peut être un peu fastidieuse dans R, cependant le *package geocmeans* propose une fonction assez pratique : `spiderPlots`.

```
library(geocmeans)

# création d'une matrice d'appartenance binaire des groupes
matrice_gp <- fastDummies::dummy_cols(resultats$cluster, remove_selected_columns = TRUE)

# réalisation du graphique
par(mfrow=c(3,2), mai = c(0.1,0.1,0.1,0.1))
plots <- spiderPlots(X, matrice_gp,
                      chartcolors = c("#E8BEE8", "#4A6A9F", "#7DB47C", "#FAF29C"))
```

Il est ainsi possible de constater, à la figure 13.23, que le groupe 3 est caractérisé par un niveau de vie élevé, mais par des niveaux de concentration de pollution atmosphérique plus élevés également. Le groupe 4 en revanche est caractérisé par un important couvert végétal, un niveau de vie médian élevé et une plus forte proportion de personnes de plus de 65 ans. Le groupe 1 est quant à lui marqué par des niveaux sonores plus élevés. Enfin, le groupe 2 se caractérise par une plus grande proportion de population ayant obtenu comme diplôme le plus élevé le brevet des collèges, d'immigrants, de jeunes de moins de 15 ans et un taux de chômage plus élevé.

Notez que ces graphiques nous permettent rapidement de nous faire une idée des caractéristiques des groupes, mais uniquement sur une échelle relative. En effet, ils ne nous indiquent à aucun moment la taille des écarts entre les groupes. Pour cela, il est nécessaire de réaliser des graphiques en violon pour chaque variable. Pour ce type de graphique, il est préférable d'utiliser les données originales non transformées

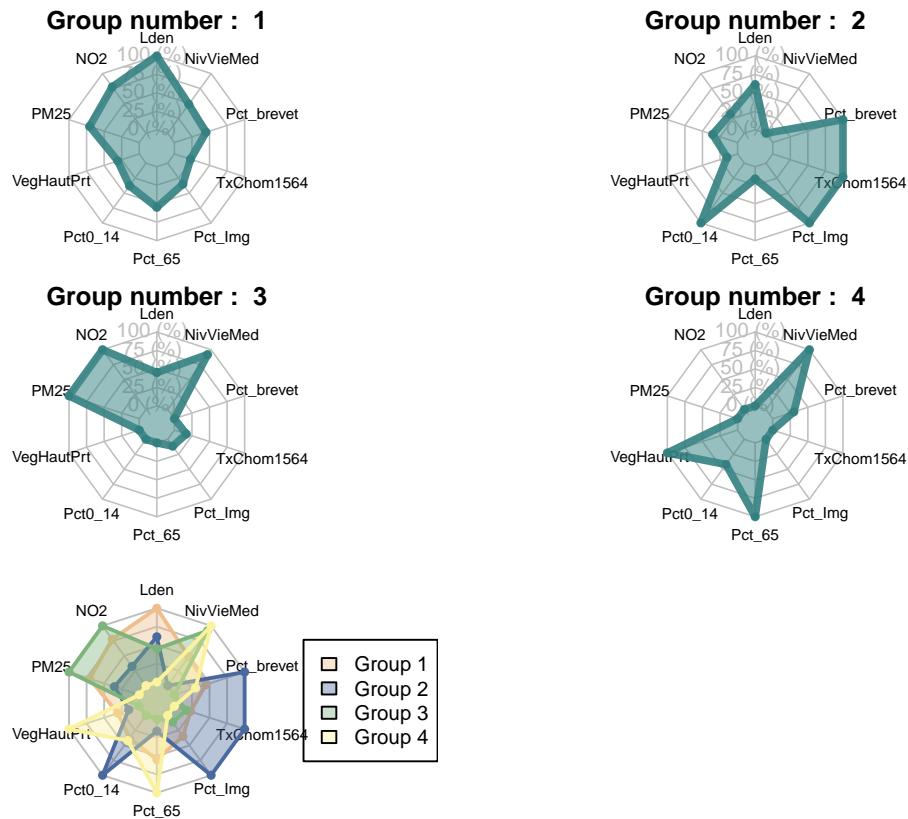


FIG. 13.23 : Graphiques en radar pour les groupes issus du k-means

pour pouvoir mieux appréhender si les différences entre les groupes sont importantes ou négligeables.

```
X2 <- LyonIris@data[c("Lden", "NO2", "PM25", "VegHautPrt", "Pct0_14", "Pct_65", "Pct_Img",
  "TxChom1564", "Pct_brevet", "NivVieMed")]

plots <- violinPlots(X2, as.character(resultats$cluster))
ggarrange(plotlist = plots, ncol = 2, nrow = 5)
```

Il est également recommandé de calculer des statistiques descriptives par groupe et de les rapporter dans un tableau.

```
# obtention d'un tableau par groupe
tableaux <- summarizeClusters(X2, matrice_gp, dec = 1, silent = TRUE)

# concaténation des tableaux
tableau_tot <- do.call(rbind, tableaux)
```

Les constats que nous avons faits précédemment sont confirmés par la figure 13.24 et le tableau 13.8. Nous retrouvons ici les groupes originaux décrits dans l'article de Gelb et Apparicio (2021b) :

- **Groupe 1** : les espaces interstitiels, formant une périphérie proche du centre et relativement hétérogène sur les variables étudiées, mais caractérisée par des niveaux de bruit importants.
- **Groupe 2** : les banlieues jeunes et défavorisées, avec des niveaux d'exposition aux pollutions atmosphérique et sonore relativement élevés comparativement à l'ensemble de la région.

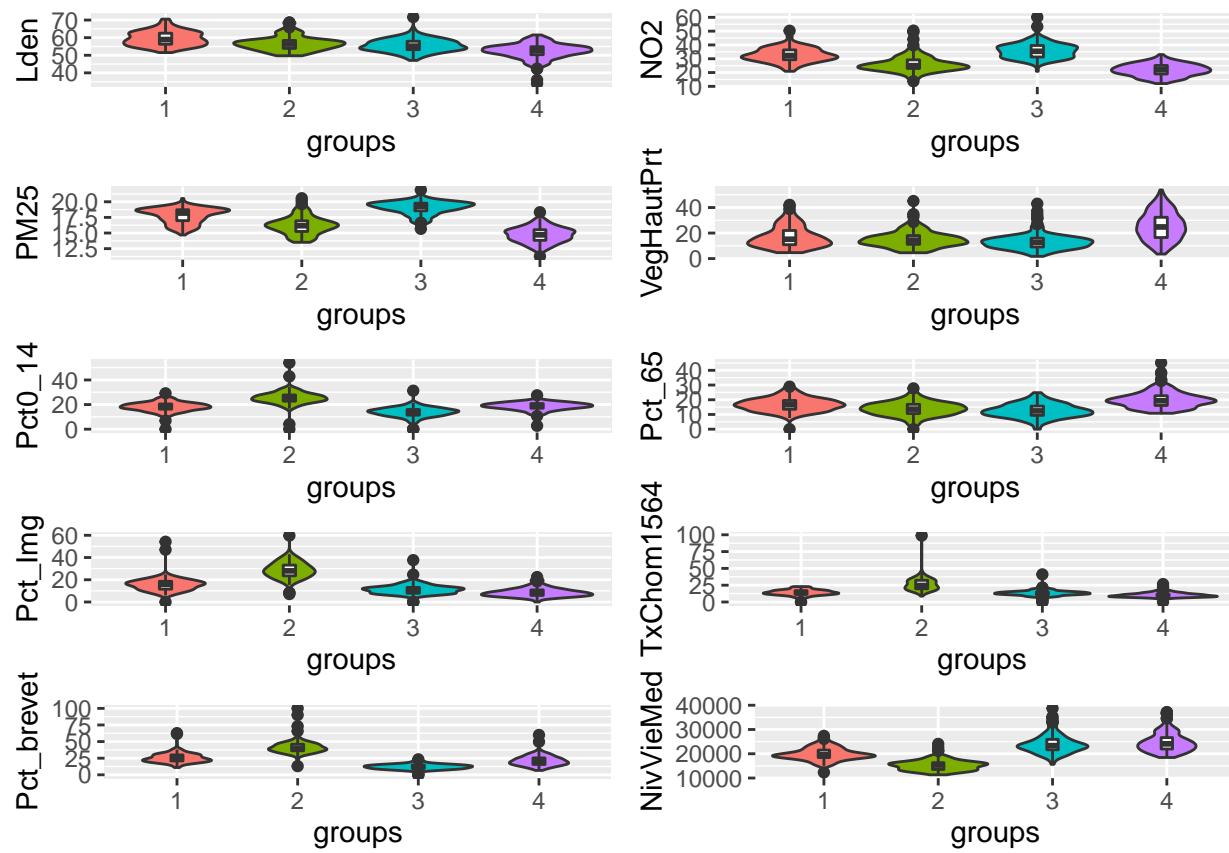


FIG. 13.24 : Graphiques en violon pour les groupes issus du k-means

- **Groupe 3 :** les quartiers centraux aisés, mais marqués par les plus hauts niveaux de pollution atmosphérique.
- **Groupe 4 :** les communes rurales, aisées et vieillissantes.



Interprétation interactive

Si, comme dans notre exemple, vos données comportent une dimension spatiale, le package `geocmeans` propose une fonction intéressante appelée `sp_clust_explorer` démarrant une application permettant d'explorer les résultats de votre classification. Le seul enjeu est de créer un objet de la classe `FCMres`. Voici un court exemple :

TAB. 13.8 : Descriptions des quatre groupes obtenus

	Lden	NO2	PM25	VegHautPrt	Pct014	Pct65	PctImg	TxChom1564	Pctbrevet	NivVieMed
groupe 1										
Q5	53,8	25,1	15,6	6,6	12,3	10,0	8,1	7,5	17,5	15 845,4
Q10	54,4	26,4	15,9	8,0	13,9	11,4	9,2	9,8	18,2	16 988,2
Q25	56,3	29,3	17,0	10,9	16,3	13,6	11,6	11,5	20,8	18 454,0
Q50	58,9	32,3	18,2	15,1	18,2	16,6	15,9	13,6	24,1	19 559,0
Q75	62,5	36,4	18,7	22,0	20,5	19,5	18,6	16,9	30,0	21 509,0
Q90	64,5	39,2	19,0	30,0	22,3	22,3	21,0	19,6	32,8	23 523,8
Q95	66,4	40,3	19,2	33,7	24,5	24,6	22,7	21,8	37,1	24 461,4
Mean	59,6	32,7	17,8	17,2	18,2	16,7	15,7	14,1	25,5	19 948,0
Std	4,2	5,2	1,2	8,5	4,0	4,7	6,5	4,3	7,6	2 637,2
groupe 2										
Q5	50,8	19,3	13,9	6,1	18,4	6,6	17,7	16,5	30,7	12 350,5
Q10	52,0	20,0	14,2	7,7	19,8	8,7	20,1	16,8	32,8	12 747,0
Q25	53,9	23,0	15,2	11,2	22,8	10,6	23,6	19,7	36,2	13 546,0
Q50	56,4	25,0	16,2	14,5	25,2	13,6	28,0	24,5	39,9	15 340,0
Q75	58,4	29,2	17,0	18,0	27,8	16,9	33,2	32,3	45,9	16 330,5
Q90	62,9	33,2	18,3	23,5	31,3	20,0	38,1	35,6	50,0	18 140,0
Q95	64,3	37,2	18,8	27,1	32,6	20,8	40,3	38,0	55,2	19 009,0
Mean	56,8	26,3	16,2	15,4	25,4	13,8	28,5	26,6	42,0	15 401,4
Std	4,1	6,3	1,5	6,8	6,2	4,8	8,0	10,5	11,3	2 340,4
groupe 3										
Q5	50,2	28,3	17,0	5,0	6,9	5,2	5,8	7,7	6,7	19 036,4
Q10	51,1	29,5	17,5	6,6	9,5	7,2	6,6	8,5	7,7	19 509,9
Q25	53,2	31,3	18,6	9,1	11,4	9,4	7,9	11,0	9,5	21 632,8
Q50	55,2	35,4	19,3	12,6	14,1	12,4	11,0	12,9	12,0	23 342,0
Q75	58,0	39,6	19,8	16,0	16,0	15,7	13,1	15,0	14,8	25 932,2
Q90	60,0	42,9	20,1	19,9	18,0	18,7	16,3	17,8	16,6	28 810,1
Q95	61,1	44,4	20,3	27,0	19,2	20,6	17,6	19,1	18,4	31 835,2
Mean	55,6	35,8	19,0	13,6	13,8	12,6	11,1	13,1	12,1	23 999,7
Std	3,7	5,6	1,0	6,8	4,2	4,7	4,6	4,4	4,0	3 870,3
groupe 4										
Q5	44,9	14,7	12,7	8,1	13,0	12,7	3,9	6,7	10,8	19 391,2
Q10	46,6	15,7	13,0	11,6	14,9	13,5	4,4	7,0	12,4	20 257,0
Q25	50,3	19,0	13,8	16,5	17,1	16,1	6,0	8,0	15,7	21 963,0
Q50	52,6	22,0	14,7	24,8	18,9	19,3	8,1	9,8	20,0	24 090,0
Q75	54,8	25,1	15,5	32,3	20,8	22,7	10,9	12,0	25,4	26 667,0
Q90	57,4	27,5	16,2	41,1	22,3	27,4	14,4	14,7	30,5	29 891,0
Q95	58,9	28,7	16,7	43,4	22,7	29,6	17,4	16,4	32,9	31 872,7
Mean	52,3	21,8	14,7	25,5	18,6	20,1	8,8	10,4	21,1	24 761,6
Std	4,3	4,4	1,2	11,0	3,1	5,6	4,2	3,4	7,6	4 008,7

```

# création d'une matrice binaire d'appartenance
kmeans_mat <- dummy_cols(resultats$cluster, remove_selected_columns = TRUE)

# extraction des centres de notre classification
centres <- resultats$centers

# création de l'objet FCMres
kmeansres <- FCMres(list(
  "Centers" = centres,
  "Belongings" = kmeans_mat,
  "Data" = X2,
  "m" = 1,
  "algo" = "kmeans"
))

# démarrage de l'application shiny
sp_clust_explorer(object = kmeansres, spatial = LyonIris)

```

13.4.4.5 K-médianes et K-médoïdes

Nous présentons simplement ici comment effectuer la même analyse en utilisant les variantes du *k-means*, soit le *k-medians* et le *k-medoids*.

Il existe relativement peu d'implémentation du *k-medians* dans R, nous optons donc ici pour la fonction `kGmedian` du package `Gmedian`. Pour le *k-medoids*, nous avons retenu la fonction `pam` du package `cluster`.

```

library(Gmedian)
k_median_res <- kGmedian(X, 4)

library(cluster)
k_mediods_res <- pam(X, 4)

```

Juste pour le plaisir des yeux, nous pouvons cartographier les trois classifications obtenues en nous assurant au préalable de faire coïncider les groupes les plus similaires de nos trois classifications.

```

matrice_gp_kmeans <- dummy_cols(resultats$cluster,
                                   remove_selected_columns = TRUE)
matrice_gp_kmedians <- dummy_cols(as.vector(k_median_res$cluster),
                                    remove_selected_columns = TRUE)
matrice_gp_kmediooids <- dummy_cols(k_mediods_res$cluster,
                                       remove_selected_columns = TRUE)

# Appariement des groupes du k-medians avec ceux du kmeans
matrice_gp_kmedians <- geocmeans::groups_matching(as.matrix(matrice_gp_kmeans),
                                                    as.matrix(matrice_gp_kmedians))

# Appariement des groupes du k-medoids avec ceux du kmeans
matrice_gp_kmediooids <- geocmeans::groups_matching(as.matrix(matrice_gp_kmeans),
                                                       as.matrix(matrice_gp_kmediooids))

# ajouts des colonnes nécessaires à LyonIris

```

```

colnames(matrice_gp_kmeans) <- paste0("groupe_", 1:4)
colnames(matrice_gp_kmedians) <- paste0("groupe_", 1:4)
colnames(matrice_gp_kmedoids) <- paste0("groupe_", 1:4)

LyonIris$kmeans <- colnames(matrice_gp_kmeans)[max.col(matrice_gp_kmeans)]
LyonIris$kmedians <- colnames(matrice_gp_kmedians)[max.col(matrice_gp_kmedians)]
LyonIris$kmédioids <- colnames(matrice_gp_kmedoids)[max.col(matrice_gp_kmedoids)]

# construction de la figure
couleurs <- c("#EFBE89", "#4A6A9F", "#7DB47C", "#FAF29C")

map1 <- tm_shape(LyonIris) +
  tm_polygons(col = "kmeans", palette = couleurs, lty = 1, lwd = 0.1)
map2 <- tm_shape(LyonIris) +
  tm_polygons(col = "kmedians", palette = couleurs, lty = 1, lwd = 0.1)
map3 <- tm_shape(LyonIris) +
  tm_polygons(col = "kmédioids", palette = couleurs, lty = 1, lwd = 0.1)

tmap_arrange(map1, map2, map3,
               ncol = 2, nrow = 2)

```

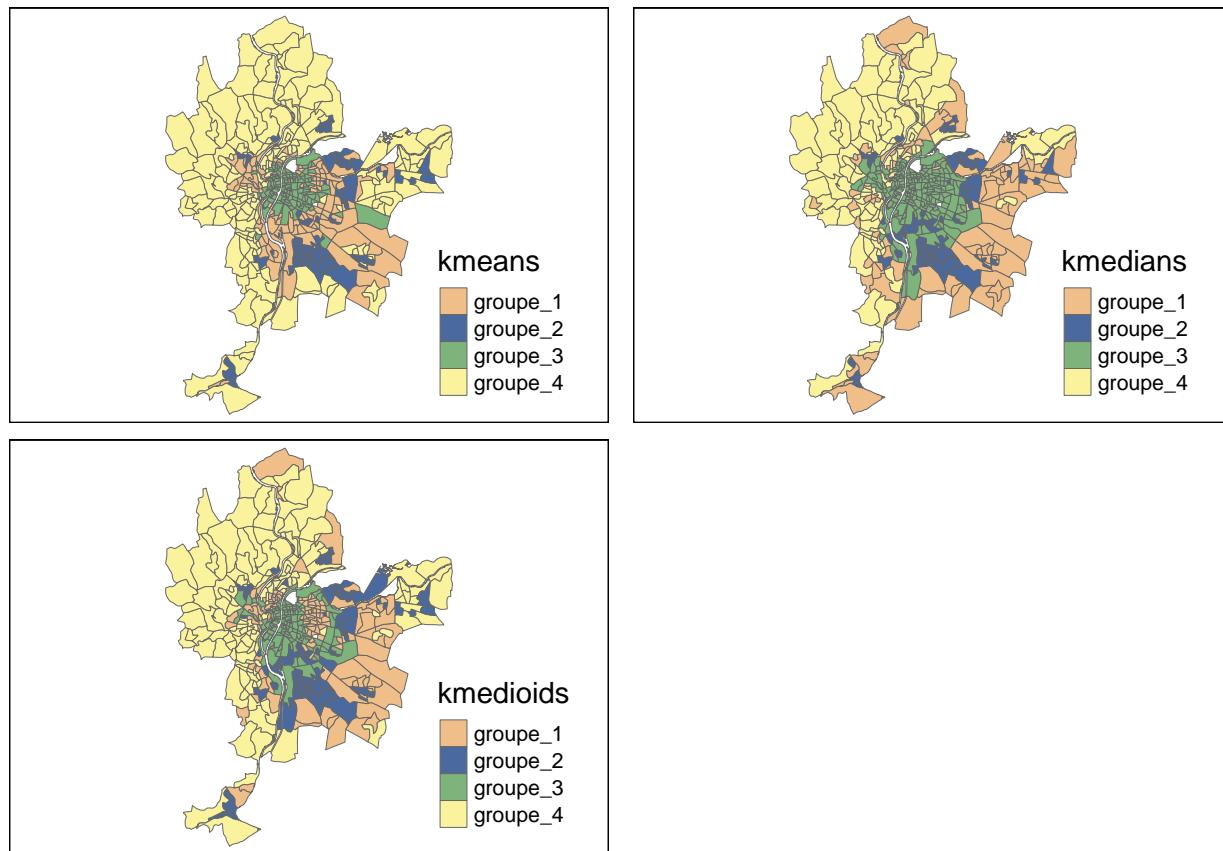


FIG. 13.25 : Comparaison géographique des résultats obtenus pour le k-means, le k-medians et le k-medoids

Les trois cartes sont très similaires (figure 13.25), ce qui signifie que les trois algorithmes tendent à attribuer les observations aux mêmes groupes. Cependant, nous observons des différences, notamment au

nord avec des observations alternant entre les groupes 2 et 3 selon la méthode employée. Cela peut notamment signifier que ces observations sont « indécises », qu'il est difficile de les attribuer définitivement à une catégorie en particulier. Pour prendre en compte cette forme d'incertitude, il est possible d'opter pour des méthodes de classification en logique floue.

13.4.5 Extensions en logique floue : *c-means*, *c-medoids*

Comme nous l'avons mentionné en introduction de cette section, les méthodes de classification floues ont pour objectif d'évaluer le degré d'appartenance de chaque observation à chaque groupe plutôt que d'attribuer chaque observation à un seul groupe. Il est ainsi possible de repérer des observations incertaines, à cheval entre plusieurs groupes. Nous présentons ici deux algorithmes appartenant à cette famille : le *c-means* et le *c-medoids*. Il s'agit dans les deux cas d'extensions des *k-means* et *k-medoids* vus précédemment.

Pour ces deux méthodes, comme pour le *k-means*, le nombre de groupes k doit être spécifié. Elles comprennent cependant un paramètre supplémentaire : m , appelé paramètre de floutage qui contrôle à quel point le résultat obtenu sera flou ou strict. Une valeur de 1 produit une classification stricte (chaque observation appartient à un seul groupe) et une valeur plus grande conduit à des classifications de plus en plus floues, jusqu'à ce que chaque observation appartienne à un degré identique à chacun des groupes. Il est recommandé de sélectionner m en même temps que k , car ces deux valeurs influencent simultanément la qualité de la classification. La meilleure approche consiste à tester un ensemble de combinaisons de m et de k et à comparer les valeurs obtenues pour différents indicateurs de qualité de classification floue. Parmi ces indicateurs, il est notamment recommandé d'utiliser le pourcentage de l'inertie expliquée, l'indice de silhouette pour classification floue, l'indice de Xie et Beni (1991), et de Fukuyama et Sugeno (Fukuyama 1989).

13.4.5.1 Mise en oeuvre du *c-means* dans R

Le package *fclust* comprend un très grand nombre de méthodes pour effectuer des classifications floues, nous l'utilisons donc en priorité ici en combinaison avec des fonctions d'interprétation du package *geocmeans*.

13.4.5.1.1 Préparation des données

Comme pour le *k-means*, cette méthode nécessite de disposer d'un jeu de données ne comprenant que des variables quantitatives dans la même échelle. Nous commençons donc à nouveau par standardiser nos données. Pour varier les plaisirs, nous optons cette fois-ci pour une transformation des variables dans une échelle allant de 0 à 100.

```
library(fclust)

data(LyonIris)

# NB : LyonIris est un objet spatial, il faut donc extraire uniquement son DataFrame
X <- LyonIris@data[c("Lden","N02","PM25","VegHautPrt","Pct0_14","Pct_65","Pct_Img",
                    "TxChom1564","Pct_brevet","NivVieMed")]

# changement d'échelle des données (0 à 100)
to_0_100 <- function(x){
  return((x-min(x)) / (max(x) - min(x)) * 100)
}
```

```

for (col in names(X)){
  X[[col]] <- to_0_100(X[[col]])
}

```

13.4.5.1.2 Sélection de k et de m

La seconde étape consiste à sélectionner les valeurs optimales pour k et m . Nous testons ici toutes les valeurs de k de 2 à 7, et les valeurs de m de 1,5 à 2,5 (avec des écarts de 0,1).

```

library(e1071)
set.seed(123)
ms <- seq(1.5,2.5,by = 0.1)
ks <- 2:7

# calcul de toutes les combinaisons
combinaisons <- expand.grid(ms,ks)

eval_indices <- c("Explained.inertia", "Silhouette.index", "FukuyamaSugeno.index")

values <- apply(combinaisons, MARGIN = 1, FUN = function(row){
  m <- row[[1]]
  k <- row[[2]]
  resultats <- FKM(X, k, m)
  idx <- geocmeans:::calcqualityIndexes(as.matrix(X),
                                         as.matrix(resultats$U),
                                         m = m,
                                         indices = eval_indices)
  return(c(k,m,unlist(idx)))
})

df_scores <- data.frame(t(values))
names(df_scores) <- c("k", "m", "inertie", "silhouette", "FukuyamaSugeno")

# changer l'échelle de l'indice pour un graphique plus joli
df_scores$FukuyamaSugeno <- round(df_scores$FukuyamaSugeno/10000,2)

# création de trois figures pour représenter les trois indicateurs
library(viridis)

plot1 <- ggplot(df_scores) +
  geom_raster(aes(x = k, y = m, fill = inertie)) +
  scale_fill_viridis() +
  scale_x_continuous(breaks = c(2,3,4,5,6,7)) +
  coord_fixed(ratio=4) +
  guides(fill = guide_colourbar(barwidth = 5, barheight = 0.5)) +
  labs(fill = "Inertie expliquée") +
  theme(legend.position = "bottom", legend.box = "horizontal",
        legend.title = element_text( size=9), legend.text=element_text(size=8))

plot2 <- ggplot(df_scores) +
  geom_raster(aes(x = k, y = m, fill = silhouette)) +
  scale_fill_viridis() +

```

```

scale_x_continuous(breaks = c(2,3,4,5,6,7)) +
coord_fixed(ratio=4) +
guides(fill = guide_colourbar(barwidth = 5, barheight = 0.5)) +
labs(fill = "Indice de silhouette") +
theme(legend.position = "bottom", legend.box = "horizontal",
legend.title = element_text( size=9), legend.text=element_text(size=8))

plot3 <- ggplot(df_scores) +
geom_raster(aes(x = k, y = m, fill = FukuyamaSugeno)) +
scale_fill_viridis() +
scale_x_continuous(breaks = c(2,3,4,5,6,7)) +
coord_fixed(ratio=4) +
guides(fill = guide_colourbar(barwidth = 5, barheight = 0.5)) +
labs(fill = "Indice de Fukuyama et Sugeno") +
theme(legend.position = "bottom", legend.box = "horizontal",
legend.title = element_text( size=9), legend.text=element_text(size=8))

ggarrange(plot1, plot2, plot3, ncol = 2, nrow = 2)

```

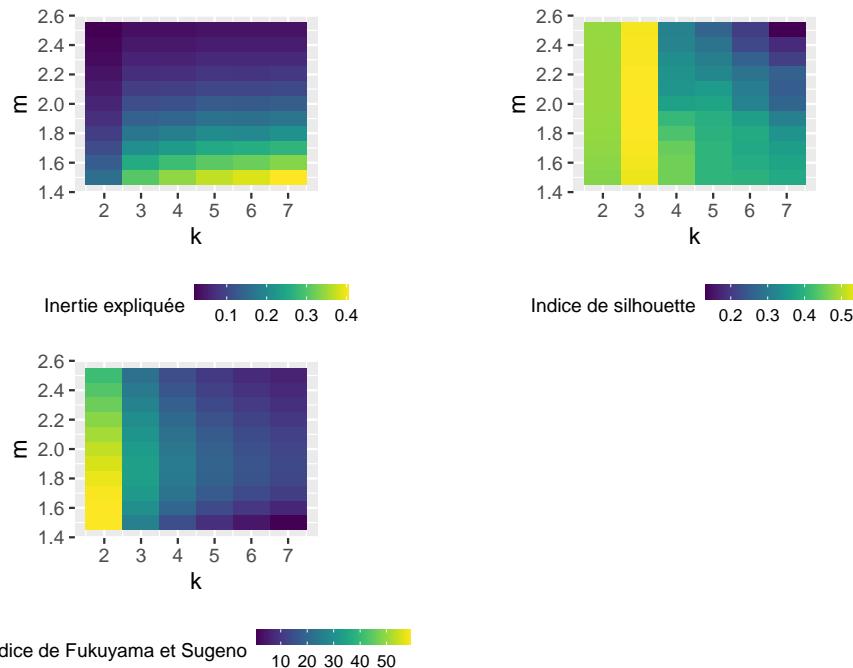


FIG. 13.26 : Sélection des paramètres k et m pour l'algorithme c -means

Les trois graphiques à la figure 13.26 semblent indiquer des solutions différentes. Sans surprise, augmenter le niveau de flou (m) réduit l'inertie expliquée, alors qu'augmenter le nombre de groupes (k) augmente l'inertie expliquée. L'indice de silhouette indique assez clairement que trois groupes serait le meilleur choix, suivi par deux ou quatre groupes, si m est inférieur à 1,8. Cependant, ne retenir que trois groupes ne permet d'expliquer que 30% de l'inertie. Afin de nous rapprocher des résultats de l'article original (Gelb et Apparicio 2021b), nous retenons $m = 1,5$ et $k = 4$.

13.4.5.1.3 Application l'algorithme c -means

Avec k et m définis, il ne reste plus qu'à appliquer l'algorithme à nos observations.

```
set.seed(123)
cmeans_resultats <- FKM(X, 4, 1.5)
```

L'objet obtenu `cmeans_resultats` contient les résultats de la classification. Plus spécifiquement, `cmeans_resultats$U` est la matrice d'appartenance, soit une matrice de taille $n \times k$, dont chaque case U_{ij} indique la probabilité pour l'observation i d'appartenir au groupe j . `cmeans_resultats$H` contient le centre des groupes, et `cmeans_resultats$Clus`, le groupe auquel chaque observation à le plus de chances d'appartenir. Pour comparer plus facilement nos résultats avec ceux du k -means, nous pouvons changer l'ordre des groupes obtenus pour les faire coïncider avec les groupes les plus similaires obtenus avec la méthode k -means.

```
# changeons l'ordre des groupes
U <- cmeans_resultats$U
U2 <- geocmeans::groups_matching(as.matrix(matrice_gp_kmeans), as.matrix(U))

# mais aussi du centre des classes
idx <- as.integer(gsub("Clus ","", colnames(U2), fixed = TRUE))
H2 <- cmeans_resultats$H[idx,]

# et recalcul du groupe le plus probable
Clus2 <- data.frame(
  "Cluster" = (1:4)[max.col(U2, ties.method="first")],
  "Membership degree" = apply(U2, MARGIN = 1, max)
)

colnames(U2) <- paste("Clus", 1:4, sep = " ")
rownames(H2) <- paste("Clus", 1:4, sep = " ")

cmeans_resultats$U <- U2
cmeans_resultats$H <- H2
cmeans_resultats$Clus <- Clus2
```

13.4.5.1.4 Interprétation des résultats

Globalement, les approches pour interpréter les résultats issus d'une classification obtenue par c -means sont les mêmes que pour une classification obtenue par k -means.

Commençons donc par créer plusieurs cartes des probabilités d'appartenir aux différents groupes.

```
maps <- mapClusters(LyonIris, cmeans_resultats$U)

ggarrange(plotlist = maps$ProbaMaps, ncol = 2, nrow = 2, legend = "none")
```

Sur les cartes de la figure 13.27, l'intensité de bleu correspond à la probabilité pour chaque IRIS d'appartenir aux différents groupes. Nous retrouvons les principales structures spatiales que nous avons identifiées avec le k -means; cependant, nous pouvons à présent constater que le groupe 1 est bien plus incertain que les autres. Nous pouvons une fois encore générer un graphique en radar pour comparer les profils des quatre groupes.

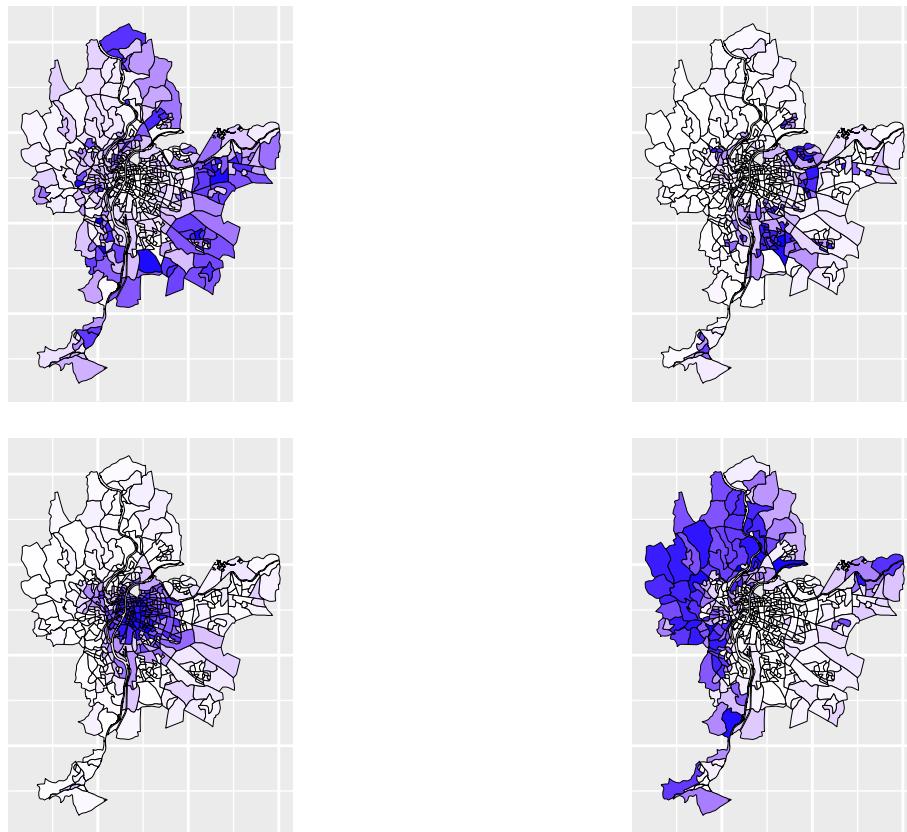


FIG. 13.27 : Cartographie des probabilités d'appartenir aux quatre groupes identifiés par l'algorithme c-means

```
par(mfrow=c(3,2), mai = c(0.1,0.1,0.1,0.1))
spiderPlots(X, cmeans_resultats$U,
            chartcolors = c("#EFBE89", "#4A6A9F", "#7DB47C", "#FAF29C"))
```

Sans surprise, nous retrouvons essentiellement les profils que nous avons obtenus avec le *k-means* dans la figure 13.28. Pour compléter la lecture des résultats, il est nécessaire de se pencher sur le tableau des statistiques descriptives des différents groupes. Une fois encore, nous proposons d'utiliser la fonction `summarizeClusters` du package `geocmeans`. Notez que cette fonction calcule les statistiques descriptives pondérées en fonction de l'appartenance des observations aux groupes. Ainsi, une observation ayant une faible chance d'appartenir à un groupe ne contribue que faiblement aux statistiques descriptives de ce groupe.

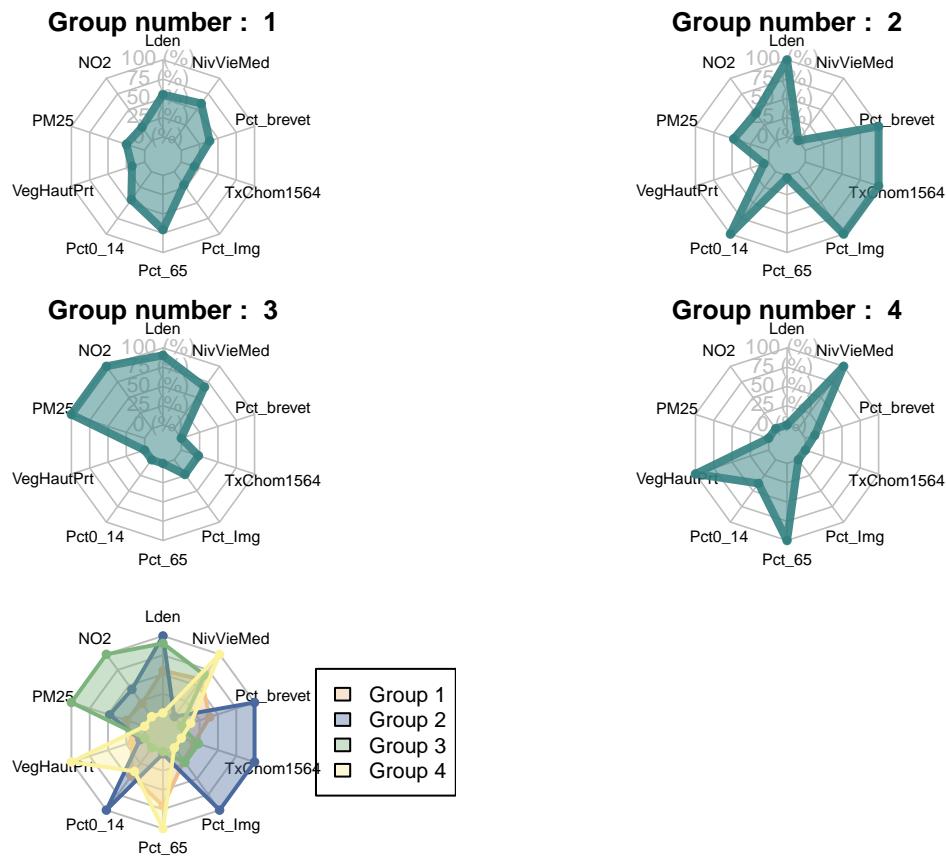


FIG. 13.28 : Graphique en radar pour les résultats du c-means

```
df <- LyonIris@data[c("Lden", "NO2", "PM25", "VegHautPrt", "Pct0_14", "Pct_65", "Pct_Img",
                    "TxChom1564", "Pct_brevet", "NivVieMed")]

tableaux <- summarizeClusters(data = df,
                                belongmatrix = cmeans_resultats$U,
                                weighted = TRUE, dec = 1)

tableau_tot <- do.call(rbind, tableaux)
```

TAB. 13.9 : Description des groupes avec la méthode c-means

	Lden	NO2	PM25	VegHautPrt	Pct014	Pct65	PctImg	TxChom1564	Pctbrevet	NivVieMed
groupe 1										
Q5	48,7	16,1	13,6	6,3	12,4	9,9	4,5	6,7	11,1	16 274,1
Q10	50,6	18,8	13,9	7,6	14,3	11,4	5,9	7,7	13,9	17 857,3
Q25	52,4	21,0	14,7	11,5	16,9	14,2	8,0	9,3	18,5	19 608,0
Q50	54,7	25,0	15,6	15,8	19,0	17,7	11,3	12,0	24,0	21 955,8
Q75	57,5	28,6	16,8	22,3	21,3	21,2	15,7	15,2	29,9	24 068,0
Q90	60,7	33,3	18,5	29,7	23,3	24,4	20,3	18,5	34,7	26 134,7
Q95	63,1	37,7	19,0	34,6	25,5	27,6	23,6	22,3	39,1	29 013,0
Mean	55,1	25,4	15,8	17,5	18,9	17,9	12,4	13,0	24,7	21 951,6
Std	4,4	6,3	1,7	8,6	4,5	5,7	6,8	6,8	10,3	3 744,1
groupe 2										
Q5	51,0	19,7	14,1	6,4	14,1	7,7	8,9	10,1	17,1	12 426,6
Q10	52,3	21,5	14,6	7,9	16,7	8,8	13,4	13,1	22,4	12 973,6
Q25	54,6	23,3	15,8	10,9	20,3	11,1	19,7	16,8	30,7	14 108,1
Q50	57,1	26,9	16,6	14,5	23,6	13,9	25,8	22,0	37,8	16 010,0
Q75	59,5	32,0	17,8	19,0	27,1	17,4	32,1	30,1	44,4	18 568,9
Q90	63,5	37,4	18,8	25,5	30,1	20,7	38,0	33,9	49,0	21 028,3
Q95	65,2	39,6	19,2	30,8	32,2	23,5	40,6	37,7	52,4	23 774,4
Mean	57,3	28,1	16,7	15,8	23,4	14,4	25,5	23,0	36,9	16 658,7
Std	4,4	6,6	1,6	7,6	6,3	5,1	9,7	9,3	12,3	3 611,9
groupe 3										
Q5	50,6	27,6	16,4	6,3	8,7	7,1	6,1	7,7	7,6	17 130,3
Q10	52,2	28,9	17,2	7,7	10,4	8,6	7,1	8,6	8,7	18 454,2
Q25	53,9	30,9	18,4	10,2	12,6	10,8	9,1	11,1	11,0	19 805,4
Q50	56,5	35,0	18,9	13,4	15,2	14,1	12,3	13,1	14,8	22 308,8
Q75	59,4	38,7	19,6	17,5	17,9	17,5	15,9	15,5	21,5	24 515,7
Q90	62,8	41,0	20,0	24,8	20,4	20,5	19,0	18,4	27,9	27 620,4
Q95	64,4	44,1	20,2	30,3	21,8	23,2	21,1	20,7	32,1	29 862,2
Mean	56,8	34,9	18,7	14,8	15,3	14,2	12,8	13,7	16,9	22 595,1
Std	4,4	5,9	1,3	7,2	4,6	5,2	5,8	5,5	9,1	3 969,2
groupe 4										
Q5	44,8	14,8	12,6	12,2	12,5	11,5	4,0	6,8	9,7	18 922,4
Q10	45,8	15,7	12,9	17,7	14,2	13,2	4,6	7,3	11,2	20 246,9
Q25	49,5	18,7	13,7	24,8	16,6	15,8	5,9	7,9	14,3	22 753,9
Q50	52,3	22,0	14,7	30,5	18,6	19,0	7,5	9,7	18,0	24 950,1
Q75	55,1	26,3	15,9	37,9	20,8	22,4	10,1	12,1	23,0	28 806,2
Q90	58,9	30,8	17,1	42,3	22,4	27,3	14,5	15,3	30,0	31 426,2
Q95	60,8	34,7	18,3	45,5	23,3	28,8	19,2	18,7	33,3	34 309,0
Mean	52,3	22,8	14,9	30,3	18,4	19,4	8,8	10,9	19,5	25 547,4
Std	5,1	6,3	1,7	10,0	4,1	5,9	5,5	6,2	9,2	4 603,5

13.4.5.2 Mise en oeuvre du *c-medoids* dans R

La méthode du *c-medoids* dans R peut être mise en oeuvre avec la fonction `FKM.med` du package `fclust`. Le processus d'analyse et de validation est identique à celui présenté ci-dessus pour le *c-means*. Nous ne donnons donc pas un exemple complet de la méthode.



Stabilité des groupes obtenus par les méthodes de nuées dynamiques :

Puisque la méthode *k-means* et ses variantes reposent sur une initialisation aléatoire de leur algorithme, les résultats peuvent varier en fonction de cet état de départ. Dans certains contextes, il est possible que les résultats obtenus varient significativement, ce qui signifie que les groupes obtenus ne sont pas représentatifs de la population étudiée. Une solution pour vérifier si ce problème se pose est simplement de relancer l'algorithme un grand nombre de fois (généralement 1000) et de comparer les résultats obtenus au cours de ces réplications.

Cette méthode est rarement implémentée directement et requiert d'écrire sa propre fonction. `geocmeans` dispose d'une fonction déjà existante, mais ne pouvant être appliquée qu'avec l'algorithme *c-means*. Nous propo-

sons ici une implémentation pour la méthode *k-means* qui peut facilement être adaptée aux autres méthodes de classifications heuristiques.

La démarche à suivre est la suivante :

1. Appliquer l'algorithme une première fois pour obtenir une classification de référence à laquelle nous comparerons toutes les réplications.
2. Effectuer 1000 itérations au cours desquelles :
 - Une nouvelle classification est calculée.
 - Les groupes obtenus sont comparés à ceux de la classification de référence.
 - L'indice de Jacard est calculé entre les groupes des deux classifications.
 - Les valeurs de l'indice de Jacard sont enregistrées.
 - Les centres des groupes sont enregistrés.

Ainsi, nous obtenons 1000 valeurs de l'indice de Jacard pour chaque groupe. Cet indice permet de mesurer le degré d'accord entre deux variables (ici les probabilités d'appartenance des observations au même groupe pour deux classifications différentes.) Une valeur moyenne en dessous de 0,5 indique qu'un groupe est très instable, car nous obtenons des résultats très différents lors des réplications. Une valeur entre 0,6 et 0,75 indique qu'un groupe semble bien exister dans les données, bien que marqué par une certaine incertitude. Une valeur au-dessus de 0,8 indique un groupe bien identifié et stable.

Nous obtenons également les centres des groupes des 1000 classifications. Il est ainsi possible de représenter leurs histogrammes et de déterminer si les centres des groupes sont stables (unimodalité et faible variance) ou incertains (plusieurs modes et/ou forte variance).

```

# X sera le jeu de données pour la classification
# clust_ref sera le vecteur indiquant le groupe de chaque observation obtenu par kmeans
# nsim sera le nombre de simulations à effectuer
kmeans_stability <- function(X, clust_ref, nsim, verbose = TRUE){

  # définition de la matrice d'appartenance originale
  k <- length(unique(clust_ref))
  ref_mat <- dummy_cols(clust_ref, remove_selected_columns = TRUE)
  colnames(ref_mat) <- paste0("groupe_", 1:k)

  # lancement des itérations
  sim_resultats <- lapply(1:nsim, function(i){

    # afficher la progression si requis
    if(verbose){
      print(paste0("iteration numero : ", i, "/", nsim))
    }
    # calculer le kmeans
    km_res <- kmeans(X, k)
    sim_mat <- dummy_cols(km_res$cluster, remove_selected_columns = TRUE)

    # ajustement de l'ordre des groupes avec geocmeans
    sim_mat <- groups_matching(as.matrix(ref_mat), as.matrix(sim_mat))

    # calcul des indices de jacard
    jac_idx <- sapply(1:k, function(j){
      calc_jaccard_idx(sim_mat[,j], ref_mat[,j])
    })

    # recuperation des centres des groupes
    idx <- as.integer(gsub(".data_","", colnames(sim_mat), fixed = TRUE))
    centers <- data.frame(km_res$centers)
    centers <- centers[idx,]
    centers$groupe <- 1:k

    return(list(
      "jac_idx" = jac_idx,
      "centers" = centers
    ))
  })

  # les simulations sont finies, nous pouvons combiner les résultats
  all_jac_values <- do.call(rbind, lapply(sim_resultats, function(x){x$jac_idx}))
  all_centers <- do.call(rbind, lapply(sim_resultats, function(x){x$centers}))
  return(list(
    "jaccard_values" = all_jac_values,
    "centers" = all_centers
  ))
}

```

Il ne nous reste plus qu'à utiliser notre nouvelle fonction pour déterminer si les groupes obtenus avec notre *k-means* sont stables.

```

data(LyonIris)
set.seed(123)

# NB : LyonIris est un objet spatial, il faut donc extraire uniquement son DataFrame
X <- LyonIris@data[c("Lden","NO2","PM25","VegHautPrt","Pct0_14","Pct_65","Pct_Img",
                    "TxChom1564","Pct_brevet","NivVieMed")]

# Centrage et réduction de chaque colonne du DataFrame
for (col in names(X)){
  X[[col]] <- scale(X[[col]], center = TRUE, scale = TRUE)
}

# calcul du kmeans de référence
kmeans_ref <- kmeans(X, 4)

# application de la fonction de stabilité
stab_results <- kmeans_stability(X, kmeans_ref$cluster, nsim = 1000, verbose = FALSE)

```

Nous pouvons à présent vérifier la stabilité de nos quatre groupes.

```

jacard_values <- data.frame(stab_results$jacard_values)
names(jacard_values) <- paste("groupe", 1:4, sep = "_")

df <- reshape2::melt(jacard_values)
df$groupes <- as.factor(df$variable)

ggplot(df) +
  geom_histogram(aes(x = value), bins = 30) +
  facet_wrap(vars(groupes), ncol=2) +
  labs(x = "", y = "Indice de Jacard")

```

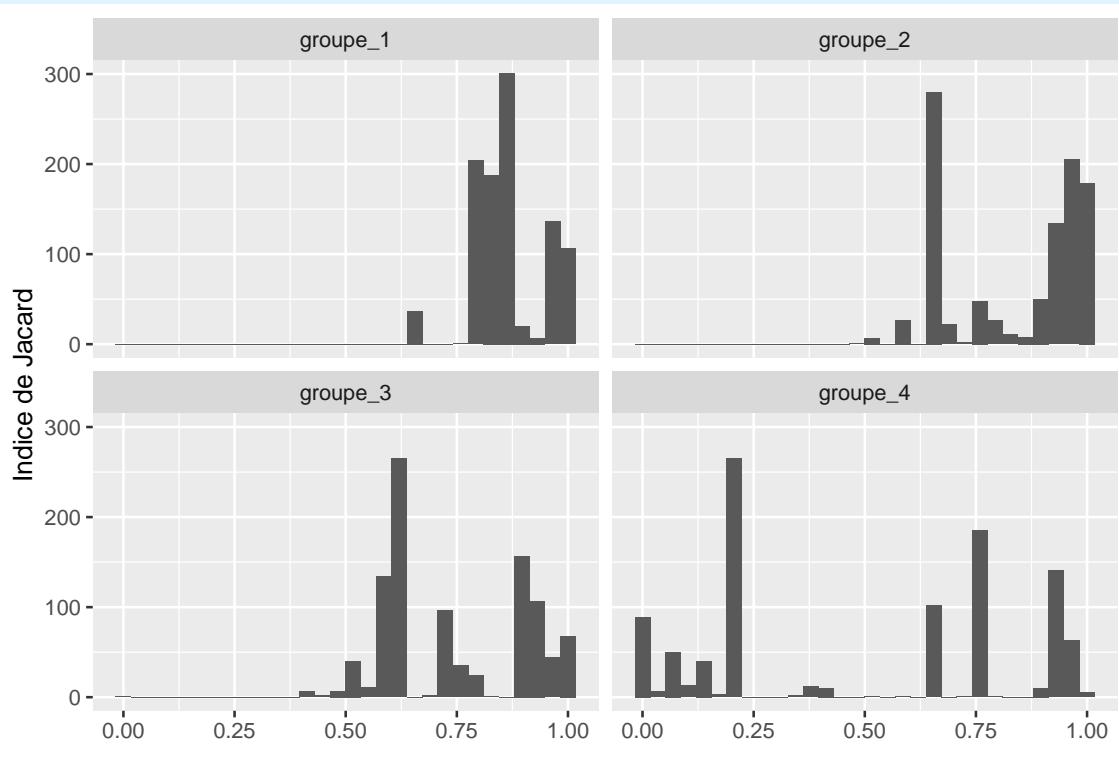


FIG. 13.29 : Indices de Jacard obtenus sur 1000 réplications du k-means

La figure 13.29 indique clairement que les groupes 1 et 2 sont très stables, car les valeurs de Jacard obtenues sont le plus souvent supérieures à 0,75. Le groupe 3 a le plus souvent des valeurs légèrement inférieures aux deux premiers groupes, mais tout de même bien supérieures à 0,5. En revanche, le groupe 4 a un grand nombre de valeurs inférieures à 0,5 indiquant une tendance du groupe à se dissoudre lors des réplications.

Considérant que le dernier groupe est le plus instable, nous décidons d'observer les valeurs des centres qu'il obtient pour les différentes réplications.

```
centers_groupe4 <- subset(stab_results$centers, stab_results$centers$groupe == 4)
centers_groupe4$groupe <- NULL

df <- reshape2::melt(centers_groupe4)
df$variable <- as.factor(df$variable)

ggplot(df) +
  geom_histogram(aes(x = value), bins = 30) +
  facet_wrap(vars(variable), ncol=3, scales="free") +
  labs(x = "", y = "")
```

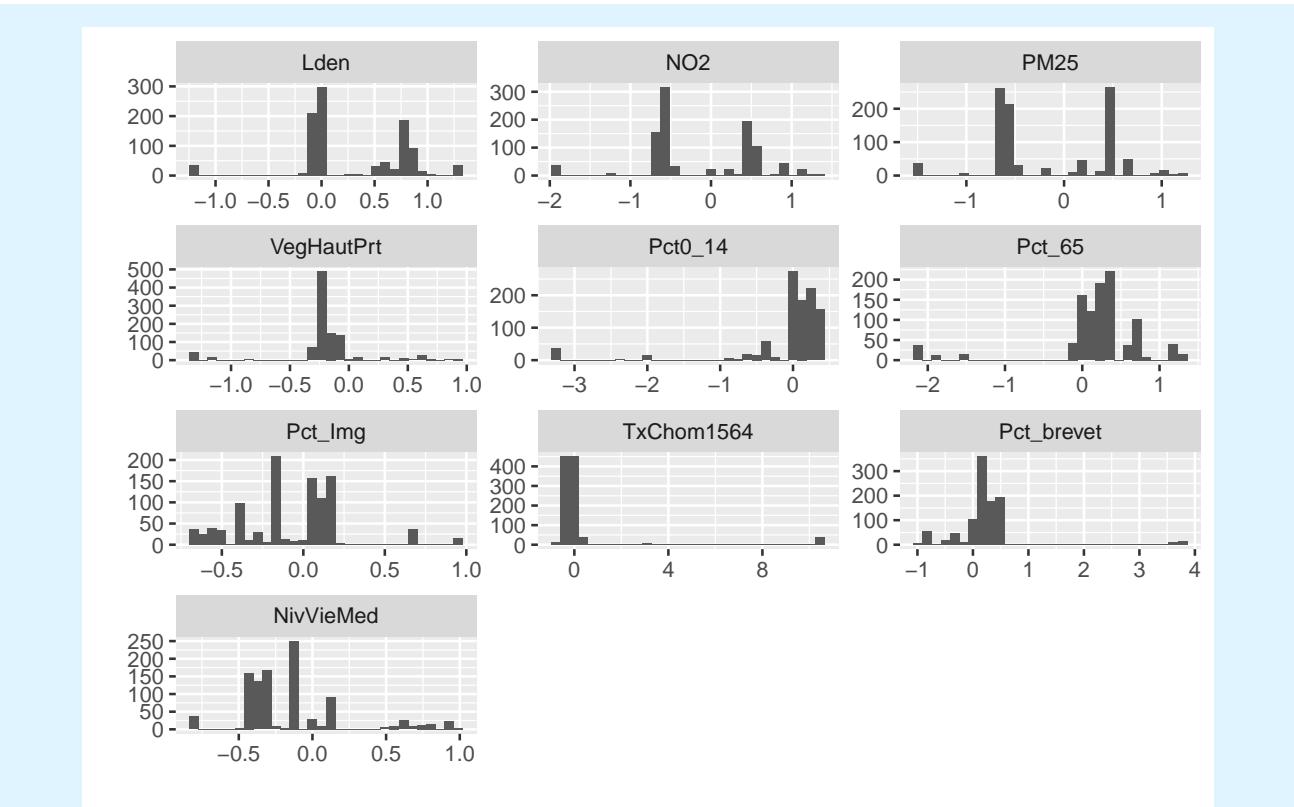


FIG. 13.30 : Distributions des valeurs des centres du groupe 4 sur 1000 itérations

Les différents histogrammes de la figure 13.30 indiquent clairement que pour plusieurs variables (Lden, NO2, PM25, Pct_Img, et NivVieMed) les caractéristiques du groupe 4 varient grandement sur l'ensemble des réplications. Nous pouvons comparer ce graphique à celui du groupe 2 qui est bien plus stable.

```
centers_groupe2 <- subset(stab_results$centers, stab_results$centers$groupe == 2)
centers_groupe2$groupe <- NULL

df <- reshape2::melt(centers_groupe2)
df$variable <- as.factor(df$variable)

ggplot(df) +
  geom_histogram(aes(x = value), bins = 30) +
  facet_wrap(vars(variable), ncol=3, scales="free") +
  labs(x = "", y = "")
```

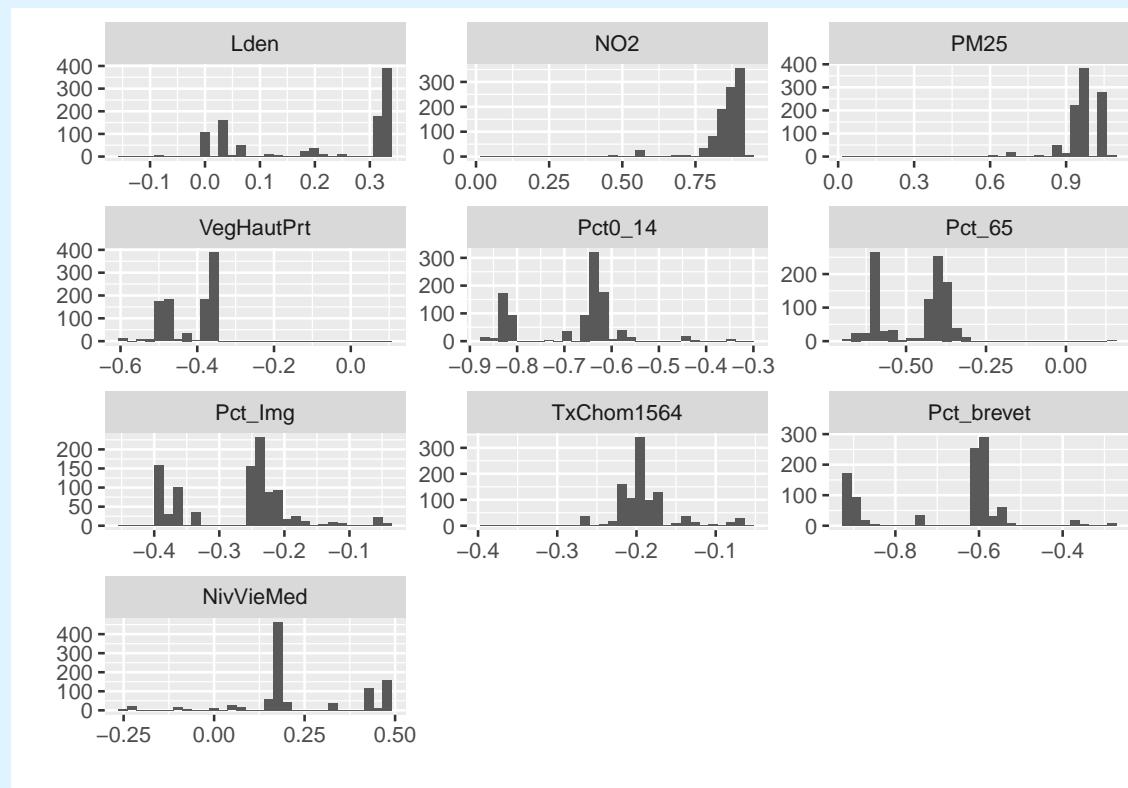


FIG. 13.31 : Distributions des valeurs des centres du groupe 2 sur 1000 itérations

Nous pouvons constater une plus faible variance des résultats obtenus (en regardant notamment l'axe horizontal) pour les centres des groupes à la figure 13.31.

13.5 Conclusion sur la cinquième partie

Dans le cadre de cette cinquième partie du livre, nous avons abordé les principales méthodes factorielles et les principales méthodes de classification non supervisée. Les premières sont des méthodes de réduction de données puisqu'elles permettent de résumer l'information d'un tableau en quelques nouvelles variables synthétiques. Les secondes permettent de regrouper les observations d'un tableau en plusieurs groupes homogènes. Il existe donc une complémentarité évidente entre ces deux groupes de méthodes : si le tableau initial comprend un grand nombre de variables, il est possible de lui appliquer une méthode factorielle produisant de nouvelles variables synthétiques qui sont ensuite utilisées pour calculer une méthode de classification non supervisée.

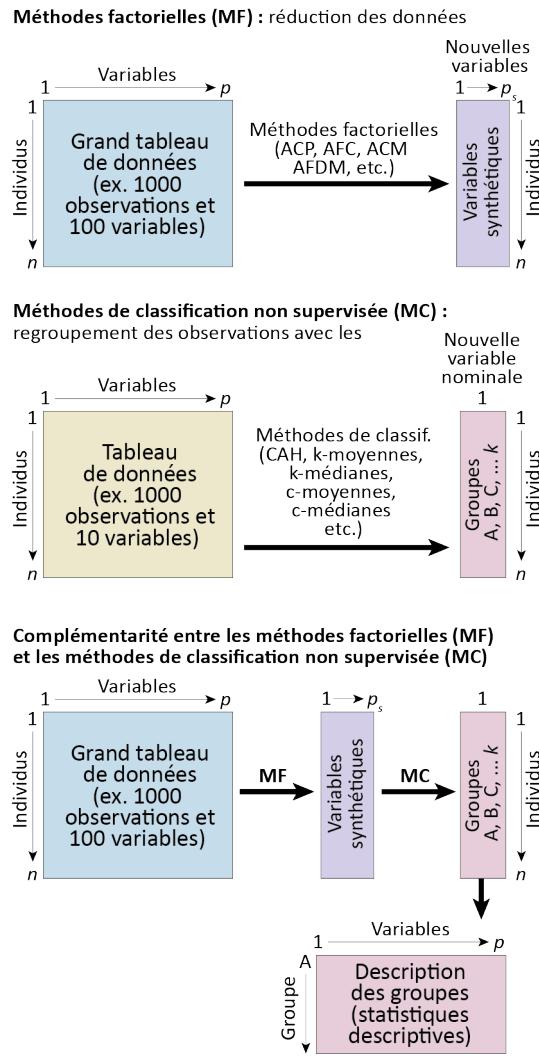


FIG. 13.32 : Complémentarité entre les méthodes factorielles et les méthodes de classification non supervisée

13.6 Quiz de révision du chapitre

```
quizz_classif <- quizz("quizzes/classification.yml", "quizz_classif")
render_quizz(quizz_classif)
```

Questions

- **Les méthodes de classification non supervisée sont appelées ainsi puisque :**
 - le nombre de groupes générés par ces méthodes est déterminé automatiquement
 - les groupes sont inconnus au préalable et formés automatiquement par ces méthodes
 - l'exécution de ces algorithmes ne nécessite pas de supervision
 - aucun paramètre n'est à définir pour exécuter ces méthodes

Relisez au besoin la section [13.1](#).
- **En quoi se distinguent les méthodes de classifications strictes et floues ?**
 - les premières attribuent chaque observation à un seul groupe alors que les secondes évaluent le degré d'appartenance des observations à chaque groupe
 - les méthodes strictes produisent des classifications comportant un moins grand nombre de degrés de liberté
 - les méthodes strictes produisent toujours les mêmes résultats, alors que les méthodes floues reposent sur un état initial aléatoire pouvant conduire à des résultats différents

Relisez au besoin la section [13.1](#).
- **Quels paramètres doivent être définis pour exécuter l'algorithme c-means ?**
 - k, soit le nombre de groupes à obtenir
 - alpha, soit le paramètre contrôlant la vitesse de convergence
 - lambda, soit la probabilité minimale qu'une observation appartient à un groupe
 - m, soit le paramètre contrôlant le niveau de flou dans une classification floue

Relisez au besoin la section [13.4.4](#).
- **La classification ascendante hiérarchique nécessite de calculer :**
 - la distance entre chaque observation et ses k plus proches voisins
 - la distance entre l'ensemble des paires d'observations
 - la probabilité pour chaque observation d'appartenir à chaque groupe
 - un dendrogramme, soit une structure hiérarchique permettant de conserver l'ordre dans lequel les groupes sont formés par l'algorithme

Relisez au besoin la section [13.3](#).
- **Pour effectuer une classification non supervisée floue utilisant comme centres de groupes de réelles observations plutôt que des moyennes fictives, quelle méthode peut-on utiliser ?**
 - ...

Relisez au besoin la section [13.4.4](#).

Réponses

- Les méthodes de classification non supervisée sont appelées ainsi puisque :
 - les groupes sont inconnus au préalable et formés automatiquement par ces méthodes

- En quoi se distinguent les méthodes de classifications strictes et floues ?
 - les premières attribuent chaque observation à un seul groupe alors que les secondes évaluent le degré d'appartenance des observations à chaque groupe
- Quels paramètres doivent être définis pour exécuter l'algorithme c-means ?
 - k, soit le nombre de groupes à obtenir
 - m, soit le paramètre contrôlant le niveau de flou dans une classification floue
- La classification ascendante hiérarchique nécessite de calculer :
 - la distance entre l'ensemble des paires d'observations
 - un dendrogramme, soit une structure hiérarchique permettant de conserver l'ordre dans lequel les groupes sont formés par l'algorithme
- Pour effectuer une classification non supervisée floue utilisant comme centres de groupes de réelles observations plutôt que des moyennes fictives, quelle méthode peut-on utiliser ?
 - c-medoids

Chapitre 14

Annexes

14.1 Table des valeurs critiques de khi-deux

La courte syntaxe R ci-dessous permet de générer le tableau 14.1 avec les valeurs critiques du khi-deux 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
tableKhi2 <- cbind(dl,
  p0.10 = round(qchisq(p=0.90, df=dl, lower.tail = TRUE),3),
  p0.05 = round(qchisq(p=0.95, df=dl, lower.tail = TRUE),3),
  p0.01 = round(qchisq(p=0.99, df=dl, lower.tail = TRUE),3),
  p0.001 = round(qchisq(p=0.999, df=dl, lower.tail = TRUE),3))
# Impression du tableau avec la library stargazer
stargazer(tableKhi2, type="text", summary=FALSE, rownames=FALSE, align = TRUE, digits = 2,
  title="Distribution des valeurs critiques du Khi2")
```

TAB. 14.1 : Distribution des valeurs critiques du khi-deux

dl	p = 0,10	p = 0,05	p = 0,01	p = 0,001
1	2,71	3,84	6,64	10,83
2	4,61	5,99	9,21	13,82
3	6,25	7,82	11,35	16,27
4	7,78	9,49	13,28	18,47
5	9,24	11,07	15,09	20,52
6	10,64	12,59	16,81	22,46
7	12,02	14,07	18,48	24,32
8	13,36	15,51	20,09	26,12
9	14,68	16,92	21,67	27,88
10	15,99	18,31	23,21	29,59
11	17,27	19,68	24,73	31,26
12	18,55	21,03	26,22	32,91
13	19,81	22,36	27,69	34,53
14	21,06	23,68	29,14	36,12
15	22,31	25,00	30,58	37,70
16	23,54	26,30	32,00	39,25
17	24,77	27,59	33,41	40,79
18	25,99	28,87	34,80	42,31
19	27,20	30,14	36,19	43,82
20	28,41	31,41	37,57	45,31
21	29,61	32,67	38,93	46,80
22	30,81	33,92	40,29	48,27
23	32,01	35,17	41,64	49,73
24	33,20	36,42	42,98	51,18
25	34,38	37,65	44,31	52,62
26	35,56	38,88	45,64	54,05
27	36,74	40,11	46,96	55,48
28	37,92	41,34	48,28	56,89
29	39,09	42,56	49,59	58,30
30	40,26	43,77	50,89	59,70
40	51,80	55,76	63,69	73,40
50	63,17	67,50	76,15	86,66
100	118,50	124,34	135,81	149,45
250	279,05	287,88	304,94	324,83
500	540,93	553,13	576,49	603,45

14.2 Table des valeurs critiques de Fisher

La courte syntaxe R ci-dessous permet de générer les tableaux 14.2, 14.3 et 14.4 avec les valeurs critiques de F avec $p = 0,05$.

```
library(stargazer)

dl1 <- c(1:10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 1000)
dl2 <- c(1:10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 500, 1000, 2000)
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 <- data.frame(dl2 = dl2, matrice)
names(tableF_p0.05) <- c("dl2", paste0("dl1=",dl1))

stargazer(tableF_p0.05, type="text", summary=FALSE, rownames=FALSE, align = TRUE, digits = 3,
          title="Distribution des valeurs critiques de F avec p = 0,05")
```

TAB. 14.2 : Distribution des valeurs critiques de F avec p = 0,05

dl2	dl1=1	dl1=2	dl1=3	dl1=4	dl1=5	dl1=6	dl1=7	dl1=8	dl1=9	dl1=10
1	161,45	199,50	215,71	224,58	230,16	233,99	236,77	238,88	240,54	241,88
2	18,51	19,00	19,16	19,25	19,30	19,33	19,35	19,37	19,38	19,40
3	10,13	9,55	9,28	9,12	9,01	8,94	8,89	8,85	8,81	8,79
4	7,71	6,94	6,59	6,39	6,26	6,16	6,09	6,04	6,00	5,96
5	6,61	5,79	5,41	5,19	5,05	4,95	4,88	4,82	4,77	4,74
6	5,99	5,14	4,76	4,53	4,39	4,28	4,21	4,15	4,10	4,06
7	5,59	4,74	4,35	4,12	3,97	3,87	3,79	3,73	3,68	3,64
8	5,32	4,46	4,07	3,84	3,69	3,58	3,50	3,44	3,39	3,35
9	5,12	4,26	3,86	3,63	3,48	3,37	3,29	3,23	3,18	3,14
10	4,96	4,10	3,71	3,48	3,33	3,22	3,14	3,07	3,02	2,98
20	4,35	3,49	3,10	2,87	2,71	2,60	2,51	2,45	2,39	2,35
30	4,17	3,32	2,92	2,69	2,53	2,42	2,33	2,27	2,21	2,16
40	4,08	3,23	2,84	2,61	2,45	2,34	2,25	2,18	2,12	2,08
50	4,03	3,18	2,79	2,56	2,40	2,29	2,20	2,13	2,07	2,03
60	4,00	3,15	2,76	2,53	2,37	2,25	2,17	2,10	2,04	1,99
70	3,98	3,13	2,74	2,50	2,35	2,23	2,14	2,07	2,02	1,97
80	3,96	3,11	2,72	2,49	2,33	2,21	2,13	2,06	2,00	1,95
90	3,95	3,10	2,71	2,47	2,32	2,20	2,11	2,04	1,99	1,94
100	3,94	3,09	2,70	2,46	2,31	2,19	2,10	2,03	1,97	1,93
200	3,89	3,04	2,65	2,42	2,26	2,14	2,06	1,98	1,93	1,88
300	3,87	3,03	2,63	2,40	2,24	2,13	2,04	1,97	1,91	1,86
500	3,86	3,01	2,62	2,39	2,23	2,12	2,03	1,96	1,90	1,85
1 000	3,85	3,00	2,61	2,38	2,22	2,11	2,02	1,95	1,89	1,84
2 000	3,85	3,00	2,61	2,38	2,22	2,10	2,01	1,94	1,88	1,84

TAB. 14.3 : Distribution des valeurs critiques de F avec p = 0,05 (suite)

dl2	dl1=15	dl1=20	dl1=30	dl1=40	dl1=50	dl1=60	dl1=70	dl1=80	dl1=90
1	245,95	248,01	250,10	251,14	251,77	252,20	252,50	252,72	252,90
2	19,43	19,45	19,46	19,47	19,48	19,48	19,48	19,48	19,48
3	8,70	8,66	8,62	8,59	8,58	8,57	8,57	8,56	8,56
4	5,86	5,80	5,75	5,72	5,70	5,69	5,68	5,67	5,67
5	4,62	4,56	4,50	4,46	4,44	4,43	4,42	4,41	4,41
6	3,94	3,87	3,81	3,77	3,75	3,74	3,73	3,72	3,72
7	3,51	3,44	3,38	3,34	3,32	3,30	3,29	3,29	3,28
8	3,22	3,15	3,08	3,04	3,02	3,01	2,99	2,99	2,98
9	3,01	2,94	2,86	2,83	2,80	2,79	2,78	2,77	2,76
10	2,85	2,77	2,70	2,66	2,64	2,62	2,61	2,60	2,59
20	2,20	2,12	2,04	1,99	1,97	1,95	1,93	1,92	1,91
30	2,01	1,93	1,84	1,79	1,76	1,74	1,72	1,71	1,70
40	1,92	1,84	1,74	1,69	1,66	1,64	1,62	1,61	1,60
50	1,87	1,78	1,69	1,63	1,60	1,58	1,56	1,54	1,53
60	1,84	1,75	1,65	1,59	1,56	1,53	1,52	1,50	1,49
70	1,81	1,72	1,62	1,57	1,53	1,50	1,49	1,47	1,46
80	1,79	1,70	1,60	1,54	1,51	1,48	1,46	1,45	1,44
90	1,78	1,69	1,59	1,53	1,49	1,46	1,44	1,43	1,42
100	1,77	1,68	1,57	1,52	1,48	1,45	1,43	1,41	1,40
200	1,72	1,62	1,52	1,46	1,41	1,39	1,36	1,35	1,33
300	1,70	1,61	1,50	1,43	1,39	1,36	1,34	1,32	1,31
500	1,69	1,59	1,48	1,42	1,38	1,35	1,32	1,30	1,29
1 000	1,68	1,58	1,47	1,41	1,36	1,33	1,31	1,29	1,27
2 000	1,67	1,58	1,46	1,40	1,36	1,32	1,30	1,28	1,27

TAB. 14.4 : Distribution des valeurs critiques de F avec p = 0,05 (suite)

dl2	dl1=100	dl1=200	dl1=300	dl1=400	dl1=500	dl1=1000
1	253,04	253,68	253,89	254,00	254,06	254,19
2	19,49	19,49	19,49	19,49	19,49	19,49
3	8,55	8,54	8,54	8,53	8,53	8,53
4	5,66	5,65	5,64	5,64	5,64	5,63
5	4,41	4,39	4,38	4,38	4,37	4,37
6	3,71	3,69	3,68	3,68	3,68	3,67
7	3,27	3,25	3,24	3,24	3,24	3,23
8	2,97	2,95	2,94	2,94	2,94	2,93
9	2,76	2,73	2,72	2,72	2,72	2,71
10	2,59	2,56	2,55	2,55	2,55	2,54
20	1,91	1,88	1,86	1,86	1,86	1,85
30	1,70	1,66	1,65	1,64	1,64	1,63
40	1,59	1,55	1,54	1,53	1,53	1,52
50	1,52	1,48	1,47	1,46	1,46	1,45
60	1,48	1,44	1,42	1,41	1,41	1,40
70	1,45	1,40	1,39	1,38	1,37	1,36
80	1,43	1,38	1,36	1,35	1,35	1,34
90	1,41	1,36	1,34	1,33	1,33	1,31
100	1,39	1,34	1,32	1,31	1,31	1,30
200	1,32	1,26	1,24	1,23	1,22	1,21
300	1,30	1,23	1,21	1,20	1,19	1,17
500	1,28	1,21	1,18	1,17	1,16	1,14
1 000	1,26	1,19	1,16	1,14	1,13	1,11
2 000	1,25	1,18	1,15	1,13	1,12	1,09

14.3 Table des valeurs critiques de t

La courte syntaxe R ci-dessous permet de générer le tableau 14.5 avec les valeurs critiques de t avec $p = 0,10, 0,05, 0,01$ et $0,001$.

```
library(stargazer)

# vecteur pour les degrés de liberté de 1 à 30, puis 40 et 50
dl <- c(1:30, 40, 50, 60, 70, 80, 90, 100, 250, 500, 1000, 2500)
# 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,
  p0.10 = round(qt(p=1 - (0.10/2), df=dl),2),
  p0.05 = round(qt(p=1 - (0.05/2), df=dl),2),
  p0.01 = round(qt(p=1 - (0.01/2), df=dl),2),
  p0.001 = round(qt(p=1 - (0.001/2), df=dl),2))
# Impression du tableau avec la library stargazer
stargazer(tableT, type="text", summary=FALSE, rownames=FALSE, align = TRUE, digits = 2,
  title="Distribution des valeurs critiques de t")
```

TAB. 14.5 : Distribution des valeurs critiques de t

dl	p = 0,10	p = 0,05	p = 0,01	p = 0,001
1	6,31	12,71	63,66	636,62
2	2,92	4,30	9,92	31,60
3	2,35	3,18	5,84	12,92
4	2,13	2,78	4,60	8,61
5	2,02	2,57	4,03	6,87
6	1,94	2,45	3,71	5,96
7	1,89	2,36	3,50	5,41
8	1,86	2,31	3,36	5,04
9	1,83	2,26	3,25	4,78
10	1,81	2,23	3,17	4,59
11	1,80	2,20	3,11	4,44
12	1,78	2,18	3,05	4,32
13	1,77	2,16	3,01	4,22
14	1,76	2,14	2,98	4,14
15	1,75	2,13	2,95	4,07
16	1,75	2,12	2,92	4,01
17	1,74	2,11	2,90	3,97
18	1,73	2,10	2,88	3,92
19	1,73	2,09	2,86	3,88
20	1,72	2,09	2,85	3,85
21	1,72	2,08	2,83	3,82
22	1,72	2,07	2,82	3,79
23	1,71	2,07	2,81	3,77
24	1,71	2,06	2,80	3,75
25	1,71	2,06	2,79	3,73
26	1,71	2,06	2,78	3,71
27	1,70	2,05	2,77	3,69
28	1,70	2,05	2,76	3,67
29	1,70	2,05	2,76	3,66
30	1,70	2,04	2,75	3,65
40	1,68	2,02	2,70	3,55
50	1,68	2,01	2,68	3,50
60	1,67	2,00	2,66	3,46
70	1,67	1,99	2,65	3,44
80	1,66	1,99	2,64	3,42
90	1,66	1,99	2,63	3,40
100	1,66	1,98	2,63	3,39
250	1,65	1,97	2,60	3,33
500	1,65	1,96	2,59	3,31
1 000	1,65	1,96	2,58	3,30
2 500	1,65	1,96	2,58	3,29

Bibliographie

- Aggarwal, Charu C, Alexander Hinneburg et Daniel A Keim. 2001. « On the surprising behavior of distance metrics in high dimensional space ». In *International conference on database theory*, 420-434. Springer.
- Allcott, Hunt et Matthew Gentzkow. 2017. « Social media and fake news in the 2016 election ». *Journal of economic perspectives* 31 (2) : 211-36. <https://doi.org/10.1257/jep.31.2.211>.
- Aly, Sharif S, Jianyang Zhao, Ben Li et Jiming Jiang. 2014. « Reliability of environmental sampling culture results using the negative binomial intraclass correlation coefficient ». *SpringerPlus* 3 (1) : 40. <https://doi.org/10.1186/2193-1801-3-40>.
- Anastasopoulos, Panagiotis C, John E Haddock, Matthew G Karlaftis et Fred L Mannerling. 2012. « Analysis of urban travel times : Hazard-based approach to random parameters ». *Transportation research record* 2302 (1) : 121-129. <https://doi.org/10.3141%2F2302-13>.
- Apparicio, Philippe. 2002. « Apport des systèmes d'information géographique à l'étude de l'insertion des HLM dans les quartiers montréalais ». Thèse de doctorat, Université du Maine. <http://www.theses.fr/2002LEMA3007>.
- Apparicio, Philippe, Mathieu Carrier, Jérémie Gelb, Anne-Marie Séguin et Simon Kingham. 2016. « Cyclists' exposure to air pollution and road traffic noise in central city neighbourhoods of Montreal ». *Journal of Transport Geography* 57 : 63-69. <https://doi.org/10.1016/j.jtrangeo.2016.09.014>.
- Apparicio, Philippe, Marie-Soleil Cloutier, Anne-Marie Séguin et Josefina Ades. 2010. « Accessibilité spatiale aux parcs urbains pour les enfants et injustice environnementale. Exploration du cas montréalais ». *Revue internationale de géomatique* 20 (3) : 363-389. <http://rig.revuesonline.com/article.jsp?articleId=15208>.
- Apparicio, Philippe et Jérémie Gelb. 2020. « Cyclists' exposure to road traffic noise : A comparison of three North American and European cities ». *Acoustics* 2 (1) : 73-86. <https://doi.org/10.3390/acoustics2010006>.
- Apparicio, Philippe, Jérémie Gelb, Mathieu Carrier, Marie-Ève Mathieu et Simon Kingham. 2018. « Exposure to noise and air pollution by mode of transportation during rush hours in Montreal ». *Journal of Transport Geography* 70 : 182-192. <https://doi.org/10.1016/j.jtrangeo.2018.06.007>.
- Apparicio, Philippe, Jérémie Gelb, Anne-Sophie Dubé, Simon Kingham, Lise Gauvin et Éric Robitaille. 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. <https://doi.org/10.1186/s12942-017-0105-9>.
- Apparicio, Philippe, Jérémie Gelb, Vincent Jarry et Élaine Lesage-Mann. 2021. « Cycling in one of the most polluted cities in the world : Exposure to noise and air pollution and potential adverse health impacts in Delhi ». *International journal of health geographics* 20 (1) : 1-16. <https://doi.org/10.1186/s12942-021-00272-2>.

- Apparicio, Philippe, Jérémy Gelb et Marie-Ève Mathieu. 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. <https://doi.org/10.4000/cybergeo.32391>.
- Apparicio, Philippe, David Maignan et Jérémy Gelb. 2021. « VIFECO : An Open-source software for counting features on a Video». *Journal of Open Research Software* 9 (1). <https://doi.org/10.5334/jors.300>.
- Apparicio, Philippe, Thi-Thanh-Hien Pham, Anne-Marie Séguin et Jean Dubé. 2016. « Spatial distribution of vegetation in and around city blocks on the Island of Montreal : A double environmental inequity?» *Applied Geography* 76 : 128-136. <http://dx.doi.org/10.1016/j.apgeog.2016.09.023>.
- Atkinson, Rowland et Keith Kintrea. 2001. « Disentangling area effects : Evidence from deprived and non-deprived neighbourhoods». *Urban studies* 38 (12) : 2277-2298. <https://doi.org/10.1080/2F00420980120087162>.
- Audate, Pierre Paul, Geneviève Cloutier et Alexandre Lebel. 2021. « The motivations of urban agriculture practitioners in deprived neighborhoods : A comparative study of Montreal and Quito». *Urban Forestry & Urban Greening* 62 : 127171. <https://doi.org/10.1016/j.ufug.2021.127171>.
- Audrin, Thomas, Philippe Apparicio et Anne-Marie Séguin. 2021. « La localisation des écoles primaires et le bruit aérien dans la région métropolitaine de Toronto : un diagnostic d'équité environnementale et une analyse des impacts sur la réussite scolaire». *Canadian Journal of Regional Science/Revue canadienne des sciences régionales* 44 (1) : 22-34. https://idjs.ca/images/rCSR/archives/V44N1_5-AUDRIN-APPARICIO-SEGUIN.pdf.
- Bair, Eric. 2013. « Semi-supervised clustering methods». *Wiley Interdisciplinary Reviews : Computational Statistics* 5 (5) : 349-361. <https://dx.doi.org/10.1002%2Fwics.1270>.
- Bendixen, Mike T. 1995. « Compositional perceptual mapping using chi-squared trees analysis and correspondence analysis». *Journal of Marketing Management* 11 (6) : 571-581. <https://doi.org/10.1080/0267257X.1995.9964368>.
- Benzécri, Jean-Paul. 1973. *L'analyse des données. Tome 1. La taxinomie. Tome 2. L'analyse des correspondances.* Vol. 2. Dunod.
- Bhatt, Vikram et Leila Marie Farah. 2016. « Cultivating Montreal : A Brief History of Citizens and Institutions Integrating Urban Agriculture in the City». *Urban Agriculture & Regional Food Systems* 1 (1) : 1-12. <http://dx.doi.org/10.2134/urbanag2015.01.1511>.
- Bolker, Benjamin M, Mollie E Brooks, Connie J Clark, Shane W Geange, John R Poulsen, M Henry H Stevens et Jada-Simone S White. 2009. « Generalized linear mixed models : a practical guide for ecology and evolution». *Trends in ecology & evolution* 24 (3) : 127-135. <https://doi.org/10.1016/j.tree.2008.10.008>.
- Boulos, Maged N Kamel et Estella M Geraghty. 2020. « Geographical tracking and mapping of coronavirus disease COVID-19/severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) epidemic and associated events around the world : how 21st century GIS technologies are supporting the global fight against outbreaks and epidemics». *International journal of health geographics* 19 (1) : 1-12. <https://doi.org/10.1186/s12942-020-00202-8>.
- Brant, Rollin. 1990. « Assessing proportionality in the proportional odds model for ordinal logistic regression». *Biometrics* : 1171-1178. <https://www.jstor.org/stable/2532457>.
- Bressoux, Pascal. 2010. *Modélisation statistique appliquée aux sciences sociales*. De boeck.
- Burdenski, Jr Thomas K. 2000. « Evaluating univariate, bivariate, and multivariate normality using graphical procedures». *ERIC* : 1-62.

- Buregeya, Jean Marie, Philippe Apparicio et Jérémie Gelb. 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. <https://dx.doi.org/10.3390%2Fijerph17041220>.
- Carrier, Mathieu, Philippe Apparicio, Anne-Marie Séguin et Dan Crouse. 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. <https://doi.org/10.1016/j.cstp.2014.06.003>.
- Chandra, Mahalanobis Prasanta. 1936. « On the generalised distance in statistics ». In *Proceedings of the National Institute of Sciences of India*, 2 :49-55. 1.
- Chang, Winston. 2018. *R Graphics Cookbook, 2nd edition*. CRC Press.
- Cloutier, Marie-soleil, Mathieu Tremblay, Patrick Morency et Philippe Apparicio. 2014. « Carrefours en milieu urbain : quels risques pour les piétons ? Exemple empirique des quartiers centraux de Montréal, Canada ». *Recherche Transports Sécurité* 30 : 3-20.
- Cohen, Jacob. 1992. « A power primer ». *Psychological bulletin* 112 (1) : 155-159. <https://doi.org/10.1037/0033-295X.112.1.155>.
- . 2013. *Statistical power analysis for the behavioral sciences*. Academic press.
- De Alvarenga, Bernardo, Philippe Apparicio et Anne-Marie Séguin. 2018. « L'accessibilité aux aires de jeux dans les parcs de la Communauté métropolitaine de Montréal ». *Cahiers de géographie du Québec* 62 (176) : 229-246. <https://doi.org/10.7202/1063104ar>.
- Delaunay, Déborah, Philippe Apparicio, Anne-Marie Séguin, Jérémie Gelb et Mathieu Carrier. 2019. « L'identification des zones calmes et un diagnostic d'équité environnementale à Montréal ». *The Canadian Geographer/Le Géographe canadien* 63 (2) : 184-197. <https://doi.org/10.1111/cag.12511>.
- Dunn, Peter K. et Gordon K. Smyth. 1996. « Randomized Quantile Residuals ». *Journal of Computational and Graphical Statistics* 5 (3) : 236-244. <https://doi.org/10.2307/1390802>.
- Escofier, Brigitte. 1979. « Traitement simultané de variables qualitatives et quantitatives en analyse factorielle ». *Cahiers de l'Analyse des Données* 4 (2) : 137-146.
- Escofier, Brigitte et Jérôme Pagès. 1998. « Analyses factorielles simples et multiples ». *Dunod, Paris*.
- Field, Andy P, Jeremy Miles et Zoë Field. 2012. « Discovering statistics using R ». Thousand Oaks.
- Fox, John et Georges Monette. 1992. « Generalized collinearity diagnostics ». *Journal of the American Statistical Association* 87 (417) : 178-183. <https://doi.org/10.2307/2290467>.
- Frank, Lawrence, Mark Bradley, Sarah Kavage, James Chapman et T Keith Lawton. 2008. « Urban form, travel time, and cost relationships with tour complexity and mode choice ». *Transportation* 35 (1) : 37-54. <https://doi.org/10.1007/s11116-007-9136-6>.
- Fukuyama, Yoshiki. 1989. « A new method of choosing the number of clusters for the fuzzy c-mean method ». In *Proc. 5th Fuzzy Syst. Symp.*, 1989, 247-250.
- Gelb, Jérémie et Philippe Apparicio. 2019. « Noise exposure of cyclists in Ho Chi Minh City : A spatio-temporal analysis using non-linear models ». *Applied Acoustics* 148 : 332-343. <https://doi.org/10.1016/j.apacoust.2018.12.031>.
- Gelb, Jérémie et Philippe Apparicio. 2020. « Modelling cyclists' multi-exposure to air and noise pollution with low-cost sensors : The case of Paris ». *Atmosphere* 11 (4) : 422. <https://doi.org/10.3390/atmos11040422>.

- . 2021a. « Cyclists' exposure to atmospheric and noise pollution : a systematic literature review ». *Transport Reviews* : 1-24. <https://doi.org/10.1080/01441647.2021.1895361>.
- . 2021b. « Apport de la classification floue c-means spatiale en géographie : essai de taxinomie socio-résidentielle et environnementale à Lyon ». *Cybergeo : European Journal of Geography*. <https://doi.org/10.4000/cybergeo.36414>.
- Gelman, Andrew. 2005. « Analysis of variance—why it is more important than ever ». *The annals of statistics* 33 (1) : 1-53. <https://doi.org/10.1214/009053604000001048>.
- Gelman, Andrew et Jennifer Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.
- Gilles, Alain et Pierre Maranda. 1994. *Éléments de méthodologie et d'analyse statistique pour les sciences sociales*. McGraw-Hill.
- Glass, Gene V, Percy D Peckham et James R Sanders. 1972. « Consequences of failure to meet assumptions underlying the fixed effects analyses of variance and covariance ». *Review of educational research* 42 (3) : 237-288.
- Gower, John C. 1971. « A general coefficient of similarity and some of its properties ». *Biometrics* : 857-871. <https://doi.org/10.2307/2528823>.
- Hanck, Christoph, Martin Arnold, Alexander Gerber et Martin Schmelzer. 2019. *Introduction to econometrics with R*. <https://www.econometrics-with-r.org/ITER.pdf>.
- Harlan, Sharon L, Anthony J Brazel, G Darrel Jenerette, Nancy S Jones, Larissa Larsen, Lela Prashad et William L Stefanov. 2007. « In the shade of affluence : the inequitable distribution of the urban heat island ». *Research in Social Problems and Public Policy* 15 : 173-202. [http://dx.doi.org/10.1016/S0196-1152\(07\)15005-5](http://dx.doi.org/10.1016/S0196-1152(07)15005-5).
- Hilbe, Joseph M. 2009. *Logistic regression models*. CRC press.
- Hotelling, Harold. 1933. « Analysis of a complex of statistical variables into principal components ». *Journal of educational psychology* 24 (6) : 417. <https://psycnet.apa.org/doi/10.1037/h0071325>.
- Huang, Ganlin, Weiqi Zhou et ML Cadenasso. 2011. « Is everyone hot in the city ? Spatial pattern of land surface temperatures, land cover and neighborhood socioeconomic characteristics in Baltimore, MD ». *Journal of environmental management* 92 (7) : 1753-1759. <https://doi.org/10.1016/j.jenvman.2011.02.006>.
- Hubert, Mia, Peter J Rousseeuw et Karlien Vanden Branden. 2005. « ROBPCA : a new approach to robust principal component analysis ». *Technometrics* 47 (1) : 64-79. <https://doi.org/10.1198/004017004000000563>.
- Ismay, Chester et Albert Y Kim. 2019. *Statistical inference via data science : a ModernDive into R and the tidyverse*. CRC Press.
- Joanes, DN et CA Gill. 1998. « Comparing measures of sample skewness and kurtosis ». *Journal of the Royal Statistical Society : Series D (The Statistician)* 47 (1) : 183-189. <https://www.jstor.org/stable/2988433>.
- Kaiser, Henry F. 1960. « The application of electronic computers to factor analysis ». *Educational and psychological measurement* 20 (1) : 141-151. <https://doi.org/10.1177%2F001316446002000116>.
- Lebart, Ludovic, Alain Morineau et Marie Piron. 1995. *Statistique exploratoire multidimensionnelle*. Dunod.
- Lix, Lisa M, Joanne C Keselman et HJ Keselman. 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. <https://doi.org/10.3102/00346543066004579>.

- MacQueen, James. 1967. « Some methods for classification and analysis of multivariate observations ». In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1 :281-297. 14. Oakland, CA, USA.
- McClintock, Nathan. 2018. « Urban agriculture, racial capitalism, and resistance in the settler-colonial city ». *Geography Compass* 12 (6) : 1-16. <https://doi.org/10.1111/gec3.12373>.
- McElreath, Richard. 2020. *Statistical rethinking : A Bayesian course with examples in R and Stan*. CRC press.
- McFadden, Brandon R. 2016. « Examining the gap between science and public opinion about genetically modified food and global warming ». *PLoS one* 11 (11) : e0166140. <https://doi.org/10.1371/journal.pone.0166140>.
- Messerli, Franz H. 2012. « Chocolate consumption, cognitive Function, and Nobel laureates ». *The new England Journal of Medicine* 367 (16) : 1563-1564. <https://doi.org/10.1056/nejmon1211064>.
- Mihalcea, Rada et Paul Tarau. 2004. « Textrank : Bringing order into text ». In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404-411.
- Nakagawa, Shinichi, Paul C. D. Johnson et Holger Schielzeth. 2017. « The coefficient of determination R² and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded ». *Journal of The Royal Society Interface* 14 (134) : 20170213. <https://doi.org/10.1098/rsif.2017.0213>.
- Nelder, John A. et Robert W. M. Wedderburn. 1972. « Generalized Linear Models ». *Journal of the Royal Statistical Society. Series A (General)* 135 (3) : 370-384. <http://www.jstor.org/stable/2344614>.
- Neyman, Jerzy, Egon Sharpe Pearson et Karl Pearson. 1933. « IX. On the problem of the most efficient tests of statistical hypotheses ». *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231 (694-706) : 289-337. <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1933.0009>.
- Pagès, Jérôme. 2002. « Analyse factorielle multiple appliquée aux variables qualitatives et aux données mixtes ». *Revue de statistique appliquée* 50 (4) : 5-37.
- . 2013. *Analyse factorielle multiple avec R*. EDP sciences.
- Pham, Thi-Thanh-Hien, Philippe Apparicio, Shawn Landry et Joseph Lewnard. 2017. « Disentangling the effects of urban form and socio-demographic context on street tree cover : A multi-level analysis from Montréal ». *Landscape and Urban Planning* 157 : 422-433. <http://dx.doi.org/10.1016/j.landurbplan.2016.09.001>.
- Pham, Thi-Thanh-Hien, Philippe Apparicio, Anne-Marie Séguin, Shawn Landry et Martin Gagnon. 2012. « Spatial distribution of vegetation in Montreal : an uneven distribution or environmental inequity ? » *Landscape and urban planning* 107 (3) : 214-224. <http://dx.doi.org/10.1016/j.landurbplan.2012.06.002>.
- Philibert, Mathieu D et Philippe Apparicio. 2007. « Statistiques spatiales appliquées à l'analyse de données de santé ». In *Géographie de la santé : un panorama*, 111-132. Economica.
- Pumain, Denise et Michèle Béguin. 1994. *La représentation des données géographiques : statistique et cartographie*. Armand Colin.
- Raudenbush, Stephen W et Anthony S Bryk. 2002. *Hierarchical linear models : Applications and data analysis methods*. Vol. 1. Sage.
- Razali, Nornadiah Mohd et Yap Bee Wah. 2011. « Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests ». *Journal of statistical modeling and analytics* 2 (1) : 21-33.

- Reed, William J. 2002. « On the rank-Size distribution for human settlements ». *Journal of Regional Science* 42 (1) : 1-17. <https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9787.00247>.
- Reed, William J. et Murray Jorgensen. 2004. « The double Pareto-Lognormal distribution : A new parametric model for size distributions ». *Communications in Statistics - Theory and Methods* 33 (8). Taylor & Francis : 1733-1753. <https://doi.org/10.1081/STA-120037438>.
- Rivest, Louis-Paul et Nathalie Plante. 1988. « L'analyse en composantes principales robuste ». *Revue de statistique appliquée* 36 (1) : 55-66. http://www.numdam.org/article/RSA_1988__36_1_55_0.pdf.
- Roback, Paul et Julie Legler. 2021. *Beyond multiple linear regression : Applied generalized linear models and multilevel models in R*. CRC Press.
- Sanchez, Lino et Tony G Reames. 2019. « Cooling Detroit : A socio-spatial analysis of equity in green roofs as an urban heat island mitigation strategy ». *Urban Forestry & Urban Greening* 44 : 126331. <https://doi.org/10.1016/j.ufug.2019.04.014>.
- Sanders, Lena. 1989. *L'analyse statistique des données en géographie*. GIP Reclus.
- SAS Institute Inc. 2020a. « SAS/STAT 15.2 User's Guide Modeling Multinomial Overdispersion ». https://documentation.sas.com/?docsetId=statug&docsetTarget=statug_fmm_examples04.htm&docsetVersion=15.2&locale=en.
- . 2020b. « SAS/STAT 15.2 User's Guide Poisson Regression | SAS Annotated Output ». <https://stats.idre.ucla.edu/sas/output/poisson-regression/>.
- Sawilowsky, Shlomo S. 2009. « New effect size rules of thumb ». *Journal of Modern Applied Statistical Methods* 8 (2) : 467-474. <https://doi.org/10.22237/jmasm/1257035100>.
- Schwarzkopf, Dietrich, Benjamin de Haas et Geraint Rees. 2012. « Better ways to improve standards in brain-behavior correlation analysis ». *Frontiers in Human Neuroscience* 6 : 200. <https://www.frontiersin.org/articles/10.3389/fnhum.2012.00200>.
- Sean, Owen. 2018. « Common probability distributions ». <https://medium.com/@srowen/common-probability-distributions-347e6b945ce4>.
- Smithson, Michael et Jay Verkuilen. 2006. « A better lemon squeezer? Maximum-likelihood regression with beta-distributed dependent variables ». *Psychological methods* 11 (1) : 54. <https://doi.org/10.1037/1082-989x.11.1.54>.
- Sobol, IM. 1993. « Sensitivity estimates for nonlinear mathematical models ». *Mathematics and Computers in Simulation* 1 (4) : 407-414.
- Spyratos, Spyridon, Michele Vespe, Fabrizio Natale, Ingmar Weber, Emilio Zagheni et Marzia Rango. 2019. « Quantifying international human mobility patterns using Facebook Network data ». *PloS one* 14 (10) : e0224134. <https://doi.org/10.1371/journal.pone.0224134>.
- Stryhn, H, J Sanchez, P Morley, C Booker et IR Dohoo. 2006. « Interpretation of variance parameters in multilevel Poisson regression models ». In *Proceedings of the 11th International Symposium on Veterinary Epidemiology and Economics*. Vol. 702.
- Tabachnick, Barbara G, Linda S Fidell et Jodie B Ullman. 2007. *Using multivariate statistics*. Pearson.
- Teubner, Timm, Florian Hawlitschek et David Dann. 2017. « Price determinants on AirBnB : How reputation pays off in the sharing economy ». *Journal of Self-Governance & Management Economics* 5 (4). <http://dx.doi.org/10.22381/JSME5420173>.

- Tibshirani, Robert, Guenther Walther et Trevor Hastie. 2001. « Estimating the number of clusters in a data set via the gap statistic ». *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 63 (2) : 411-423. <https://doi.org/10.1111/1467-9868.00293>.
- Wang, Dan et Juan L Nicolau. 2017. « Price determinants of sharing economy based accommodation rental : A study of listings from 33 cities on Airbnb.com ». *International Journal of Hospitality Management* 62 : 120-131. <https://doi.org/10.1016/j.ijhm.2016.12.007>.
- Ward, Joe H. 1963. « Hierarchical grouping to optimize an objective function ». *Journal of the American statistical association* 58 (301) : 236-244. <https://doi.org/10.1080/01621459.1963.10500845>.
- Wickham, Hadley. 2010. « A layered grammar of graphics ». *Journal of Computational and Graphical Statistics* 19 (1) : 3-28. <http://dx.doi.org/10.1198/jcgs.2009.07098>.
- Wilcox, Rand R. 1994. « The percentage bend correlation coefficient ». *Psychometrika* 59 (4) : 601-616. <https://doi.org/10.1007/BF02294395>.
- Wood, Simon N. 2004. « Stable and efficient multiple smoothing parameter estimation for generalized additive models ». *Journal of the American Statistical Association* 99 (467) : 673-686. <https://doi.org/10.1198/0162145040000000980>.
- Wu, Sheng, Catherine M Crespi et Weng Kee Wong. 2012. « Comparison of methods for estimating the intraclass correlation coefficient for binary responses in cancer prevention cluster randomized trials ». *Contemporary clinical trials* 33 (5) : 869-880. <https://doi.org/10.1016/j.cct.2012.05.004>.
- Xie, Xuanli Lisa et Gerardo Beni. 1991. « A validity measure for fuzzy clustering ». *IEEE Transactions on pattern analysis and machine intelligence* 13 (8) : 841-847. <https://doi.org/10.1109/34.85677>.
- Xie, Yihui. 2016. *Bookdown : authoring books and technical documents with R markdown*. CRC Press.
- Yap, Bee Wah et Chiaw Hock Sim. 2011. « Comparisons of various types of normality tests ». *Journal of Statistical Computation and Simulation* 81 (12) : 2141-2155. <https://doi.org/10.1080/00949655.2010.520163>.
- Zeileis, Achim. 2004. *Econometric computing with HC and HAC covariance matrix estimators*. Institut für Statistik und Mathematik.
- Zhang, Zhihua, Rachel JC Chen, Lee D Han et Lu Yang. 2017. « Key factors affecting the price of Airbnb listings : A geographically weighted approach ». *Sustainability* 9 (9) : 1635. <http://dx.doi.org/10.3390/su9091635>.