

Faculté des sciences

Modélisation de la mortalité en tenant compte des facteurs climatiques

Auteurs : Zeumo Stanis Lassi

Promoteur : Karim Barigou

Lecteur : Donatien Hainaut

Année académique 2025-2026

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
INSTITUT DE STATISTIQUE, BIOSTATISTIQUE ET ACTUARIAT

Modélisation de la mortalité en tenant compte des facteurs climatiques

Mémoire présenté en vue de l'obtention du
Master en Actuariat

par
Stanis Lassi Zeumo

Promoteur : Pr. Karim Barigou

Année universitaire 2025–2026

Résumé

L'espérance de vie de la population européenne est en pleine croissance, en raison des progrès médicaux et de l'amélioration des conditions de vie. Cependant, l'impact des facteurs tels que l'environnement, le statut social, l'éducation ou encore le climat se fait de plus en plus ressentir sur la mortalité à travers des périodes de surmortalité telles que les vagues de chaleur, les épisodes de froid intense, les pandémies (comme la Covid-19) ou encore les inégalités socio-économiques. Dans ce mémoire, nous évaluons l'impact de la température sur la mortalité à travers l'analyse de deux modèles : le modèle de Lee-Carter (1992), largement utilisé et reposant sur un facteur temporel, et le modèle de Seklecka et al.(2017) qui prend en compte un facteur lié à la température via un coefficient de corrélation entre la mortalité et la température. Notre étude s'appuie sur des données de la population française couvrant la période 2000-2019, disponibles par classe d'âge et par sexe. Nous analysons cet impact sur trois types de périodes : annuelle (toutes les semaines de l'année), froide (les semaines allant d'octobre à avril) et chaude (les semaines allant de mai à septembre). Les résultats montrent que le modèle de Seklecka et al. donne de meilleures performances en termes d'ajustement et de prédiction que celui de Lee-Carter. Enfin, une application à la tarification d'un produit de rente est proposée à titre illustratif de la pertinence de la prise en compte de ce facteur.

Mots clés : Changement climatique, modèle de Seklecka et al., modèle de Lee-Carter, prédiction, tarification d'un produit de rente, population française.

Abstract

Life expectancy of the European population is growing rapidly due to medical advances and improvements in living conditions. However, the impact of factors such as the environment, social status, education, and climate change is increasingly affecting mortality through periods of excess mortality such as heatwaves, episodes of extreme cold, pandemics (such as COVID-19), and socio-economic inequality. In this thesis, we assess the impact of temperature on mortality through the analysis of two models : the widely used Lee-Carter model (1992), which relies on a time factor, and the Seklecka et al. (2017) model, which incorporates a temperature-related factor through a correlation coefficient between mortality and temperature. Our study is based on data from the French population covering the period 2000-2019, available by age group and sex. We analyse this impact over three types of periods : annual (for every week of the year), cold (for weeks from October to April), and warm (for weeks from May to September). The results show that the Seklecka et al. model provides better performance in terms of fit and prediction than the Lee-Carter model. Finally, an application to the pricing of an annuity product is proposed as an illustration of the relevance of taking this factor into account.

Keywords : Climate change, Seklecka et al. model, Lee-Carter model, prediction, annuity pricing, French population.

Remerciements

Je remercie Dieu pour le souffle de vie, pour toutes les grâces dont j'ai bénéficié tout au long de la production de ce mémoire et dont je continue de bénéficier. Je tiens à exprimer mes sincères remerciements à toutes les personnes qui ont contribué à la réalisation de ce mémoire grâce à leur soutien multiforme.

- ♣ Tout d'abord, je remercie mon directeur de mémoire Pr. Karim Barigou, pour l'encadrement qu'il m'a offert, pour son soutien constant et ses précieux conseils tout au long de ce projet. Sa patience, sa disponibilité et sa rigueur m'ont permis d'avancer sereinement dans mes recherches et de réaliser ce mémoire dans de très bonnes conditions. Votre présence à mes côtés a été un véritable moteur pour moi.
- ♣ Je remercie les éminents membres du jury qui ont sacrifié leur temps pour juger la qualité de ce travail.
- ♣ Je souhaite également remercier tous les enseignants de l'institut des Statistiques, Biostatistique et Sciences Actuarielles pour les enseignements de qualité dont j'ai bénéficié tout au long de mon parcours universitaire.

Je remercie,

- ♣ Mes très chers parents Dongmo Benoît et Nzangue Eveline Philippine Françoise pour l'amour inconditionnel qu'ils me portent, pour la bonne éducation que j'ai reçue et pour tout le soutien tant sur le plan financier que moral.
- ♣ Mon frère Azangue Dongmo Lionel Ulrich et mes sœurs Djiofack Dongmo Aïcha Georgette et Mafodem Emma Princesse pour votre présence continue et pour l'amour que vous m'avez apporté.
- ♣ Mes oncles et tantes, en particulier, Maadoumgang Djiofack Cecile, Tsopmo Esther pour les encouragements et le soutien financier.
- ♣ Ma grand-mère et ma copine pour toutes les prières qu'elles font pour moi.
- ♣ Je remercie ensuite mes amis et camarades de promotion, en particulier, Fongang Rhoderic, Benjamin Minko, Leukemo Marselo, Makeng Dimitri Angelo, Gerard Dongmo pour l'assistance académique et pour l'esprit d'équipe.
- ♣ Enfin, je voudrais exprimer ma gratitude envers toutes les personnes qui m'ont soutenu de près ou de loin dans l'élaboration de ce mémoire.

Table des matières

Introduction générale	1
1 Évolution des modèles de mortalité et Mise en contexte	3
1.1 Introduction	3
1.2 Définitions et notations	3
1.2.1 Définitions	3
1.2.2 Notations	4
1.3 Modèles de mortalité	4
1.3.1 Évolution de la première génération des modèles de mortalité : modèles non-stochastiques	4
1.3.2 Évolution de la deuxième génération des modèles de mortalité : modèles stochastiques	5
1.4 Formule générale des modèles de mortalité stochastiques	8
1.5 Climat et mortalité	8
1.5.1 Le modèle de Seklecka-Pantelous-O'hare (SPO)	8
1.5.2 Modèle de Gavakshi et Jason	8
1.5.3 Modèle de Jens Robben, Katrien Antonio et Torsten Kleinow	9
1.6 Conclusion	10
2 Étude de la corrélation entre la mortalité et la température	11
2.1 Base de données de mortalité	11
2.2 Base de données de température	12
2.3 Évaluation de la corrélation entre mortalité et température	14
2.4 Conclusion	16
3 Application et estimation des paramètres des modèles LC et SPO	17
3.1 Introduction	17
3.2 Estimation des paramètres du modèle de Lee-Carter(LC)	17
3.3 Estimation des paramètres du modèle SPO	19
3.4 Qualité d'ajustement des modèles LC et SPO	23

TABLE DES MATIÈRES

3.4.1	Visualisation de l'ajustement des deux modèles par classe d'âge	23
3.4.2	Evaluation des erreurs des modèles	24
3.5	Robustesse du modèle SPO mixte	25
3.6	Projection des taux de mortalité des modèles LC et SPO	27
3.6.1	Projection du modèle LC	28
3.6.2	Projection du modèle SPO	33
3.6.3	Qualité d'ajustement hors-échantillon des modèles LC et SPO	35
3.7	Conclusion	36
4	Impact des modèles de mortalité intégrant la température sur la tarification d'un portefeuille de rentiers	37
4.1	Introduction	37
4.2	Construction du produit de rente	37
4.3	Méthodologie de la comparaison	38
4.4	Calcul Numérique de la rente	39
4.4.1	Comparaison des prix de rente	39
4.4.2	Analyse de l'erreur relative	42
4.5	Conclusion	43
	Conclusion générale	45
	Annexe	iii
	Code R complet	vii

Table des figures

2.1	Évolution des taux de mortalité brutes par sexe et par classe d'âge.	12
2.2	Évolution des températures (TAVG, TMIN, TMAX).	14
2.3	Coefficients de corrélation de Pearson et de Spearman en fonction de l'âge. . . .	15
3.1	Estimation des paramètres du modèle de LC chez les hommes	19
3.2	Estimation des paramètres du modèle de LC chez les femmes	19
3.3	Comparaison des quatre composantes κ_t chez les hommes pour les cas annuel, chaud et froid.	22
3.4	Comparaison des quatre composantes κ_t chez les femmes pour les cas annuel, chaud et froid.	22
3.5	Ajustement des modèles LC et SPO.	23
3.6	Coefficients de corrélation de Pearson pour homme et femme.	26
3.7	Comparaison entre les Coefficients de corrélation de Pearson pour homme et femme avec et sans perturbation.	27
3.8	ACF des séries brutes chez les hommes et femmes pour les cas annuel, chaud et froid.	29
3.9	Séries sans saisonnalités chez les hommes et femmes pour les cas annuel, chaud et froid.	29
3.10	Séries stationnaires chez les hommes et femmes pour les cas annuel, chaud et froid.	30
3.11	ACF et PCAF des séries nettoyées.	30
3.12	Résidus standardisés pour homme froid.	32
3.13	Projections hors-échantillon des taux de mortalité chez les hommes et femmes pour le cas annuel avec LC sur 2017-2019.	33
3.14	Projections Out-of-sample des taux de mortalité chez les hommes et femmes pour le cas annuel avec SPO sur 2017-2019.	34
4.1	Évolution des prix de rentes chez les hommes sur la période 2000-2016 en fonction des horizons	41
4.2	Évolution des prix de rentes chez les femmes sur la période 2000-2016 en fonction des horizons.	41

TABLE DES FIGURES

4.3	Erreur relative des valeurs actuarielles des rentes en fonction des horizons pour différentes classes d'âges issues des modèles LC et SPO sur la période 2000-2016 chez les hommes.	42
4.4	Erreur relative des valeurs actuarielles des rentes en fonction des horizons pour différentes classes d'âges issues des modèle LC et SPO sur la période 2000-2016 chez les femmes.	43
4.5	Résidus standardisés chez les hommes et femmes pour les cas annuel, chaud et froid.	v

Liste des tableaux

3.1	Meilleur choix de a pour le modèle SPO mixte	21
3.2	Qualité d'ajustement des modèles LC et SPO	24
3.3	Tableau des meilleurs modèles de SARIMA	31
3.4	Structure des processus ARIMA/SARIMA pour la projection des séries stochas- tiques κ_t du modèle SPO	34
3.5	Qualité d'ajustement des modèles LC et SPO sur les données test 2017-2019 . .	35
4.1	Prix des rentes mensuelles (payant un euro à la fin de chaque mois) issus des données brutes et des modèles LC et SPO à horizon différent pour les âges 30ans et 50 ans.	40
4.3	Qualité d'ajustement des modèles LC et SPO sur la période 2000-2010	iii
4.2	Corrélations mortalité-température avec p-values significatives : *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$	iv
4.4	Tableau récapitulatif des meilleurs modèles	vi

Introduction générale

Au cours des dernières décennies, les progrès médicaux et technologiques ont conduit à une augmentation significative de l'espérance de vie. Cette évolution contraint les assureurs à adapter régulièrement leurs modèles de mortalité. En effet, une mauvaise estimation de la longévité peut avoir un impact important sur la valorisation des produits tels que les rentes pour lesquels l'assureur s'engage à verser des paiements réguliers jusqu'au décès de l'assuré. Cairns, Blake et Dowd (2008) ont étudié ce risque de longévité et soulignent dans leur article que la modélisation et la gestion du risque de mortalité constituent des enjeux majeurs pour les assureurs et les fonds de pensions.

De nombreux modèles de mortalité ont été développés au cours du temps afin de modéliser et prévoir l'évolution de la mortalité. Parmi eux, le modèle de Lee-Carter introduit par Lee et Carter (1992), constitue une référence largement utilisée en pratique. Ce modèle a par la suite connu plusieurs extensions. D'autre part, le modèle de Cairns-Blake-Dowd (CBD), formulé par Cairns, Blake et Dowd (2006), s'est également imposé comme une alternative importante, notamment pour la modélisation de la mortalité aux âges élevés, et a lui aussi fait l'objet de nombreuses extensions.

Cependant, ces modèles ne prennent pas en compte certains facteurs exogènes tels que le changement climatique, les facteurs environnementaux, ou encore les déterminants socio-économiques. Pourtant, les estimations de la surmortalité liées au changement climatique, notamment celles de l'Organisation mondiale de la Santé (OMS), mettent en évidence un impact non-négligeable sur la mortalité humaine. L'OMS estime ainsi que le changement climatique pourrait entraîner environ 250 000 décès supplémentaires par an entre 2030 et 2050 à l'échelle mondiale, dus à des facteurs tels que le paludisme, la malnutrition, la diarrhée et le stress thermique.

Par ailleurs, de nombreux travaux empiriques soulignent l'influence des facteurs socio-économiques sur la mortalité. À ce titre, Feliciangeli, Kerkhof et Vanduffel([5]) (2025) montrent que celle-ci dépend fortement des variables telles que le niveau de revenu, le niveau d'éducation ou encore le statut professionnel. L'ensemble de ces éléments remet en question l'hypothèse d'un allongement continu et uniforme de l'espérance de vie. Plus récemment, la crise du Covid-19 a également conduit certains chercheurs comme Cairns, Blake, Kessler(2020) à s'interroger sur

ses effets à court et à moyen terme sur la mortalité, d'où l'intérêt d'introduire des variables explicatives comme la température.

Dans ce mémoire, nous cherchons à analyser l'impact du changement climatique sur la mortalité de la population française, en étudiant l'apport de l'intégration de variables liées à la température dans la modélisation du risque de mortalité. Pour cela, nous allons faire une étude comparative entre le modèle populaire de Lee-Carter et le modèle de Seklecka et al. (2017) qui est le premier modèle dans la littérature à prendre en compte un facteur climatique, plus précisément la température.

Cette étude est menée à partir de données hebdomadaires françaises couvrant la période 2000-2019 et distinguées selon trois sous-ensembles : annuel, froid et chaud.

Le mémoire s'articule autour de quatre chapitres. Le premier chapitre présente une revue des modèles de mortalité, en distinguant les approches statiques et stochastiques. Il met également en évidence l'intérêt d'utiliser des données de fréquence plus élevée (hebdomadaire ou mensuelle) plutôt que des données annuelles comme dans Seklecka et al. En effet, les données annuelles cachent de nombreuses informations telles que la saisonnalité, le phénomène de compensation, les effets à court terme. Dans ce mémoire, nous essayons de capter au mieux ces effets. Le deuxième chapitre est consacré à l'étude de la corrélation entre la mortalité et la température. Cette analyse constitue une étape essentielle, dans la mesure où elle permet de justifier l'intégration de la température comme variable explicative dans la modélisation de la mortalité. Le troisième chapitre débute par une analyse des corrélations par âge, afin d'identifier les classes d'âge pour lesquelles l'effet de la température est significatif. Les modèles de Lee-Carter et de Seklecka et al. sont ensuite ajustés sur la période 2000-2016 (in-sample), et leur performance est évaluée à l'aide des critères tels que le RMSE, le MAD et le MAPE. Une analyse de robustesse est également menée, en considérant des sous-périodes et en introduisant des perturbations sur la température. Par ailleurs, un réajustement du modèle est effectué afin d'évaluer la stabilité des résultats. Enfin, une modélisation des paramètres par des processus ARIMA ou SARIMA est réalisée, en fonction de la saisonnalité afin de projeter les taux de mortalité sur la période 2017-2019. Le quatrième chapitre est consacré à l'application actuarielle, à travers la tarification d'une rente temporaire versant un montant unitaire à la fin de chaque mois, tant que l'assuré est en vie.

ÉVOLUTION DES MODÈLES DE MORTALITÉ ET MISE EN CONTEXTE

1.1 Introduction

La mortalité constitue un élément fondamental en assurance vie et en démographie, car elle détermine la durée de versement des prestations. Toutefois, son évolution future reste incertaine et peut s'écarter des prévisions établies à partir des données passées. Cette incertitude est appelée risque de mortalité. Depuis plusieurs décennies, de nombreux modèles statistiques ont été développés afin de décrire et de prévoir les taux de mortalité selon l'âge et le temps. L'objectif de ce chapitre est de présenter par ordre chronologique les principaux modèles de mortalité de la littérature scientifique et actuarielle, et décrire la manière dont les paramètres des modèles sont estimés à partir des données d'expérience.

1.2 Définitions et notations

1.2.1 Définitions

La modélisation de la mortalité est une approche statistique et mathématique permettant d'analyser, de décrire et de prévoir les taux de mortalité d'une population donnée. Son objectif est de permettre la compréhension des tendances et des facteurs qui influencent la mortalité. Ces facteurs peuvent être : l'âge, le sexe, la période historique, les cohortes, les progrès médicaux, les catastrophes naturelles, les changements climatiques. Elle permet de faire des projections utiles dans plusieurs domaines notamment la santé publique, les sciences sociales, la démographie et l'assurance vie.

Le risque de mortalité est, par définition, le risque lié à l'incertitude sur l'évolution des taux de mortalité. Selon la Directive Solvabilité II, une variation défavorable des taux de mortalité peut entraîner une hausse ou une baisse des provisions techniques selon la nature des contrats considérés, par exemple les contrats d'épargne ou de retraite.

1.2.2 Notations

Dans toute la suite de ce mémoire, nous considérerons les notations suivantes :

- $N(x, t)$ est le nombre d'individus ayant survécu à l'âge x au début de la période t considérée ;
- $D(x, t)$ est le nombre d'individus décédés à l'âge x dans la période t ;
- $E(x, t)$ est l'exposition au risque, i.e., le nombre d'individus exposés au risque à l'âge x pendant l'année t .
- μ_x est le taux instantané de mortalité ;
- $m_{x,t}$ est le taux central de mortalité et se calcule comme le rapport entre le nombre de décès $D(x, t)$ et l'exposition $E(x, t)$ à l'âge x dans la période t :

$$m_{x,t} = \frac{D(x, t)}{E(x, t)}. \quad (1.1)$$

1.3 Modèles de mortalité

1.3.1 Évolution de la première génération des modèles de mortalité : modèles non-stochastiques

Abraham De Moivre est le premier à avoir proposé une formule mathématique pour la modélisation de la mortalité en 1725. Il a proposé un modèle reposant sur une distribution uniforme des décès et a présenté des méthodes simplifiées de calcul des rentes. La loi de De Moivre a un seul paramètre appelé l'âge ultime. Beaucoup de critiques ont été émises dans la littérature sur cette hypothèse. Son modèle est donné par l'équation (1.2) avec w l'âge ultime et x l'âge des individus.

$$\mu_x = \frac{1}{w - x}. \quad (1.2)$$

Ensuite vient le modèle de Gompertz, proposé en 1825 par Benjamin Gompertz qui stipule que la mortalité augmente de manière exponentielle avec l'âge. L'équation de base du modèle est donnée par (1.3) avec α et β des paramètres positifs, x l'âge. Ce modèle permet de prévoir la mortalité future en fonction des tendances passées.

$$\mu_x = \alpha \exp(\beta x). \quad (1.3)$$

William Makeham, en 1867, ajoute une constante supplémentaire au modèle de Gompertz pour tenir compte des causes de décès non liées à l'âge. Ces deux modèles ont beaucoup été utilisés comme modèles standards. Thorvald Thiele, en 1871, a introduit les concepts d'équations différentielles dans la modélisation de la mortalité. Le modèle est donné par l'équation (1.4).

$$\mu_x = \alpha_1 \exp(-\beta x) + \alpha_2 \exp\left(-\frac{1}{2}\beta_2(x - C)^2\right) + \alpha_3 \exp(\beta_3 x). \quad (1.4)$$

En 1998, Thatcher et ses collaborateurs ont réalisé des études visant à ajuster différents modèles mathématiques à différents ensembles de données fiables sur la mortalité des adultes et des personnes âgées (80 ans et plus). Ils ont par la suite établi le modèle logistique comme le meilleur modèle mathématique de la mortalité aux âges adultes et élevés, remplaçant les modèles de Gompertz et Makeham, largement utilisés à ce stade. Le modèle logistique suppose que la mortalité μ_x est une fonction logistique de l'âge x . Ce modèle s'écrit :

$$\mu_x = \frac{\alpha \exp(\beta x)}{1 + \alpha \exp(\beta x)}. \quad (1.5)$$

La principale limite que l'on peut formuler à l'égard des modèles de mortalité dits statiques, qui décrivent la mortalité à un instant donné, réside dans leur incapacité à intégrer explicitement son évolution dans le temps. En effet, leurs paramètres doivent être réajustés régulièrement afin de refléter les changements observés, car ils ne prennent pas en compte le caractère dynamique et incertain de la mortalité réelle, laquelle évolue continuellement sous l'influence de nombreux facteurs.

1.3.2 Évolution de la deuxième génération des modèles de mortalité : modèles stochastiques

Les modèles stochastiques de mortalité ont tous un intérêt commun : ils reposent sur le fait qu'il existe une continuité entre l'évolution de la mortalité passée et son évolution future. Ainsi, ils supposent que les taux de mortalité suivent un processus stochastique. Grâce à cette hypothèse, il est possible de produire une distribution de prévisions, et non plus uniquement une valeur déterministe. Sous ces hypothèses, les modèles stochastiques répondent donc à la critique émise sur les modèles statiques dans la mesure où ils supposent que le taux de mortalité $\mu_{x,t}$ est aléatoire et donc, $\mu_{x,t}$ est un processus stochastique. Le taux de mortalité effectivement observé pour un âge et une période donnés est alors la réalisation d'une variable aléatoire. Ces modèles sont regroupés en deux catégories à savoir les modèles de type Lee-Carter ([9]), et les modèles de type Cairns, Blake et Dowd ([3]).

1.3.2.1 Modèle de Lee-Carter (LC)

Le modèle LC, est l'un des modèles stochastiques de mortalité les plus connus, et l'un des plus utilisés à ce stade de la littérature. Il a été introduit par Ronald Lee et Lawrence Carter en 1992. En observant les taux de mortalité antérieurs, ils utilisent des séries chronologiques pour extrapoler la tendance temporelle à l'aide d'un modèle stochastique à un facteur. En raison de son mécanisme endogène de calcul des paramètres année après année, il permet de capter de manière satisfaisante les évolutions tendancielle de la mortalité. Ce modèle s'écrit :

$$\ln(m_{x,t}) = \alpha_x + \beta_x \kappa_t + \varepsilon_{x,t}. \quad (1.6)$$

Dans ce modèle, nous avons :

- α_x décrit le niveau moyen du logarithme du taux central de mortalité à chaque âge. En outre, il garantit que la forme de la courbe de mortalité est conforme à l'expérience. En utilisant les contraintes applicables à l'estimation des paramètres, ce terme non paramétrique représente la moyenne arithmétique de $m_{x,t}$ sur toute la période observée ;
- β_x est le terme qui explique comment $\ln(m_{x,t})$ réagit au passage du temps. C'est un paramètre de sensibilité de la vitesse de réaction du taux de mortalité à κ_t pour chaque âge. Le modèle permet d'obtenir des valeurs positives et négatives. Le signe de β_x indique le sens de réaction du logarithme du taux de mortalité aux variations de κ_t . Lorsque κ_t diminue au cours du temps, une valeur positive de β_x traduit généralement une baisse de la mortalité à l'âge considéré ;
- κ_t représente l'indice temporel d'évolution de la mortalité dans l'espace-temps ;
- $\epsilon_{x,t}$ fait référence aux termes d'erreurs, à savoir les effets non saisis par le modèle.

Le modèle de Lee-Carter est réputé pour sa simplicité d'utilisation et sa qualité d'ajustement. Il est parcimonieux, c'est-à-dire avec peu de paramètres, il capte l'essentiel des tendances de mortalité. Aussi, il est flexible dans le sens où il est applicable même si on ne dispose pas de tous les taux par âge car on peut travailler à partir des décès totaux. Cependant, il a plusieurs limites. Tout d'abord, il y a un seul paramètre stochastique qui capte l'évolution de la mortalité à tous les âges, sauf que les dynamiques de mortalité peuvent différer entre les jeunes âges et les âges élevés. Aussi, il ne prend pas en compte l'effet cohorte car les personnes nées pendant les périodes de guerre ou de pandémie peuvent avoir une mortalité différente des autres.

1.3.2.2 Modèle de Renshaw et Haberman (RH)

En 2006, Arthur Renshaw et Peter Haberman ([14]) proposent une extension du modèle LC permettant d'intégrer un facteur de cohorte au modèle. Le modèle RH s'écrit sous la forme suivante :

$$\ln(m_{x,t}) = \alpha_x + \beta_x^1 \kappa_t^1 + \beta_x^0 \gamma_{t-x} + \epsilon_{x,t}. \quad (1.7)$$

Ici, en plus des paramètres de Lee-Carter, nous avons :

- β_x^0 est le terme qui explique les interactions à l'âge, autrement dit ce paramètre mesure la sensibilité de la mortalité à l'effet de cohorte selon l'âge ;
- γ_{t-x} représente l'effet de cohorte qui est fonction de l'année de naissance $t - x$.

Pour les populations où un effet de cohorte est observé dans les données historiques, le modèle RH offre une meilleure adéquation aux données historiques. Cependant, il est constaté que le modèle RH souffre d'un manque de robustesse. En outre, bien que le modèle dispose d'un facteur stochastique supplémentaire pour l'effet de cohorte, pour la plupart des taux de mortalité simulés, la structure de corrélation est encore insignifiante. En particulier, lorsque l'on utilise une large gamme d'âge, les paramètres de la cohorte simulée ne sont pertinents que pour les âges les plus élevés dans la partie la plus éloignée de la projection.

1.3.2.3 Modèle Age-Période-Cohorte (APC)

Le modèle Age-Period-Cohort (APC), est une sous-structure du modèle RH. Il simplifie dans sa forme le modèle RH en imposant les contraintes suivantes sur les paramètres : $\beta_x^1 = 1$; $\beta_x^0 = 1$. Le modèle APC s'écrit sous la forme suivante :

$$\ln(m_{x,t}) = \alpha_x + \kappa_t^1 + \gamma_{t-x} + \epsilon_{x,t}. \quad (1.8)$$

Cette variante du modèle permet de résoudre certains problèmes d'instabilité du modèle RH.

1.3.2.4 Modèle de Cairns, Blake et Dowd (CBD)

Publié pour la première fois en 2006, le modèle stochastique qui sort de façon nette du paradigme des modèles de type LC est celui d'Andrew Cairns, David Blake, et Kevin Dowd([3]) (CBD). Il part de l'hypothèse qu'il n'y a pas de comportement moyen du taux de mortalité, et qu'il n'y a pas d'effet de cohorte. Il y a cependant selon les auteurs une tendance dans l'âge de naissance des cohortes évaluées. Le modèle de CBD s'écrit sous la forme suivante :

$$\log\left(\frac{q_{x,t}}{1 - q_{x,t}}\right) = \kappa_t^1 + (x - \bar{x})\kappa_t^2 + \varepsilon_{x,t}, \quad (1.9)$$

où $q_{x,t}$ est la probabilité qu'une personne d'âge x décède dans l'année qui vient et κ_t^2 permet aux changements de mortalité de varier entre les âges reflétant les observations historiques selon lesquelles les taux d'améliorations peuvent différer pour différentes classes d'âges.

1.3.2.5 Modèle de Plat

Plat([12]) a proposé un modèle qui tient compte à la fois de l'effet cohorte et de l'effet age-période dit âge-période-cohorte. Ce modèle est donné par :

$$\ln(m_{x,t}) = \alpha_x + \kappa_t^1 + (\bar{x} - x)\kappa_t^2 + (\bar{x} - x)^+\kappa_t^3 + \gamma_{t-x} + \epsilon_{x,t}. \quad (1.10)$$

Dans ce modèle, les paramètres β_x^1 , κ_t^1 et κ_t^2 ont la même signification que dans les modèles précédents. κ_t^3 quant à lui, modélise l'effet de la mortalité spécifique aux jeunes âges.

1.3.2.6 Modèle de O'Hare et Li

Colin O'hare et Youwei Li ([11]) ont amélioré le modèle de Plat en prenant en compte des effets non linéaires de la mortalité logarithmique aux jeunes âges. Il offre également un meilleur ajustement pour l'ensemble des populations considérées. Le modèle est donné par :

$$\ln(m_{x,t}) = \alpha_x + \kappa_t^1 + (\bar{x} - x)\kappa_t^2 + ((\bar{x} - x)^+ + [(\bar{x} - x)]^2)\kappa_t^3 + \gamma_{t-x} + \epsilon_{x,t}. \quad (1.11)$$

1.4 Formule générale des modèles de mortalité stochastiques

En résumé de la littérature actuelle, la structure mathématique générale des modèles de mortalité stochastiques peut-être résumée par l'équation suivante :

$$\ln(m_{x,t}) = \alpha_x + \sum_{i=1}^N \beta_x^i \kappa_t^i + \beta_x^0 \gamma_{t-x} + \epsilon_{x,t}. \quad (1.12)$$

Cependant, tous les modèles précédents ne prennent pas en compte les effets liés aux facteurs exogènes tels que le changement climatique.

1.5 Climat et mortalité

De nombreuses études dans la littérature ont cherché à décrire et à quantifier la relation entre la température et la mortalité.

1.5.1 Le modèle de Seklecka-Pantelous-O'hare (SPO)

Seklecka, Pantelous et O'Hare([15]) étendent les approches de Plat([12]), O'Hare et Li([11]) en introduisant un facteur lié à la température. Ils ont été parmi les premiers à avoir proposé un modèle qui tient compte du changement climatique, plus précisément des fluctuations de la température. Leur objectif principal était d'examiner pour la première fois la relation entre les tendances de la mortalité et les tendances de la température. Ils ont travaillé avec des données annuelles distinguant les périodes chaudes et froides au Royaume-Uni allant de 1974 à 2011. Dans leur travail, ils ont démontré qu'il existe bien une relation à long terme entre les variations de la température moyenne et l'indice de mortalité du modèle LC. Le modèle est donné par :

$$\ln(m_{x,t}) = \alpha_x + \kappa_t^1 + (\bar{x} - x)\kappa_t^2 + (\bar{x} - x)^+ \kappa_t^3 + [(a - x)^+ + ct_x(x - a)^+]^2 \kappa_t^4 + \gamma_{t-x} + \epsilon_{x,t}. \quad (1.13)$$

Dans ce modèle, le paramètre ct_x capte la relation entre la température et la mortalité selon l'âge, plus particulièrement pour les âges avancés car ils sont plus vulnérables aux changements de la température et κ_t^4 capte la dynamique temporelle associée à cet effet supplémentaire, notamment les effets non linéaires liés aux âges élevés et à la température.

1.5.2 Modèle de Gavakshi et Jason

Gavakshi et Jason([6]) ont modifié le modèle Lee-Carter afin de capturer les événements extrêmes de mortalité. Leur objectif est d'améliorer la capacité du modèle LC à prendre en compte

les épisodes de surmortalité extrême, tels que les pandémies ou les vagues de chaleur, qui engendrent des ruptures dans les tendances de mortalité et ne sont pas correctement modélisés par les approches standards. Le modèle est intitulé EVT-Modified Lee-Carter Model et utilise la distribution de Pareto généralisée. Ils utilisent les approches telles que Peak-Over-Threshold. Ils travaillent sur des données divisées en deux périodes : une période instable de 1900 à 1960, marquée par des événements de mortalité extrême et une période stable de 1961 à 2017, où aucun saut significatif de mortalité n'est observé. Ils évaluent les prévisions de mortalité sur la base des mesures de précision RMSE (erreur quadratique moyenne), MAE (erreur absolue moyenne), MPE (erreur en pourcentage moyenne) et MAPE (erreur en pourcentage absolue moyenne).

Ces modèles sont calibrés sur des données annuelles. Cependant les effets des variations de température ne sont pas les mêmes pour les périodes froides et les périodes chaudes et en plus, ces effets ne durent pas nécessairement longtemps, ils peuvent aller jusqu'à une semaine tout au plus car il existe un effet de compensation au cours de l'année par exemple, une hausse soudaine de mortalité due à une canicule peut être suivie d'une période de diminution de la mortalité, car en raison de leur fragilité, certaines des personnes décédées auraient de toute façon succombé peu après. C'est d'ailleurs de ça que Cairns, Blake, Kessler([2]) parlent dans leur article intitulé "The Impact of Covid-19 on Future Higher-Age Mortality". Ils disent qu'une partie des décès causés par le Covid-19 correspond à des décès qui auraient eu lieu dans un futur proche mais qui sont avancés dans le temps en raison de la fragilité des individus. Ainsi, si on s'en tient uniquement aux statistiques annuelles, cette compensation rend difficile l'identification des impacts spécifiques des températures extrêmes car elles ne permettent pas de capter correctement ces fluctuations de court terme. Un autre aspect important à prendre en compte est l'impact de la croissance du changement climatique, en particulier les conséquences de la hausse des températures et des périodes de chaleur extrême.

1.5.3 Modèle de Jens Robben, Katrien Antonio et Torsten Kleinow

Jens Robben, Katrien Antonio et Torsten Kleinow([8]) ont étudié la relation entre la mortalité et les facteurs environnementaux en utilisant des données hebdomadaires. Ils ont construit un modèle de référence de mortalité hebdomadaire pour estimer les variations saisonnières des taux de mortalité, telles qu'observées dans les données historiques d'une région spécifique. ils ont observé que la surmortalité était plus ressentie lors des semaines extrêmement chaudes. En considérant la structure suivante $\mu_{t,w}^{(r)}$ pour chaque région $r \in \mathcal{R}$, leur modèle est donné par :

$$\begin{aligned} \log(\mu_{t,w}^{(r)}) = & \beta_0^{(r)} + \beta_1^{(r)}t + \beta_2^{(r)} \sin\left(\frac{2\pi w}{52, 18}\right) + \beta_3^{(r)} \cos\left(\frac{2\pi w}{52, 18}\right) \\ & + \beta_4^{(r)} \sin\left(\frac{2\pi w}{26, 09}\right) + \beta_5^{(r)} \cos\left(\frac{2\pi w}{26, 09}\right), \end{aligned} \quad (1.14)$$

1.6 Conclusion

où la force hebdomadaire de mortalité spécifique à la région, notée $(\mu_{t,w}^{(r)})$ représente le taux instantané de mortalité dans la région r au groupe d'âge x et en semaine ISO w de l'année t . 52,18 correspond au nombre moyen de semaines par année. Notez que cette structure de base est spécifique à la région, à la semaine et à l'année. De plus, les paramètres spécifiques à chaque région $\beta_p^{(r)}$, pour $p = 0, \dots, 5$, contrôlent le niveau, la tendance et les variations saisonnières.

1.6 Conclusion

Dans ce chapitre, nous avons vu de manière chronologique l'évolution des modèles de mortalité, depuis les modèles statiques jusqu'aux modèles stochastiques. Le modèle de base qui est celui de Lee-Carter a connu plusieurs extensions mais reste tout de même l'un des meilleurs modèles de mortalité grâce à sa simplicité. Ensuite vient le modèle de Cairns, Blake et Dowd (CBD) qui a aussi plusieurs extensions. Ces modèles se basaient sur la structure historique des données et ne prenaient pas en compte des facteurs exogènes tels que les facteurs climatiques, les facteurs socio-économiques, les facteurs environnementaux, etc. Ainsi, Aujourd'hui, une partie importante de la recherche vise donc à intégrer ces différents facteurs dans les modèles de mortalité.

ÉTUDE DE LA CORRÉLATION ENTRE LA MORTALITÉ ET LA TEMPÉRATURE

L'objectif de ce chapitre est de justifier l'étude de l'impact du changement climatique, et plus précisément des variations de la température, sur la mortalité humaine. Pour cela, nous présenterons tout d'abord les données de mortalité, puis les données de température, avant d'étudier la corrélation à l'aide des coefficients de Pearson et de Spearman, ainsi que de tests statistiques.

2.1 Base de données de mortalité

Les données relatives à la mortalité humaine « Human Mortality Database » sont disponibles pour plusieurs pays dans la base STMF (Short-Term Mortality Fluctuations Database) accessible via le site internet <https://www.mortality.org/Data/STMF>. Nous avons choisi la population française, dont les données hebdomadaires sont disponibles de 2000 à 2025. Afin de limiter l'impact dans l'analyse des populations ayant vécu la période du Covid-19, nous avons limité notre période d'observation à 2000-2019. Les décès sont disponibles par groupe d'âge (0, 1-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59, 60-64, 65-69, 70-74, 75-79, 80-84, 85-89, 90-94, 95+) et les expositions vont jusqu'à 110+. Nous avons sommé les expositions entre 95 ans et 110 ans et plus afin d'obtenir les mêmes classes d'âge pour les décès et les expositions. Pour obtenir les taux de mortalité, nous avons procédé comme dans ([4]), c'est-à-dire, les taux de mortalité aux âges spécifiques sont calculés comme suit :

$$m_y^w(x, x+a) = \frac{D_y^w(x, x+a)}{E_y(x, x+a)/52},$$

où a est la limite supérieure de l'intervalle des âges ; $D_y^w(x, x+a)$ est le nombre de décès dans l'intervalle d'âge $[x, x+a]$ pour la semaine w de l'année y ; $E_y(x, x+a)$ est la population exposée dans l'intervalle d'âge $(x, x+a]$ de l'année y .

Les graphiques ci-dessous montrent l'évolution des taux de mortalité chez les hommes et chez les femmes par classes d'âges :

2.2 Base de données de température

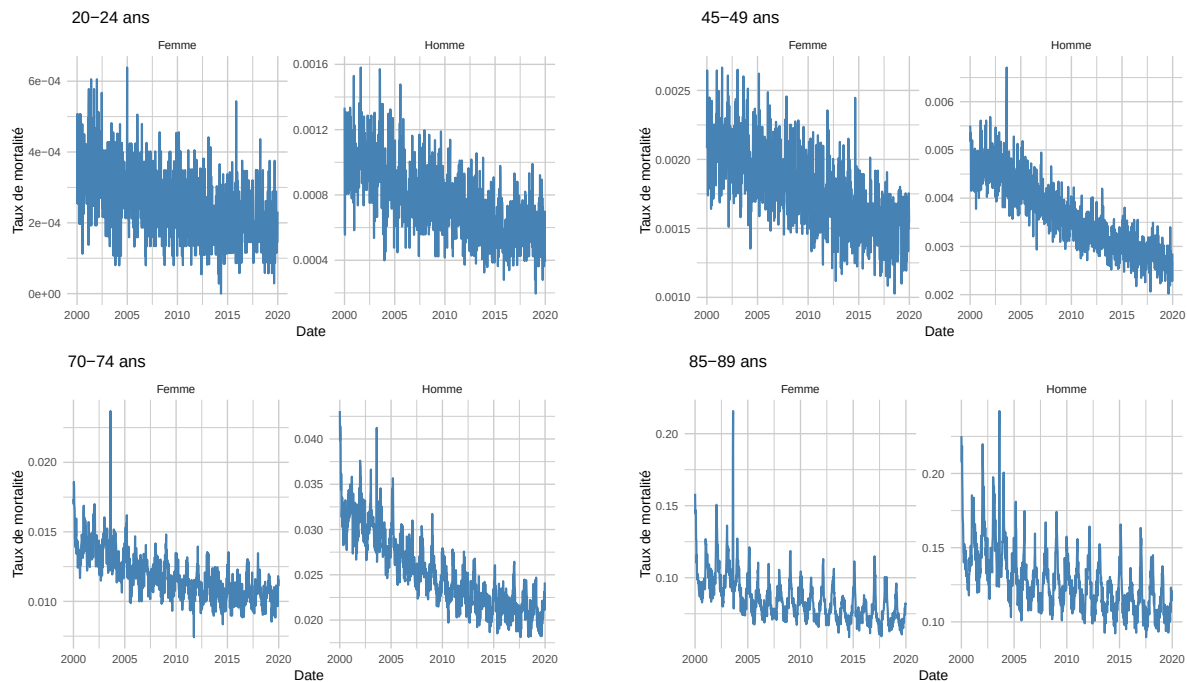


FIGURE 2.1 – Évolution des taux de mortalité brutes par sexe et par classe d'âge.

Ces graphiques mettent en évidence trois phénomènes fondamentaux en démographie de la mortalité : la dépendance avec l'âge, la saisonnalité et la tendance temporelle. On observe une croissance du taux de mortalité avec l'âge, ce qui correspond à la description de Gompertz (1.3) avec des taux plus élevés chez les hommes que chez les femmes. On observe également des pics réguliers chaque année sur toutes les courbes, ce qui correspond à la saisonnalité avec une mortalité plus élevée en hiver et plus faible en été. Cette saisonnalité est beaucoup plus forte pour les âges élevés, on voit une forte variation chez les 85-89 ans et une faible variation chez les 20-24 ans, c'est-à-dire, la vulnérabilité augmente avec l'âge. De manière globale, on remarque une amélioration de la mortalité dans le temps tant chez les femmes que chez les hommes, ce qui est dû aux progrès de la médecine, à l'amélioration des conditions de vie, etc. Cette tendance est précisément ce que le paramètre κ_t du modèle de Lee-Carter essaye de capturer. D'autre part, on remarque des pics anormaux vers 2003 ce qui correspond probablement à la canicule européenne de 2003 au cours de laquelle les températures ont atteint $40,7^\circ\text{C}$ à Bordeaux. Ces caractéristiques nous permettent de justifier l'utilisation des modèles démographiques tels que Lee-Carter(1.6) pour capturer la tendance temporelle et SPO(1.13) pour intégrer un facteur lié aux variations de la température.

2.2 Base de données de température

Les données de température proviennent du NCEI (National Centers for Environmental Information) qui est une institution officielle gérée par la NOAA (National Oceanic and At-

2.2 Base de données de température

mospheric Administration) des États-Unis. Elle fournit des données fiables car elles sont issues de stations météorologiques validées. Aussi, les données climatiques qu'elle offre sont universelles et sur plusieurs stations de nombreux pays dont la France. Elles suivent des méthodologies standardisées pour permettre des comparaisons entre pays ou régions.

Pour notre étude, nous avons sélectionné les 5 meilleures stations (Bordeaux, Lyon, Lille, Nice, STRASBOURG) entre 2000-2019 en fonction du pourcentage de couverture. Pour chacune de ces stations, les données disponibles sont journalières et donnent accès à la température moyenne, minimale et maximale. Nous commençons par convertir les températures exprimées en degrés Fahrenheit en degrés Celsius par la formule :

$$C = \frac{(F - 32) * 5}{9}.$$

Ensuite, une première agrégation est faite au niveau de chaque station : on calcule, pour chaque semaine, la moyenne hebdomadaire de TAVG, TMIN, TMAX. Après cela, une seconde agrégation est réalisée entre les stations, afin d'obtenir une seule valeur hebdomadaire nationale pour chaque indicateur. Cela permet d'avoir les données de température hebdomadaires et agrégées à l'échelle nationale, comparables aux données de mortalité hebdomadaires utilisées dans l'étude. Enfin, une variable de saison est créée pour distinguer les semaines chaudes et froides de la façon suivante :

- mois allant de mai à septembre : période chaude ;
- mois allant de octobre à avril : période froide.

TAVG, TMAX, TMIN représentent respectivement les températures moyennes, maximales et minimales hebdomadaires de l'air, mesurées en degrés Celsius (°C) à 2 mètres au-dessus de la surface.

Le graphique suivant représente l'évolution des températures minimales, moyennes et maximales pour les 5 stations françaises agrégées à l'échelle nationale : Les courbes montrent un cycle régulier qui se répète chaque année avec les températures moyennes qui varient autour de 17–23°C en période chaude et 5–12°C en période de froid. Les températures minimales descendent jusqu'à -7.22°C correspondant à la dernière vague de froid la plus meurtrière qui a touché l'Europe en 2012.

On observe également que les températures maximales dépassent 35°C correspondant à la canicule de 2003. L'Europe tout entière a enregistré des températures exceptionnelles. Le bilan humain selon l'Institut de Veille Sanitaire serait évalué à un total de 11 435 décès supplémentaires. Cette canicule a été suivie par d'autres épisodes de chaleur intense, notamment en 2006, 2012, 2017 et 2018, d'où l'intérêt de notre étude par saison froide et chaude séparément.

2.3 Évaluation de la corrélation entre mortalité et température

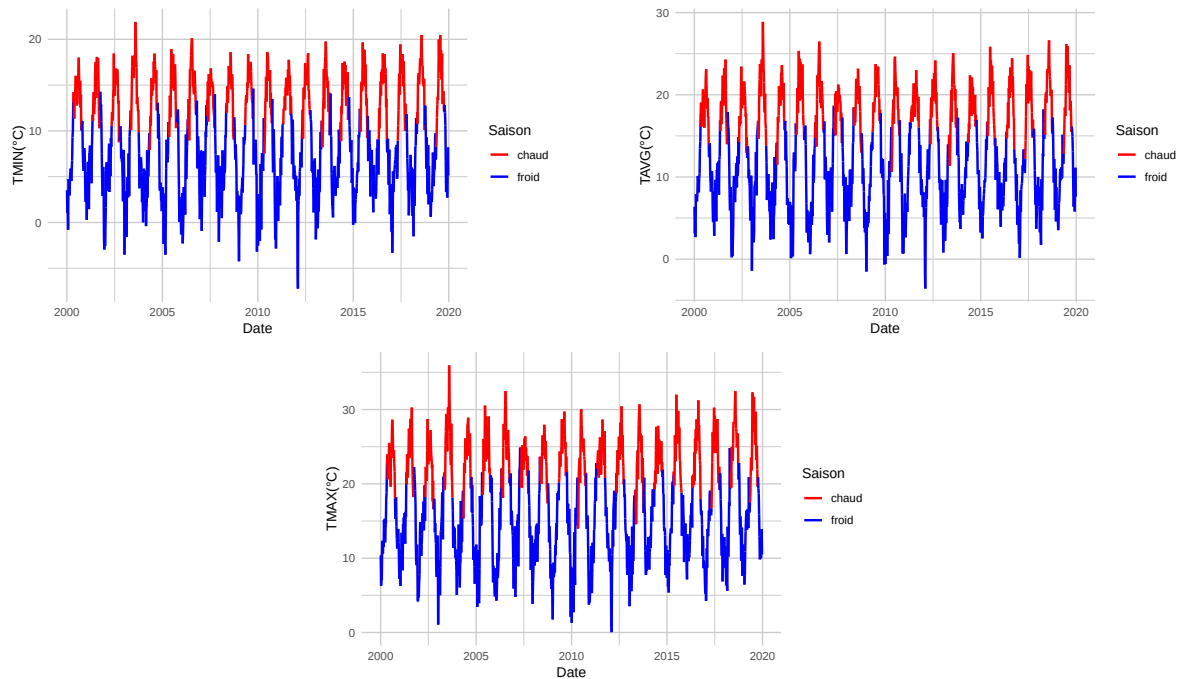


FIGURE 2.2 – Évolution des températures (TAVG, TMIN, TMAX).

2.3 Évaluation de la corrélation entre mortalité et température

Les analyses précédentes montrent que la mortalité a tendance à diminuer. La saisonnalité ne nous permet pas de voir clairement une tendance à long terme de la température mais on observe quand même que les températures élevées semblent devenir plus fréquentes. Ainsi, on pourrait s'attendre à une corrélation négative entre la température et la mortalité. Nous allons donc étudier cette corrélation à l'aide des méthodes de Pearson et de Spearman comme dans l'approche de Seklecka et al.

La corrélation de Pearson évalue la relation linéaire entre deux variables continues. Une relation est dite linéaire lorsqu'une modification de l'une des variables est associée à une modification proportionnelle de l'autre variable. Le coefficient de corrélation de Pearson pour deux variables X et Y s'écrit sous la forme suivante :

$$r(X, Y) = \frac{Cov(X, Y)}{\sigma_X * \sigma_Y}.$$

Ainsi, si $r(X, Y)$ est proche de 0, il n'y a pas de relation linéaire entre les deux. si $r(X, Y) = 1$, la relation linéaire est parfaite positive ; si $r(X, Y) = -1$, elle est parfaite négative.

La corrélation de Spearman est une corrélation basée sur les rangs. Elle examine s'il existe une relation entre le rang des observations pour deux variables X et Y , ce qui permet de détecter l'existence de relations dites monotones (c'est-à-dire de forme croissante ou décroissante). Cette

2.3 Évaluation de la corrélation entre mortalité et température

dernière corrélation permet donc d'analyser les relations non linéaires monotones. Elle s'écrit sous la forme :

$$\rho(X, Y) = 1 - \frac{6 \sum_{i=1}^N [r(X_i^2) - r(Y_i^2)]}{N^3 - N},$$

où $r(X_i)$ et $r(Y_i)$ représentent les rangs de X_i et Y_i dans leurs distributions respectives. Il s'interprète de la même manière que celui de Pearson.

Nous avons calculé ces corrélations par âge et par période de l'année chez les hommes et chez les femmes.

La figure (2.3) nous présente les coefficients de corrélation de Pearson et Spearman pour tous les âges par sexe et par période sur toute la période considérée. Les deux mesures présentent une allure similaire et des valeurs relativement proches. Cette corrélation est donc principalement monotone et très proche d'une relation linéaire.

On remarque qu'en annuelle, tant chez les hommes que chez les femmes, la corrélation est proche de zéro aux jeunes âges, puis diminue avec l'âge jusqu'à devenir fortement négative à partir d'environ 70 ans avec une sensibilité un peu plus marquée chez les hommes que chez les femmes.

Pendant la période froide, la corrélation devient encore plus négative, ce qui montre que la mortalité augmente lorsque la température baisse et cela se ressent encore plus chez les personnes âgées, ce qui montre que les personnes âgées sont plus vulnérables.

Dans la période chaude, la corrélation est soit positive, soit très proche de zéro. Cela signifie que pendant l'été, l'augmentation de la température peut s'accompagner d'une légère augmentation de la mortalité mais beaucoup moins marquée qu'en hiver sauf lors des épisodes de canicule.

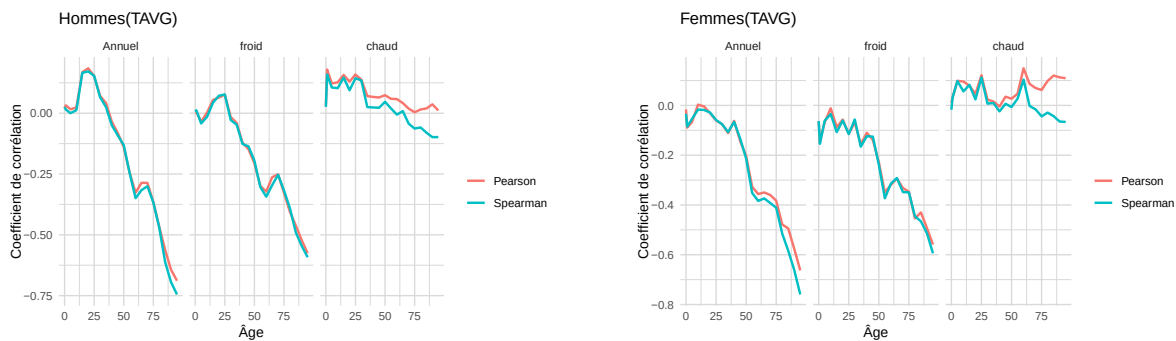


FIGURE 2.3 – Coefficients de corrélation de Pearson et de Spearman en fonction de l'âge.

Afin de déterminer le degré de significativité de cette corrélation, nous avons procédé à un test d'hypothèse en posant :

- H_0 : « Il n'existe pas de relation significative entre la température et la mortalité ».
- H_1 : « Il existe une relation significative entre la température et la mortalité ».

Nous avons calculé les p-valeurs de chaque coefficient de corrélation entre la mortalité et la température des âges correspondant via les méthodes de Pearson et de Spearman. Les valeurs

2.4 Conclusion

sont présentées dans le tableau en annexe (4.2). L'interprétation correspond à celle des graphes avec des p-valeurs significatives (<0.05) pour les périodes froide et annuelle et non significatives pour la période chaude, excepté à certains âges, probablement en raison des épisodes de canicule. Cependant, une majorité de p-valeurs sont inférieures à 0.05 et donc sur cette base, nous rejetons l'hypothèse nulle (H_0).

2.4 Conclusion

Dans ce chapitre, nous avons étudié la corrélation entre les taux de mortalité et la température. Dans un premier temps, l'analyse des taux de mortalité a mis en évidence trois caractéristiques importantes : la dépendance à l'âge, la saisonnalité et la tendance temporelle. Cette saisonnalité apparaît plus marquée aux âges élevés, ce qui traduit une plus grande vulnérabilité des personnes âgées aux variations saisonnières. Dans un second temps, nous avons analysé l'évolution de la température sur la période 2000-2019. L'étude des corrélations de Pearson et Spearman montre l'existence d'une relation négative entre la température et la mortalité pendant les périodes froides et annuelles. Cela signifie que la mortalité tend à augmenter lorsque la température diminue, en particulier aux âges élevés. Pour les périodes chaudes, la corrélation est généralement proche de zéro ou légèrement positive, ce qui suggère que la mortalité peut augmenter lorsque les températures atteignent des niveaux élevés, notamment lors des canicules. Les tests d'hypothèses confirment globalement ces résultats, avec des corrélations significatives principalement pendant les périodes froides et annuelles. Ces éléments justifient donc l'étude plus approfondie de l'impact de la température sur la mortalité. Dans le chapitre suivant, nous analyserons cet impact en comparant le modèle de Lee-Carter au modèle SPO, qui permet d'intégrer explicitement un facteur lié à la température.

APPLICATION ET ESTIMATION DES PARAMÈTRES DES MODÈLES LC ET SPO

3.1 Introduction

Dans le chapitre précédent, nous avons étudié la corrélation entre la mortalité et la température et nous avons montré qu'il existe une corrélation négative pour les périodes annuelles et froides, dont l'intensité augmente avec l'âge. Pour les périodes chaudes, la corrélation est proche de zéro ou positive, probablement en raison des épisodes de canicules. On note donc que l'impact du froid et de la chaleur n'est pas le même, c'est pourquoi nous réaliserons l'étude sur trois périodes à savoir : annuelle, froide et chaude et par sexe pour tous les âges. Pour la suite de ce mémoire, nous nous limitons à l'étude des modèles de Lee-Carter (1.2) et de Seklecka, Pantelous et O'Hare([15]).

3.2 Estimation des paramètres du modèle de Lee-Carter(LC)

Dans toute la suite de notre travail, nous avons scindé nos données en :

- In-sample : 2000–2016 pour entraîner les modèles ;
- Out-of-sample : 2017–2019 pour tester les projections.

Tout d'abord, rappelons qu'un modèle est dit identifiable s'il est possible d'apprendre la vraie valeur des différents paramètres du modèle à partir des données d'observations disponibles. En effet, l'identification d'un système nécessite la résolution de deux problèmes correspondant d'une part au choix de la structure du modèle et d'autre part à l'estimation des paramètres de la structure choisie.

Le modèle de Lee-Carter (LC) $:= \ln(m_{x,t}) = \alpha_x + \beta_x \kappa_t^1 + \epsilon_{x,t}$ comme défini dans la section (1.3.2.1) est l'un des modèles de mortalité les plus utilisés à ce jour. Cependant, il souffre d'un problème d'identifiabilité des paramètres. En d'autres termes, l'estimation de ses paramètres ne donne pas d'unique solution. Par exemple, si on suppose une constante $c \neq 0$ et une solution

3.2 Estimation des paramètres du modèle de Lee-Carter(LC)

$(\alpha_x, \beta_x, \kappa_t)$, alors on peut définir de nouveaux paramètres $\beta_x^* = c\beta_x$, $\kappa_t^{1*} = \frac{\kappa_t^1}{c}$, alors on a :

$$\beta_x^* \kappa_t^{1*} = (c\beta_x) \left(\frac{\kappa_t^1}{c} \right) = \beta_x \kappa_t^1,$$

donc $\alpha_x + \beta_x \kappa_t^1 = \alpha_x + \beta_x^* \kappa_t^{1*}$. Ainsi, la mortalité prédite est la même. On a donc $(\alpha_x, \beta_x, \kappa_t)$ et $(\alpha_x, c\beta_x, \frac{\kappa_t^1}{c})$ qui donnent exactement les mêmes données. Donc les paramètres ne sont pas uniques et le modèle n'est donc pas identifiable. Pour cette raison, les auteurs Lee et Carter ont défini les deux contraintes suivantes :

- $\sum \beta_x = 1$,
- $\sum \kappa_t = 0$.

Par conséquent le paramètre α_x peut être traduit comme étant la moyenne temporelle du logarithme des taux de mortalité centraux pour chaque classe d'âge :

$$a_x = \frac{1}{T} \sum_{t=1}^T \ln(m_{x,t}).$$

Dans l'article de base, les auteurs utilisent la méthode dite de décomposition en valeur singulière (SVD) pour l'estimation des paramètres. Cependant, l'estimation des paramètres β_x et κ_t^1 ne garantit pas que le nombre de décès estimé est identique à celui observé car elle donne le même poids aux décès observés aux jeunes âges qu'aux âges élevés. C'est pourquoi une ré-estimation du paramètre κ_t^1 est requise afin de s'assurer qu'on a exactement :

$$D(t) = \sum_x N(x, t) e^{\alpha_x + \beta_x \kappa_t^1},$$

où $N(x, t)$ est la population exposée à l'âge x et à la période t . $D(x, t)$ désigne le nombre total de décès observés pendant la période t . Cette équation peut ne pas avoir de solution unique ou ne pas du tout présenter de solution. Le modèle devient alors incohérent.

Les paramètres du modèle LC sont donc estimés par maximum de vraisemblance sous l'hypothèse que le nombre de décès $D_{x,t}$ suit une loi de Poisson de paramètre $E_{x,t}m_{x,t}$. Cette approche est celle proposée par Natacha Brouhns, Michel Denuit et Vermunt (2002). Elle constitue une amélioration de la méthode originale de Lee-Carter (1.6). Elle permet de tenir compte de l'hétéroscédasticité inhérente aux données de mortalité et est aujourd'hui largement utilisée dans la littérature. La fonction à maximiser est la suivante :

$$L(\psi, D, E) = \sum_{x,t} \{D_{x,t} \ln [E_{x,t}m_{x,t}(\psi)] - E_{x,t}m_{x,t}(\psi) - \ln(D_{x,t})\}. \quad (3.1)$$

Nous avons utilisé le package *StMoMo* afin d'estimer les paramètres. Les graphiques (3.1) et (3.2) présentent les résultats de l'estimation des paramètres du modèle LC. Les courbes en bleu représentent les résultats de l'estimation des paramètres pour la population masculine, alors

3.3 Estimation des paramètres du modèle SPO

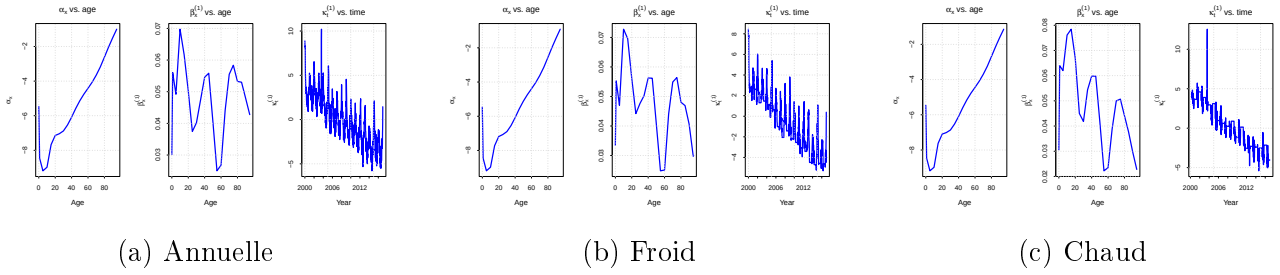


FIGURE 3.1 – Estimation des paramètres du modèle de LC chez les hommes

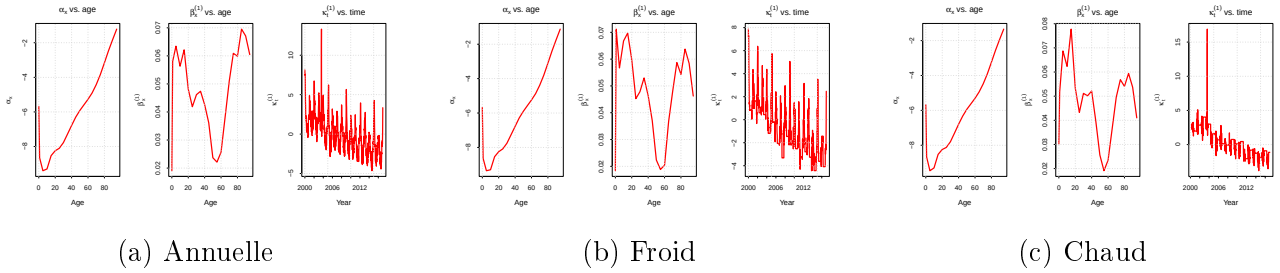


FIGURE 3.2 – Estimation des paramètres du modèle de LC chez les femmes

que celles en rouge représentent les résultats de l'estimation des paramètres pour la population féminine.

Les paramètres estimés du modèle de Lee-Carter présentent des comportements cohérents avec la littérature démographique. Le paramètre α_x qui est le niveau moyen de la mortalité augmente fortement avec l'âge et se caractérise par un pic de mortalité infantile, ainsi que par une légère surmortalité autour de 18–25 ans, plus marquée chez les hommes. La courbe du paramètre κ_t^1 est quant à elle décroissante avec le temps traduisant une amélioration globale de la mortalité. Cette décroissance est beaucoup plus marquée chez les hommes que chez les femmes. Aussi, les estimations saisonnières montrent également une variabilité plus marquée durant les périodes froides et chaudes ce qui reflète l'impact de la température sur la mortalité. Enfin le paramètre β_x qui représente la sensibilité des âges aux variations de la mortalité, dans la littérature, a tendance à être lisse sur des données annuelles. Cependant, plusieurs informations sont cachées dans les données annuelles, notamment, les vagues de froid, de chaleur, épidémies de grippe, les semaines avec des mortalités très faibles. Ces effets peuvent provoquer des pics de mortalité à certaines semaines, ce qui influence l'estimation du modèle et peut créer des irrégularités dans β_x .

3.3 Estimation des paramètres du modèle SPO

Dans cette partie, nous allons estimer les paramètres du modèle SPO(1.13) version mixte, i.e., nous considérons : l'indice temporel κ_t^1 et le facteur qui capte les effets de la température κ_t^4 à l'échelle hebdomadaire ; le facteur permettant au changement de la mortalité de varier en

3.3 Estimation des paramètres du modèle SPO

fonction des âges κ_t^2 , le facteur lié aux âges jeunes κ_t^3 et la cohorte à l'échelle annuelle. Ainsi le modèle devient :

$$\ln(m_{x,t}) = \alpha_x + \kappa_{t_h}^1 + (\bar{x} - x)\kappa_{t_a}^2 + (\bar{x} - x)^+\kappa_{t_a}^3 + [(a - x)^+ + c_x(x - a)^+]^2\kappa_{t_h}^4 + \gamma_{t_a-x} + \epsilon_{x,t}. \quad (3.2)$$

Le paramètre de température c_x permet de relier l'intensité de la sensibilité thermique à l'âge. Il représente ici le coefficient de corrélation de Pearson entre la température et la mortalité.

Comme pour tous les modèles de mortalité stochastique, en particulier le modèle de Plat, 2009([12])(1.11) le modèle proposé a un problème d'identifiabilité, ie, des paramétrisations différentes peuvent conduire à des valeurs identiques pour $\ln(m_{x,t})$. Par exemple, si on considère la paramétrisation suivante :

$$(\alpha_x, \kappa_t^{(1)}, \kappa_t^{(2)}, \kappa_t^{(3)}, \kappa_t^{(4)}, \gamma_{t-x}) \rightarrow (\alpha_x + \phi_1 - \phi_2 x + \phi_3 x^2, \kappa_t^{(1)} + \phi_2 t + \phi_3(t^2 - 2xt), \kappa_t^{(2)} + 2\phi_3 t, \kappa_t^{(3)}, \kappa_t^{(4)}, \gamma_{t-x} - \phi_1 - \phi_2(t - x) - \phi_3(t - x)^2),$$

où ϕ_1, ϕ_2, ϕ_3 et d sont des constantes. Ainsi, l'identification du modèle s'obtient en posant les mêmes contraintes que le modèle de Plat([12]) :

$$\left\{ \begin{array}{l} \sum_{c=c_0}^{c_1} \gamma_c = 0 \\ \sum_{c=c_0}^{c_1} c \gamma_c = 0 \\ \sum_t k_t^1 = 0 \\ \sum_t k_t^2 = 0, \end{array} \right. \quad \left\{ \begin{array}{l} \sum_{c=c_0}^{c_1} c^2 \gamma_c = 0 \\ \sum_t k_t^3 = 0 \\ \sum_t k_t^4 = 0, \end{array} \right.$$

où c_0 et c_1 sont respectivement les premières et dernières années de naissance auxquelles la cohorte est ajustée et $c = t - x$. Les contraintes sur les kappas sont utilisées pour normaliser l'estimation des facteurs.

Ici on suppose que les décès suivent une loi de Poisson :

$$D_{x,t} \sim \text{Poisson}(E_{x,t} m_{x,t}).$$

Sous cette hypothèse de Poisson, les paramètres sont donc estimés par la méthode de *backfitting* qui consiste en la maximisation par itération de la log-vraisemblance de Poisson donnée par (3.3) par rapport à des blocs de paramètres, les autres étant maintenus fixes. Chaque étape correspond à une maximisation conditionnelle de la log-vraisemblance, obtenue soit par régression de Poisson avec offset, soit par expression analytique. L'algorithme est répété jusqu'à convergence de la log-vraisemblance.

$$L = \prod_{x,t} \frac{\mu_{x,t}^{D_{x,t}} \exp(-\mu_{x,t})}{D_{x,t}!}. \quad (3.3)$$

Choix du paramètre a du modèle SPO

Selon Seklecka, Pantelous et O'Hare([15]), l'impact de la température sur les fluctuations de la mortalité se fait davantage ressentir sur les populations âgées, c'est pourquoi ils ont émis cette contrainte sur le paramètre a ($a = 50$ dans SPO). Cette valeur peut être déterminée à partir des coefficients de corrélation de Pearson et de Spearman présentés dans la section (2.1) du chapitre précédent et on remarque que ces derniers paraissent stables autour de 40 ans. Afin de déterminer le meilleur a , nous avons simulé différentes valeurs dans l'intervalle $[40, 60]$ et sur base des critères MAPE, AIC et BIC, nous avons retenu les valeurs de a dans le tableau(3.1) pour les hommes et les femmes et pour les périodes annuelle, froide et chaude.

	Froid	Chaud	Annuel
Hommes	60	60	40
Femmes	40	60	40

TABLE 3.1 – Meilleur choix de a pour le modèle SPO mixte

Les figures (3.3) et (3.4) présentent l'évolution des facteurs du modèle SPO chez les hommes et les femmes pour les période annuelle, froide et chaude.

On observe que le facteur κ_t^1 a une tendance à la baisse tant chez les femmes que chez les hommes pour toutes les périodes, ce qui traduit l'amélioration progressive de la mortalité. Les oscillations hebdomadaires traduisent la saisonnalité de la mortalité et les variations intra-annuelles, telles que les épidémies hivernales ou les vagues de chaleur. On remarque que ce facteur est beaucoup plus irrégulier en période de froid qu'en période chaude.

Le facteur κ_t^2 décrit comment la mortalité évolue différemment entre les âges jeunes et les âges élevés. S'il augmente, alors l'amélioration de la mortalité est plus forte aux âges élevés qu'aux âges jeunes. C'est ce qu'on appelle la rotation du profil de mortalité par âge. De manière générale, homme/femme, annuelle/froid/chaud, on observe une augmentation croissante au début de la période suivie d'une baisse, ce qui veut dire que les améliorations de la mortalité ont été plus fortes aux âges élevés pendant la première partie de la période d'étude.

κ_t^3 capte les effets non linéaires de la mortalité chez les jeunes. Dans le cas annuel, on observe chez les hommes et chez les femmes une tendance globalement décroissante sur toute la période, ce qui pourrait traduire une amélioration de la mortalité des jeunes.

3.3 Estimation des paramètres du modèle SPO

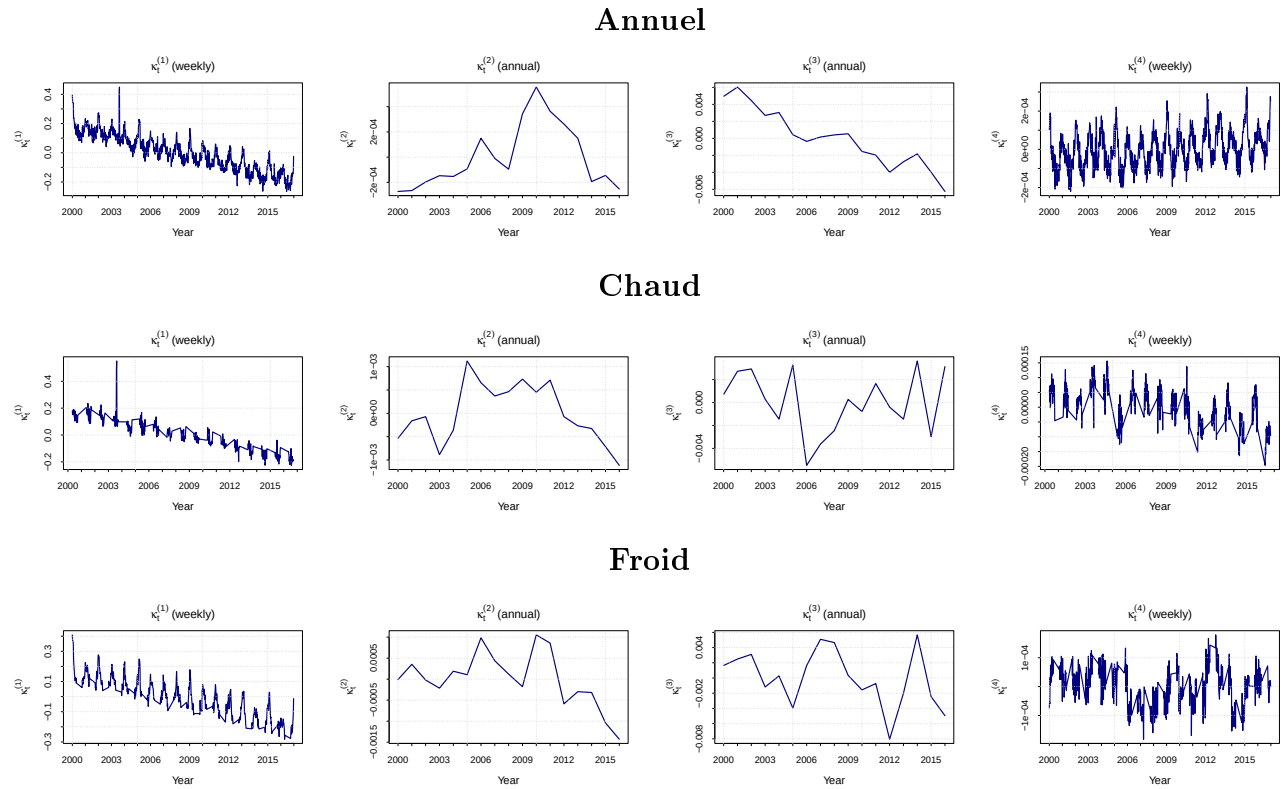


FIGURE 3.3 – Comparaison des quatre composantes κ_t chez les hommes pour les cas annuel, chaud et froid.

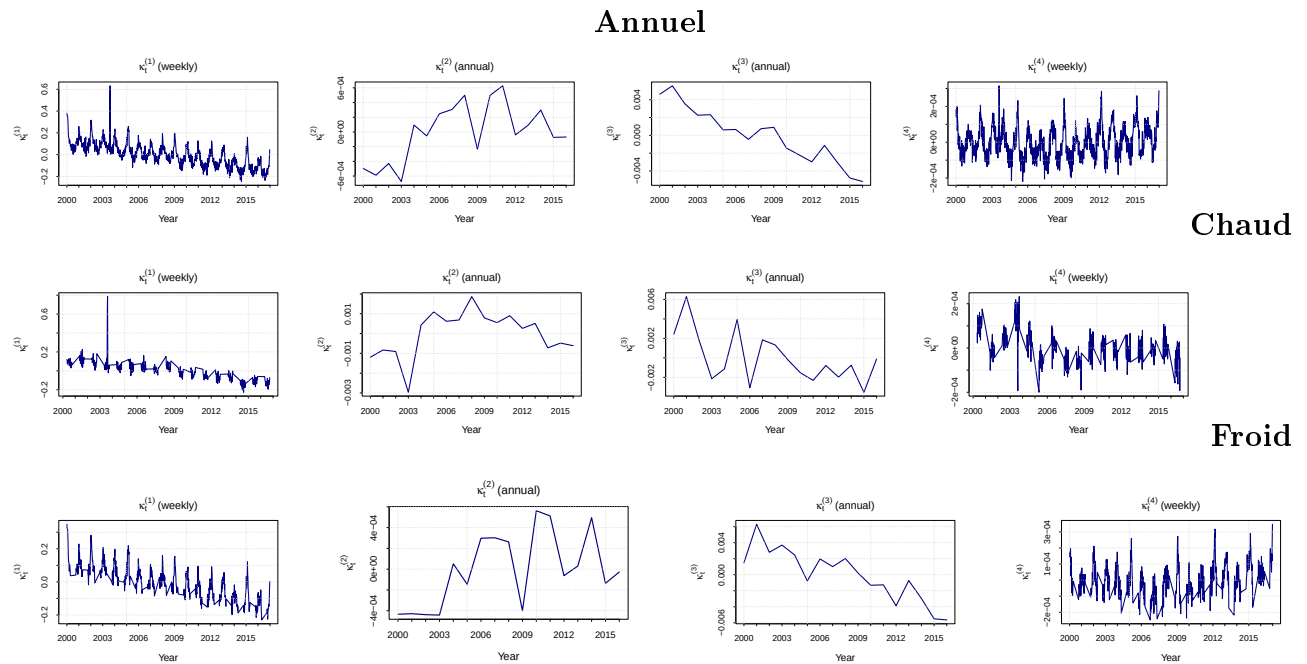


FIGURE 3.4 – Comparaison des quatre composantes κ_t chez les femmes pour les cas annuel, chaud et froid.

3.4 Qualité d'ajustement des modèles LC et SPO

Pour les cas chaud et froid, les graphiques sont plus irréguliers ce qui pourrait être lié aux épisodes climatiques.

Le paramètre κ_t^4 capte les effets de température sur la mortalité. Ils agissent à court terme et affectent plus fortement certaines classes d'âges. Sur les graphiques, on observe une forte volatilité du paramètre, des oscillations rapides. On voit une légère tendance à la hausse pour les hommes en période annuelle et des pics remarquables sur les autres reflétant les canicules ou les périodes de froid intense.

3.4 Qualité d'ajustement des modèles LC et SPO

3.4.1 Visualisation de l'ajustement des deux modèles par classe d'âge

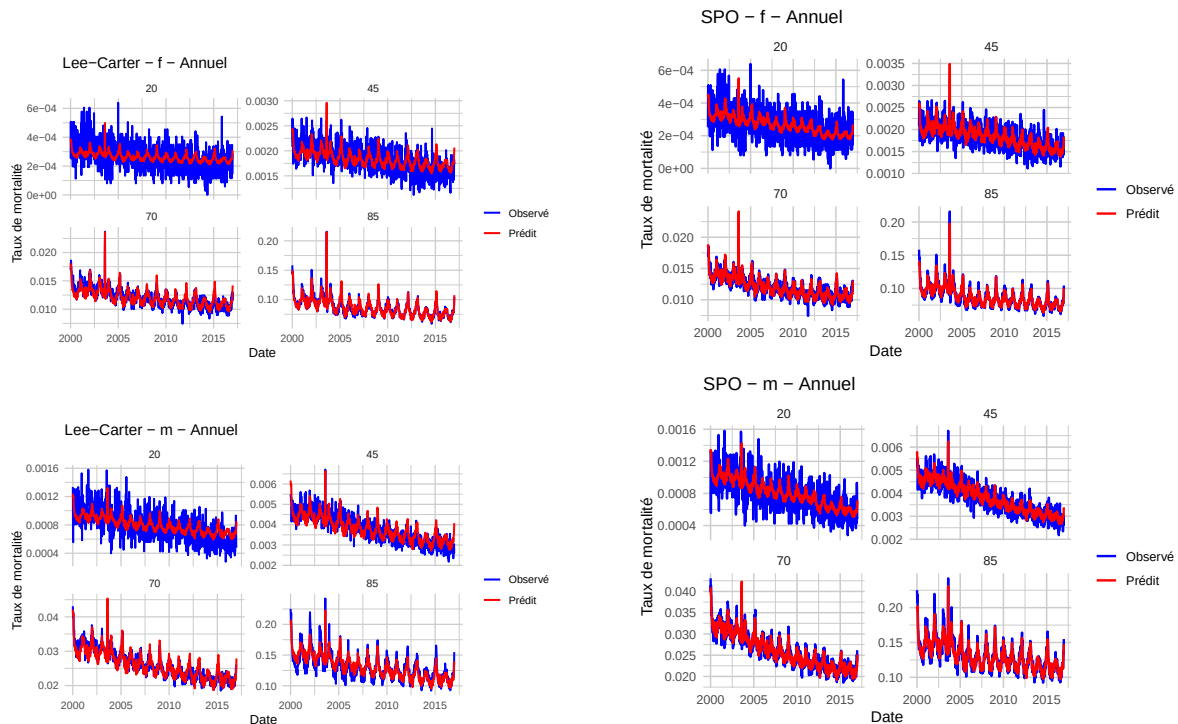


FIGURE 3.5 – Ajustement des modèles LC et SPO.

La figure (3.5) compare l'ajustement du modèle LC et du modèle SPO chez les femmes et les hommes pour le cas annuel selon différentes classes d'âge. On observe que les deux modèles reproduisent correctement la tendance générale de la mortalité, mais le modèle LC peine un peu à reproduire la tendance pour les âges jeunes, notamment 20–24 ans et 45–49 ans. Par contre le modèle SPO capte mieux la tendance à ces âges. Même aux âges élevés, le modèle SPO capte mieux la tendance que le modèle LC même si le modèle LC capte mieux certains pics que le modèle SPO. Nous allons faire une évaluation quantitative pour savoir lequel des deux s'ajuste le mieux aux données.

3.4.2 Evaluation des erreurs des modèles

Dans la section précédente, nous avons estimé les paramètres du modèle LC et SPO. L'objectif maintenant est de voir quel modèle ajuste le mieux les données et pour cela nous allons utiliser les critères tels que le RMSE, MAPE, MAD, BIC, AIC avec :

$$\text{MAPE} = \frac{1}{NM} \sum_{x,t} \frac{\|\hat{m}_{x,t} - m_{x,t}\|}{m_{x,t}}, \quad \text{MAD} = \sum_{x,t} \frac{\|\hat{m}_{x,t} - m_{x,t}\|}{NM}, \quad \text{RMSE} = \sqrt{\frac{1}{N} (m_{x,t} - \hat{m}_{x,t})^2},$$

où N est la dimension temporelle et M est la dimension d'âge. Le BIC et l'AIC sont calculés avec les formules :

$$\text{BIC} = -k \ln(n) + 2L(\psi), \quad \text{AIC} = 2k - 2 \ln(L),$$

avec $L(\psi)$ le maximum de vraisemblance du paramètre estimé ψ , n est le nombre d'observations et k le nombre de paramètres à estimer.

Sex	Period	MAD	RMSE	MAPE	AIC	BIC	N
LC							
Femmes	Annuel	0.0017843	0.0064215	20.22203	140604.52	147865.13	927
Femmes	froid	0.0019334	0.0068691	19.81356	82613.11	86666.70	556
Femmes	chaud	0.0013058	0.0048637	19.66464	56348.13	59208.99	411
Hommes	Annuel	0.0035118	0.0123618	16.53709	151428.22	158688.82	927
Hommes	froid	0.0034554	0.0121944	15.85223	86912.92	90966.51	556
Hommes	chaud	0.0023429	0.0087066	15.01347	59796.14	62656.99	411
SPO							
Femmes	Annuel	0.0010933	0.0032104	19.69238	136593.56	151741.36	1934
Femmes	froid	0.0012140	0.0035197	19.34386	80205.95	88896.38	1192
Femmes	chaud	0.0010353	0.0033754	18.35626	55793.21	62071.78	902
Hommes	Annuel	0.0020865	0.0066948	15.64044	144610.19	159757.99	1934
Hommes	froid	0.0028325	0.0095547	14.41055	84667.80	93358.23	1192
Hommes	chaud	0.0020357	0.0070006	13.85304	58979.82	65258.39	902

TABLE 3.2 – Qualité d'ajustement des modèles LC et SPO

Le tableau (3.2) présente le résultat du MAPE, MAD, RMSE, AIC et BIC pour les hommes et les femmes. En comparaison avec le modèle LC, le modèle SPO a beaucoup plus de paramètres. Ainsi, en termes de MAD, MAPE, RMSE, le modèle SPO produit des erreurs de prédiction plus faibles que le modèle classique de LC (19.62% et 20.22% pour le MAPE, 0.001 et 0.0017 pour le MAD, et 0.0032 et 0.0064 pour RMSE respectivement pour les femmes en période annuelle). Cela suggère que la prise en compte d'une composante temporelle additionnelle liée à la mortalité saisonnière améliore la représentation de la dynamique de la mortalité.

Cependant, le critère BIC indique que le modèle LC reste plus parcimonieux en raison de son faible nombre de paramètres. Cette différence s'explique par le fait que le critère BIC pénalise plus fortement la complexité du modèle que l'AIC. Outre ça, de manière générale, les résultats d'ajustements montrent que le modèle SPO produit des estimations de meilleure qualité pour les taux de mortalité.

3.5 Robustesse du modèle SPO mixte

Après avoir ajusté le modèle, il est toujours important de montrer que le modèle est robuste. Dans cette section, nous montrons la robustesse de notre modèle en regardant tout d'abord la corrélation entre la mortalité par âge et la température dans le temps, ensuite nous analysons l'impact de l'ajout d'un bruit aléatoire sur le coefficient de corrélation et enfin nous faisons un nouvel ajustement de notre modèle sur des données de 2000 à 2010 afin de voir si le modèle reste meilleur sur cette période réduite.

Nous avons scindé notre période d'étude en sous-période (2000-2008, 2000-2010, 2000-2016, 2000-2019) et nous avons calculé le coefficient de corrélation de Pearson afin de voir s'il reste stable dans le temps. Le graphique (3.6) nous présente l'évolution du coefficient de corrélation de Pearson chez les hommes et les femmes pour différentes fenêtres temporelles, pour les périodes annuelle, chaude et froide. De manière globale, on observe une tendance similaire sur toutes les périodes de temps considérées. Chez les hommes comme chez les femmes, on observe pour les données annuelles et froides des formes identiques avec une légère variation de niveau. Cependant, on remarque une différence un peu plus marquée pour l'ensemble des données chaudes avec le reste des données. On remarque qu'il y a un peu plus de dispersion, surtout chez les hommes, mais l'effet reste quand même stable dans le temps. On remarque aussi plus de volatilité pour les périodes chaudes et en particulier chez les femmes aux âges élevés. Nous observons également que la réduction du nombre d'observations pour l'ensemble des données annuelles, froides et chaudes fait baisser la valeur du coefficient de corrélation, néanmoins, la forme générale reste la même. Nous pouvons donc conclure que la corrélation entre la mortalité et la température ne dépend pas significativement du choix de la fenêtre temporelle.

D'autre part, dans la réalité, les données de température ne sont jamais mesurées parfaitement. Pour cette raison, nous introduisons une perturbation aléatoire additive afin d'évaluer la sensibilité du coefficient de corrélation à des erreurs de mesure. Plus précisément, pour chaque semaine t , nous définissons une température perturbée par :

$$\text{TAVG}_t^{\text{noise}} = \text{TAVG}_t + \epsilon_t,$$

où $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ est une variable aléatoire gaussienne centrée, indépendante d'une semaine à l'autre. L'écart-type σ est défini comme une fraction de l'écart-type empirique de la tempéra-

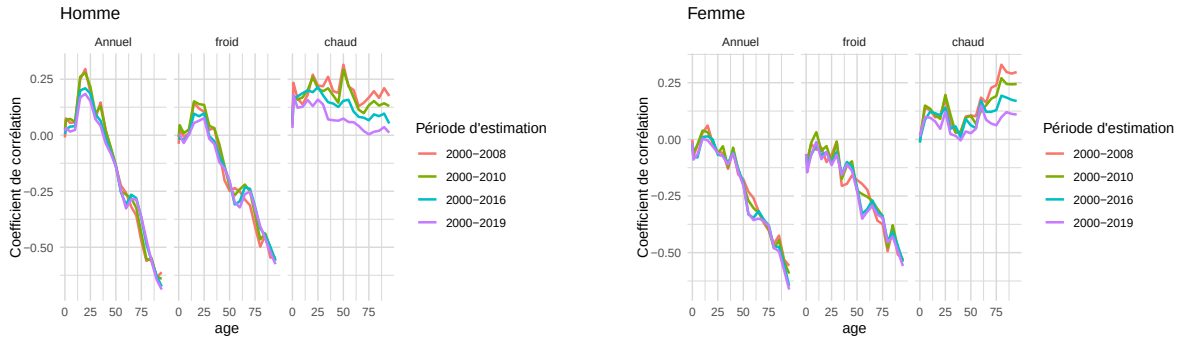


FIGURE 3.6 – Coefficients de corrélation de Pearson pour homme et femme.

ture, soit :

$$\sigma = \alpha \cdot \sigma_T, \quad \text{avec } \alpha \in \{5\%, 20\%\}.$$

σ_T est l'écart-type empirique et vaut 6.332762 °C.

Cette spécification permet de simuler des perturbations réalistes allant d'une faible incertitude (5%) à une perturbation plus importante (20%), proportionnelles à la variabilité observée dans les données de température, tout en conservant une moyenne nulle. Le coefficient de corrélation associé est alors donné par :

$$c_{tx}^{noise} = \text{corr}(\text{TAVG}_t^{noise}, m_{x,t}),$$

que nous comparons à la corrélation de référence donnée par :

$$c_{tx} = \text{corr}(\text{TAVG}_t, m_{x,t}).$$

Le graphique (3.7) nous présente différentes évolutions des coefficients de corrélations de Pearson avec et sans perturbation selon le sexe et la période (annuel/froid/chaud) en fonction de l'âge. On observe que les courbes simulées restent proches de la courbe de base et conservent la forme globale. Plus précisément, avec une perturbation de 5%, les courbes sont quasiment confondues, elles ne présentent aucune dispersion visible, ce qui montre que le coefficient de corrélation reste insensible aux petites variations. Pour la perturbation de 20%, on note une légère dispersion des courbes, surtout visible en période chaude et plus volatile chez les femmes. Le bruit commence à avoir un effet mais la structure reste intacte. De manière générale, en période annuelle et froide, l'impact du bruit est très faible. Cependant, en période chaude, lorsque la perturbation devient importante, les courbes s'écartent un peu plus. Néanmoins, la structure globale des coefficients de corrélation reste inchangée, ce qui confirme la stabilité du modèle face à des erreurs de mesure.

Durant l'ajustement, nous avons utilisé les données sur la période allant de 2000 à 2016 afin de réserver les 3 dernières années pour les tests hors-échantillons. Pour montrer que notre modèle est robuste, nous avons séparé cette période entre 2000 et 2010 et nous avons à nouveau ajusté les paramètres sur cette période et il en ressort que les erreurs (voir annexe 4.3) du

3.6 Projection des taux de mortalité des modèles LC et SPO

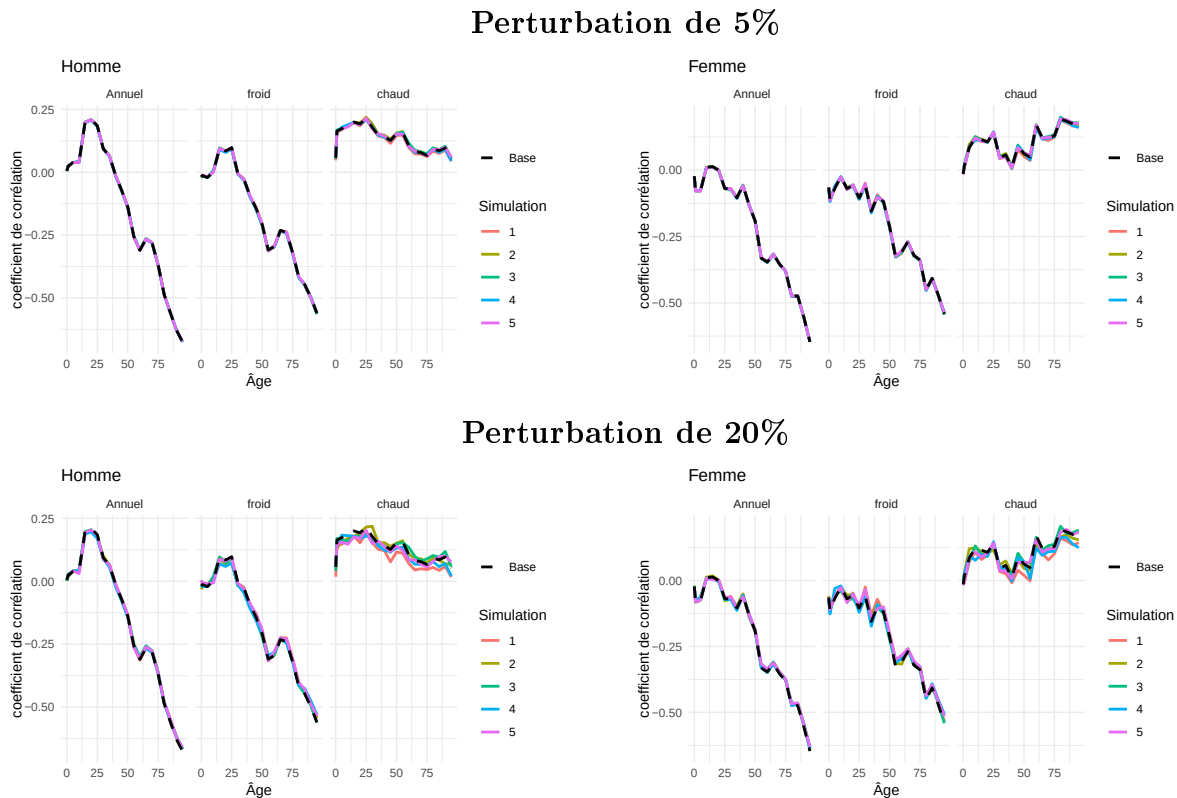


FIGURE 3.7 – Comparaison entre les Coefficients de corrélation de Pearson pour homme et femme avec et sans perturbation.

modèle SPO mixte restent tout autant plus faibles que celles du modèle LC avec toujours une pénalisation plus forte du critère BIC du modèle SPO.

Au vu de toutes ces analyses, nous pouvons conclure que le coefficient de corrélation entre le taux de mortalité et la température est robuste. Il est stable dans le temps et reste presque insensible aux éventuelles erreurs liées à la température. De plus, l'ajustement du modèle sur une période plus courte conserve de meilleurs résultats en termes d'erreurs que le modèle LC. D'où, le modèle SPO est un modèle robuste.

3.6 Projection des taux de mortalité des modèles LC et SPO

Dans les sections précédentes, nous avons fait l'étude in-sample des modèles LC et SPO et nous avons conclu que le modèle SPO s'ajuste mieux aux données que le modèle LC malgré sa complexité et son nombre élevé de paramètres. Nous allons par la suite faire une étude hors-échantillon, c'est-à-dire, nous allons faire une projection des taux de mortalité sur une période des données non utilisée lors de l'estimation des modèles (2017-2019). Pour ce faire, nous devons tout d'abord déterminer quel modèle (ARIMA (p,d,q)) ou (SARIMA(p,d,q)(P,D,Q)_s)

correspond à nos différents paramètres, pour cela nous allons utiliser la méthodologie de Box-Jenkins dont les étapes sont les suivantes :

1. Identification du modèle

- Analyse des données : Examiner les données de la série temporelle pour détecter les tendances, les saisonnalités et les comportements anormaux.
- Différenciation : Si la série temporelle présente une tendance ou une saisonnalité, appliquer une différenciation pour rendre les données stationnaires.
- Identification des ordres : Utiliser les graphiques ACF (fonction d'autocorrélation) et PACF (fonction d'autocorrélation partielle) pour déterminer les ordres p , d et q du modèle ARIMA.

2. Estimation du modèle

- Estimation des paramètres : Utiliser les méthodes d'estimation (telles que la méthode des moindres carrés) pour estimer les paramètres du modèle ARIMA.

3. Vérification du modèle

- Diagnostic du modèle : Vérifier si les résidus du modèle ARIMA sont des bruits blancs (c'est-à-dire s'ils ne présentent pas de structure ou de corrélation significative).
- Réajustement : Si le modèle ne satisfait pas les critères de bruit blanc, ajuster les ordres du modèle ARIMA et répéter les étapes précédentes.
- Validation : Valider les performances du modèle en effectuant des prédictions sur des données de validation ou en utilisant des mesures d'évaluation telles que l'erreur quadratique moyenne (RMSE) ou le critère d'information d'Akaike (AIC).

3.6.1 Projection du modèle LC

La projection du modèle LC repose sur l'étude des caractéristiques de la série temporelle κ_t pour les deux sexes et pour les trois périodes considérées.

3.6.1.1 Retrait de la saisonnalité

Tout d'abord, si nous visualisons les différentes séries (H/F*A/F/C) des figures (3.1) et (3.2), on observe une tendance saisonnière très marquée pour les périodes froides et annuelles et un peu moins marquée pour la période chaude. Pour s'en convaincre, nous pouvons observer sur la figure(3.8) représentant l'ACF brute des séries pour toutes les périodes que l'ACF des cas froid et annuel présente une saisonnalité avec $s = 52$ pour le cas annuel et $s = 30$ pour froid. Par contre, l'ACF du cas chaud est juste décroissante sans saisonnalité, donc les périodes chaudes n'ont pas de saisonnalité. Sur cette base, nous allons donc projeter les séries temporelles annuelles et froides avec un SARIMA(p,d,q)(P,D,Q) $_s$ et chaudes par ARIMA(p,d,q). Les séries sans saisonnalité sont données par la figure (3.9).

3.6 Projection des taux de mortalité des modèles LC et SPO

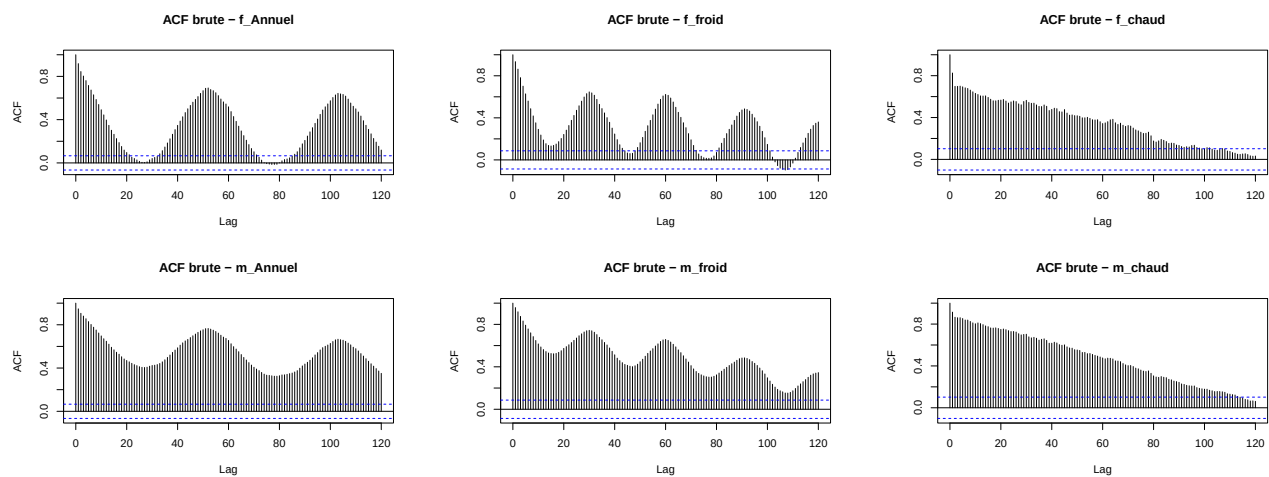


FIGURE 3.8 – ACF des séries brutes chez les hommes et femmes pour les cas annuel, chaud et froid.

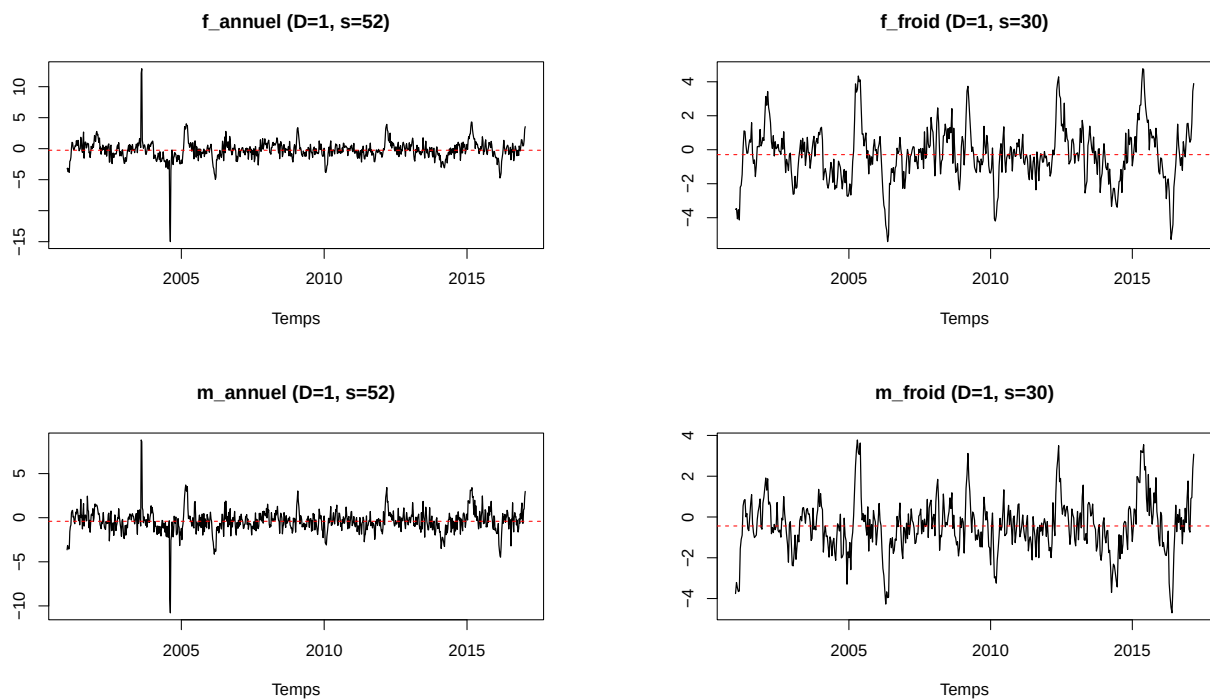


FIGURE 3.9 – Séries sans saisonnalités chez les hommes et femmes pour les cas annuel, chaud et froid.

3.6.1.2 Retrait de la tendance

Après avoir retiré la saisonnalité, il faut rendre la série stationnaire en retirant la tendance pour pouvoir appliquer la méthode de Box-Jenkins. Afin de rendre nos séries stationnaires,

3.6 Projection des taux de mortalité des modèles LC et SPO

nous avons fait une différenciation $d=1$. La figure (3.10) nous présente les différentes séries stationnaires.

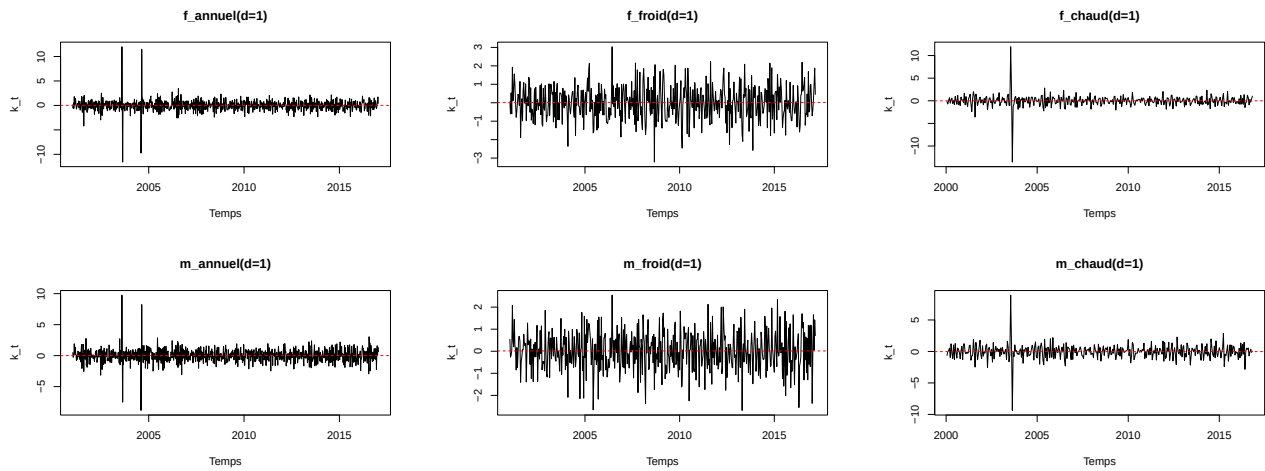


FIGURE 3.10 – Séries stationnaires chez les hommes et femmes pour les cas annuel, chaud et froid.

3.6.1.3 Analyse de l'ACF et de la PACF

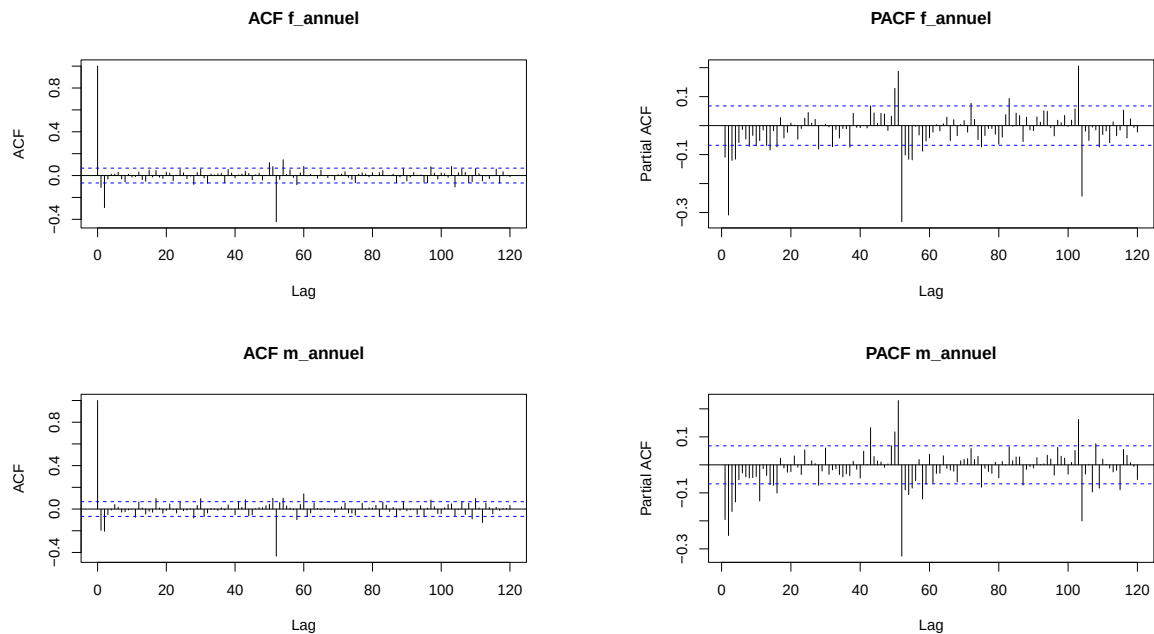


FIGURE 3.11 – ACF et PACF des séries nettoyées.

La figure (3.11) donne l'ACF et le PACF de la série κ_t annuelle chez les femmes et les hommes.

3.6 Projection des taux de mortalité des modèles LC et SPO

Chez les hommes comme chez les femmes, au niveau de la dépendance annuelle, on remarque au lag 1 (i.e. $s = 52$) un pic significatif sur l'ACF et deux pics sur le PACF aux lags 52 et 104. Donc on pourrait avoir une combinaison de paramètre ($p = 0, 1, 2; q = 0, 1, 2$). Pour la dépendance hebdomadaire, on voit sur l'ACF un pic significatif à la première position et deux petits pics ensuite. Sur le PACF, on observe également des pics, ce qui justifie la présence d'un AR(P). Ainsi, une combinaison de paramètre candidat pourrait être ($P = 0, 1, 2; Q = 0, 1, 2$). Nous avons fait le même raisonnement pour les cas froids et chauds afin de trouver des candidats pour les différents paramètres. Par la suite, nous avons mis en place une procédure de sélection du meilleur modèle SARIMA. Les ordres de différenciation non saisonnière d et saisonnière D , ainsi que la fréquence saisonnière S , ont été fixés au préalable à partir des diagnostics de stationnarité et de saisonnalité que nous avons fait précédemment. Ensuite, plusieurs combinaisons des ordres auto-régressifs et moyennes mobiles, non saisonnière (p, q) et saisonnière (P, Q) ont été testées. Chaque modèle a été estimé par maximum de vraisemblance. Les modèles ont été classés par ordre croissant d'AIC. Nous avons également regardé le nombre de paramètre estimés, la log-vraisemblance, la significativité statistique des coefficients et la p-value du test de Ljung-Box appliqué aux résidus.

Sur base de ce processus, nous avons choisi les modèles en fonction de l'AIC et pour ceux dont l'AIC est assez proche, nous avons choisi les plus parcimonieux.

id	p	d	q	P	D	Q	s	ncoef	AIC	lb_pvalue	all_coef_sig
f_froid	0	1	0	0	1	1	30	1	1093.264	0.2443859	TRUE
f_froid	1	1	1	0	1	1	30	3	1095.993	0.2092013	TRUE
f_chaud	0	1	2	0	0	0	22	2	1135.942	0.9899110	TRUE
f_chaud	1	1	2	1	0	0	22	4	1139.688	0.9890752	TRUE
f_annuel	1	1	2	0	1	1	52	4	2134.117	0.5717846	TRUE
f_annuel	0	1	2	2	1	2	52	6	2138.795	0.2554083	TRUE
m_froid	1	1	1	0	1	1	30	3	1006.114	0.4936628	TRUE
m_froid	0	1	1	0	1	1	30	2	1035.770	0.0633300	TRUE
m_chaud	0	1	2	0	0	0	22	2	1033.421	0.6445448	TRUE
m_chaud	0	1	2	1	0	1	22	4	1033.966	0.6713380	TRUE
m_annuel	1	1	1	0	1	1	52	3	1986.359	0.0767172	TRUE
m_annuel	0	1	2	0	1	1	52	3	2007.813	0.0559648	TRUE

TABLE 3.3 – Tableau des meilleurs modèles de SARIMA

3.6.1.4 Validation

Pour valider le choix de nos paramètres, nous avons représenté les résidus standardisés en utilisant la fonction *tsdiag()*. Cette fonction permet de vérifier si les résidus se comportent comme un bruit blanc. La figure (3.3) nous présente la sortie pour les hommes en période

3.6 Projection des taux de mortalité des modèles LC et SPO

de froid (voir annexe(4.5) pour les autres cas). On peut observer que le graphe des résidus standardisés oscille autour de zéro avec une variance approximativement constante et sans tendance apparente, ce qui est compatible avec un bruit blanc. Sur l'ACF, on voit un pic au lag 1 et plus rien ensuite, les autres barres restent dans les bandes, ce qui suggère qu'il ne reste plus de dépendances temporelles significative. Enfin, les p-valeurs du test de Ljung-Box sont toutes supérieures à 0.05, ce qui signifie que l'hypothèse nulle selon laquelle il n'y a pas d'autocorrélation significative dans les résidus n'est pas rejetée.

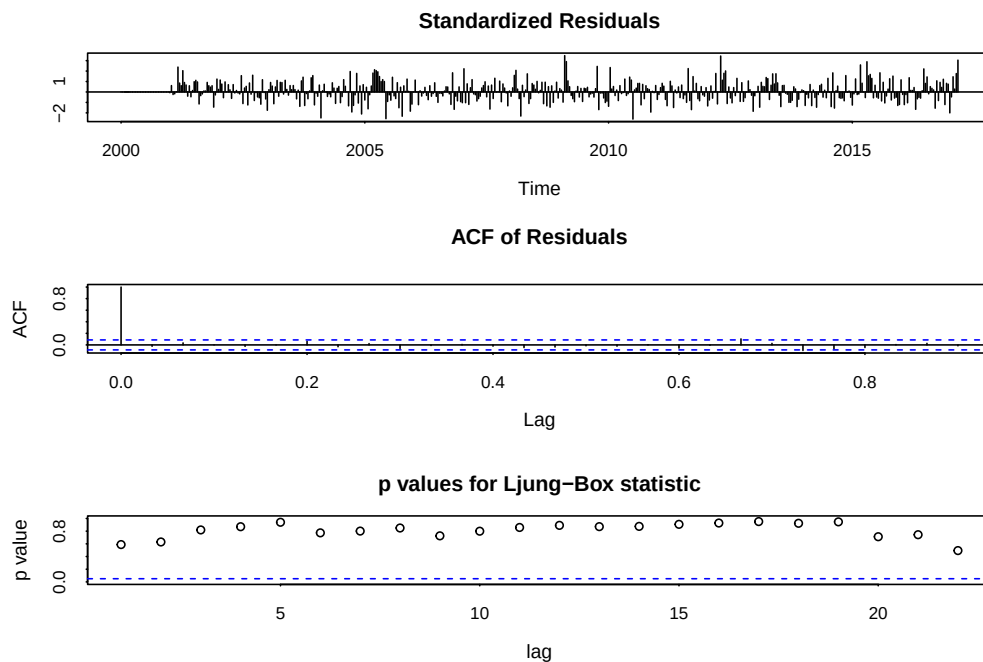


FIGURE 3.12 – Résidus standardisés pour homme froid.

3.6 Projection des taux de mortalité des modèles LC et SPO

3.6.1.5 Prédiction

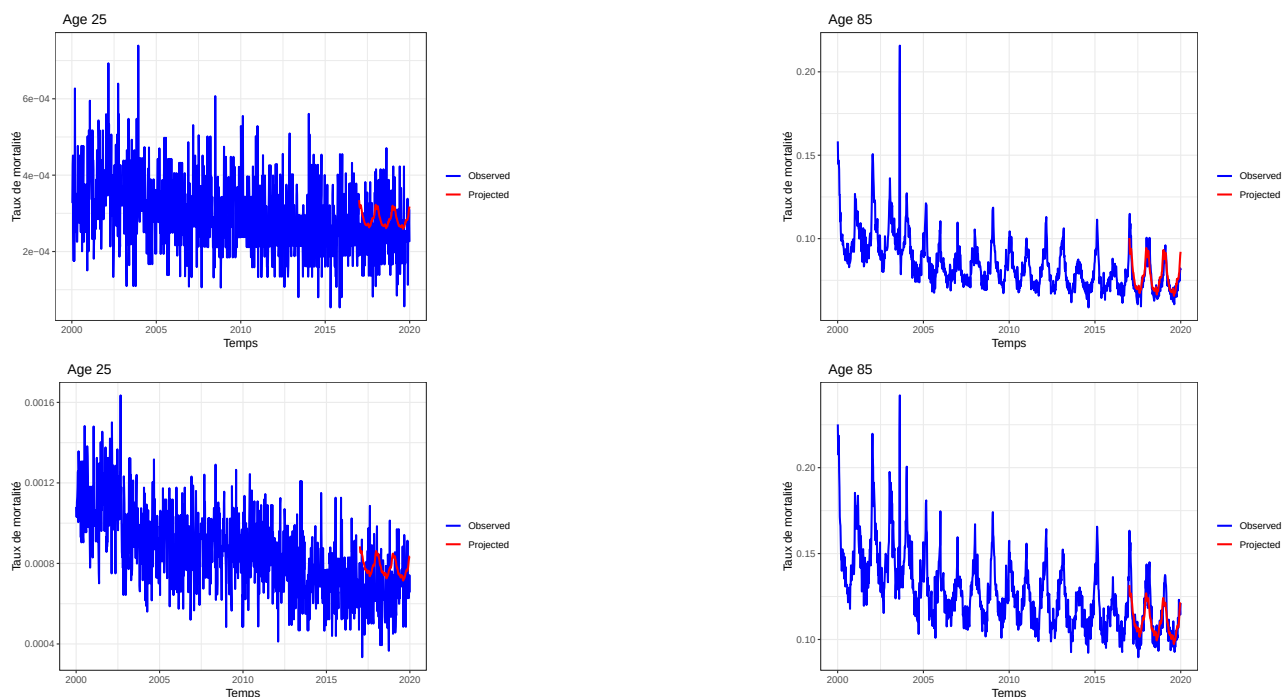


FIGURE 3.13 – Projections hors-échantillon des taux de mortalité chez les hommes et femmes pour le cas annuel avec LC sur 2017-2019.

La figure (3.13) présente la projection des taux de mortalité sur l'échantillon 2017-2019 pour la période annuelle chez les hommes(bas) et les femmes(haut) par le modèle de LC. Pour les âges jeunes (25-29 ans), on remarque que le modèle LC s'ajuste mieux aux données chez les femmes que chez les hommes. De même chez les personnes âgées, LC semblent mieux s'ajuster aux données chez les femmes que chez les hommes.

3.6.2 Projection du modèle SPO

3.6.2.1 Paramètres du modèle et Projection hors-échantillon

Nous avons appliqué les mêmes étapes de la même méthode de Box-Jenkins énoncée dans la section précédente pour déterminer les paramètres du modèle.

3.6 Projection des taux de mortalité des modèles LC et SPO

Sex	κ_t^1	κ_t^2	κ_t^3	κ_t^4
f_annuel	(1,1,1) x (0,1,1)	(0,1,1)	(0,1,0)	(1,1,2) x (0,1,1)
f_chaud	(0,1,2)	(0,1,1)	(0,1,1)	(0,1,1) x (2,0,0)
f_froid	(2,1,1) x (0,1,1)	(0,1,1)	(1,1,0)	(2,1,1) x (1,0,2)
m_annuel	(0,1,2) x (1,0,1)	(0,1,0)	(0,1,0)	(1,1,2) x (0,1,1)
m_chaud	(0,1,2)	(0,1,0)	(0,1,1)	(2,1,2)
m_froid	(1,1,1) x (0,1,1)	(0,1,0)	(0,1,2)	(0,1,1)

TABLE 3.4 – Structure des processus ARIMA/SARIMA pour la projection des séries stochastiques κ_t du modèle SPO

Le tableau (3.4) présente la structure des modèles ARIMA/SARIMA retenus pour la projection des paramètres $(\kappa_t^1, \kappa_t^2, \kappa_t^3, \kappa_t^4)$ du modèle SPO, pour chaque sexe et chaque période (annuelle, chaude et froide). Pour chaque série, le modèle a été sélectionné à l'aide de la méthodologie de Box-Jenkins, en comparant plusieurs modèles possibles et en retenant celui qui minimise le critère d'information d'Akaike (AIC), tout en vérifiant la significativité des coefficients et l'absence d'auto-corrélation résiduelle à l'aide du test de Ljung-Box. On observe que les paramètres κ_t^1 et κ_t^4 , définis à fréquence hebdomadaire dans le modèle SPO, sont généralement modélisés par des processus SARIMA incluant une composante saisonnière, ce qui reflète la présence d'une structure périodique dans les séries temporelles.

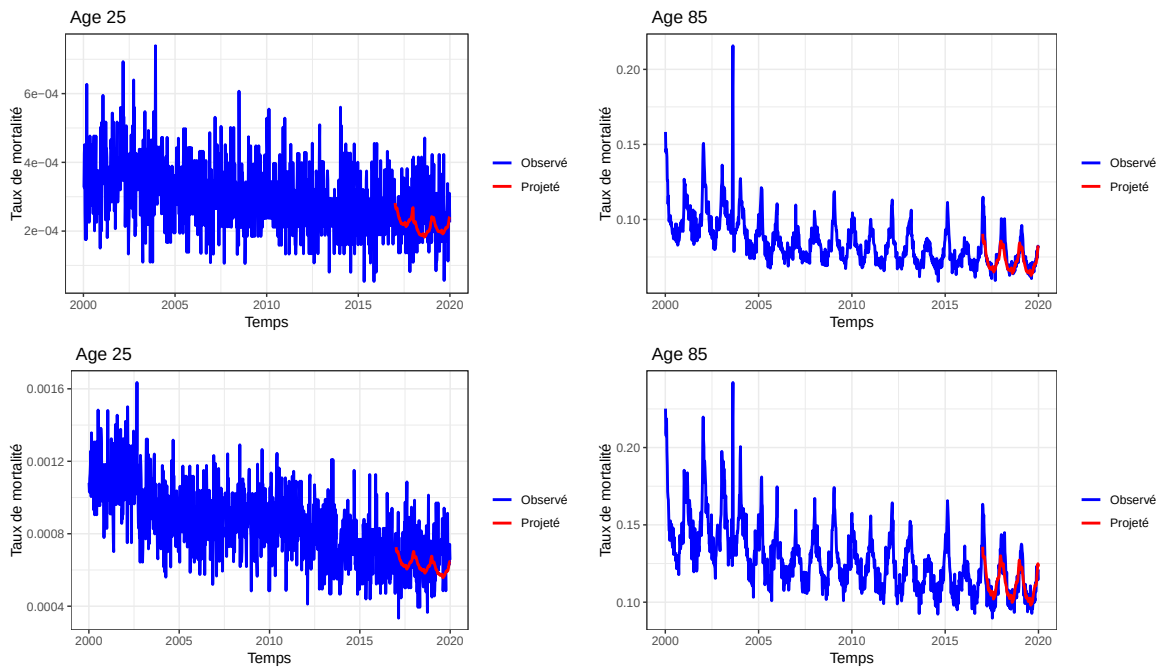


FIGURE 3.14 – Projections Out-of-sample des taux de mortalité chez les hommes et femmes pour le cas annuel avec SPO sur 2017-2019.

3.6 Projection des taux de mortalité des modèles LC et SPO

La figure (3.14) présente la projection des taux de mortalité sur l'échantillon 2017-2019 pour la période annuelle chez les hommes(bas) et les femmes(haut) par le modèle SPO. On observe que, pour les jeunes âges (25-29 ans), le modèle SPO s'aligne mieux sur les données pour les deux sexes que le modèle LC. Pour les plus âgés (85-89 ans), le modèle SPO suit parfaitement la tendance mais ne capte pas totalement les pics, ce qui pourrait laisser croire qu'il sous-estime un peu contrairement à LC qui montre plutôt une tendance à la hausse. Dans la section suivante, nous allons faire une analyse quantitative des erreurs de prédiction afin de savoir quel modèle parmi les deux donne de meilleurs résultats.

3.6.3 Qualité d'ajustement hors-échantillon des modèles LC et SPO

Le tableau(3.5) nous présente les résultats des mesures de prédictions hors-échantillon avec les modèles SARIMA/ARIMA sur la période 2017-2019. Comme nous pouvons le voir dans le tableau, les résultats sont très bons pour les périodes annuelle et froide chez les hommes et chez les femmes car la saisonnalité se fait plus ressentir pour ces périodes malgré la courte période de validation considérée. On voit également que le modèle ajuste mieux chez les femmes que chez les hommes avec un MAPE qui part de 0.26 pour le modèle LC à 0.21 pour le modèle SPO en annuel (et plus particulièrement, de 0.39 à 0.20, et de 0.26 à 0.23 pour froid et chaud respectivement). Cependant, pour le cas chaud, la prédiction est en faveur du modèle de LC du au fait que sur la période considérée, les événements chauds ne sont pas assez représentés pour être modélisés par un paramètre externe, et comme le modèle est assez complexe avec son nombre élevé de paramètres, LC donne un meilleur ajustement.

Sex	Period	RMSE	MAD	MAPE
LC				
Femmes	Annuel	0.006928	0.002035	0.268793
Femmes	froid	0.016143	0.006364	0.398744
Femmes	chaud	0.003707	0.001289	0.266255
Hommes	Annuel	0.013123	0.003612	0.218542
Hommes	froid	0.013535	0.003815	0.188973
Hommes	chaud	0.006788	0.002246	0.182522
SPO				
Femmes	Annuel	0.006340	0.001803	0.217879
Femmes	froid	0.007416	0.002296	0.208539
Femmes	chaud	0.003907	0.001286	0.237278
Hommes	Annuel	0.009006	0.002857	0.184352
Hommes	froid	0.010913	0.003203	0.188407
Hommes	chaud	0.006988	0.002316	0.186719

TABLE 3.5 – Qualité d'ajustement des modèles LC et SPO sur les données test 2017-2019

3.7 Conclusion

Dans ce chapitre, nous avons étudié l'impact de la prise en compte d'un facteur lié à la température dans les modèles de mortalité en faisant une étude comparative entre le modèle Lee-Carter([9]) et le modèle de Seklecka & Al.([15]) qui prend en compte la température à travers un coefficient ct_x qui représente la corrélation entre la mortalité et la température que nous avons étudié au chapitre précédent. Nous avons vu que la température a beaucoup plus d'influence sur les personnes âgées. Nous avons ajusté les deux modèles sur nos données et nous avons calculé les erreurs d'ajustement et il en ressort que le modèle SPO s'ajuste mieux aux données sur la période considérée, ici, 2000-2016. Par la suite, nous avons projeté les taux de mortalité sur la période de test 2017-2019 et mesuré les erreurs de projection. Nous avons vu que le modèle SPO donne également de meilleurs résultats de projection. Cependant, le critère BIC pénalise fortement le modèle SPO en raison de sa complexité due à son nombre élevé de paramètres. Outre ça, le modèle SPO est meilleur en matière de prédiction que le modèle LC. Nous avons également montré que le modèle SPO est un modèle robuste. L'objectif maintenant est de voir de manière concrète quel est l'avantage de prendre en compte une composante thermique dans la tarification des produits d'assurance.

IMPACT DES MODÈLES DE MORTALITÉ INTÉGRANT LA TEMPÉRATURE SUR LA TARIFICATION D'UN PORTEFEUILLE DE RENTIERS

4.1 Introduction

La tarification en assurance vie, et en particulier des rentes repose sur l'estimation des taux de mortalité. Ainsi, la qualité de ces estimations est primordiale à cause du fait qu'elle impacte directement sur la valorisation des engagements de l'assureur ainsi que sur l'équilibre financier du portefeuille. Il est donc crucial que les modèles de mortalité utilisés reflètent au mieux la réalité observée en prenant en compte les facteurs susceptibles d'influencer la mortalité humaine. Nous avons montré dans le chapitre précédent que l'introduction d'un facteur lié à la température améliore l'ajustement du modèle aux données observées et nous avons vérifié cela en évaluant les erreurs entre les taux de mortalité bruts et ceux estimés par les modèles LC et SPO. Dans ce chapitre, nous nous intéressons à la traduction financière de ces améliorations. Plus précisément, nous analysons l'impact de la prise en compte de la température sur la valorisation d'un portefeuille de rentiers. Pour cela, nous comparons les prix de rente obtenus à partir des données brutes et ceux issus des modèles LC et SPO.

4.2 Construction du produit de rente

Bien que les compagnies d'assurance proposent de nombreuses variantes de produits financiers, la plupart d'entre eux sont basés sur trois produits principaux : les rentes, l'assurance vie et le capital différé ou une combinaison d'une assurance vie et d'un capital différé appelé endowment. Dans cette section, nous allons présenter une analyse plus approfondie de l'un de ces produits, à savoir une rente sur n ans, dans laquelle un montant de 1 est versé à la fin de chaque année pendant n ans, et montrer l'impact de la température sur sa tarification. Bien

4.3 Méthodologie de la comparaison

que les modèles de mortalité soient formulés sur une base hebdomadaire, nous allons considérer un produit de rente avec des paiements mensuels.

Nous considérons un produit de rente temporaire versant un montant unitaire à la fin de chaque mois, tant que l'assuré est en vie, sur un horizon de (M) mois. Pour un individu appartenant à une classe d'âge (x) à la semaine initiale (S_0) , les paiements sont effectués à des instants correspondant à des multiples de semaines. Soit i le taux d'intérêt annuel constant. Le facteur d'actualisation hebdomadaire est donné par :

$$v = \frac{1}{(1+i)^{1/52}}.$$

Donc, la valeur actualisée d'un paiement effectué après t semaines est :

$$v = \frac{1}{(1+i)^{t/52}}.$$

Les probabilités de survie sont construites à partir des taux de mortalité hebdomadaires $m_{x,S}$ selon :

$$p_{x,S} = \exp(-m_{x,S}). \quad (4.1)$$

La probabilité de survie cumulée sur (t) semaines est alors donnée par :

$${}_t p_{x,S_0} = \prod_{j=0}^{t-1} p_{x_j, S_0+j}. \quad (4.2)$$

La valeur actuarielle de la rente est donc donnée par :

$$\ddot{a}_{x:\overline{n}|} = \sum_{s=1}^n v^{t_k} \cdot {}_{t_k} p_{x,S_0}, \quad (4.3)$$

où (t_k) représente le nombre de semaines correspondant au (k) -ième paiement mensuel.

Cette formulation permet de combiner une modélisation fine de la mortalité à l'échelle hebdomadaire avec une structure de paiement réaliste. Chaque terme de la somme correspond à un paiement mensuel actualisé, pondéré par la probabilité de survie jusqu'à la date correspondante.

4.3 Méthodologie de la comparaison

L'objectif de cette section est de comparer l'impact des différents modèles de mortalité sur la valorisation d'un portefeuille de rentiers. Plus précisément, nous analysons les différences de prix de rente obtenues à partir des taux de mortalité issus des modèles de Lee-Carter (LC) et du modèle de Seklecka et al., ainsi que des données observées.

La comparaison repose sur le calcul de la valeur actuarielle d'une rente mensuelle, définie à la section précédente. Pour un individu appartenant à une classe d'âge (x) à la semaine initiale (S_0) , la valeur de la rente est calculée selon trois sources de mortalité :

4.4 Calcul Numérique de la rente

- les taux observés (données brutes), notés $m_{x,S}^{obs}$;
- les taux estimés par le modèle de Lee-Carter, notés $m_{x,S}^{LC}$;
- les taux estimés par le modèle SPO, notés $m_{x,S}^{SPO}$;

Ces taux sont transformés en probabilités de survie hebdomadaire, puis agrégés afin de calculer les probabilités de survie aux dates de paiement mensuelles.

Ainsi, pour chaque combinaison d'âge initial et d'horizon de rente, en remplaçant :

$$mod \in obs, LC, SPO$$

dans la formule :

$$\ddot{a}_{x:\overline{n}|} = \sum_{s=1}^n V^{t_k} \cdot {}_{t_k}P_{x,S_0}^{(mod)}, \quad (4.4)$$

nous obtenons les trois valeurs suivantes :

$$PV^{obs}, \quad PV^{LC}, \quad PV^{SPO}.$$

Les horizons de projections sont choisis de manière à rester compatible avec la période d'observation des données. En effet, les données étant disponibles jusqu'à l'année 2019, la comparaison avec les données brutes n'est possible que sur un horizon pour lequel les trajectoires de mortalité observées sont disponibles. Par conséquent, pour une semaine initiale S_0 , l'horizon maximal (H) est déterminé par la dernière observation disponible. Dans la pratique, nous considérons plusieurs horizons (par exemple, 1 an, 2 ans, 5 ans, 15 ans) afin d'analyser la sensibilité des résultats à la durée de la rente.

4.4 Calcul Numérique de la rente

4.4.1 Comparaison des prix de rente

Le calcul de la rente s'est fait à l'aide de la formule (4.4). Pour ce faire, comme nous sommes en discret, nous avons, dans un premier temps, déterminé les probabilités de survie à chaque semaine à l'aide de la formule (4.1) et ensuite cumulé à l'aide de (4.2) pour avoir la probabilité de survie jusqu'en 2016. Le taux d'intérêt i est fixé à 2% et la date de départ pour tout le monde est le 03/01/2000. Nous avons fait varier les horizons n allant de 1 à 15 ans et la rente est payée à la fin du mois. Les calculs ont été faits par sexe et sur une période annuelle. Ainsi, le tableau (4.1) nous donne les différents prix à horizons de 1 an, 5 ans et 10 ans pour les personnes âgées de 30, 50 et 60 ans chez les hommes et les femmes.

On voit que les prix issus du modèle SPO (PV_SPO) sont, dans la plupart des cas, plus bas que les prix issus des données brutes(PV_raw) et de ceux issus du modèle LC(PV_LC). Cependant, les prix du modèle SPO sont de manière générale plus proches des prix de référence que ceux du modèle LC. Cela montre l'importance de prendre en compte le facteur température

4.4 Calcul Numérique de la rente

dans la modélisation de la mortalité car la prise en compte de ce facteur permet d'avoir des prix de rentes qui se rapprochent un peu plus de la réalité.

Sex	Age	n	PV_raw	PV_LC	PV_SPO
Femmes					
	30	1	11.682	11.708	11.685
	30	5	53.376	53.715	53.344
	30	10	94.686	95.589	94.623
	50	1	10.977	10.939	10.946
	50	5	40.470	40.266	40.294
	50	10	56.307	56.053	56.043
Hommes					
	30	1	11.442	11.468	11.431
	30	5	48.718	49.151	48.573
	30	10	79.939	80.786	79.617
	50	1	9.864	9.774	9.830
	50	5	26.998	26.752	26.752
	50	10	30.569	30.343	30.538

TABLE 4.1 – Prix des rentes mensuelles (payant un euro à la fin de chaque mois) issus des données brutes et des modèles LC et SPO à horizon différent pour les âges 30ans et 50 ans.

Les figures (4.1) et (4.2) montrent l'évolution des prix chez les hommes et les femmes en fonction des horizons pour les 4 classes d'âges (30-34, 40-44, 50-54, 60-64) issues des données brutes et des modèles LC et SPO. On observe que de manière générale, les modèles reproduisent très bien les prix obtenus avec les données brutes. Comme attendu, plus l'horizon est grand, et plus les prix augmentent, puis se stabilisent avec la diminution de la probabilité de survie. Aussi, le prix de la rente décroît avec l'âge car plus on vieillit, moins il y a de paiement. On observe également que les courbes des femmes sont au-dessus de celles des hommes, ce qui traduit le fait que l'espérance de vie des femmes est plus élevée que celle des hommes et donc elles perçoivent plus de rente. Enfin, on remarque que les courbes issues de LC sont légèrement au-dessus de celles obtenues avec les données brutes, ce qui signifie que LC a tendance à surestimer la valeur de la rente, probablement en raison d'une sous-estimation de la mortalité, contrairement à Seklecka qui est légèrement en dessous mais plus proche que LC. Afin de mieux analyser ces différences, une étude des erreurs relatives est présentée dans la sous-section suivante.

4.4 Calcul Numérique de la rente

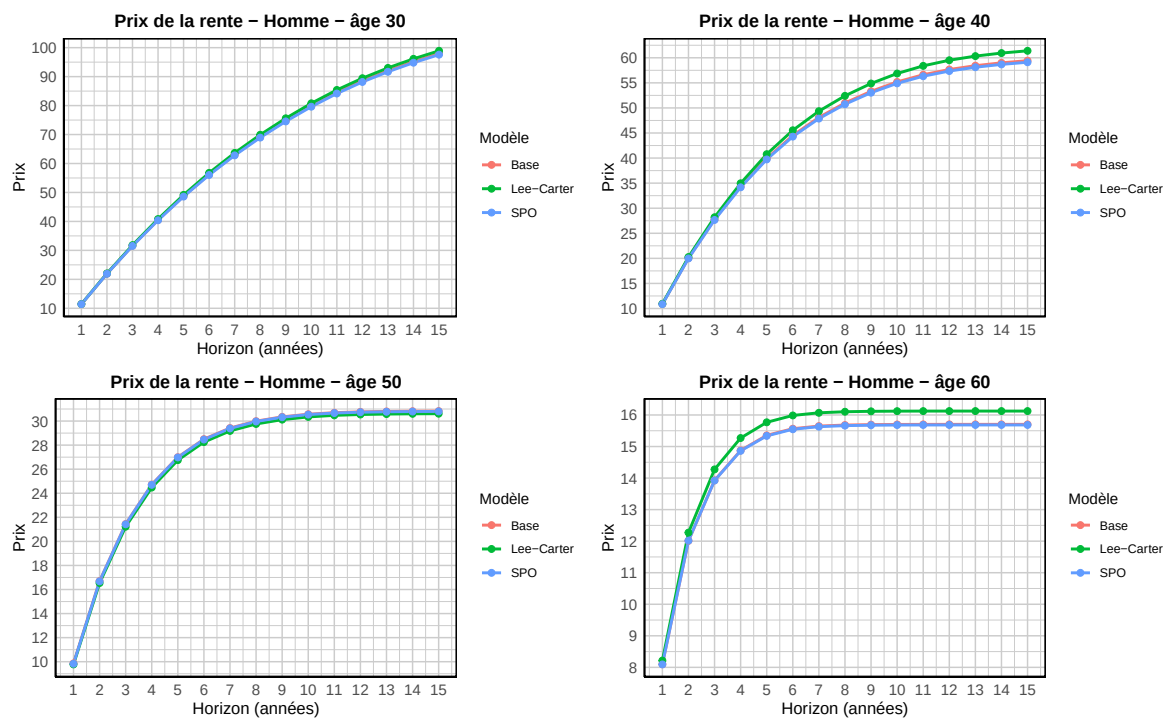


FIGURE 4.1 – Évolution des prix de rentes chez les hommes sur la période 2000-2016 en fonction des horizons

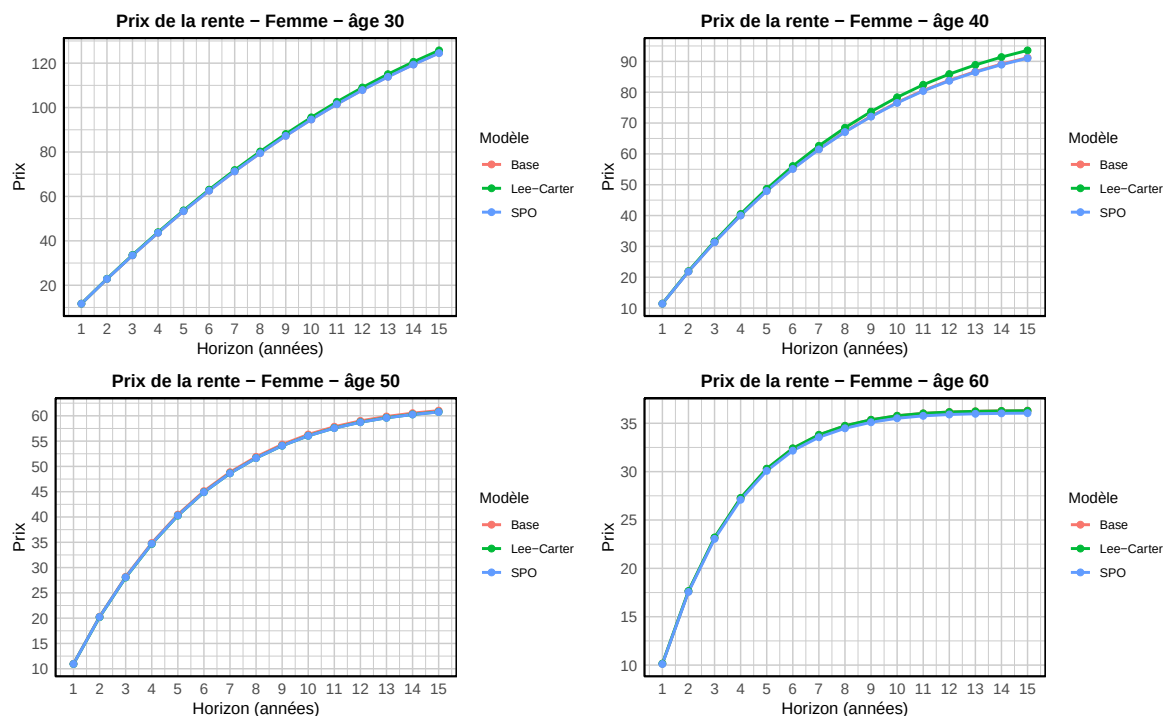


FIGURE 4.2 – Évolution des prix de rentes chez les femmes sur la période 2000-2016 en fonction des horizons.

4.4.2 Analyse de l'erreur relative

Afin d'évaluer l'impact des modèles de mortalité sur la valorisation, nous avons calculé l'erreur relative définie par :

$$\text{Erreur}^{(mod)} = \frac{PV^{(mod)} - PV^{raw}}{PV^{raw}}, \quad mod \in LC, SPO.$$

Cette mesure nous permet de savoir si un modèle tend à surestimer ou sous-estimer le prix de la rente par rapport à la référence empirique. Si l'erreur est positive, alors le modèle surestime le prix, tandis qu'une erreur négative indique une sous-évaluation.

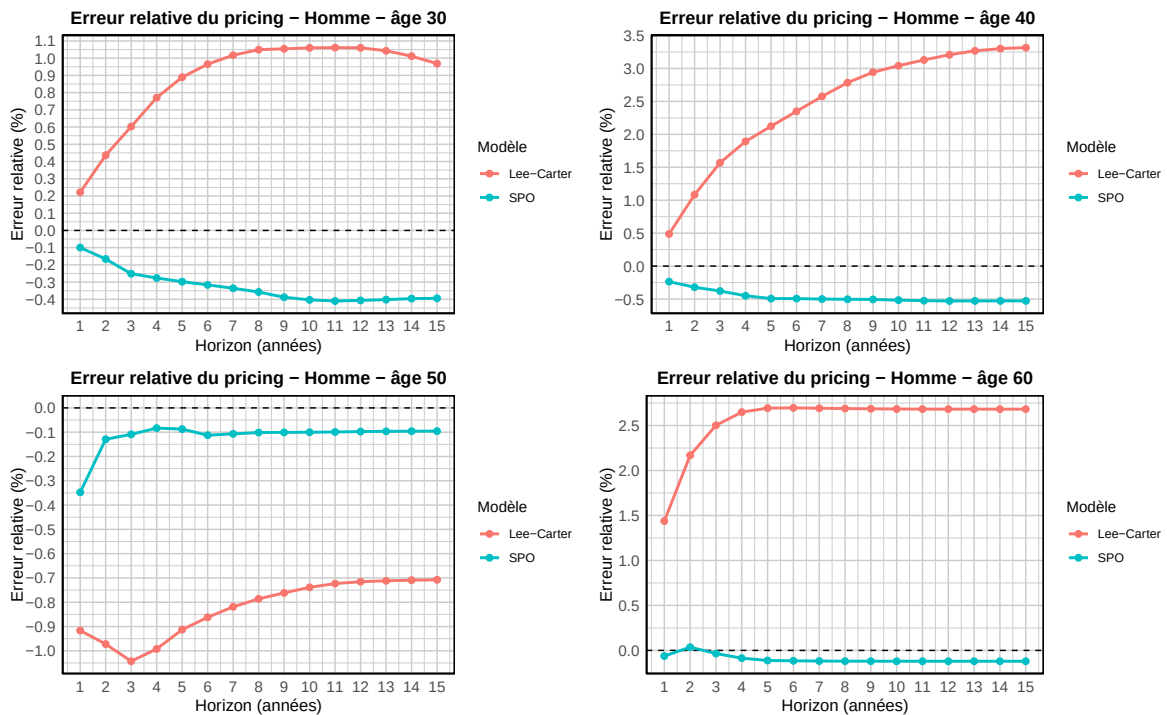


FIGURE 4.3 – Erreur relative des valeurs actuarielles des rentes en fonction des horizons pour différentes classes d'âges issues des modèles LC et SPO sur la période 2000-2016 chez les hommes.

Les figures (4.3) et (4.4) présentent les erreurs relatives des prix de rente obtenus à partir des modèles LC et SPO par rapport aux données brutes. On voit sur les graphiques que le modèle LC présente un biais significatif, généralement positif, ce qui est signe d'une surestimation des prix, particulièrement marquée pour les horizons longs. De plus, ce biais varie beaucoup avec l'âge ce qui veut dire que le modèle n'est pas forcément stable. Le modèle SPO quant à lui se distingue par une grande stabilité des erreurs, qui restent proches de zéro pour la majorité des cas. Cela indique une meilleure capacité à reproduire les valeurs actuarielles observées. Toutefois, une légère tendance à la sous-estimation est observée, notamment pour les âges élevés. Cependant, on observe chez les femmes à 60 ans que le modèle LC est meilleur que le modèle SPO.

4.5 Conclusion

De manière générale, le modèle SPO qui prend en compte le facteur température en plus donne des erreurs beaucoup plus faibles que le modèle LC.

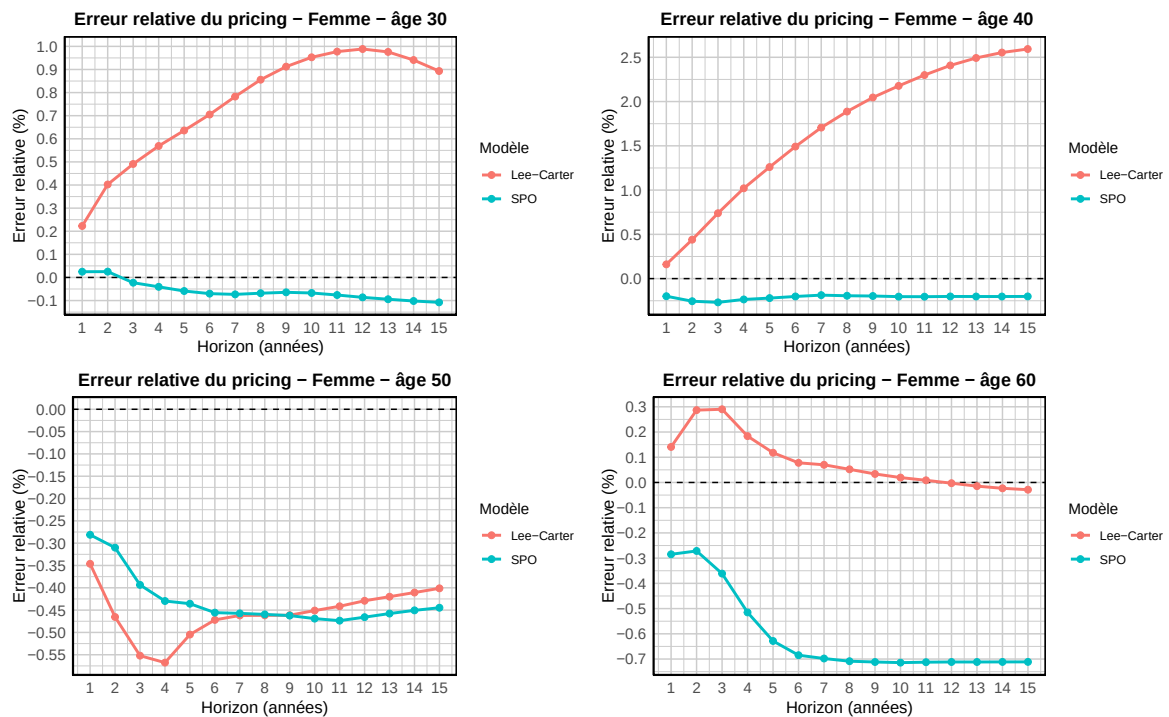


FIGURE 4.4 – Erreur relative des valeurs actuarielles des rentes en fonction des horizons pour différentes classes d'âges issues des modèle LC et SPO sur la période 2000-2016 chez les femmes.

4.5 Conclusion

Cette étude a consisté à mesurer l'impact de la prise en compte d'un facteur lié à la température dans les modèles de mortalité sur la tarification d'une rente temporaire mensuelle. Cette rente verse une unité monétaire à la fin de chaque mois tant que l'assuré est en vie, avec un taux d'intérêt annuel fixé à 2%, pour différents horizons et différentes classes d'âges. Nous avons tout d'abord déterminé les probabilités de survie et par la suite calculé les prix des rentes à l'aide du modèle LC et du modèle SPO. Il en ressort que le modèle SPO donne des prix beaucoup plus proches des prix obtenus avec les données brutes que le modèle LC qui surestime beaucoup les prix. Afin de s'en assurer, nous avons calculé les erreurs relatives liées à chaque modèle et le modèle SPO donne des erreurs plus faibles que le modèle LC.

Conclusion générale

Dans ce mémoire, nous avons étudié l'impact de la température sur la mortalité. Bien que cette problématique ait déjà été abordée dans la littérature, il subsiste encore un manque de modèles de mortalité intégrant de manière explicite cette relation. Seklecka et al. ont été parmi les premiers à proposer une approche satisfaisante, en mettant en évidence un effet significatif de la température sur la mortalité, particulièrement en période de froid.

Notre étude s'est appuyée sur des données hebdomadaires de la population française sur la période 2000-2019, en distinguant trois sous-périodes : les périodes froides (d'octobre à avril), les périodes chaudes (de mai à septembre) et les périodes annuelles (de janvier à décembre). Les données de température, initialement disponibles à l'échelle journalière et par station, ont été agrégées à l'échelle hebdomadaire puis nationale à l'aide d'une moyenne.

Dans un premier temps, nous avons présenté les principaux modèles de mortalité existants dans la littérature, notamment le modèle de Lee-Carter et ses extensions, le modèle Cairns-Blake_dowd, ainsi que le modèle de Seklecka et al., lui-même extension du modèle de O'Hare et Li intégrant un facteur lié à la température. Une analyse de la corrélation entre les taux de mortalité et la température a ensuite été réalisée. Celle-ci met en évidence une relation négative entre la mortalité et la température en période froide, tandis qu'une relation positive apparaît lors des périodes chaudes, notamment en cas d'augmentation brutale des températures, comme lors de la canicule de 2003. Par ailleurs, cette corrélation s'avère robuste aux variations de la période d'étude ainsi qu'à de légères perturbations des données de température.

Les paramètres des modèles de Lee-Carter et de Seklecka ont ensuite été estimés par la méthode du maximum de vraisemblance sur la période 2000-2016. Les critères de comparaison (RMSE, MAPE, MAD) montrent que le modèle de Seklecka offre un meilleur ajustement que le modèle de Lee-Carter, malgré une complexité plus élevée et un nombre de paramètres plus important. Ce résultat souligne l'intérêt d'intégrer un facteur exogène tel que la température dans la modélisation de la mortalité.

Les projections réalisées sur la période 2017-2019 confirment ces résultats, le modèle de Seklecka présente des erreurs plus faibles. Toutefois, le critère BIC pénalise davantage ce modèle en raison de sa complexité, en particulier dans le cadre des données hebdomadaires.

Afin d'évaluer l'impact de ces modèles en pratique, nous avons procédé à la tarification

d'un produit de rente versant une unité monétaire mensuelle sur différents horizons et pour différents âges. Les prix obtenus à partir des modèles ont été comparés à ceux calculés à partir des données observées. Il en ressort que le modèle de Lee-Carter tend à surestimer les prix, tandis que le modèle de Seklecka fournit des estimations plus proches des données réelles, bien qu'il présente une légère sous-estimation.

Ces résultats mettent en évidence l'importance croissante d'intégrer les facteurs climatiques dans les modèles actuariels, dans un contexte marqué par le changement climatique et l'augmentation des événements extrêmes. Cependant, malgré ses bonnes performances, le modèle Seklecka présente certaines limites. En particulier, il repose sur un paramètre (ct_x) dépendant lui-même des taux de mortalité, ce qui peut poser des problèmes d'interprétation. De plus, la prise en compte d'un unique facteur lié à la température moyenne peut s'avérer insuffisante. En effet, la littérature met en évidence une relation non linéaire de type U entre température et mortalité. Dans ce contexte, une perspective intéressante consisterait à distinguer explicitement les effets des températures froides et chaudes, et à introduire un facteur dépendant du temps permettant de capter les fluctuations futures du climat. Une telle approche permettrait d'améliorer la capacité du modèle à représenter de manière réaliste l'impact du changement climatique sur la mortalité.

Bibliographie

- [1] Cairns, A. J. G., D. Blake, and K. Dowd, 2008, Modelling and Management of Mortality Risk : A Review, *Scandinavian Actuarial Journal*, 2-3 : 79–113.
- [2] Cairns, A.J.G., Blake, D.P., Kessler, A.R. & Kessler, M. (2020). The Impact of Covid-19 on Future Higher-Age Mortality.
- [3] Cairns, A. J. G., Blake, D., & Dowd, K. (2006). A two-factor model for stochastic mortality with parameter uncertainty : Theory and calibration. *Journal of Risk and Insurance*, 73, 687–718.
- [4] Dmitri Jdanov, Vladimir M. Shkolnikov, Ainhua Alustiza Galarza (2020) : Short-Term Mortality Fluctuations Data series (STMF).
- [5] Feliciangeli, M., Kerkhof, J., & Van Duffel, S. (2025). Socio-economic mortality curves : the Belgian case. *EUROPEAN ACTUARIAL JOURNAL*, 15(3), 773-806. <http://hdl.handle.net/1854/LU-01K0C2TWD98YG36280HZQXWHBV>
- [6] Gavakshi Gungah & Jason Narsoo (2022) A novel EVT-modified Lee-Carter model for mortality forecasting : An application to extreme mortality events, *Journal of Statistics and Management Systems*, 25 :1, 211-243.
- [7] Gasparrini, A, Armstrong, B, and Kenward, M. G. (2010). Distributed lag non-linear models. *Statistics in Medicine* 29.21, pp. 2224–2234.
- [8] Jens Robben, Katrien Antonio & Torsten Kleinow (2024). The short-term association between environmental variables and mortality : evidence from Europe, 1-52.
- [9] Lee, R. D., & Carter, L. (1992). Modeling and Forecasting U.S. mortality. *Journal of the American Statistical Association*, 87, 659–671.
- [10] Li, N., & Lee, R. (2005). COHERENT MORTALITY FORECASTS FOR A GROUP OF POPULATIONS : AN EXTENSION OF THE LEE-CARTER METHOD*. *Demography* (Pre-2011), 42(3), 575-94.
- [11] O'Hare, C., & Li, Y. (2012). Explaining young mortality. *Insurance : Mathematics and Economics*, 50, 12–25.
- [12] Plat, R. (2009). On stochastic mortality modelling. *Insurance : Mathematics and Economics*, 45, 393–404.

BIBLIOGRAPHIE

- [13] Quentin Guibert, Gaëlle Pincemin & Frédéric Planchet (2024). Impacts of Climate Change on Mortality : An extrapolation of temperature effects based on time series data in France, 1-54.
- [14] Renshaw, A. E., & Haberman, S. (2006). A cohort-based extension to the Lee–Carter model for mortality reduction factors. *Insurance : Mathematics and Economics*, 38, 556–570.
- [15] Seklecka, M., Pantelous, A. A., & O’Hare, C. (2017). Mortality effects of temperature changes in the United Kingdom. *Journal of Forecasting*, 36(7), 824-841.
- [16] Steven Haberman, Maria Russolillo, November (2005). Lee-Carter Mortality Forecasting : Application to the Italian Population.

Annexe

Sex	Period	MAD	RMSE	MAPE	AIC	BIC	N
LC							
Femmes	Annuel	0.0015357	0.0048244	18.98100	89493.47	94035.32	614
Femmes	froid	0.0016921	0.0051515	18.62294	52733.56	55297.58	374
Femmes	chaud	0.0012402	0.0040063	18.91379	36325.47	38152.51	280
Hommes	Annuel	0.0031781	0.0104474	15.86066	96701.20	101243.06	614
Hommes	froid	0.0032990	0.0108566	15.40117	55991.00	58555.02	374
Hommes	chaud	0.0024305	0.0083667	14.60272	38744.44	40571.49	280
SPO							
Femmes	Annuel	0.0012317	0.0036002	19.09980	89655.13	99197.46	1290
Femmes	froid	0.0013018	0.0037000	18.61148	52524.02	58077.11	810
FemmesFemmes	chaud	0.0010706	0.0034900	17.61862	36414.25	40472.91	622
Hommes	Annuel	0.0023107	0.0072177	15.50666	94778.74	104321.06	1290
Hommes	froid	0.0030034	0.0097122	13.97720	55316.94	60870.02	810
Hommes	chaud	0.0021590	0.0072641	13.45765	38486.57	42545.22	622

TABLE 4.3 – Qualité d'ajustement des modèles LC et SPO sur la période 2000-2010

TABLE 4.2 – Corrélations mortalité-température avec p-values significatives : *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

Age	Period	Pearson_Male	p_Pearson_Male	Spearman_Male	p_Spearman_Male	Pearson_Female	p_Pearson_Female	Spearman_Female	p_Spearman_Female
0	Annuel	0.01968	0.5255	0.02529	0.4146	-0.01614	0.6025	-0.03374	0.2762
0	chaud	0.03222	0.5022	0.02607	0.5872	0.01184	0.8052	-0.01741	0.7170
0	froid	0.00315	0.9383	0.00691	0.8652	-0.06478	0.1108	-0.06280	0.1222
1	Annuel	0.03237	0.2963	0.01530	0.6216	-0.08928	0.0039**	-0.08479	0.0062**
1	chaud	0.17969	0.0002***	0.15910	0.0009***	0.03245	0.4991	0.03177	0.5082
1	froid	0.00381	0.9254	0.01191	0.7696	-0.14544	0.0003***	-0.15416	0.0001***
5	Annuel	0.01612	0.6030	0.00010	0.9974	-0.06681	0.0310*	-0.05427	0.0798
5	chaud	0.12257	0.0104*	0.10561	0.0274*	0.09879	0.0392*	0.09814	0.0405*
5	froid	-0.03484	0.3915	-0.04222	0.2991	-0.06720	0.0981	-0.06154	0.1299
10	Annuel	0.02353	0.4477	0.01261	0.6842	0.00313	0.9197	-0.01673	0.5894
10	chaud	0.12690	0.0080**	0.10291	0.0317*	0.09582	0.0455*	0.05621	0.2415
10	froid	0.00293	0.9425	-0.01478	0.7164	-0.01149	0.7776	-0.03457	0.3952
15	Annuel	0.16876	0.0000***	0.16640	0.0000***	-0.00409	0.8950	-0.01827	0.5555
15	chaud	0.15731	0.0010***	0.14675	0.0021**	0.07933	0.0981	0.08248	0.0854
15	froid	0.05433	0.1813	0.04344	0.2853	-0.08742	0.0313*	-0.10727	0.0082**
20	Annuel	0.18461	0.0000***	0.17211	0.0000***	-0.03110	0.3157	-0.02862	0.3559
20	chaud	0.12990	0.0066**	0.09414	0.0495*	0.04623	0.3355	0.02412	0.6154
20	froid	0.06474	0.1110	0.07198	0.0764	-0.05692	0.1613	-0.06069	0.1353
25	Annuel	0.15176	0.0000***	0.15355	0.0000***	-0.06061	0.0504	-0.05863	0.0584
25	chaud	0.15901	0.0009***	0.14392	0.0026**	0.12110	0.0114*	0.11203	0.0193*
25	froid	0.07628	0.0603	0.07690	0.0583	-0.11420	0.0048**	-0.11510	0.0045**
30	Annuel	0.07136	0.0212*	0.06597	0.0332*	-0.07540	0.0149*	-0.07539	0.0149*
30	chaud	0.13783	0.0039**	0.13438	0.0049**	0.02334	0.6269	0.00714	0.8819
30	froid	-0.01597	0.6946	-0.02806	0.4902	-0.06811	0.0937	-0.05654	0.1642
35	Annuel	0.04174	0.1779	0.02509	0.4182	-0.11094	0.0003***	-0.10713	0.0005***
35	chaud	0.07050	0.1416	0.02520	0.5998	0.01578	0.7425	0.00964	0.8409
35	froid	-0.03968	0.3291	-0.04713	0.2463	-0.15658	0.0001***	-0.16583	0.0000***
40	Annuel	-0.03618	0.2430	-0.05053	0.1029	-0.06212	0.0449*	-0.06617	0.0326*
40	chaud	0.06682	0.1637	0.02362	0.6228	-0.00450	0.9253	-0.02408	0.6161
40	froid	-0.12162	0.0027**	-0.12654	0.0018**	-0.10996	0.0067**	-0.12349	0.0023**
45	Annuel	-0.08179	0.0082**	-0.09130	0.0032**	-0.14288	0.0000***	-0.13273	0.0000***
45	chaud	0.06470	0.1775	0.02238	0.6413	0.03512	0.4645	0.00661	0.8906
45	froid	-0.14791	0.0003***	-0.13622	0.0008***	-0.13861	0.0006***	-0.12533	0.0020**
50	Annuel	-0.13932	0.0000***	-0.13111	0.0000***	-0.20326	0.0000***	-0.21336	0.0000***
50	chaud	0.07423	0.1217	0.04704	0.3271	0.02692	0.5750	-0.00634	0.8950
50	froid	-0.20585	0.0000***	-0.19522	0.0000***	-0.22821	0.0000***	-0.23867	0.0000***
55	Annuel	-0.24518	0.0000***	-0.24623	0.0000***	-0.32677	0.0000***	-0.35172	0.0000***
55	chaud	0.05917	0.2176	0.01977	0.6806	0.04715	0.3260	0.02679	0.5769
55	froid	-0.29888	0.0000***	-0.30336	0.0000***	-0.34992	0.0000***	-0.37314	0.0000***
60	Annuel	-0.32665	0.0000***	-0.34928	0.0000***	-0.35626	0.0000***	-0.38388	0.0000***
60	chaud	0.05799	0.2269	-0.00586	0.9029	0.14909	0.0018**	0.10390	0.0301*
60	froid	-0.32241	0.0000***	-0.34329	0.0000***	-0.31919	0.0000***	-0.31452	0.0000***
65	Annuel	-0.28620	0.0000***	-0.31626	0.0000***	-0.34970	0.0000***	-0.37354	0.0000***
65	chaud	0.04172	0.3848	0.00864	0.8573	0.08641	0.0715	-0.00201	0.9667
65	froid	-0.26307	0.0000***	-0.29587	0.0000***	-0.29380	0.0000***	-0.29166	0.0000***
70	Annuel	-0.28647	0.0000***	-0.30009	0.0000***	-0.35980	0.0000***	-0.39207	0.0000***
70	chaud	0.01873	0.6965	-0.04327	0.3674	0.06982	0.1455	-0.01600	0.7391
70	froid	-0.25146	0.0000***	-0.25293	0.0000***	-0.33174	0.0000***	-0.34812	0.0000***
75	Annuel	-0.36546	0.0000***	-0.36698	0.0000***	-0.38291	0.0000***	-0.41087	0.0000***
75	chaud	0.00394	0.9346	-0.06316	0.1881	0.06203	0.1961	-0.04397	0.3598
75	froid	-0.32707	0.0000***	-0.31753	0.0000***	-0.34690	0.0000***	-0.34955	0.0000***
80	Annuel	-0.46664	0.0000***	-0.47032	0.0000***	-0.47825	0.0000***	-0.51516	0.0000***
80	chaud	0.01532	0.7497	-0.05850	0.2228	0.09816	0.0405*	-0.02885	0.5480
80	froid	-0.40532	0.0000***	-0.39046	0.0000***	-0.45382	0.0000***	-0.44567	0.0000***
85	Annuel	-0.56259	0.0000***	-0.61221	0.0000***	-0.49435	0.0000***	-0.58378	0.0000***
85	chaud	0.01962	0.6829	-0.08011	0.0948	0.12016	0.0120*	-0.04324	0.3677
85	froid	-0.46194	0.0000***	-0.49027	0.0000***	-0.43009	0.0000***	-0.46569	0.0000***
90	Annuel	-0.64388	0.0000***	-0.69489	0.0000***	-0.57639	0.0000***	-0.66142	0.0000***
90	chaud	0.03641	0.4482	-0.09843	0.0399*	0.11253	0.0187*	-0.06487	0.1764
90	froid	-0.52230	0.0000***	-0.54567	0.0000***	-0.49288	0.0000***	-0.51388	0.0000***
95	Annuel	-0.68868	0.0000***	-0.74513	0.0000***	-0.66281	0.0000***	-0.75956	0.0000***
95	chaud	0.01116	0.8163	-0.09823	0.0403*	0.10947	0.0222*	-0.06640	0.1664
95	froid	-0.57549	0.0000***	-0.59224	0.0000***	-0.55918	0.0000***	-0.59437	0.0000***

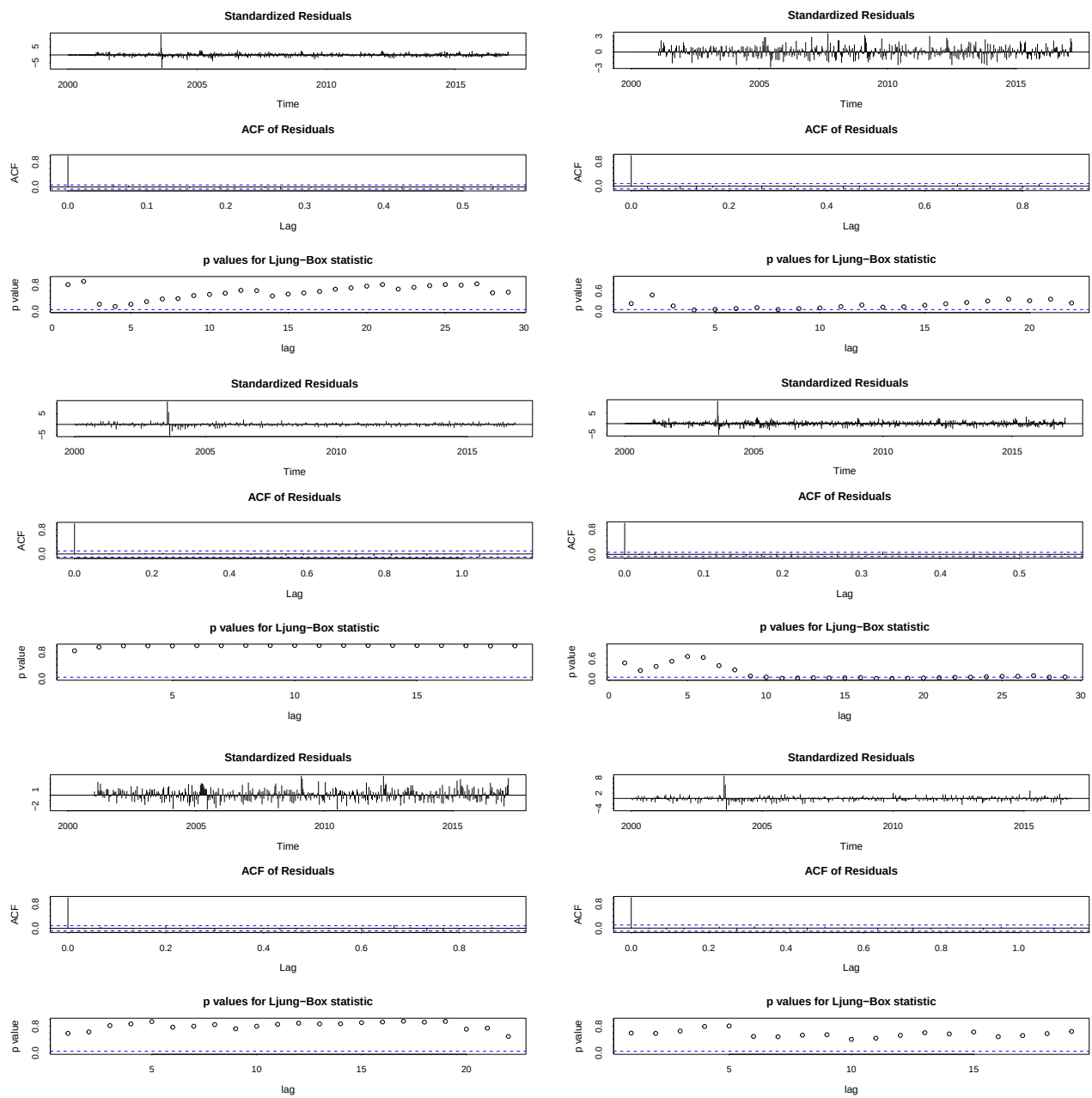


FIGURE 4.5 – Résidus standardisés chez les hommes et femmes pour les cas annuel, chaud et froid.

id	param	model	ncoef	AIC	lb_pvalue	all_coef_sig
f_annuel	k1	(1,1,1) x (0,1,1)[52]	3	-2815.3280	0.1421441	TRUE
f_annuel	k2	(0,1,1) x (0,0,0)[1]	1	-205.3675	0.6605492	TRUE
f_annuel	k3	(0,1,0) x (0,0,0)[1]	0	-164.0358	0.5257602	TRUE
f_annuel	k4	(1,1,2) x (0,1,1)[52]	4	-14045.0234	0.6059925	TRUE
f_chaud	k1	(0,1,2) x (0,0,0)[22]	2	-1061.2132	0.9995898	TRUE
f_chaud	k2	(0,1,1) x (0,0,0)[1]	1	-169.6573	0.8464881	FALSE
f_chaud	k3	(0,1,1) x (0,0,0)[1]	1	-140.2971	0.6212561	TRUE
f_chaud	k4	(0,1,1) x (2,0,0)[22]	3	-6169.1374	0.3730708	FALSE
f_froid	k1	(2,1,1) x (0,1,1)[30]	4	-1771.7367	0.5357764	TRUE
f_froid	k2	(0,1,1) x (0,0,0)[1]	1	-205.2094	0.3209183	TRUE
f_froid	k3	(1,1,0) x (0,0,0)[1]	1	-147.6369	0.5187052	TRUE
f_froid	k4	(2,1,1) x (1,0,2)[30]	6	-8422.0291	0.2464396	FALSE
m_annuel	k1	(0,1,2) x (1,0,1)[52]	4	-3353.4117	0.3301681	TRUE
m_annuel	k2	(0,1,0) x (0,0,0)[1]	0	-228.1081	0.6434690	TRUE
m_annuel	k3	(0,1,0) x (0,0,0)[1]	0	-160.9081	0.4996091	TRUE
m_annuel	k4	(1,1,2) x (0,1,1)[52]	4	-14112.6160	0.5781532	TRUE
m_chaud	k1	(0,1,2) x (0,0,0)[22]	2	-1262.0612	0.5847170	TRUE
m_chaud	k2	(0,1,0) x (0,0,0)[1]	0	-193.0000	0.1730668	TRUE
m_chaud	k3	(0,1,1) x (0,0,0)[1]	1	-137.1196	0.8314234	TRUE
m_chaud	k4	(2,1,2) x (0,0,0)[22]	4	-6349.1180	0.3293865	FALSE
m_froid	k1	(1,1,1) x (0,1,1)[30]	3	-1912.1079	0.3700580	TRUE
m_froid	k2	(0,1,0) x (0,0,0)[1]	0	-188.9471	0.1753406	TRUE
m_froid	k3	(0,1,2) x (0,0,0)[1]	2	-127.2906	0.7488201	FALSE
m_froid	k4	(0,1,1) x (0,0,0)[30]	1	-8814.0051	0.3298952	TRUE

TABLE 4.4 – Tableau récapitulatif des meilleurs modèles

Code R complet

```
1 #####
2 #   Packages utiles           #
3 #####
4
5 library(StMoMo)
6 library(dplyr)
7 library(tidyr)
8 library(forecast)
9 library(tseries)
10 library(knitr)
11 library(ggplot2)
12 library(purrr)
13 library(tibble)
14 library(stringr)
15 library(gridExtra)
16 library(stats)
17 library(ISOweek)
18 library(lubridate)
19
20 #####
21 #   Traitement de la base de données   #
22 #####
23
24 setwd("C:/Users/Zeumo Stanis Lassi/Desktop/Mon mémoire/Données")
25
26 ## Donnée sur la mortalité
27
28 exposure = read.table("Exposures_5x1.txt",
29                       skip=grep("Year", readLines("Exposures_5x1.txt"))[1]-1,
30                       header=TRUE, sep=" ", stringsAsFactors = TRUE)
31 Exposur <- exposure %>%
32   filter(Year < 2020 & Year > 1999) %>%
33   mutate(Age= as.numeric(sub("-", "",gsub("\\+", "", Age)))) %>%
34   mutate(Age=ifelse(Age>=95, 95, Age)) %>%
35   group_by(Year, Age) %>%
36   summarise(across(c(Female, Male), sum),
37             .groups = "drop") %>%
38   pivot_longer(cols = c(Female, Male),
39               names_to = "Sex", values_to = "Exposure") %>%
40   mutate(Sex= recode(Sex, "Female" = "f", "Male" = "m" )) %>%
41   arrange(Year, Sex)
42
43 base <- read.csv("décès.csv")
44
45 # jointure entre exposition et décès
```

```

46 base_joint <- base %>%
47   inner_join(Exposur,by=c("Year", "Sex", "Age"))
48
49 #création de l'exposition par semaine
50 base_joint <- base_joint %>%
51   mutate(Exposure_week = Exposure/52)
52
53 # Création des taux de mortalité hebdomadaire
54
55 mortality <- base_joint %>%
56   mutate(Deaths_rate = Deaths/Exposure_week)
57
58 ## Donnée sur la température
59 temp <- read.csv("4233121.csv", header = T)
60
61 f_to_c <- function(F) (as.numeric(F)-32)*5/9
62 temp <- temp %>%
63   mutate(
64     DATE= as.Date(DATE, "%d/%m/%Y"),
65     TMIN = f_to_c(TMIN),
66     TMAX = f_to_c(TMAX),
67     TAVG = f_to_c(TAVG),
68     Year = isoyear(DATE),
69     Week = isoweek(DATE)
70   ) %>%
71   group_by(STATION, Year, Week) %>%
72   summarise(
73     TAVG = mean(TAVG, na.rm = TRUE),
74     TMIN = mean(TMIN, na.rm = TRUE),
75     TMAX = mean(TMAX, na.rm = TRUE),
76     .groups = "drop"
77   )
78
79 température <- temp %>%
80   filter(Year>1999 & Year < 2020)%>%
81   group_by( Year, Week) %>%
82   summarise(
83     TAVG = mean(TAVG, na.rm = TRUE),
84     TMIN = mean(TMIN, na.rm = TRUE),
85     TMAX = mean(TMAX, na.rm = TRUE),
86     .groups = "drop"
87   ) %>%
88   arrange(Year, Week)
89
90
91 iso_monday <- function(year, week) {
92   ISOweek2date(paste0(year, "-W", sprintf("%02d", week), "-1"))
93 }
94
95 température <- température %>%
96   mutate(
97     date = iso_monday(Year, Week),
98     month = month(date),
99     saison = ifelse(month %in% 5:9, "chaud", "froid")
100   ) %>%
101   select(-date, -month) # on garde uniquement ce qui sert ensuite
102
103 température <- température %>%
104   mutate(
105     date = ISOweek2date(paste0(Year, "-W", sprintf("%02d", Week), "-1")),
106   )

```

```
107 ### Jointure de la base de données de températures et de mortalité
108
109 data <- mortality %>%
110   left_join(température %>% select(Year, Week, saison, TAVG),
111     by = c("Year", "Week"))
112
113 write.csv(data, "data.csv")
114
115 #####
116 #       Analyse exploratoire       #
117 #####
118
119 dataset <- read.csv("data.csv")
120 dataset <- dataset %>%
121   mutate(
122     Year_week = paste0(Year, "-", sprintf("%02d", Week)),
123     date = ISOweek2date(paste0(Year, "-W", sprintf("%02d", Week), "-1"))
124   )
125
126 ## Evolution de la mortalité brute
127 Age_class <- 20
128 mort_week_age <- dataset %>%
129   filter(Age == Age_class) %>%
130   group_by(date, Sex) %>%
131   summarise(
132     Deaths = sum(Deaths, na.rm = TRUE),
133     Exposure = sum(Exposure_week, na.rm = TRUE),
134     mx = Deaths / Exposure,
135     .groups = "drop"
136   ) %>%
137   mutate(
138     Sex = recode(Sex, "m" = "Homme", "f" = "Femme")
139   )
140
141 ggplot(mort_week_age, aes(x = date, y = mx)) +
142   facet_wrap(~Sex, ncol = 2, scales = "free_y") +
143   geom_line(color = "steelblue", linewidth = 0.8) +
144   labs(
145     title = " 20-24 ans", x = "Date", y = "Taux de mortalité") +
146   theme_minimal() +
147   theme(panel.grid = element_line(color = "grey80"))
148
149 #### Température
150 temp_week <- dataset %>%
151   group_by(date, saison) %>%
152   summarise(
153     TAVG = mean(TAVG, na.rm = TRUE),
154     .groups = "drop"
155   )
156
157 ggplot(température, aes(x = date, y = TAVG, color = saison, group = 1)) +
158   geom_line(linewidth = 0.9) +
159   scale_color_manual(
160     values = c("froid" = "blue", "chaud" = "red")) +
161   labs(title = NULL, x = "Date", y = "TAVG(°C)", color = "Saison") +
162   theme_minimal() +
163   theme(panel.grid = element_line(color = "grey80"))
164
165
166 #####
167 ##       Calcul de la corrélation       ##
```



```
168 #####
169
170
171 ##### Donnée d'entrainement du modèle
172
173 dataset_train <- dataset %>%
174   filter(Year<=2016)
175
176 # 1) On impose le même ordre d'âges que df_rates (22 classes ordonnées)
177 periods <- c("Annuel", "froid", "chaud")
178
179 grid <- tidyr::crossing(
180   Age      = sort(unique(dataset_train$Age)),
181   Period = periods
182 )
183
184 fmt_p_stars <- function(p){
185
186   stars <- case_when(
187     is.na(p) ~ "",
188     p < 0.001 ~ "$~{***}$",
189     p < 0.01  ~ "$~{**}$",
190     p < 0.05  ~ "$~{*}$",
191     TRUE ~ ""
192   )
193
194   paste0(sprintf("%.4f", p), stars)
195
196 }
197
198 ## Calcul du coefficient de corrélation de Pearson et de Spearman
199
200 calc_corr <- function(df, sex, age, period) {
201
202   dd <- df %>%
203     filter(Sex == sex, Age == age) %>%
204     filter(!is.na(Deaths_rate), !is.na(TAVG))
205
206   if (period != "Annuel") {
207     dd <- dd %>% filter(saison == period)
208   }
209
210   # sécurité : il faut assez de points
211   if (nrow(dd) < 3) {
212     return(tibble(
213       Pearson = NA_real_, p_Pearson = NA_real_,
214       Spearman = NA_real_, p_Spearman = NA_real_
215     ))
216   }
217
218   # Tests
219   tp <- cor.test(dd$Deaths_rate, dd$TAVG, method = "pearson")
220   ts <- cor.test(dd$Deaths_rate, dd$TAVG, method = "spearman", exact = FALSE)
221
222   tibble(
223     Pearson      = round(unname(tp$estimate), 5),
224     p_Pearson    = round(tp$p.value, 5),
225
226     Spearman     = round(unname(ts$estimate), 5),
227     p_Spearman   = round(ts$p.value, 5))
228 }
```

```

229
230 results_season <- purrr::pmap_dfr(grid, function(Age, Period) {
231
232   cm <- calc_corr(dataset_train, "m", Age, Period)
233   cf <- calc_corr(dataset_train, "f", Age, Period)
234
235   tibble(
236     Age = Age,
237     Period = Period,
238
239     Pearson_Male = cm$Pearson,
240     p_Pearson_Male = cm$p_Pearson,
241
242     Spearman_Male = cm$Spearman,
243     p_Spearman_Male = cm$p_Spearman,
244
245     Pearson_Female = cf$Pearson,
246     p_Pearson_Female = cf$p_Pearson,
247
248     Spearman_Female = cf$Spearman,
249     p_Spearman_Female = cf$p_Spearman
250   )
251 })
252
253 ### exportation en tableau latex
254 kable(
255   results_season %>%
256     mutate(
257       p_Pearson_Male = fmt_p_stars(p_Pearson_Male),
258       p_Spearman_Male = fmt_p_stars(p_Spearman_Male),
259       p_Pearson_Female = fmt_p_stars(p_Pearson_Female),
260       p_Spearman_Female = fmt_p_stars(p_Spearman_Female)
261     ),
262   format = "latex", booktabs = TRUE, escape = FALSE, align = "c",
263   caption = "Corrélations mortalité-température avec p-values significatives"
264 )
265
266
267 # =====
268 # 2) Pearson corr -> c_x par âge / sexe / période (train uniquement !)
269 # =====
270
271 calc_cx_pearson <- function(df_train, temperature) {
272
273   periods <- c("Annuel", "froid", "chaud")
274
275   grid <- tidyr::crossing(
276     Sex = c("m", "f"),
277     Age_i = sort(unique(df_train$Age)),
278     Period = periods
279   )
280
281   calc_one <- function(Sex_i, Age_i, Period_i) {
282
283     dd <- df_train %>%
284       filter(Sex == Sex_i,
285             Age == Age_i,
286             is.finite(Deaths_rate),
287             is.finite(TAVG))
288
289     if (Period_i != "Annuel")

```

```

290     dd <- dd %>% filter(season == Period_i)
291
292     if (nrow(dd) < 3)
293       return(tibble(ct_x = 0))
294
295     tibble(ct_x = cor(dd$Deaths_rate, dd[[temperature]], method = "pearson"))
296   }
297
298   purrr::pmap_dfr(grid, function(Sex, Age_i, Period) {
299
300     out <- calc_one(Sex, Age_i, Period)
301
302     tibble(Sex = Sex,
303            Age = Age_i,
304            Period = Period,
305            ct_x = out$ct_x)
306   })
307 }
308
309
310 # Pearson c_x calculé sur TRAIN uniquement
311 cx_tbl <- calc_cx_pearson(dataset_train, "TAVG")
312
313 #####
314 #           Robustesse           #
315 #####
316
317 data_2008 <- dataset %>%
318   filter(Year <= 2008)
319 cx_tbl_2008 <- calc_cx_pearson(data_2008, "TAVG" )
320 cx_tbl_2008 <- cx_tbl_2008 %>%
321   mutate(window = "2000-2008")
322
323 data_2010 <- dataset %>%
324   filter(Year <= 2010)
325 cx_tbl_2010 <- calc_cx_pearson(data_2010, "TAVG")
326 cx_tbl_2010 <- cx_tbl_2010 %>%
327   mutate(window = "2000-2010")
328
329 data_2016 <- dataset %>%
330   filter(Year <= 2016)
331 cx_tbl_2016 <- calc_cx_pearson(data_2016, "TAVG")
332 cx_tbl_2016 <- cx_tbl_2016 %>%
333   mutate(window = "2000-2016")
334
335 data_2019 <- dataset %>%
336   filter(Year <= 2019)
337 cx_tbl_2019 <- calc_cx_pearson(data_2019, "TAVG")
338 cx_tbl_2019 <- cx_tbl_2019 %>%
339   mutate(window = "2000-2019")
340
341 compare_corr <- bind_rows(cx_tbl_2008, cx_tbl_2010, cx_tbl_2016, cx_tbl_2019, id=NULL)
342
343 compare_corr <- compare_corr %>%
344   mutate(Period = factor(Period, levels = c("Annuel", "froid", "chaud")))
345
346
347 ggplot(compare_corr %>% filter(Sex=="m"), aes(Age, ct_x, group = window, color = window))+
348   geom_line(linewidth = 0.9)+
349   facet_wrap(~Period, ncol = 3)+
350   labs(title = "Homme",

```

```

351     x = "age", y = "Coefficient de corrélation", color="Période d'estimation")+
352     theme_minimal()+
353     theme(panel.grid = element_line("gray85"))
354
355
356 ##### random noise on temperature
357 sigma_TAVG <- sd(dataset_train$TAVG)
358 n_sim = 5
359 compare_list <- list()
360 set.seed(123)
361 for(i in 1:n_sim) {
362   # Unique temperature
363   data_temp <- dataset_train %>%
364     distinct(Year, Week, TAVG)
365
366   # Noise
367   data_temp <- data_temp %>%
368     mutate(noise = rnorm(n(), mean = 0, sd=0.2*sigma_TAVG),
369            TAVG_noise = TAVG + noise)
370
371   dataset_noise <- dataset_train %>%
372     left_join(data_temp %>% select(Year, Week, TAVG_noise ),
373              by = c("Year", "Week"))
374   cx_noise <- calc_cx_pearson(dataset_noise, "TAVG_noise")%>%
375     rename(ct_x_noise = ct_x) %>% mutate(simulation = i)
376
377   compare_i <- left_join(cx_tbl, cx_noise, by = c("Sex", "Age", "Period"))%>%
378     mutate(diff = ct_x_noise - ct_x)
379   compare_list[[i]] <- compare_i
380
381 }
382
383
384 compare_all <- bind_rows(compare_list)
385
386 compare_5 <- compare_all %>%
387   filter(simulation %in% 1:5)%>%
388   mutate(Period = factor(Period, levels = c("Annuel", "froid", "chaud")))
389
390 ggplot()+
391   geom_line(data=compare_5 %>% filter(Sex=="m"),
392            aes(x = Age, y = ct_x_noise, color = factor(simulation)), linewidth = 1) +
393   geom_line(data = cx_tbl %>% filter(Sex=="m")%>%
394            mutate(Period = factor(Period, levels = c("Annuel", "froid", "chaud")),
395                   type = "Base"),
396            aes(x=Age, y = ct_x, linetype = type), linewidth = 1, color = "black")+
397   facet_wrap(~Period, ncol = 3)+
398   scale_linetype_manual(values = c( "Base"="dashed"))+
399   labs(
400     title = "Homme",
401     x = "Âge",
402     y = "coefficient de corrélation",
403     color = "Simulation",
404     linetype=""
405   ) +
406   theme_minimal()
407
408 #####
409 #       code représentation graphique       #
410 #####
411

```

```

412
413
414 # --- Style publication (base R)
415 pub_par <- function() {
416   par(
417     mar = c(4.8, 5.0, 3.2, 1.2),
418     mgp = c(2.8, 0.9, 0),
419     tck = -0.02,
420     cex.axis = 1.0,
421     cex.lab = 1.15,
422     cex.main = 1.25
423   )
424 }
425
426 # Convertit "YYYY-WW" ou "YYYY-W" -> année fractionnaire
427 yearweek_to_yearfrac <- function(yw) {
428   yw <- as.character(yw)
429   m <- regexec("^\\d{4}\\D*(\\d{1,2})$", yw)
430   parts <- regmatches(yw, m)
431   out <- rep(NA_real_, length(yw))
432   ok <- lengths(parts) == 3
433   if (any(ok)) {
434     y <- as.integer(vapply(parts[ok], `[[`, "", 2))
435     ww <- as.integer(vapply(parts[ok], `[[`, "", 3))
436     out[ok] <- y + (pmax(ww, 1) - 1) / 52
437   }
438   out
439 }
440
441 axis_years <- function(x_yearfrac, by = 2) {
442   x_yearfrac <- x_yearfrac[is.finite(x_yearfrac)]
443   y0 <- floor(min(x_yearfrac))
444   y1 <- ceiling(max(x_yearfrac))
445   at <- seq(y0, y1, by = by)
446   axis(1, at = at, labels = at)
447 }
448
449
450 #####
451 # 2. FONCTIONS UTILITAIRES COMMUNES
452 #####
453
454 safe_mean <- function(x) mean(x, na.rm = TRUE)
455
456 safe_mad <- function(obs, pred) {
457   mean(abs(obs - pred), na.rm = TRUE)
458 }
459
460 safe_rmse <- function(obs, pred) {
461   sqrt(mean((obs - pred)^2, na.rm = TRUE))
462 }
463
464 safe_mape <- function(obs, pred) {
465   idx <- is.finite(obs) & is.finite(pred) & obs != 0
466   if (sum(idx) == 0) return(NA_real_)
467   mean(abs((obs[idx] - pred[idx]) / obs[idx]), na.rm = TRUE) * 100
468 }
469
470 pois_loglik <- function(D, mu) {
471   mu <- pmax(mu, 1e-30)
472   sum(D * log(mu) - mu - lgamma(D + 1), na.rm = TRUE)

```

```

473 }
474
475 pois_deviance <- function(D, mu) {
476   mu <- pmax(mu, 1e-30)
477   term <- ifelse(D == 0, mu, D * log(D / mu) - (D - mu))
478   2 * sum(term, na.rm = TRUE)
479 }
480
481 yearweek_to_yearfrac <- function(yw) {
482   yw <- as.character(yw)
483   m <- regexec("^\\d{4}\\D*(\\d{1,2})$", yw)
484   parts <- regmatches(yw, m)
485   out <- rep(NA_real_, length(yw))
486   ok <- lengths(parts) == 3
487
488   if (any(ok)) {
489     y <- as.integer(vapply(parts[ok], `[`, "", 2))
490     ww <- as.integer(vapply(parts[ok], `[`, "", 3))
491     out[ok] <- y + (pmax(ww, 1) - 1) / 52
492   }
493
494   out
495 }
496
497 pub_par <- function() {
498   par(
499     mar = c(4.8, 5.0, 3.2, 1.2),
500     mgp = c(2.8, 0.9, 0),
501     tck = -0.02,
502     cex.axis = 1.0,
503     cex.lab = 1.15,
504     cex.main = 1.25
505   )
506 }
507
508 axis_years <- function(x_yearfrac, by = 2) {
509   x_yearfrac <- x_yearfrac[is.finite(x_yearfrac)]
510   y0 <- floor(min(x_yearfrac))
511   y1 <- ceiling(max(x_yearfrac))
512   at <- seq(y0, y1, by = by)
513   axis(1, at = at, labels = at)
514 }
515
516 coef_pvalues <- function(coef, var_diag) {
517   z <- coef / sqrt(var_diag)
518   2 * (1 - pnorm(abs(z)))
519 }
520
521 stabilize_series <- function(x_ts) {
522   x <- as.numeric(x_ts)
523
524   if (all(is.finite(x)) && all(x > 0)) {
525     list(ts = log(x_ts), transform = "log")
526   } else {
527     list(ts = x_ts, transform = "none")
528   }
529 }
530
531 invert_transform <- function(x, transform_used) {
532   if (transform_used == "log") exp(x) else x
533 }

```

```

534
535 choose_D <- function(x_ts) {
536   tryCatch(
537     forecast::nsdiffs(x_ts),
538     error = function(e) 0L
539   )
540 }
541
542
543 #####
544 # 3. TABLE DES CAS : SEXE x PERIODE
545 #####
546
547 cases_tbl <- tibble(
548   id      = c("f_froid", "f_chaud", "f_annuel", "m_froid", "m_chaud", "m_annuel"),
549   sex     = c("f", "f", "f", "m", "m", "m"),
550   saison  = c("froid", "chaud", NA, "froid", "chaud", NA),
551   Period  = c("froid", "chaud", "Annuel", "froid", "chaud", "Annuel")
552 )
553
554 #####
555 #####      Application du modèle de Lee-Carter      #####
556 #####
557
558 #####
559 #      Ajustement      #
560 #####
561
562 lc_model <- lc(link = "log")
563
564 fit_LC <- function(df, sex_label, periode) {
565
566   # Filtrage par sexe
567   if(!is.null(periode)){
568     df_sub <- df %>% filter(Sex == sex_label & saison==periode)
569   }
570   else {
571     df_sub <- df %>% filter(Sex==sex_label)
572   }
573
574   # On trie pour être sûr de l'ordre chronologique
575   df_sub <- df_sub %>% arrange(Year, Week, Age)
576   # Extraction des vecteurs
577
578   ages <- sort(unique(df_sub$Age))
579   periods <- sort(unique(df_sub$Year_week))
580
581   # Création de la matrice des Décès (Dxt)
582   # On s'assure que le pivot ne crée pas de colonnes étranges
583   Dxt <- df_sub %>%
584     select(Age, Year_week, Deaths) %>%
585     pivot_wider(names_from = Year_week, values_from = Deaths) %>%
586     arrange(Age) %>%
587     select(-Age) %>%
588     as.matrix()
589
590   # Création de la matrice d'Exposition (Ext)
591   Ext <- df_sub %>%
592     select(Age, Year_week, Exposure_week) %>%
593     pivot_wider(names_from = Year_week, values_from = Exposure_week) %>%
594     arrange(Age) %>%

```

```

595     select(-Age) %>%
596     as.matrix()
597
598     # --- SOLUTION À L'ERREUR : Construction manuelle de l'objet ---
599     # Si StMoMoData() rejette les arguments nommés, on lui passe
600     # les objets dans l'ordre exact qu'il attend (Dxt, Ext, ages, years)
601     # ou on crée une liste de classe StMoMoData.
602
603     data_lc <- list(
604       Dxt = Dxt,
605       Ext = Ext,
606       ages = ages,
607       years = 1:length( periods ),
608       type = "mortality"
609     )
610     class(data_lc) <- "StMoMoData"
611
612     # On nomme les lignes et colonnes des matrices pour la sécurité
613     dimnames(data_lc$Dxt) <- list(ages, 1:length( periods ))
614     dimnames(data_lc$Ext) <- list(ages, 1:length( periods ))
615
616     # 2. Ajustement (Fit)
617     # On ajoute une petite sécurité pour les valeurs NA ou nulles dans l'exposition
618     data_lc$Ext[data_lc$Ext <= 0] <- 0.0001
619
620     fit_result <- fit(lc_model, data = data_lc)
621
622     return(fit_result)
623 }
624
625
626 # femme
627 fit_f_f <- fit_LC(dataset_train, "f","froid")
628 fit_f_c <- fit_LC(dataset_train, "f", "chaud")
629 fit_f_a <- fit_LC(dataset_train, "f", NULL)
630
631 # homme
632 fit_m_f <- fit_LC(dataset_train, "m","froid")
633 fit_m_c <- fit_LC(dataset_train, "m", "chaud")
634 fit_m_a <- fit_LC(dataset_train, "m", NULL)
635
636
637 fits <- list(
638   f_froid = fit_f_f,
639   f_chaud = fit_f_c,
640   f_annuel = fit_f_a,
641   m_froid = fit_m_f,
642   m_chaud = fit_m_c,
643   m_annuel = fit_m_a
644 )
645
646
647
648 # -----
649 # Données LC pour la représentation graphiques des paramètres
650 # -----
651
652 get_lc_plot_data <- function(fit, data, sex, season_label = NULL, fill = c("locf", "NA")) {
653
654   fill <- match.arg(fill)
655

```



```

656 # calendrier complet
657 all_yw <- data %>%
658   distinct(Year_week) %>%
659   arrange(Year_week) %>%
660   pull(Year_week)
661
662 # semaines utilisées dans le fit
663 if (is.null(season_label)) {
664   used_yw <- data %>%
665     filter(Sex == sex) %>%
666     distinct(Year_week) %>%
667     arrange(Year_week) %>%
668     pull(Year_week)
669 } else {
670   used_yw <- data %>%
671     filter(Sex == sex, saison == season_label) %>%
672     distinct(Year_week) %>%
673     arrange(Year_week) %>%
674     pull(Year_week)
675 }
676
677 kt <- as.numeric(fit$kt[1, ])
678
679 # remplacer kt sur le calendrier complet
680 kt_full <- rep(NA_real_, length(all_yw))
681 idx <- match(used_yw, all_yw)
682 kt_full[idx] <- kt
683
684 # prolongation visuelle pour obtenir des segments plats
685 if (fill == "locf") {
686   for (i in 2:length(kt_full)) {
687     if (is.na(kt_full[i])) kt_full[i] <- kt_full[i - 1]
688   }
689 }
690
691 list(
692   age = fit$ages,
693   ax = as.numeric(fit$ax),
694   bx = as.numeric(fit$bx),
695   yw = all_yw,
696   x = yearweek_to_yearfrac(all_yw),
697   kt = kt_full
698 )
699 }
700
701 # -----
702 # Tracé alpha_x
703 # -----
704 plot_lc_alpha_pub <- function(fit, col_line = "navy") {
705   pub_par()
706   plot(fit$ages, as.numeric(fit$ax),
707     type = "l", lwd = 2, col = col_line,
708     xlab = "Age", ylab = expression(alpha[x]),
709     main = expression(alpha[x] ~ "vs. age"))
710   grid()
711 }
712
713 # -----
714 # Tracé beta_x
715 # -----
716 plot_lc_beta_pub <- function(fit, col_line = "navy") {

```

```

717   pub_par()
718   plot(fit$ages, as.numeric(fit$bx),
719         type = "l", lwd = 2, col = col_line,
720         xlab = "Age", ylab = expression(beta[x]^{(1)}),
721         main = expression(beta[x]^{(1)} ~ "vs. age"))
722   grid()
723 }
724
725 # -----
726 # Tracé kappa_t remis sur le calendrier complet
727 # -----
728 plot_lc_kappa_pub <- function(fit, data, sex, season_label = NULL,
729                               fill = c("locf", "NA"),
730                               year_tick_by = 2,
731                               col_line = "navy") {
732
733   fill <- match.arg(fill)
734   res <- get_lc_plot_data(fit, data, sex, season_label, fill = fill)
735
736   keep <- is.finite(res$x) & is.finite(res$kt)
737   x <- res$x[keep]
738   y <- res$kt[keep]
739
740   pub_par()
741   plot(x, y, type = "l", lwd = 2, col = col_line,
742        xlab = "Year", ylab = expression(kappa[t]^{(1)}),
743        main = expression(kappa[t]^{(1)} ~ "vs. time"),
744        xaxt = "n")
745   axis_years(x, by = year_tick_by)
746   grid()
747 }
748
749 # -----
750 # Tracé des 3 composantes LC sur une ligne
751 # -----
752 plot_lc_pub <- function(fit, data, sex, season_label = NULL,
753                         fill = c("locf", "NA"),
754                         year_tick_by = 2,
755                         col_line = "navy") {
756
757   fill <- match.arg(fill)
758   oldpar <- par(no.readonly = TRUE)
759   on.exit(par(oldpar))
760
761   par(mfrow = c(1, 3))
762
763   plot_lc_alpha_pub(fit, col_line)
764   plot_lc_beta_pub(fit, col_line)
765   plot_lc_kappa_pub(fit, data, sex, season_label, fill, year_tick_by, col_line)
766 }
767
768 #Hommes froid
769 plot_lc_pub(fit_m_f, dataset_train, sex = "m", season_label = "froid",
770            fill = "locf", col_line = "blue")
771 #Hommes chaud
772 plot_lc_pub(fit_m_c, dataset_train, sex = "m", season_label = "chaud",
773            fill = "locf", col_line = "blue")
774 #Hommes annuel
775 plot_lc_pub(fit_m_a, dataset_train, sex = "m", season_label = NULL,
776            fill = "locf", col_line = "blue")
777 #Femmes froid

```

```

778 plot_lc_pub(fit_f_f, dataset_train, sex = "f", season_label = "froid",
779             fill = "locf", col_line = "red")
780 #Femmes chaud
781 plot_lc_pub(fit_f_c, dataset_train, sex = "f", season_label = "chaud",
782             fill = "locf", col_line = "red")
783 #Femmes annuel
784 plot_lc_pub(fit_f_a, dataset_train, sex = "f", season_label = NULL,
785             fill = "locf", col_line = "red")
786
787
788 # =====
789 # Evaluation in-sample du modèle Lee-Carter
790 # - décès prédits
791 # - taux de mortalité prédits
792 # - erreurs : MAD, RMSE, MAPE, AIC, BIC, Deviance
793 # - tableau final par sexe et période
794 # =====
795
796
797 # Nombre effectif de paramètres pour Lee-Carter :
798
799 count_params_lc <- function(n_ages, n_periods) {
800   max(2 * n_ages + n_periods - 2, 1)
801 }
802
803 # -----
804 # 2) Préparation des données
805 # -----
806 prepare_lc_data <- function(df, sex_label, periode = NULL) {
807
808   if (!is.null(periode)) {
809     df_sub <- df %>% filter(Sex == sex_label, saison == periode)
810     period_key <- periode
811   } else {
812     df_sub <- df %>% filter(Sex == sex_label)
813     period_key <- "Annuel"
814   }
815
816   df_sub <- df_sub %>% arrange(Year, Week, Age)
817
818   ages <- sort(unique(df_sub$Age))
819   periods <- sort(unique(df_sub$Year_week))
820
821   Dxt <- df_sub %>%
822     select(Age, Year_week, Deaths) %>%
823     group_by(Age, Year_week) %>%
824     summarise(Deaths = sum(Deaths, na.rm = TRUE), .groups = "drop") %>%
825     pivot_wider(names_from = Year_week, values_from = Deaths, values_fill = 0) %>%
826     arrange(Age)
827
828   Ext <- df_sub %>%
829     select(Age, Year_week, Exposure_week) %>%
830     group_by(Age, Year_week) %>%
831     summarise(Exposure_week = sum(Exposure_week, na.rm = TRUE), .groups = "drop") %>%
832     pivot_wider(names_from = Year_week, values_from = Exposure_week, values_fill = 0) %>%
833     arrange(Age)
834
835   age_vec <- Dxt$Age
836   period_vec <- names(Dxt)[-1]
837
838   Dxt_mat <- Dxt %>% select(-Age) %>% as.matrix()

```

```

839 Ext_mat <- Ext %>% select(-Age) %>% as.matrix()
840
841 Ext_mat[Ext_mat <= 0] <- 0.0001
842
843 list(
844   df_sub = df_sub,
845   ages = age_vec,
846   periods = period_vec,
847   Dxt = Dxt_mat,
848   Ext = Ext_mat,
849   Sex = sex_label,
850   Period = period_key
851 )
852 }
853
854 #####
855 # 3. Evaluation du modèle de lee-carter ###
856 #####
857
858 evaluate_one_lc_fit <- function(fit_obj, dataset, sex_label, periode = NULL) {
859
860   prep <- prepare_lc_data(dataset, sex_label, periode)
861
862   ages <- prep$ages
863   periods <- prep$periods
864   Dxt <- prep$Dxt
865   Ext <- prep$Ext
866
867   ax <- as.numeric(fit_obj$ax)
868   bx <- as.numeric(fit_obj$bx)
869   kt <- as.numeric(fit_obj$kt)
870
871   if (length(ax) != length(ages)) {
872     stop("La longueur de ax ne correspond pas au nombre d'âges.")
873   }
874   if (length(bx) != length(ages)) {
875     stop("La longueur de bx ne correspond pas au nombre d'âges.")
876   }
877   if (length(kt) != length(periods)) {
878     stop("La longueur de kt ne correspond pas au nombre de périodes.")
879   }
880
881   eta <- outer(ax, rep(1, length(kt))) + outer(bx, kt)
882
883   mx_hat <- exp(eta)
884   D_hat <- Ext * mx_hat
885
886   mx_obs <- Dxt / Ext
887   D_obs <- Dxt
888
889   logLik_val <- pois_loglik(D_obs, D_hat)
890   dev_val <- pois_deviance(D_obs, D_hat)
891   k_param <- count_params_lc(n_ages = length(ages), n_periods = length(periods))
892   n_obs <- length(D_obs)
893
894   AIC_val <- -2 * logLik_val + 2 * k_param
895   BIC_val <- -2 * logLik_val + log(n_obs) * k_param
896
897   summary_tbl <- tibble(
898     Sex = prep$Sex,
899     Period = prep$Period,

```

```

900   MAD_Deaths = safe_mad(D_obs, D_hat),
901   RMSE_Deaths = safe_rmse(D_obs, D_hat),
902   MAPE_Deaths = safe_mape(D_obs, D_hat),
903   MAD_Rate = safe_mad(mx_obs, mx_hat),
904   RMSE_Rate = safe_rmse(mx_obs, mx_hat),
905   MAPE_Rate = safe_mape(mx_obs, mx_hat),
906   LogLik = logLik_val,
907   AIC = AIC_val,
908   BIC = BIC_val,
909   Deviance = dev_val,
910   Nb_obs = n_obs,
911   Nb_param = k_param
912 )
913
914 # IMPORTANT : Age en premier, Year_week en second
915 # car as.vector(matrix âge x période) lit colonne par colonne
916 pred_tbl <- expand.grid(
917   Age = ages,
918   Year_week = periods,
919   KEEP.OUT.ATTRS = FALSE,
920   stringsAsFactors = FALSE
921 ) %>%
922   mutate(
923     Sex = prep$Sex,
924     Period = prep$Period,
925     Exposure = as.vector(Ext),
926     Deaths_obs = as.vector(D_obs),
927     Deaths_hat = as.vector(D_hat),
928     mx_obs = as.vector(mx_obs),
929     mx_hat = as.vector(mx_hat),
930     eta = as.vector(eta)
931   ) %>%
932   relocate(Sex, Period, Year_week, Age)
933
934 date_map <- prep$df_sub %>%
935   distinct(Year_week, Year, Week, date) %>%
936   arrange(Year, Week)
937
938 pred_tbl <- pred_tbl %>%
939   left_join(date_map, by = "Year_week") %>%
940   relocate(Year, Week, date, .after = Year_week)
941
942 list(
943   summary = summary_tbl,
944   predictions = pred_tbl
945 )
946 }
947 # -----
948 # 4) Appliquer aux 6 ajustements
949 # -----
950
951 lc_results <- list(
952   f_annuel = evaluate_one_lc_fit(fit_f_a, dataset_train, sex_label = "f", periode = NULL),
953   f_froid = evaluate_one_lc_fit(fit_f_f, dataset_train, sex_label = "f", periode = "froid"),
954   f_chaud = evaluate_one_lc_fit(fit_f_c, dataset_train, sex_label = "f", periode = "chaud"),
955   m_annuel = evaluate_one_lc_fit(fit_m_a, dataset_train, sex_label = "m", periode = NULL),
956   m_froid = evaluate_one_lc_fit(fit_m_f, dataset_train, sex_label = "m", periode = "froid"),
957   m_chaud = evaluate_one_lc_fit(fit_m_c, dataset_train, sex_label = "m", periode = "chaud")
958 )
959
960 lc_summary_table <- purrr::map_dfr(lc_results, "summary") %>%

```

```

961   mutate(
962     Period = factor(Period, levels = c("Annuel", "froid", "chaud")),
963     Sex     = factor(Sex, levels = c("f", "m"))
964   ) %>%
965   arrange(Sex, Period)
966
967 lc_predictions_table <- purrr::map_dfr(lc_results, "predictions") %>%
968   mutate(
969     Period = factor(Period, levels = c("Annuel", "froid", "chaud")),
970     Sex     = factor(Sex, levels = c("f", "m"))
971   ) %>%
972   arrange(Sex, Period, Year, Week, Age, date)
973 # -----
974 # 5) Affichage des résultats
975 # -----
976
977 print(lc_summary_table)
978 print(head(lc_predictions_table))
979
980 # Importation en fichier latex
981 kable(lc_summary_table, format = "latex", caption = "Qualité d'ajustement du modèle LC", label = "tab2")
982
983 ##### Visualisation de l'ajustement du modèles aux données brutes
984
985 plot_lc_rates_by_age_date <- function(pred_table,
986                                       sex_label,
987                                       period_label,
988                                       ages_to_plot = c(20, 45, 70, 85)) {
989
990   df_plot <- pred_table %>%
991     filter(Sex == sex_label, Period == period_label, Age %in% ages_to_plot) %>%
992     select(date, Age, mx_obs, mx_hat) %>%
993     pivot_longer(
994       cols = c(mx_obs, mx_hat),
995       names_to = "Series",
996       values_to = "mx"
997     ) %>%
998     mutate(
999       Series = recode(
1000         Series,
1001         mx_obs = "Observé",
1002         mx_hat = "Prédit"
1003       )
1004     )
1005
1006   ggplot(df_plot, aes(x = date, y = mx, linetype = Series, colour = Series)) +
1007     geom_line(linewidth = 0.8, na.rm = TRUE) +
1008     facet_wrap(~ Age, scales = "free_y") +
1009     scale_linetype_manual(
1010       values = c("Observé" = "solid", "Prédit" = "solid")
1011     ) +
1012     scale_color_manual(
1013       values = c("Observé" = "blue", "Prédit" = "red")
1014     ) +
1015     theme_minimal() +
1016     labs(
1017       title = paste0("Lee-Carter - ", sex_label, " - ", period_label),
1018       x = "Date",
1019       y = "Taux de mortalité",
1020       linetype = "",

```

```

1022     colour = ""
1023   ) +
1024   theme(panel.grid = element_line(color = "grey80"))
1025 }
1026
1027 plot_lc_rates_by_age_date(
1028   lc_predictions_table,
1029   sex_label = "m",
1030   period_label = "chaud",
1031   ages_to_plot = c(20, 45, 70, 85)
1032 )
1033
1034
1035
1036 #####
1037 #####
1038 # Application de la méthode de box-jenskin pour le choix des paramètres de SARIMA#
1039 #####
1040 #####
1041
1042
1043 # -----
1044 # OUTILS UTILITAIRES: Fonctions utilitaires pour Box-Jenkins
1045 # -----
1046
1047 # p-value des coefficients comme dans le document
1048 coef_pvalues <- function(coef, var_diag) {
1049   z <- coef / sqrt(var_diag)
1050   2 * (1 - pnorm(abs(z)))
1051 }
1052
1053 # Extraire proprement le k_t de Lee-Carter
1054 extract_lc_kt <- function(fit_obj) {
1055   kt <- fit_obj$kt
1056   kt <- as.numeric(kt[1, ])
1057   return(kt)
1058 }
1059
1060 # Déterminer automatiquement la fréquence saisonnière s
1061 # à partir du nombre médian de semaines observées par an
1062 infer_seasonal_frequency <- function(df_sub) {
1063   freq_tab <- df_sub %>%
1064     distinct(Year, Year_week) %>%
1065     count(Year, name = "n_weeks")
1066
1067   s <- round(median(freq_tab$n_weeks))
1068   return(s)
1069 }
1070
1071 # Construire la série temporelle d'une combinaison
1072 build_lc_series <- function(Dataset, fit_obj, sex_label, periode_label) {
1073
1074   if (!is.na(periode_label)) {
1075     df_sub <- Dataset %>%
1076       filter(Sex == sex_label, saison == periode_label) %>%
1077       arrange(Year, Week, Age)
1078   } else {
1079     df_sub <- Dataset %>%
1080       filter(Sex == sex_label) %>%
1081       arrange(Year, Week, Age)
1082   }

```

```

1083
1084 periods <- sort(unique(df_sub$Year_week))
1085
1086 kt <- extract_lc_kt(fit_obj)
1087
1088 cat("sex =", sex_label, "| periode =", periode_label,
1089     "| length(kt) =", length(kt),
1090     "| length(periods) =", length(periods), "\n")
1091
1092 stopifnot(length(kt) == length(periods))
1093
1094 s <- infer_seasonal_frequency(df_sub)
1095 start_year <- min(df_sub$Year, na.rm = TRUE)
1096
1097 kt_ts <- ts(kt, start = c(start_year, 1), frequency = s)
1098
1099 list(
1100     ts = kt_ts,
1101     periods = periods,
1102     s = s,
1103     raw_df = df_sub
1104 )
1105 }
1106
1107 # Stabilisation de variance :
1108 # si série > 0 partout, on applique log
1109 # sinon on garde la série brute
1110 stabilize_variance <- function(x_ts) {
1111     x <- as.numeric(x_ts)
1112
1113     if (all(is.finite(x)) && all(x > 0)) {
1114         list(
1115             ts = log(x_ts),
1116             transform = "log"
1117         )
1118     } else {
1119         list(
1120             ts = x_ts,
1121             transform = "none"
1122         )
1123     }
1124 }
1125
1126 # Appliquer les différenciations saisonnière et ordinaire
1127 make_clean_series <- function(x_ts, D, d, s) {
1128     out <- x_ts
1129
1130     if (D > 0) {
1131         out <- diff(out, lag = s, differences = D)
1132     }
1133     if (d > 0) {
1134         out <- diff(out, lag = 1, differences = d)
1135     }
1136
1137     return(out)
1138 }
1139
1140 ##### Construction des les 6 séries de Lee-Carter
1141
1142 # -----
1143 # TABLE DES 6 CAS

```



```

1144 # -----
1145 cases_lc <- tibble(
1146   id      = c("f_froid", "f_chaud", "f_annuel", "m_froid", "m_chaud", "m_annuel"),
1147   sex     = c("f", "f", "f", "m", "m", "m"),
1148   saison  = c("froid", "chaud", NA, "froid", "chaud", NA)
1149 )
1150
1151 # Associer chaque fit à son cas
1152 fits_lc <- list(
1153   f_froid = fit_f_f,
1154   f_chaud = fit_f_c,
1155   f_annuel = fit_f_a,
1156   m_froid = fit_m_f,
1157   m_chaud = fit_m_c,
1158   m_annuel = fit_m_a
1159 )
1160
1161 # Construire les séries temporelles
1162 series_lc <- cases_lc %>%
1163   mutate(
1164     serie_obj = pmap(
1165       list(sex, saison, id),
1166       function(sex, saison, id) {
1167         build_lc_series(
1168           Dataset = dataset_train,
1169           fit_obj = fits_lc[[id]],
1170           sex_label = sex,
1171           periode_label = saison
1172         )
1173       }
1174     )
1175   )
1176
1177 # -----
1178 # ETAPE 1 : VISUALISATION DE LA SERIE BRUTE
1179 # -----
1180 plot_one_lc_series <- function(series_lc, case_id) {
1181
1182   i <- which(series_lc$id == case_id)
1183
1184   if (length(i) == 0) {
1185     stop("case_id introuvable dans series_lc")
1186   }
1187
1188   x <- series_lc$serie_obj[[i]]$ts
1189   s <- series_lc$serie_obj[[i]]$s
1190
1191   plot(
1192     x,
1193     type = "o",
1194     pch = 16,
1195     cex = 0.4,
1196     lwd = 1.5,
1197     main = paste0("Série brute - ", case_id, " (s=", s, ")"),
1198     xlab = "Temps",
1199     ylab = expression(k[t])
1200   )
1201
1202   abline(h = mean(x, na.rm = TRUE), col = "red", lty = 2)
1203 }
1204

```

```
1205 #acf(fit_f_f$kt, lag=30)
1206
1207 par(mfrow=c(1,3))
1208 plot_one_lc_series(series_lc, "f_froid")
1209 plot_one_lc_series(series_lc, "f_chaud")
1210 plot_one_lc_series(series_lc, "f_annuel")
1211
1212 plot_one_lc_series(series_lc, "m_froid")
1213 plot_one_lc_series(series_lc, "m_chaud")
1214 plot_one_lc_series(series_lc, "m_annuel")
1215
1216 ### ACF et PCAF sur les données brutes
1217
1218 x <- series_lc$serie_obj[[which(series_lc$id == "m_chaud")]]$ts
1219
1220
1221 acf(as.numeric(x), lag.max = 120, main = "ACF brute - m_chaud")
1222 pacf(as.numeric(x), lag.max = 120, main = "PACF brute - f_chaud")
1223
1224
1225 # -----
1226 # ETAPE 2 : STABILISATION DE LA VARIANCE
1227 # -----
1228 series_lc <- series_lc %>%
1229   mutate(
1230     stab_obj = map(serie_obj, ~ stabilize_variance(.x$ts)),
1231     ts_stab = map(stab_obj, "ts"),
1232     transform_used = map_chr(stab_obj, "transform")
1233   )
1234
1235 # Vérifier quelle transformation a été utilisée
1236 series_lc %>%
1237   select(id, transform_used)
1238
1239 # -----
1240 # ETAPE 3 : CHOIX DE D ET SUPPRESSION DE LA SAISONNALITE
1241 # -----
1242 x <- series_lc$serie_obj[[which(series_lc$id=="f_chaud")]]$ts
1243 s <- 22
1244
1245 plot(x[1:s], type="l", ylim=c(min(x), max(x)),
1246      main="Cycle saisonnier - f_chaud")
1247
1248 for(i in 2:floor(length(x)/s)){
1249   lines(x[s*(i-1)+(1:s)], col=i)
1250 }
1251
1252
1253
1254 choose_D <- function(x_ts) {
1255   # nsdiffs peut parfois échouer sur certaines séries
1256   out <- tryCatch(
1257     forecast::nsdiffs(x_ts),
1258     error = function(e) 0
1259   )
1260   return(out)
1261 }
1262
1263 series_lc <- series_lc %>%
1264   mutate(
1265     s= map_int(ts_stab, frequency),
```

```

1266     D = map_int(ts_stab, choose_D),
1267     ts_noseason = map2(ts_stab, D, function(x, D) {
1268       s <- frequency(x)
1269       if (D > 0) diff(x, lag = s, differences = D) else x
1270     })
1271   )
1272
1273 series_lc %>%
1274   select(id, s, D)
1275
1276 ### plot de la série désaisonnalisé
1277 plot_noseason_one <- function(series_lc, case_id) {
1278
1279   i <- which(series_lc$id == case_id)
1280
1281   if (length(i) == 0) {
1282     stop("case_id introuvable dans series_lc")
1283   }
1284
1285   x <- series_lc$ts_noseason[[i]]
1286   D <- series_lc$D[i]
1287   s <- series_lc$s[i]
1288
1289   plot(
1290     x,
1291     type = "l",
1292     pch = 16,
1293     cex = 0.4,
1294     lwd = 1.2,
1295     main = paste0(case_id, " (D=", D, ", s=", s, ")"),
1296     xlab = "Temps",
1297     ylab = ""
1298   )
1299
1300   abline(h = mean(x, na.rm = TRUE), col = "red", lty = 2)
1301 }
1302
1303 plot_noseason_one(series_lc, "f_froid")
1304 plot_noseason_one(series_lc, "f_chaud")
1305 plot_noseason_one(series_lc, "f_annuel")
1306
1307 plot_noseason_one(series_lc, "m_froid")
1308 plot_noseason_one(series_lc, "m_chaud")
1309 plot_noseason_one(series_lc, "m_annuel")
1310
1311 x_ns <- series_lc$ts_noseason[[which(series_lc$id == "f_froid")]]
1312
1313 acf(as.numeric(x_ns), lag.max = 120, main = "ACF sans saisonnalité - f_froid")
1314 pacf(as.numeric(x_ns), lag.max = 120, main = "PACF sans saisonnalité - f_froid")
1315
1316 # -----
1317 # ETAPE 4 : CHOIX DE d ET SUPPRESSION DE LA TENDANCE
1318 # -----
1319 series_lc <- series_lc %>%
1320   mutate(
1321     d = 1L,
1322     ts_clean = map(ts_noseason, ~ diff(.x, lag = 1))
1323   )
1324
1325 ### plot de la série sans tendance
1326 x <- series_lc$ts_clean[[which(series_lc$id=="m_annuel")]]

```

```

1327
1328 plot(x, type="l", pch=16, cex=0.4,
1329       main="m_annuel(d=1)",
1330       ylab="k_t",
1331       xlab="Temps")
1332
1333 abline(h=mean(x), col="red", lty=2)
1334
1335 acf(as.numeric(x), lag.max=120,
1336     main="ACF m_annuel")
1337
1338 pacf(as.numeric(x), lag.max=120,
1339      main="PACF m_annuel")
1340
1341 # -----
1342 # ETAPE 5 : ANALYSE ACF / PACF
1343 # -----
1344 plot_acf_pacf_all <- function(series_lc, lag_max = 120) {
1345   for (i in seq_len(nrow(series_lc))) {
1346     x <- series_lc$ts_clean[[i]]
1347     id_case <- series_lc$id[i]
1348     par(mfrow=c(1,2))
1349     acf(x, lag.max = lag_max, main = paste("ACF -", id_case))
1350     pacf(x, lag.max = lag_max, main = paste("PACF -", id_case))
1351   }
1352   par(mfrow=c(1,1))
1353 }
1354
1355 plot_acf_pacf_all(series_lc, lag_max = 120)
1356
1357 # -----
1358 # ETAPE 6 : ESTIMATION DES MODELES CANDIDATS
1359 # -----
1360 fit_sarima_grid <- function(x_ts, d, D, s,
1361                             p_max = 2, q_max = 2,
1362                             P_max = 2, Q_max = 2,
1363                             include_mean = FALSE) {
1364   grid <- expand.grid(
1365     p = 0:p_max,
1366     q = 0:q_max,
1367     P = 0:P_max,
1368     Q = 0:Q_max
1369   )
1370
1371   results <- vector("list", nrow(grid))
1372
1373   for (i in seq_len(nrow(grid))) {
1374     p <- grid$p[i]
1375     q <- grid$q[i]
1376     P <- grid$P[i]
1377     Q <- grid$Q[i]
1378
1379     fit_try <- tryCatch(
1380       arima(
1381         x = x_ts,
1382         order = c(p, d, q),
1383         seasonal = list(order = c(P, D, Q), period = s),
1384         include.mean = include_mean,
1385         method = "ML"
1386       ),
1387       error = function(e) NULL

```

```

1388   )
1389
1390   if (!is.null(fit_try)) {
1391     coef_vec <- fit_try$coef
1392     var_diag <- diag(fit_try$var.coef)
1393
1394     # sécurité si la matrice de variance est singulière
1395     if (length(coef_vec) > 0 && length(var_diag) == length(coef_vec)) {
1396       pvals <- coef_pvalues(coef_vec, var_diag)
1397     } else {
1398       pvals <- rep(NA_real_, length(coef_vec))
1399     }
1400
1401     # test de Ljung-Box sur les résidus
1402     lb_lag <- min(floor(sqrt(length(residuals(fit_try)))), length(residuals(fit_try)) - 1)
1403     lb_pvalue <- tryCatch(
1404       Box.test(residuals(fit_try), lag = lb_lag, type = "Ljung-Box")$p.value,
1405       error = function(e) NA_real_
1406     )
1407
1408     results[[i]] <- tibble(
1409       p = p, d = d, q = q,
1410       P = P, D = D, Q = Q,
1411       s = s,
1412       ncoef = length(coef_vec),
1413       AIC = fit_try$aic,
1414       loglik = as.numeric(fit_try$loglik),
1415       lb_pvalue = lb_pvalue,
1416       all_coef_sig = ifelse(length(pvals) == 0, TRUE, all(pvals < 0.05, na.rm = TRUE)),
1417       fit = list(fit_try),
1418       coef_table = list(
1419         tibble(
1420           param = names(coef_vec),
1421           estimate = as.numeric(coef_vec),
1422           p_value = as.numeric(pvals)
1423         )
1424       )
1425     )
1426   }
1427 }
1428
1429 bind_rows(results) %>%
1430   arrange(AIC)
1431 }
1432
1433 # Application
1434
1435 series_lc <- series_lc %>%
1436   mutate(
1437     sarima_table = pmap(
1438       list(ts_stab, d, D, s),
1439       function(ts_stab, d, D, s) {
1440         fit_sarima_grid(
1441           x_ts = ts_stab,
1442           d = d,
1443           D = D,
1444           s = s,
1445           p_max = 2,
1446           q_max = 2,
1447           P_max = 2,
1448           Q_max = 2,

```

```

1449         include_mean = FALSE
1450     )
1451 }
1452 )
1453 )
1454
1455
1456 # -----
1457 # ETAPE 7 : TOP MODELES PAR AIC
1458 # -----
1459 show_top_models <- function(series_lc, top_n = 10) {
1460   for (i in seq_len(nrow(series_lc))) {
1461     cat("\n===== \n")
1462     cat("CAS :", series_lc$id[i], "\n")
1463     cat("===== \n")
1464
1465     print(
1466       series_lc$sarima_table[[i]] %>%
1467         select(p, d, q, P, D, Q, s, ncoef, AIC, lb_pvalue, all_coef_sig) %>%
1468         slice_head(n = top_n)
1469     )
1470   }
1471 }
1472
1473 show_top_models(series_lc, top_n = 10)
1474
1475
1476 # -----
1477 # ETAPE 8 : SIGNIFICATIVITE DES COEFFICIENTS
1478 # -----
1479 show_coef_significance <- function(series_lc, top_n = 3) {
1480   for (i in seq_len(nrow(series_lc))) {
1481     cat("\n##### \n")
1482     cat("SIGNIFICATIVITE - CAS :", series_lc$id[i], "\n")
1483     cat("##### \n")
1484
1485     tab <- series_lc$sarima_table[[i]] %>% slice_head(n = top_n)
1486
1487     for (j in seq_len(nrow(tab))) {
1488       cat("\nModèle : (",
1489         tab$p[j], ",", tab$d[j], ",", tab$q[j], ") x (",
1490         tab$P[j], ",", tab$D[j], ",", tab$Q[j], ")_", tab$s[j], "\n", sep = ""
1491       )
1492       print(tab$coef_table[[j]])
1493     }
1494   }
1495 }
1496
1497 show_coef_significance(series_lc, top_n = 3)
1498
1499
1500 # -----
1501 # ETAPE 9 : ANALYSE DES RESIDUS
1502 # -----
1503 #####
1504 #           Validation           #
1505 #####
1506
1507
1508 fit_obj <- series_lc$best_model[[6]]$fit[[2]]
1509 tsdiag(fit_obj, floor(sqrt(length(residuals(fit_obj)))))

```

```

1510
1511
1512
1513 # -----
1514 # ETAPE 10 : CHOIX DU MEILLEUR MODELE
1515 # -----
1516 choose_best_model <- function(tab, n_best = 2) {
1517
1518   valid_tab <- tab %>%
1519     filter(all_coef_sig, !is.na(lb_pvalue), lb_pvalue > 0.05) %>%
1520     arrange(AIC, ncoef)
1521
1522   if (nrow(valid_tab) > 0) {
1523     return(slice_head(valid_tab, n = n_best))
1524   }
1525
1526   fallback <- tab %>%
1527     arrange(AIC, ncoef)
1528
1529   return(slice_head(fallback, n = n_best))
1530 }
1531
1532 series_lc <- series_lc %>%
1533   mutate(
1534     best_model = map(sarima_table, choose_best_model, n_best=2)
1535   )
1536
1537 # -----
1538 # TABLEAU FINAL DES MEILLEURS MODELES
1539 # -----
1540 best_models_lc <- series_lc %>%
1541   transmute(
1542     id = id,
1543     best = map(best_model, ~ .x %>%
1544       select(p, d, q, P, D, Q, s, ncoef, AIC, lb_pvalue, all_coef_sig))
1545   ) %>%
1546   tidyr::unnest(best)
1547
1548 best_models_lc
1549 kable(best_models_lc, format = "latex", label = "", caption = "Tableau des meilleurs modèles de SARIMA")
1550
1551
1552
1553 dataset_test <- dataset %>%
1554   filter(Year>=2017 & Year <= 2019)
1555
1556 #####
1557 #       Projection       ##
1558 #####
1559
1560 best_models_lc_final <- best_models_lc %>%
1561   filter(all_coef_sig, lb_pvalue > 0.05) %>%
1562   group_by(id) %>%
1563   arrange(AIC, ncoef, .by_group = TRUE) %>%
1564   mutate(rank = row_number()) %>%
1565   ungroup() %>%
1566   filter(
1567     (id != "m_annuel" & rank == 1) |
1568     (id == "m_annuel" & rank == 2)
1569   ) %>%
1570   select(-rank)

```

```

1571
1572 forecast_lc_case_full <- function(case_id,
1573                                   dataset,
1574                                   dataset_train,
1575                                   dataset_test,
1576                                   cases_lc,
1577                                   series_lc,
1578                                   fits_lc,
1579                                   best_order_lc) {
1580
1581   # =====
1582   # 1) Infos du cas
1583   # =====
1584   case_row <- cases_lc %>% filter(id == case_id)
1585   if (nrow(case_row) == 0) stop("case_id introuvable dans cases_lc")
1586
1587   sex_i <- case_row$sex[1]
1588   saison_i <- case_row$saison[1]
1589
1590   # =====
1591   # 2) Sous-données complètes (2000-2019)
1592   # =====
1593   if (!is.na(saison_i)) {
1594     df_all <- dataset %>%
1595       filter(Sex == sex_i, saison == saison_i) %>%
1596       arrange(Year, Week, Age)
1597
1598     df_test <- dataset_test %>%
1599       filter(Sex == sex_i, saison == saison_i) %>%
1600       arrange(Year, Week, Age)
1601
1602     period_label <- saison_i
1603   } else {
1604     df_all <- dataset %>%
1605       filter(Sex == sex_i) %>%
1606       arrange(Year, Week, Age)
1607
1608     df_test <- dataset_test %>%
1609       filter(Sex == sex_i) %>%
1610       arrange(Year, Week, Age)
1611
1612     period_label <- "Annuel"
1613   }
1614
1615   ages_all <- sort(unique(df_all$Age))
1616
1617   # =====
1618   # 3) Observé sur toute la période
1619   # =====
1620   obs_all <- df_all %>%
1621     group_by(Age, Year_week, Year, Week) %>%
1622     summarise(
1623       Deaths_obs = sum(Deaths, na.rm = TRUE),
1624       Exposure     = sum(Exposure_week, na.rm = TRUE),
1625       .groups = "drop"
1626     ) %>%
1627     mutate(
1628       mx_obs = ifelse(Exposure > 0, Deaths_obs / Exposure, NA_real_),
1629       year_t = Year + (Week - 0.5) / 52
1630     ) %>%
1631     arrange(Age, Year, Week)

```



```

1632
1633 # =====
1634 # 4) Série k_t train
1635 # =====
1636 i_case <- which(series_lc$id == case_id)
1637 if (length(i_case) == 0) stop("case_id introuvable dans series_lc")
1638
1639 x_train <- series_lc$ts_stab[[i_case]]
1640 transform_used <- series_lc$transform_used[i_case]
1641
1642 # =====
1643 # 5) Modèle SARIMA choisi
1644 # =====
1645 ord <- best_order_lc %>% filter(id == case_id)
1646 if (nrow(ord) == 0) stop("Aucun SARIMA choisi pour ce cas")
1647
1648 p <- ord$p[1]; d <- ord$d[1]; q <- ord$q[1]
1649 P <- ord$P[1]; D <- ord$D[1]; Q <- ord$Q[1]
1650 s <- ord$s[1]
1651
1652 # =====
1653 # 6) Prévoir k_t sur 2017-2019
1654 # =====
1655 periods_test <- sort(unique(df_test$Year_week))
1656 h <- length(periods_test)
1657
1658 sar_fit <- forecast::Arima(
1659   x_train,
1660   order = c(p, d, q),
1661   seasonal = list(order = c(P, D, Q), period = s),
1662   include.mean = FALSE,
1663   method = "ML"
1664 )
1665
1666 fc_kt <- forecast::forecast(sar_fit, h = h)
1667 kt_proj <- as.numeric(fc_kt$mean)
1668
1669 if (transform_used == "log") {
1670   kt_proj <- exp(kt_proj)
1671 }
1672
1673 # =====
1674 # 7) Paramètres Lee-Carter
1675 # =====
1676 fit_obj <- fits_lc[[case_id]]
1677 ax <- as.numeric(fit_obj$ax)
1678 bx <- as.numeric(fit_obj$bx)
1679 ages_fit <- fit_obj$ages
1680
1681 stopifnot(length(ax) == length(ages_fit))
1682 stopifnot(length(bx) == length(ages_fit))
1683
1684 # =====
1685 # 8) Taux projetés m_hat(x,t)
1686 # matrice âge x temps
1687 # =====
1688 mx_proj_mat <- exp(matrix(ax, nrow = length(ages_fit), ncol = h) + bx %o% kt_proj)
1689
1690 # IMPORTANT : ordre correct
1691 proj_test <- expand.grid(
1692   Age = ages_fit,

```

```

1693   Year_week = periods_test
1694 ) %>%
1695   mutate(
1696     mx_proj = as.vector(mx_proj_mat)
1697   ) %>%
1698   left_join(
1699     df_test %>%
1700       group_by(Age, Year_week, Year, Week) %>%
1701       summarise(
1702         Exposure = sum(Exposure_week, na.rm = TRUE),
1703         .groups = "drop"
1704       ),
1705     by = c("Age", "Year_week")
1706   ) %>%
1707   mutate(
1708     D_proj = Exposure * mx_proj,
1709     year_t = Year + (Week - 0.5) / 52
1710   ) %>%
1711   arrange(Age, Year, Week)
1712
1713   # =====
1714   # 9) Fusion finale
1715   # observé 2000-2019 + projeté seulement 2017-2019
1716   # =====
1717   final_df <- obs_all %>%
1718     left_join(
1719       proj_test %>% select(Age, Year_week, mx_proj, D_proj),
1720       by = c("Age", "Year_week")
1721     ) %>%
1722     mutate(
1723       Sex = sex_i,
1724       Period = period_label
1725     )
1726
1727   return(final_df)
1728 }
1729
1730
1731 plot_lc_projection_full <- function(proj_df,
1732                                     ages_to_plot = c(25, 45, 65, 85),
1733                                     title_prefix = "") {
1734
1735   df_plot <- proj_df %>%
1736     filter(Age %in% ages_to_plot)
1737
1738   plots <- lapply(ages_to_plot, function(a){
1739
1740     df_age <- df_plot %>% filter(Age == a)
1741
1742     ggplot(df_age, aes(x = year_t)) +
1743       geom_line(aes(y = mx_obs, colour = "Observed"), linewidth = 0.9, na.rm = TRUE) +
1744       geom_line(aes(y = mx_proj, colour = "Projected"),
1745                 linewidth = 0.9,
1746                 linetype = "solid",
1747                 na.rm = TRUE) +
1748       scale_color_manual(values = c("Observed"="blue","Projected"= "red" ))+
1749       theme_bw() +
1750       labs(
1751         title = paste(title_prefix, " Age", a),
1752         x = "Temps",
1753         y = "Taux de mortalité",

```

```

1754         colour = ""
1755     )
1756 })
1757
1758     return(plots)
1759 }
1760
1761 proj_f_ann_full <- forecast_lc_case_full(
1762     case_id = "m_annuel",
1763     dataset = dataset,
1764     dataset_train = dataset_train,
1765     dataset_test = dataset_test,
1766     cases_lc = cases_lc,
1767     series_lc = series_lc,
1768     fits_lc = fits_lc,
1769     best_order_lc = best_models_lc_final
1770 )
1771
1772 plots <- plot_lc_projection_full(
1773     proj_f_ann_full,
1774     ages_to_plot = c(25, 45, 65, 85)
1775 )
1776
1777 plots[[1]]
1778 plots[[2]]
1779 plots[[3]]
1780 plots[[4]]
1781
1782
1783 proj_f_ann_full %>%
1784     filter(!is.na(mx_proj), mx_obs>0)%>%
1785     summarise(
1786         RMSE = sqrt(mean((mx_obs-mx_proj)^2)),
1787         MAE = mean(abs(mx_obs-mx_proj)),
1788         MAPE = mean(abs((mx_obs-mx_proj)/mx_obs))
1789     )
1790
1791 case_ids_lc <- best_models_lc_final$id
1792
1793 #####
1794 #     Compute error     #
1795 #####
1796
1797 compute_errors_one_case_lc <- function(case_id) {
1798
1799     proj <- forecast_lc_case_full(
1800         case_id = case_id,
1801         dataset = dataset,
1802         dataset_train = dataset_train,
1803         dataset_test = dataset_test,
1804         cases_lc = cases_lc,
1805         series_lc = series_lc,
1806         fits_lc = fits_lc,
1807         best_order_lc = best_models_lc_final
1808     )
1809
1810     proj %>%
1811         filter(!is.na(mx_proj), mx_obs > 0) %>%
1812         summarise(
1813             RMSE = sqrt(mean((mx_obs - mx_proj)^2, na.rm = TRUE)),
1814             MAE = mean(abs(mx_obs - mx_proj), na.rm = TRUE),

```

```

1815     MAD   = safe_mad(mx_obs, mx_proj),
1816     MAPE = mean(abs((mx_obs - mx_proj) / mx_obs), na.rm = TRUE)
1817   ) %>%
1818   mutate(case_id = case_id)
1819 }
1820
1821 results_lc_errors <- map_dfr(case_ids_lc, compute_errors_one_case_lc) %>%
1822   mutate(
1823     Sex = substr(case_id, 1, 1),
1824     Period = sub("^([mf])_", "", case_id)
1825   ) %>%
1826   select(Sex, Period, RMSE, MAE, MAD, MAPE) %>%
1827   arrange(Sex, Period) %>%
1828   mutate(
1829     RMSE = round(RMSE, 6),
1830     MAE = round(MAE, 6),
1831     MAD = round(MAD, 6),
1832     MAPE = round(MAPE, 6)
1833   )
1834
1835
1836 print(results_lc_errors)
1837 kable(results_lc_errors, format = "latex")
1838
1839
1840 #####
1841 #      Ajustement du modèle de Seklecka/SPO      #
1842 #####
1843
1844
1845 # =====
1846 # 3) Préparation des données pour le fit (train)
1847 # =====
1848
1849 prepare_data_seklecka_mixed <- function(dataset, cx_tbl,
1850                                         sex_label = "f",
1851                                         periode = NULL,
1852                                         a = 50) {
1853
1854   # 3.1 Filtrage sexe + saison
1855   if (!is.null(periode)) {
1856     df <- dataset %>% filter(Sex == sex_label, saison == periode)
1857     period_key <- periode
1858   } else {
1859     df <- dataset %>% filter(Sex == sex_label)
1860     period_key <- "Annuel"
1861   }
1862
1863   # 3.2 Joindre c_x (Pearson) par âge
1864   cx_sub <- cx_tbl %>%
1865     filter(Sex == sex_label, Period == period_key) %>%
1866     select(Age, ct_x)
1867
1868   df <- df %>%
1869     left_join(cx_sub, by = "Age") %>%
1870     mutate(ct_x = ifelse(is.na(ct_x), 0, ct_x))
1871
1872   # 3.3 Temps hebdo proprement ordonné
1873   # t_id: "YYYY-WW" avec WW sur 2 digits
1874   df <- df %>%
1875     mutate(

```

```

1876     Year = as.integer(Year),
1877     Week = as.integer(Week),
1878     t_id = sprintf("%04d-%02d", Year, Week),
1879     age_num = as.numeric(Age),
1880     cohort = Year - age_num,      # cohorte annuelle
1881     D = as.integer(round(Deaths)),
1882     E = as.numeric(Exposure_week)
1883 ) %>%
1884 filter(is.finite(D), is.finite(E), E > 0, is.finite(age_num), is.finite(cohort))
1885
1886 # Ordre chronologique des semaines
1887 t_levels <- df %>%
1888   distinct(Year, Week, t_id) %>%
1889   arrange(Year, Week) %>%
1890   pull(t_id)
1891
1892 df <- df %>% mutate(t = factor(t_id, levels = t_levels))
1893
1894 # 3.4 Bases âge
1895 ages <- sort(unique(df$age_num))
1896 xbar <- mean(ages)
1897
1898 df <- df %>%
1899   mutate(
1900     bx2 = xbar - age_num,
1901     bx3 = pmax(xbar - age_num, 0),
1902     # g_x = (a-x)^+ + c_x (x-a)^+
1903     gx  = ifelse(age_num <= a, a - age_num, ct_x * (age_num - a)),
1904     gx2 = gx^2
1905   )
1906
1907 list(df = df, xbar = xbar, t_levels = t_levels)
1908 }
1909
1910 # =====
1911 # 4) Updates (backfitting)
1912 # =====
1913
1914 update_kappa_weekly <- function(df, alpha_tbl, gamma_tbl, kappa_year_tbl) {
1915
1916   df2 <- df %>%
1917     left_join(alpha_tbl, by = "age_num") %>%
1918     left_join(gamma_tbl, by = "cohort") %>%
1919     left_join(kappa_year_tbl, by = "Year") %>%
1920     mutate(
1921       alpha_x = ifelse(is.na(alpha_x), 0, alpha_x),
1922       gamma_c = ifelse(is.na(gamma_c), 0, gamma_c),
1923       k2 = ifelse(is.na(k2), 0, k2),
1924       k3 = ifelse(is.na(k3), 0, k3)
1925     )
1926
1927   split_idx <- split(seq_len(nrow(df2)), df2$t)
1928
1929   out <- lapply(names(split_idx), function(tt) {
1930     idx <- split_idx[[tt]]
1931     dft <- df2[idx, , drop = FALSE]
1932
1933     off <- log(dft$E) + dft$alpha_x + dft$gamma_c + dft$bx2 * dft$k2 + dft$bx3 * dft$k3
1934
1935     fit <- glm(D ~ 1 + gx2, family = poisson(), offset = off, data = dft)
1936

```

```

1937     b <- coef(fit)
1938     b <- c(b, rep(0, 2 - length(b)))
1939
1940     data.frame(t = tt, k1 = b[1], k4 = b[2])
1941   })
1942
1943   bind_rows(out)
1944 }
1945
1946 update_kappa_annual <- function(df, alpha_tbl, gamma_tbl, kappa_week_tbl) {
1947
1948   df2 <- df %>%
1949     left_join(alpha_tbl, by = "age_num") %>%
1950     left_join(gamma_tbl, by = "cohort") %>%
1951     left_join(kappa_week_tbl, by = "t") %>%
1952     mutate(
1953       alpha_x = ifelse(is.na(alpha_x), 0, alpha_x),
1954       gamma_c = ifelse(is.na(gamma_c), 0, gamma_c),
1955       k1 = ifelse(is.na(k1), 0, k1),
1956       k4 = ifelse(is.na(k4), 0, k4)
1957     )
1958
1959   split_idx <- split(seq_len(nrow(df2)), df2$Year)
1960
1961   out <- lapply(names(split_idx), function(yy) {
1962     idx <- split_idx[[yy]]
1963     dfy <- df2[idx, , drop = FALSE]
1964
1965     off <- log(dfy$E) + dfy$alpha_x + dfy$gamma_c + dfy$k1 + dfy$gx2 * dfy$k4
1966
1967     # Pas d'intercept : k1_t (hebdo) porte le niveau
1968     fit <- glm(D ~ 0 + bx2 + bx3, family = poisson(), offset = off, data = dfy)
1969
1970     b <- coef(fit)
1971     b <- c(b, rep(0, 2 - length(b)))
1972
1973     data.frame(Year = as.integer(yy), k2 = b[1], k3 = b[2])
1974   })
1975
1976   bind_rows(out) %>% arrange(Year)
1977 }
1978
1979 update_gamma_mixed <- function(df, alpha_tbl, kappa_week_tbl, kappa_year_tbl) {
1980
1981   df2 <- df %>%
1982     left_join(alpha_tbl, by = "age_num") %>%
1983     left_join(kappa_week_tbl, by = "t") %>%
1984     left_join(kappa_year_tbl, by = "Year") %>%
1985     mutate(
1986       alpha_x = ifelse(is.na(alpha_x), 0, alpha_x),
1987       k1 = ifelse(is.na(k1), 0, k1),
1988       k4 = ifelse(is.na(k4), 0, k4),
1989       k2 = ifelse(is.na(k2), 0, k2),
1990       k3 = ifelse(is.na(k3), 0, k3)
1991     )
1992
1993   # eta sans gamma
1994   eta_wo_gamma <- with(df2, alpha_x + k1 + bx2*k2 + bx3*k3 + gx2*k4)
1995
1996   gamma_new <- df2 %>%
1997     mutate(eta0 = eta_wo_gamma) %>%

```

```

1998   group_by(cohort) %>%
1999   summarise(
2000     num = sum(D, na.rm = TRUE),
2001     den = sum(E * exp(eta0), na.rm = TRUE),
2002     .groups = "drop"
2003   ) %>%
2004   mutate(gamma_c = log(pmax(num, 1e-30) / pmax(den, 1e-30))) %>%
2005   arrange(cohort)
2006
2007   gamma_new
2008 }
2009
2010 update_alpha_mixed <- function(df, kappa_week_tbl, kappa_year_tbl, gamma_tbl) {
2011
2012   df2 <- df %>%
2013     left_join(kappa_week_tbl, by = "t") %>%
2014     left_join(kappa_year_tbl, by = "Year") %>%
2015     left_join(gamma_tbl, by = "cohort") %>%
2016     mutate(
2017       gamma_c = ifelse(is.na(gamma_c), 0, gamma_c),
2018       k1 = ifelse(is.na(k1), 0, k1),
2019       k4 = ifelse(is.na(k4), 0, k4),
2020       k2 = ifelse(is.na(k2), 0, k2),
2021       k3 = ifelse(is.na(k3), 0, k3)
2022     )
2023
2024   # eta sans alpha
2025   eta_wo_alpha <- with(df2, k1 + bx2*k2 + bx3*k3 + gx2*k4 + gamma_c)
2026
2027   alpha_new <- df2 %>%
2028     mutate(eta0 = eta_wo_alpha) %>%
2029     group_by(age_num) %>%
2030     summarise(
2031       num = sum(D, na.rm = TRUE),
2032       den = sum(E * exp(eta0), na.rm = TRUE),
2033       .groups = "drop"
2034     ) %>%
2035     mutate(alpha_x = log(pmax(num, 1e-30) / pmax(den, 1e-30))) %>%
2036     arrange(age_num)
2037
2038   alpha_new
2039 }
2040
2041 normalize_params_mixed <- function(alpha_tbl, kappa_week_tbl, kappa_year_tbl, gamma_tbl) {
2042
2043   # Centrer k1 et pousser le shift dans alpha
2044   tmp <- center_col(kappa_week_tbl, "k1")
2045   kappa_week_tbl <- tmp$df
2046   alpha_tbl$alpha_x <- alpha_tbl$alpha_x + tmp$shift
2047
2048   # Centrer k4 (contrainte sum_t k4 = 0)
2049   kappa_week_tbl$k4 <- kappa_week_tbl$k4 - safe_mean(kappa_week_tbl$k4)
2050
2051   # Centrer k2,k3 (stabilité / identifiabilité dans ta version mixte)
2052   kappa_year_tbl$k2 <- kappa_year_tbl$k2 - safe_mean(kappa_year_tbl$k2)
2053   kappa_year_tbl$k3 <- kappa_year_tbl$k3 - safe_mean(kappa_year_tbl$k3)
2054
2055   # Contrainte gamma ordre 2
2056   gamma_tbl <- detrend_gamma_poly2(gamma_tbl)
2057
2058   list(alpha_tbl = alpha_tbl,

```

```

2059     kappa_week_tbl = kappa_week_tbl,
2060     kappa_year_tbl = kappa_year_tbl,
2061     gamma_tbl = gamma_tbl)
2062 }
2063
2064 # =====
2065 # 5) Fit complet : backfitting + logLik + convergence
2066 # =====
2067
2068 fit_seklecka_backfit_mixed_best <- function(dataset_train, cx_tbl,
2069                                           sex_label = "f",
2070                                           periode = NULL,
2071                                           a = 50,
2072                                           max_iter = 50,
2073                                           tol_ll = 1e-7,
2074                                           tol_par = 1e-6,
2075                                           verbose = TRUE) {
2076
2077   prep <- prepare_data_seklecka_mixed(dataset_train, cx_tbl,
2078                                       sex_label = sex_label,
2079                                       periode = periode,
2080                                       a = a)
2081
2082   df <- prep$df
2083
2084   # Initialisations
2085   alpha_tbl <- df %>%
2086     group_by(age_num) %>%
2087     summarise(alpha_x = log(sum(D, na.rm = TRUE) / sum(E, na.rm = TRUE)), .groups = "drop") %>%
2088     arrange(age_num)
2089
2090   gamma_tbl <- tibble(cohort = sort(unique(df$cohort)), gamma_c = 0)
2091
2092   kappa_week_tbl <- df %>% distinct(t) %>% mutate(k1 = 0, k4 = 0)
2093   kappa_year_tbl <- df %>% distinct(Year) %>% arrange(Year) %>% mutate(k2 = 0, k3 = 0)
2094
2095   # logLik initial
2096   eta0 <- df %>%
2097     left_join(alpha_tbl, by="age_num") %>%
2098     left_join(gamma_tbl, by="cohort") %>%
2099     left_join(kappa_week_tbl, by="t") %>%
2100     left_join(kappa_year_tbl, by="Year") %>%
2101     mutate(
2102       alpha_x = ifelse(is.na(alpha_x), 0, alpha_x),
2103       gamma_c = ifelse(is.na(gamma_c), 0, gamma_c),
2104       k1 = ifelse(is.na(k1), 0, k1),
2105       k4 = ifelse(is.na(k4), 0, k4),
2106       k2 = ifelse(is.na(k2), 0, k2),
2107       k3 = ifelse(is.na(k3), 0, k3),
2108       eta = alpha_x + gamma_c + k1 + bx2*k2 + bx3*k3 + gx2*k4,
2109       mu = E * exp(eta)
2110     )
2111   ll_old <- pois_loglik(eta0$D, eta0$mu)
2112
2113   trace <- tibble(iter = 0, logLik = ll_old, dLL = NA_real_)
2114
2115   for (it in 1:max_iter) {
2116
2117     kW_new <- update_kappa_weekly(df, alpha_tbl, gamma_tbl, kappa_year_tbl)
2118     kY_new <- update_kappa_annual(df, alpha_tbl, gamma_tbl, kW_new)
2119     g_new <- update_gamma_mixed(df, alpha_tbl, kW_new, kY_new)
2120     a_new <- update_alpha_mixed(df, kW_new, kY_new, g_new)

```



```

2120
2121 tmp <- normalize_params_mixed(a_new, kW_new, kY_new, g_new)
2122 alpha_new <- tmp$alpha_tbl
2123 kW_new <- tmp$kappa_week_tbl
2124 kY_new <- tmp$kappa_year_tbl
2125 gamma_new <- tmp$gamma_tbl
2126
2127 # logLik nouveau
2128 etal <- df %>%
2129   left_join(alpha_new, by="age_num") %>%
2130   left_join(gamma_new, by="cohort") %>%
2131   left_join(kW_new, by="t") %>%
2132   left_join(kY_new, by="Year") %>%
2133   mutate(
2134     alpha_x = ifelse(is.na(alpha_x),0,alpha_x),
2135     gamma_c = ifelse(is.na(gamma_c),0,gamma_c),
2136     k1 = ifelse(is.na(k1),0,k1),
2137     k4 = ifelse(is.na(k4),0,k4),
2138     k2 = ifelse(is.na(k2),0,k2),
2139     k3 = ifelse(is.na(k3),0,k3),
2140     eta = alpha_x + gamma_c + k1 + bx2*k2 + bx3*k3 + gx2*k4,
2141     mu = E * exp(eta)
2142   )
2143 ll_new <- pois_loglik(etal$D, etal$mu)
2144 dLL <- ll_new - ll_old
2145
2146 if (verbose) {
2147   cat(sprintf("iter=%d | logLik=%.3f | dLL=%.6g\n",
2148     it, ll_new, dLL))
2149 }
2150
2151 # =====
2152 # CONDITION D'ARRET IMPORTANTE
2153 # =====
2154
2155 if (!is.finite(dLL) || dLL < 0) {
2156   if (verbose) cat("LogLik decreased. Stop at previous iteration.\n")
2157   break
2158 }
2159
2160 # Sinon on accepte la mise à jour
2161 alpha_tbl <- alpha_new
2162 gamma_tbl <- gamma_new
2163 kappa_week_tbl <- kW_new
2164 kappa_year_tbl <- kY_new
2165 ll_old <- ll_new
2166
2167 # Critère secondaire : très petite amélioration
2168 if (abs(dLL) < tol_ll) {
2169   if (verbose) cat("Convergence reached (small dLL).\n")
2170   break
2171 }
2172 }
2173
2174 list(
2175   alpha = alpha_tbl,
2176   kappa_weekly = kappa_week_tbl, # k1,k4 hebdo
2177   kappa_annual = kappa_year_tbl, # k2,k3 annuel
2178   gamma = gamma_tbl,
2179   data = df,
2180   trace = trace,

```

```

2181     meta = list(sex = sex_label,
2182                 periode = ifelse(is.null(periode), "Annuel", periode),
2183                 a = a)
2184   )
2185 }
2186
2187 # =====
2188 # 6) Exemple d'utilisation : TRAIN 2000-2016
2189 # =====
2190
2191 # Fit femmes
2192 res_f_ann <- fit_seklecka_backfit_mixed_best(dataset_train, cx_tbl, sex_label="f", periode=NULL, a=40)
2193 res_f_cha <- fit_seklecka_backfit_mixed_best(dataset_train, cx_tbl, sex_label="f", periode="chaud", a=60)
2194 res_f_fro <- fit_seklecka_backfit_mixed_best(dataset_train, cx_tbl, sex_label="f", periode="froid", a=40)
2195
2196 # Fit hommes
2197 res_m_ann <- fit_seklecka_backfit_mixed_best(dataset_train, cx_tbl, sex_label="m", periode=NULL, a=40)
2198 res_m_cha <- fit_seklecka_backfit_mixed_best(dataset_train, cx_tbl, sex_label="m", periode="chaud", a=60)
2199 res_m_fro <- fit_seklecka_backfit_mixed_best(dataset_train, cx_tbl, sex_label="m", periode="froid", a=60)
2200
2201
2202 par(mfrow=c(2,2))
2203 ## plot
2204
2205 plot_kappa1_weekly_pub(res_f_fro, year_tick_by = 1)
2206 plot_kappa2_annual_pub(res_f_fro, year_tick_by = 1)
2207 plot_kappa3_annual_pub(res_f_fro, year_tick_by = 1)
2208 plot_kappa4_weekly_pub(res_f_fro, year_tick_by = 1)
2209
2210 plot_alpha_pub(res_m_ann)
2211
2212
2213
2214
2215 #####
2216 #   Choix du a   ####
2217 #####
2218
2219 # =====
2220 # 1) Evaluer un fit Seklecka pour une valeur donnée de a
2221 # =====
2222
2223 evaluate_seklecka_fit <- function(fit_obj) {
2224
2225   df <- fit_obj$data
2226
2227   alpha_tbl <- fit_obj$alpha
2228   gamma_tbl <- fit_obj$gamma
2229   kW_tbl    <- fit_obj$kappa_weekly
2230   kY_tbl    <- fit_obj$kappa_annual
2231
2232   # -----
2233   # Reconstruire eta, mu, mx_obs, mx_hat
2234   # -----
2235   eval_df <- df %>%
2236     left_join(alpha_tbl, by = "age_num") %>%
2237     left_join(gamma_tbl, by = "cohort") %>%
2238     left_join(kW_tbl, by = "t") %>%
2239     left_join(kY_tbl, by = "Year") %>%
2240     mutate(
2241       alpha_x = ifelse(is.na(alpha_x), 0, alpha_x),

```

```

2242     gamma_c = ifelse(is.na(gamma_c), 0, gamma_c),
2243     k1 = ifelse(is.na(k1), 0, k1),
2244     k4 = ifelse(is.na(k4), 0, k4),
2245     k2 = ifelse(is.na(k2), 0, k2),
2246     k3 = ifelse(is.na(k3), 0, k3),
2247     eta = alpha_x + gamma_c + k1 + bx2 * k2 + bx3 * k3 + gx2 * k4,
2248     mu = E * exp(eta),
2249     mx_obs = ifelse(E > 0, D / E, NA_real_),
2250     mx_hat = exp(eta)
2251   )
2252
2253   # -----
2254   # Cellules valides
2255   # -----
2256   ok <- is.finite(eval_df$mx_obs) & is.finite(eval_df$mx_hat) & (eval_df$mx_obs > 0) & (eval_df$E > 0)
2257
2258   # -----
2259   # MAPE
2260   # -----
2261   MAPE <- mean(abs(eval_df$mx_hat[ok] - eval_df$mx_obs[ok]) / eval_df$mx_obs[ok], na.rm = TRUE) * 100
2262
2263   # -----
2264   # logLik Poisson
2265   # -----
2266   mu_safe <- pmax(eval_df$mu, 1e-12)
2267   logLik_val <- sum(eval_df$D * log(mu_safe) - mu_safe - lgamma(eval_df$D + 1), na.rm = TRUE)
2268
2269   # -----
2270   # Nombre de paramètres k
2271   # alpha_x : un par âge
2272   # gamma_c : un par cohorte
2273   # k1, k4 : hebdomadaire
2274   # k2, k3 : annuel
2275   # -----
2276   k <- nrow(alpha_tbl) + nrow(gamma_tbl) + 2 * nrow(kW_tbl) + 2 * nrow(kY_tbl)
2277
2278   # -----
2279   # Taille d'échantillon
2280   # -----
2281   n <- sum(ok)
2282
2283   # -----
2284   # AIC / BIC
2285   # -----
2286   AIC <- 2 * k - 2 * logLik_val
2287   BIC <- log(n) * k - 2 * logLik_val
2288
2289   tibble(
2290     MAPE = MAPE,
2291     logLik = logLik_val,
2292     k = k,
2293     n = n,
2294     AIC = AIC,
2295     BIC = BIC
2296   )
2297 }
2298
2299
2300 # =====
2301 # 2) Simuler plusieurs valeurs de a pour un cas donné
2302 # =====

```

```

2303
2304 simulate_a_seklecka <- function(dataset_train, cx_tbl,
2305                               sex_label,
2306                               periode = NULL,
2307                               a_grid = 40:60,
2308                               max_iter = 50,
2309                               tol_ll = 1e-7,
2310                               tol_par = 1e-6,
2311                               verbose_fit = FALSE) {
2312
2313   results <- map_dfr(a_grid, function(a_val) {
2314
2315     cat("Sex =", sex_label, "| Period =", ifelse(is.null(periode), "Annuel", periode), "| a =", a_val, "\n")
2316
2317     fit_obj <- fit_seklecka_backfit_mixed_best(
2318       dataset_train = dataset_train,
2319       cx_tbl = cx_tbl,
2320       sex_label = sex_label,
2321       periode = periode,
2322       a = a_val,
2323       max_iter = max_iter,
2324       tol_ll = tol_ll,
2325       tol_par = tol_par,
2326       verbose = verbose_fit
2327     )
2328
2329     crit <- evaluate_seklecka_fit(fit_obj)
2330
2331     crit %>%
2332       mutate(
2333         Sex = sex_label,
2334         Period = ifelse(is.null(periode), "Annuel", periode),
2335         a = a_val
2336       ) %>%
2337       select(Sex, Period, a, n, k, logLik, MAPE, AIC, BIC)
2338   })
2339
2340   results
2341 }
2342
2343 # =====
2344 # 3) Simulation de a pour tous les cas
2345 # =====
2346
2347 a_grid <- 40:60
2348
2349 res_a_f_ann <- simulate_a_seklecka(dataset_train, cx_tbl, sex_label = "f", periode = NULL, a_grid = a_grid)
2350 res_a_f_cha <- simulate_a_seklecka(dataset_train, cx_tbl, sex_label = "f", periode = "chaud", a_grid = a_grid)
2351 res_a_f_fro <- simulate_a_seklecka(dataset_train, cx_tbl, sex_label = "f", periode = "froid", a_grid = a_grid)
2352
2353 res_a_m_ann <- simulate_a_seklecka(dataset_train, cx_tbl, sex_label = "m", periode = NULL, a_grid = a_grid)
2354 res_a_m_cha <- simulate_a_seklecka(dataset_train, cx_tbl, sex_label = "m", periode = "chaud", a_grid = a_grid)
2355 res_a_m_fro <- simulate_a_seklecka(dataset_train, cx_tbl, sex_label = "m", periode = "froid", a_grid = a_grid)
2356
2357 results_a_all <- bind_rows(
2358   res_a_f_ann, res_a_f_cha, res_a_f_fro,
2359   res_a_m_ann, res_a_m_cha, res_a_m_fro
2360 )
2361
2362 # =====
2363 # 4) Choix du meilleur a par cas

```

```

2364 # =====
2365
2366 best_a_by_case <- results_a_all %>%
2367   group_by(Sex, Period) %>%
2368   mutate(
2369     rank_MAPE = rank(MAPE, ties.method = "first"),
2370     rank_AIC  = rank(AIC,  ties.method = "first"),
2371     rank_BIC  = rank(BIC,  ties.method = "first"),
2372     score_total = rank_MAPE + rank_AIC + rank_BIC
2373   ) %>%
2374   arrange(score_total, MAPE, AIC, BIC) %>%
2375   slice(1) %>%
2376   ungroup() %>%
2377   select(Sex, Period, a, MAPE, AIC, BIC, score_total)
2378
2379 best_a_by_case
2380
2381
2382 # =====
2383 # 5) Choix d'un a global
2384 # =====
2385
2386 best_a_global <- results_a_all %>%
2387   group_by(a) %>%
2388   summarise(
2389     mean_MAPE = mean(MAPE, na.rm = TRUE),
2390     mean_AIC  = mean(AIC,  na.rm = TRUE),
2391     mean_BIC  = mean(BIC,  na.rm = TRUE),
2392     .groups = "drop"
2393   ) %>%
2394   mutate(
2395     rank_MAPE = rank(mean_MAPE, ties.method = "first"),
2396     rank_AIC  = rank(mean_AIC,  ties.method = "first"),
2397     rank_BIC  = rank(mean_BIC,  ties.method = "first"),
2398     score_total = rank_MAPE + rank_AIC + rank_BIC
2399   ) %>%
2400   arrange(score_total, mean_MAPE, mean_AIC, mean_BIC) %>%
2401   slice(1)
2402
2403 best_a_global
2404
2405
2406 # =====
2407 # EVALUATION COMPLETE DU MODELE DE SEKLECKA
2408 # - décès prédits
2409 # - taux de mortalité prédits
2410 # - erreurs : MAD, RMSE, MAPE, AIC, BIC, Deviance
2411 # - tableau final par sexe et période
2412 # =====
2413
2414 # Nombre "effectif" de paramètres tenant compte des contraintes
2415 count_params_seklecka <- function(fit_obj) {
2416   n_alpha <- nrow(fit_obj$alpha)
2417   n_kw    <- nrow(fit_obj$kappa_weekly) # k1, k4
2418   n_ky    <- nrow(fit_obj$kappa_annual)  # k2, k3
2419   n_gamma <- nrow(fit_obj$gamma)
2420
2421   # Contraintes d'identifiabilité appliquées dans ton code :
2422   # - k1 centré      -> -1
2423   # - k4 centré      -> -1
2424   # - k2 centré      -> -1

```

```

2425 # - k3 centré          -> -1
2426 # - gamma detrend ordre 2 -> -3
2427 k_eff <- n_alpha + (2 * n_kw - 2) + (2 * n_ky - 2) + (n_gamma - 3)
2428
2429 max(k_eff, 1)
2430 }
2431
2432 # -----
2433 # 2) Fonction principale : prédictions + métriques pour 1 fit
2434 # -----
2435
2436 evaluate_one_seklecka_fit <- function(fit_obj) {
2437
2438   df <- fit_obj$data
2439
2440   # Jointure des paramètres estimés
2441   df_eval <- df %>%
2442     left_join(fit_obj$alpha,      by = "age_num") %>%
2443     left_join(fit_obj$gamma,      by = "cohort") %>%
2444     left_join(fit_obj$kappa_weekly, by = "t") %>%
2445     left_join(fit_obj$kappa_annual, by = "Year") %>%
2446     mutate(
2447       alpha_x = coalesce(alpha_x, 0),
2448       gamma_c = coalesce(gamma_c, 0),
2449       k1      = coalesce(k1, 0),
2450       k4      = coalesce(k4, 0),
2451       k2      = coalesce(k2, 0),
2452       k3      = coalesce(k3, 0),
2453
2454       # Prédicteur linéaire du modèle Seklecka
2455       eta = alpha_x + gamma_c + k1 + bx2 * k2 + bx3 * k3 + gx2 * k4,
2456
2457       # Quantités prédites
2458       D_hat = E * exp(eta),
2459       mx_hat = D_hat / E,
2460
2461       # Quantités observées
2462       D_obs = D,
2463       mx_obs = D / E,
2464
2465       # Infos utiles pour sortie
2466       Sex = fit_obj$meta$sex,
2467       Period = fit_obj$meta$periode
2468     )
2469
2470   # Log-vraisemblance, AIC, BIC, Deviance
2471   logLik_val <- pois_loglik(df_eval$D_obs, df_eval$D_hat)
2472   dev_val <- pois_deviance(df_eval$D_obs, df_eval$D_hat)
2473   k_param <- count_params_seklecka(fit_obj)
2474   n_obs <- nrow(df_eval)
2475
2476   AIC_val <- -2 * logLik_val + 2 * k_param
2477   BIC_val <- -2 * logLik_val + log(n_obs) * k_param
2478
2479   # Tableau résumé des métriques
2480   summary_tbl <- tibble(
2481     Sex = unique(df_eval$Sex),
2482     Period = unique(df_eval$Period),
2483
2484     MAD_Deaths = safe_mad(df_eval$D_obs, df_eval$D_hat),
2485     RMSE_Deaths = safe_rmse(df_eval$D_obs, df_eval$D_hat),

```

```

2486     MAPE_Deaths    = safe_mape(df_eval$D_obs, df_eval$D_hat),
2487
2488     MAD_Rate       = safe_mad(df_eval$mx_obs, df_eval$mx_hat),
2489     RMSE_Rate      = safe_rmse(df_eval$mx_obs, df_eval$mx_hat),
2490     MAPE_Rate      = safe_mape(df_eval$mx_obs, df_eval$mx_hat),
2491
2492     LogLik         = logLik_val,
2493     AIC            = AIC_val,
2494     BIC            = BIC_val,
2495     Deviance       = dev_val,
2496     Nb_obs         = n_obs,
2497     Nb_param       = k_param
2498   )
2499
2500   # Tableau détaillé des valeurs observées/prédites
2501   pred_tbl <- df_eval %>%
2502     mutate(
2503       date = as.Date(date)
2504     ) %>%
2505     transmute(
2506       Sex,
2507       Period,
2508       Year,
2509       Week,
2510       date,
2511       t_id,
2512       Age = age_num,
2513       cohort,
2514       Exposure = E,
2515       Deaths_obs = D_obs,
2516       Deaths_hat = D_hat,
2517       mx_obs,
2518       mx_hat,
2519       eta
2520     )
2521
2522   list(
2523     summary = summary_tbl,
2524     predictions = pred_tbl
2525   )
2526 }
2527
2528 # -----
2529 # 3) Appliquer à tous les fits
2530 # -----
2531
2532 sek_fits <- list(
2533   f_annuel = res_f_ann,
2534   f_chaud  = res_f_cha,
2535   f_froid  = res_f_fro,
2536   m_annuel = res_m_ann,
2537   m_chaud  = res_m_cha,
2538   m_froid  = res_m_fro
2539 )
2540
2541 sek_results <- purrr::map(sek_fits, evaluate_one_seklecka_fit)
2542
2543 sek_summary_table <- purrr::map_dfr(sek_results, "summary") %>%
2544   mutate(
2545     Period = factor(Period, levels = c("Annuel", "froid", "chaud")),
2546     Sex     = factor(Sex, levels = c("f", "m"))

```

```

2547   ) %>%
2548   arrange(Sex, Period)
2549
2550 sek_predictions_table <- purrr::map_dfr(sek_results, "predictions") %>%
2551   mutate(
2552     Period = factor(Period, levels = c("Annuel", "froid", "chaud")),
2553     Sex     = factor(Sex, levels = c("f", "m"))
2554   ) %>%
2555   arrange(Sex, Period, Year, Week, Age, date)
2556   # -----
2557   # 4) Affichage
2558   # -----
2559
2560 print(sek_summary_table)
2561
2562 kable(sek_summary_table, format = "latex", caption="Qualité d'ajustement du modèle SP0", label="tab3")
2563 # Si tu veux voir les premières lignes du détail :
2564 print(head(sek_predictions_table))
2565
2566 # =====
2567 # DECES OBSERVES VS PREDITS PAR SEMAINE
2568 # GRAPHIQUES SEPARES : Annuel / froid / chaud
2569 # =====
2570
2571 plot_sek_rates_by_age_date <- function(pred_table,
2572                                       sex_label,
2573                                       period_label,
2574                                       ages_to_plot = c(20, 45, 70, 85)) {
2575
2576   df_plot <- pred_table %>%
2577     filter(Sex == sex_label, Period == period_label, Age %in% ages_to_plot) %>%
2578     select(date, Age, mx_obs, mx_hat) %>%
2579     pivot_longer(
2580       cols = c(mx_obs, mx_hat),
2581       names_to = "Series",
2582       values_to = "mx"
2583     ) %>%
2584     mutate(
2585       Series = recode(
2586         Series,
2587         mx_obs = "Observé",
2588         mx_hat = "Prédit"
2589       )
2590     )
2591
2592   ggplot(df_plot, aes(x = date, y = mx, colour = Series, linetype = Series)) +
2593     geom_line(linewidth = 0.8, na.rm = TRUE) +
2594     facet_wrap(~ Age, scales = "free_y") +
2595     scale_color_manual(
2596       values = c("Observé" = "blue", "Prédit" = "red")
2597     ) +
2598     scale_linetype_manual(
2599       values = c("Observé" = "solid", "Prédit" = "solid")
2600     ) +
2601     theme_minimal() +
2602     labs(
2603       title = paste0("SP0 - ", sex_label, " - ", period_label),
2604       x = "Date",
2605       y = "Taux de mortalité",
2606       colour = "",
2607       linetype = ""

```



```
2608     ) +
2609     theme(panel.grid = element_line(color = "grey80"))
2610 }
2611
2612 plot_sek_rates_by_age_date(
2613   sek_predictions_table,
2614   sex_label = "f",
2615   period_label = "Annuel",
2616   ages_to_plot = c(20, 45, 70, 85)
2617 )
2618
2619
2620 # =====
2621 # BOX-JENKINS SUR LES PARAMETRES DU MODELE DE SEKLECKA
2622 # Version simple et complète
2623 # =====
2624 # Etapes :
2625 # 1) Construction des séries
2626 # 2) Visualisation : série brute + ACF/PACF
2627 # 3) Désaisonnalisation si nécessaire + visualisation
2628 # 4) Stationnarisation avec d = 1 + visualisation
2629 # 5) Recherche automatique des paramètres
2630 # 6) Affichage des 2 meilleurs modèles
2631 # =====
2632
2633
2634 # -----
2635 # 1) HYPOTHESE : les 6 fits Seklecka existent déjà
2636 # -----
2637 sek_fits <- list(
2638   f_annuel = res_f_ann,
2639   f_chaud  = res_f_cha,
2640   f_froid  = res_f_fro,
2641   m_annuel = res_m_ann,
2642   m_chaud  = res_m_cha,
2643   m_froid  = res_m_fro
2644 )
2645
2646 # -----
2647 # 2) OUTILS UTILITAIRES
2648 # -----
2649 coef_pvalues <- function(coef, var_diag) {
2650   z <- coef / sqrt(var_diag)
2651   2 * (1 - pnorm(abs(z)))
2652 }
2653
2654 get_saison_length <- function(case_id) {
2655   if (grepl("annuel", case_id)) return(52L)
2656   if (grepl("froid", case_id)) return(30L)
2657   if (grepl("chaud", case_id)) return(22L)
2658   stop("Impossible de déterminer la saison pour : ", case_id)
2659 }
2660
2661 stabilize_series <- function(x_ts) {
2662   x <- as.numeric(x_ts)
2663
2664   if (all(is.finite(x)) && all(x > 0)) {
2665     list(ts = log(x_ts), transform = "log")
2666   } else {
2667     list(ts = x_ts, transform = "none")
2668   }
2669 }
```

```
2669 }
2670
2671 choose_D <- function(x_ts) {
2672   tryCatch(
2673     forecast::nsdiffs(x_ts),
2674     error = function(e) 0L
2675   )
2676 }
2677
2678 # -----
2679 # 3) CONSTRUCTION DES SERIES TEMPORELLES
2680 # -----
2681
2682 # ----- Séries hebdomadaires : k1 et k4
2683 build_weekly_series <- function(res_fit, param_name, case_id) {
2684
2685   s <- get_saison_length(case_id)
2686
2687   dfw <- res_fit$kappa_weekly %>%
2688     mutate(
2689       t = as.character(t),
2690       Year = as.integer(substr(t, 1, 4)),
2691       Week = as.integer(substr(t, 6, 7))
2692     ) %>%
2693     arrange(Year, Week)
2694
2695   x <- dfw[[param_name]]
2696
2697   ts(
2698     x,
2699     start = c(min(dfw$Year, na.rm = TRUE), 1),
2700     frequency = s
2701   )
2702 }
2703
2704 # ----- Séries annuelles : k2 et k3
2705 build_annual_series <- function(res_fit, param_name) {
2706
2707   dfy <- res_fit$kappa_annual %>%
2708     arrange(Year)
2709
2710   x <- dfy[[param_name]]
2711
2712   ts(
2713     x,
2714     start = min(dfy$Year, na.rm = TRUE),
2715     frequency = 1
2716   )
2717 }
2718
2719 # -----
2720 # 4) FONCTIONS DE VISUALISATION
2721 # -----
2722 plot_series_acf_pacf <- function(x_ts, title_prefix = "", lag_max = 60) {
2723
2724   plot(
2725     x_ts,
2726     type = "l",
2727     pch = 16,
2728     cex = 0.5,
```

```

2730     lwd = 1.2,
2731     main = paste("Série -", title_prefix),
2732     xlab = "Temps",
2733     ylab = ""
2734 )
2735 abline(h = mean(x_ts, na.rm = TRUE), col = "red", lty = 2)
2736
2737 acf(
2738   as.numeric(x_ts),
2739   lag.max = lag_max,
2740   main = paste("ACF -", title_prefix),
2741   na.action = na.pass
2742 )
2743
2744 pacf(
2745   as.numeric(x_ts),
2746   lag.max = lag_max,
2747   main = paste("PACF -", title_prefix),
2748   na.action = na.pass
2749 )
2750
2751 }
2752
2753 plot_seasonal_cycle <- function(x_ts, s, main_title = "Cycle saisonnier") {
2754   n <- length(x_ts)
2755   n_blocks <- floor(n / s)
2756
2757   if (n_blocks < 2) {
2758     warning("Pas assez de cycles complets pour tracer le cycle saisonnier.")
2759     return(invisible(NULL))
2760   }
2761
2762   plot(
2763     x_ts[1:s],
2764     type = "l",
2765     ylim = c(min(x_ts, na.rm = TRUE), max(x_ts, na.rm = TRUE)),
2766     main = main_title,
2767     xlab = "Index dans le cycle",
2768     ylab = "Valeur"
2769   )
2770
2771   for (i in 2:n_blocks) {
2772     lines(x_ts[s * (i - 1) + (1:s)], col = i)
2773   }
2774 }
2775
2776 # -----
2777 # 5) PREPARATION D'UNE SERIE
2778 # -----
2779 # On fixe toujours d = 1
2780 # D est automatique seulement pour les séries hebdomadaires
2781 # -----
2782 prepare_bj_series <- function(x_ts, is_weekly = TRUE) {
2783
2784   stab_obj <- stabilize_series(x_ts)
2785   ts_stab <- stab_obj$ts
2786   transform_used <- stab_obj$transform
2787
2788   if (is_weekly) {
2789     D <- choose_D(ts_stab)
2790   } else {

```

```

2791     D <- 0L
2792   }
2793
2794   ts_deseason <- if (D > 0) {
2795     diff(ts_stab, lag = frequency(ts_stab), differences = D)
2796   } else {
2797     ts_stab
2798   }
2799
2800   d <- 1L
2801   ts_stationary <- diff(ts_deseason, differences = d)
2802
2803   list(
2804     ts_raw = x_ts,
2805     ts_stab = ts_stab,
2806     transform_used = transform_used,
2807     D = D,
2808     d = d,
2809     ts_deseason = ts_deseason,
2810     ts_stationary = ts_stationary
2811   )
2812 }
2813
2814 # -----
2815 # 6) DIAGNOSTIC D'UNE SERIE
2816 # -----
2817 diagnose_one_series <- function(x_ts,
2818                                series_name = "Série",
2819                                is_weekly = TRUE,
2820                                lag_raw = 120,
2821                                lag_clean = 120) {
2822
2823   prep <- prepare_bj_series(
2824     x_ts = x_ts,
2825     is_weekly = is_weekly
2826   )
2827
2828   cat("\n===== \n")
2829   cat("Diagnostic Box-Jenkins :", series_name, "\n")
2830   cat("Transformation :", prep$transform_used, "\n")
2831   cat("D choisi automatiquement :", prep$D, "\n")
2832   cat("d fixé à :", prep$d, "\n")
2833   cat("===== \n")
2834
2835   # 1) Série brute / stabilisée
2836   plot_series_acf_pacf(
2837     prep$ts_stab,
2838     title_prefix = paste(series_name, "- brute"),
2839     lag_max = min(lag_raw, length(prep$ts_stab) - 1)
2840   )
2841
2842   # Cycle saisonnier si hebdomadaire
2843   if (is_weekly) {
2844     plot_seasonal_cycle(
2845       prep$ts_stab,
2846       s = frequency(prep$ts_stab),
2847       main_title = paste("Cycle saisonnier -", series_name)
2848     )
2849   }
2850
2851   # 2) Série désaisonnalisée

```

```
2852 plot_series_acf_pacf(
2853   prep$ts_deseason,
2854   title_prefix = paste(series_name, "- désaisonnalisée"),
2855   lag_max = min(lag_clean, length(prep$ts_deseason) - 1)
2856 )
2857
2858 # 3) Série stationnaire
2859 plot_series_acf_pacf(
2860   prep$ts_stationary,
2861   title_prefix = paste(series_name, "- stationnaire"),
2862   lag_max = min(lag_clean, length(prep$ts_stationary) - 1)
2863 )
2864
2865 invisible(prep)
2866 }
2867
2868 # -----
2869 # 7) DIAGNOSTIC D'UN CAS COMPLET
2870 # -----
2871 plot_full_diagnostics_sek_case <- function(res_fit, case_id) {
2872
2873   # k1
2874   k1_ts <- build_weekly_series(res_fit, "k1", case_id)
2875   diagnose_one_series(
2876     x_ts = k1_ts,
2877     series_name = paste(case_id, "- k1"),
2878     is_weekly = TRUE,
2879     lag_raw = 120,
2880     lag_clean = 120
2881   )
2882
2883   # k4
2884   k4_ts <- build_weekly_series(res_fit, "k4", case_id)
2885   diagnose_one_series(
2886     x_ts = k4_ts,
2887     series_name = paste(case_id, "- k4"),
2888     is_weekly = TRUE,
2889     lag_raw = 120,
2890     lag_clean = 120
2891   )
2892
2893   # k2
2894   k2_ts <- build_annual_series(res_fit, "k2")
2895   diagnose_one_series(
2896     x_ts = k2_ts,
2897     series_name = paste(case_id, "- k2"),
2898     is_weekly = FALSE,
2899     lag_raw = 10,
2900     lag_clean = 10
2901   )
2902
2903   # k3
2904   k3_ts <- build_annual_series(res_fit, "k3")
2905   diagnose_one_series(
2906     x_ts = k3_ts,
2907     series_name = paste(case_id, "- k3"),
2908     is_weekly = FALSE,
2909     lag_raw = 10,
2910     lag_clean = 10
2911   )
2912 }
```

```
2913
2914 # -----
2915 # 8) GRILLES D'ESTIMATION
2916 # -----
2917
2918 # ----- SARIMA pour séries hebdomadaires
2919 fit_sarima_grid <- function(x_ts, d, D, s,
2920                             p_max = 2, q_max = 2,
2921                             P_max = 2, Q_max = 2,
2922                             include_mean = FALSE) {
2923
2924   grid <- expand.grid(
2925     p = 0:p_max,
2926     q = 0:q_max,
2927     P = 0:P_max,
2928     Q = 0:Q_max
2929   )
2930
2931   results <- vector("list", nrow(grid))
2932
2933   for (i in seq_len(nrow(grid))) {
2934
2935     p <- grid$p[i]
2936     q <- grid$q[i]
2937     P <- grid$P[i]
2938     Q <- grid$Q[i]
2939
2940     fit_try <- tryCatch(
2941       arima(
2942         x = x_ts,
2943         order = c(p, d, q),
2944         seasonal = list(order = c(P, D, Q), period = s),
2945         include.mean = include_mean,
2946         method = "ML"
2947       ),
2948       error = function(e) NULL
2949     )
2950
2951     if (!is.null(fit_try)) {
2952
2953       coef_vec <- fit_try$coef
2954       var_diag <- tryCatch(diag(fit_try$var.coef), error = function(e) numeric(0))
2955
2956       if (length(coef_vec) > 0 && length(var_diag) == length(coef_vec)) {
2957         pvals <- coef_pvalues(coef_vec, var_diag)
2958       } else {
2959         pvals <- rep(NA_real_, length(coef_vec))
2960       }
2961
2962       lb_lag <- min(
2963         floor(sqrt(length(residuals(fit_try)))),
2964         length(residuals(fit_try)) - 1
2965       )
2966
2967       lb_pvalue <- tryCatch(
2968         Box.test(residuals(fit_try), lag = lb_lag, type = "Ljung-Box")$p.value,
2969         error = function(e) NA_real_
2970       )
2971
2972       results[[i]] <- tibble(
2973         p = p, d = d, q = q,
```

```

2974     P = P, D = D, Q = Q,
2975     s = s,
2976     ncoef = length(coef_vec),
2977     AIC = fit_try$aic,
2978     BIC = BIC(fit_try),
2979     lb_pvalue = lb_pvalue,
2980     all_coef_sig = ifelse(length(pvals) == 0, TRUE, all(pvals < 0.05, na.rm = TRUE)),
2981     fit = list(fit_try),
2982     coef_table = list(
2983       tibble(
2984         param = names(coef_vec),
2985         estimate = as.numeric(coef_vec),
2986         p_value = as.numeric(pvals)
2987       )
2988     )
2989   }
2990 }
2991 }
2992
2993 bind_rows(results) %>%
2994   arrange(AIC, ncoef)
2995 }
2996
2997 # ----- ARIMA pour séries annuelles
2998 fit_arima_grid <- function(x_ts, d,
2999                           p_max = 2, q_max = 2,
3000                           include_mean = FALSE) {
3001
3002   grid <- expand_grid(
3003     p = 0:p_max,
3004     q = 0:q_max
3005   )
3006
3007   results <- vector("list", nrow(grid))
3008
3009   for (i in seq_len(nrow(grid))) {
3010
3011     p <- grid$p[i]
3012     q <- grid$q[i]
3013
3014     fit_try <- tryCatch(
3015       arima(
3016         x = x_ts,
3017         order = c(p, d, q),
3018         include.mean = include_mean,
3019         method = "ML"
3020       ),
3021       error = function(e) NULL
3022     )
3023
3024     if (!is.null(fit_try)) {
3025
3026       coef_vec <- fit_try$coef
3027       var_diag <- tryCatch(diag(fit_try$var.coef), error = function(e) numeric(0))
3028
3029       if (length(coef_vec) > 0 && length(var_diag) == length(coef_vec)) {
3030         pvals <- coef_pvalues(coef_vec, var_diag)
3031       } else {
3032         pvals <- rep(NA_real_, length(coef_vec))
3033       }
3034

```

```

3035     lb_lag <- min(
3036       floor(sqrt(length(residuals(fit_try)))),
3037       length(residuals(fit_try)) - 1
3038     )
3039
3040     lb_pvalue <- tryCatch(
3041       Box.test(residuals(fit_try), lag = lb_lag, type = "Ljung-Box")$p.value,
3042       error = function(e) NA_real_
3043     )
3044
3045     results[[i]] <- tibble(
3046       p = p, d = d, q = q,
3047       P = 0L, D = 0L, Q = 0L,
3048       s = 1L,
3049       ncoef = length(coef_vec),
3050       AIC = fit_try$aic,
3051       BIC = BIC(fit_try),
3052       lb_pvalue = lb_pvalue,
3053       all_coef_sig = ifelse(length(pvals) == 0, TRUE, all(pvals < 0.05, na.rm = TRUE)),
3054       fit = list(fit_try),
3055       coef_table = list(
3056         tibble(
3057           param = names(coef_vec),
3058           estimate = as.numeric(coef_vec),
3059           p_value = as.numeric(pvals)
3060         )
3061       )
3062     )
3063   }
3064 }
3065
3066 bind_rows(results) %>%
3067   arrange(AIC, ncoef)
3068 }
3069
3070 # -----
3071 # 9) GARDER SEULEMENT LES 2 MEILLEURS MODELES
3072 # -----
3073 choose_top2_models <- function(tab) {
3074   tab %>%
3075     filter(!is.na(AIC)) %>%
3076     arrange(AIC, ncoef) %>%
3077     slice_head(n = 2)
3078 }
3079
3080 # -----
3081 # 10) CONSTRUCTION DES 4 FAMILLES DE SERIES
3082 # -----
3083 cases_tbl <- tibble(
3084   id = names(sek_fits),
3085   fit = sek_fits
3086 )
3087
3088 sek_k1 <- cases_tbl %>%
3089   mutate(
3090     param = "k1",
3091     x_ts = map2(fit, id, ~ build_weekly_series(.x, "k1", .y)),
3092     s = map_int(id, get_saison_length)
3093   )
3094
3095 sek_k4 <- cases_tbl %>%

```



```

3096   mutate(
3097     param = "k4",
3098     x_ts = map2(fit, id, ~ build_weekly_series(.x, "k4", .y)),
3099     s = map_int(id, get_saison_length)
3100   )
3101
3102 sek_k2 <- cases_tbl %>%
3103   mutate(
3104     param = "k2",
3105     x_ts = map(fit, ~ build_annual_series(.x, "k2")),
3106     s = 1L
3107   )
3108
3109 sek_k3 <- cases_tbl %>%
3110   mutate(
3111     param = "k3",
3112     x_ts = map(fit, ~ build_annual_series(.x, "k3")),
3113     s = 1L
3114   )
3115
3116 # -----
3117 # 11) PIPELINE AUTOMATIQUE : SERIES HEBDOMADAIRES
3118 # -----
3119 run_bj_weekly <- function(df_cases,
3120                           p_max = 2, q_max = 2,
3121                           P_max = 2, Q_max = 2,
3122                           include_mean = FALSE) {
3123
3124   df_cases %>%
3125     mutate(
3126       prep = map(x_ts, ~ prepare_bj_series(.x, is_weekly = TRUE)),
3127       ts_stab = map(prepare_bj_series, "ts_stab"),
3128       transform_used = map_chr(prepare_bj_series, "transform_used"),
3129       D = map_int(prepare_bj_series, "D"),
3130       d = map_int(prepare_bj_series, "d"),
3131       ts_deseason = map(prepare_bj_series, "ts_deseason"),
3132       ts_stationary = map(prepare_bj_series, "ts_stationary"),
3133
3134       sarima_table = pmap(
3135         list(ts_stab, d, D, s),
3136         function(ts_stab, d, D, s) {
3137           fit_sarima_grid(
3138             x_ts = ts_stab,
3139             d = d,
3140             D = D,
3141             s = s,
3142             p_max = p_max,
3143             q_max = q_max,
3144             P_max = P_max,
3145             Q_max = Q_max,
3146             include_mean = include_mean
3147           )
3148         }
3149       ),
3150
3151       top2_models = map(sarima_table, choose_top2_models)
3152     )
3153 }
3154
3155 # -----
3156 # 12) PIPELINE AUTOMATIQUE : SERIES ANNUELLES

```

```

3157 # -----
3158 run_bj_annual <- function(df_cases,
3159                           p_max = 2, q_max = 2,
3160                           include_mean = FALSE) {
3161
3162   df_cases %>%
3163     mutate(
3164       prep = map(x_ts, ~ prepare_bj_series(.x, is_weekly = FALSE)),
3165       ts_stab = map(prep, "ts_stab"),
3166       transform_used = map_chr(prep, "transform_used"),
3167       D = map_int(prep, "D"),
3168       d = map_int(prep, "d"),
3169       ts_deseason = map(prep, "ts_deseason"),
3170       ts_stationary = map(prep, "ts_stationary"),
3171
3172       sarima_table = map(
3173         ts_stab,
3174         ~ fit_arima_grid(
3175           x_ts = .x,
3176           d = 1L,
3177           p_max = p_max,
3178           q_max = q_max,
3179           include_mean = include_mean
3180         )
3181       ),
3182
3183       top2_models = map(sarima_table, choose_top2_models)
3184     )
3185 }
3186
3187 # -----
3188 # 13) LANCEMENT AUTOMATIQUE
3189 # -----
3190 sek_k1_bj <- run_bj_weekly(sek_k1, p_max = 2, q_max = 2, P_max = 2, Q_max = 2)
3191 sek_k4_bj <- run_bj_weekly(sek_k4, p_max = 2, q_max = 2, P_max = 2, Q_max = 2)
3192
3193 sek_k2_bj <- run_bj_annual(sek_k2, p_max = 2, q_max = 2)
3194 sek_k3_bj <- run_bj_annual(sek_k3, p_max = 2, q_max = 2)
3195
3196 # -----
3197 # 14) TABLEAUX DES 2 MEILLEURS MODELES
3198 # -----
3199 extract_top2_summary <- function(df_bj) {
3200   df_bj %>%
3201     transmute(
3202       id = id,
3203       param = param,
3204       transform_used = transform_used,
3205       top2 = map(top2_models, ~ .x %>%
3206         select(p, d, q, P, D, Q, s, ncoef, AIC, BIC, lb_pvalue, all_coef_sig))
3207     ) %>%
3208     unnest(top2)
3209 }
3210
3211 top2_k1 <- extract_top2_summary(sek_k1_bj)
3212 top2_k4 <- extract_top2_summary(sek_k4_bj)
3213 top2_k2 <- extract_top2_summary(sek_k2_bj)
3214 top2_k3 <- extract_top2_summary(sek_k3_bj)
3215
3216
3217 # -----

```

```

3218 # 15) AFFICHER LES 2 MEILLEURS MODELES D'UN CAS
3219 # -----
3220 show_top2_models <- function(df_bj, case_id) {
3221   i <- which(df_bj$id == case_id)
3222   if (length(i) == 0) stop("case_id introuvable.")
3223
3224   df_bj$top2_models[[i]] %>%
3225     select(p, d, q, P, D, Q, s, ncoef, AIC, BIC, lb_pvalue, all_coef_sig)
3226 }
3227
3228 # Exemples :
3229 show_top2_models(sek_k1_bj, "m_annuel")
3230 show_top2_models(sek_k4_bj, "f_froid")
3231 show_top2_models(sek_k2_bj, "m_chaud")
3232
3233 # -----
3234 # 16) AFFICHER LES COEFFICIENTS DES 2 MEILLEURS MODELES
3235 # -----
3236 show_top2_with_coef <- function(df_bj, case_id) {
3237   i <- which(df_bj$id == case_id)
3238   if (length(i) == 0) stop("case_id introuvable.")
3239
3240   top2 <- df_bj$top2_models[[i]]
3241
3242   for (j in seq_len(nrow(top2))) {
3243     cat("\n===== \n")
3244     cat("Cas :", df_bj$id[i], "| Paramètre :", df_bj$param[i], "| Rang :", j, "\n")
3245     cat("Modèle : (",
3246         top2$p[j], ",", top2$d[j], ",", top2$q[j], ") x (",
3247         top2$P[j], ",", top2$D[j], ",", top2$Q[j], ")_[" , top2$s[j], "]\n", sep = "")
3248     cat("AIC =", top2$AIC[j], "\n")
3249     cat("BIC =", top2$BIC[j], "\n")
3250     cat("Ljung-Box p-value =", top2$lb_pvalue[j], "\n")
3251     cat("Tous coefficients significatifs =", top2$all_coef_sig[j], "\n")
3252
3253     print(top2$coef_table[[j]])
3254   }
3255 }
3256
3257 # -----
3258 # 17) OBJET FINAL POUR LA SUITE
3259 # -----
3260 sek_boxjenkins_results <- list(
3261   k1 = sek_k1_bj,
3262   k4 = sek_k4_bj,
3263   k2 = sek_k2_bj,
3264   k3 = sek_k3_bj,
3265   top2_tables = list(
3266     top2_k1 = top2_k1,
3267     top2_k4 = top2_k4,
3268     top2_k2 = top2_k2,
3269     top2_k3 = top2_k3
3270   )
3271 )
3272
3273 #####
3274 #           Validation           #
3275 #####
3276
3277
3278 i <- which(sek_boxjenkins_results$k4$id == "m_annuel")

```

```
3279
3280 fit_obj_sek <- sek_boxjenkins_results$sek4$top2_models[[i]]$fit[[1]]
3281
3282 K <- floor(sqrt(length(residuals(fit_obj_sek))))
3283
3284 tsdiag(fit_obj_sek, gof.lag = K)
3285
3286
3287
3288 # -----
3289 # 18) EXEMPLES D'UTILISATION
3290 # -----
3291
3292 # 1) Diagnostic complet d'un cas
3293 plot_full_diagnostics_sek_case(sek_fits$f_annuel, "f_annuel")
3294
3295 # 2) Voir les 2 meilleurs modèles
3296 show_top2_models(sek_k1_bj, "m_annuel")
3297
3298 # 3) Voir les coefficients des 2 meilleurs modèles
3299 show_top2_with_coef(sek_k1_bj, "m_annuel")
3300
3301 # 4) Accéder directement aux tableaux finaux
3302 sek_boxjenkins_results$top2_tables$top2_k1
3303 sek_boxjenkins_results$sek1$sarima_table[[1]]
3304
3305
3306
3307 #####
3308 #      Projection      #
3309 #####
3310
3311
3312 invert_transform <- function(x, transform_used) {
3313   if (transform_used == "log") {
3314     exp(x)
3315   } else {
3316     x
3317   }
3318 }
3319
3320
3321 #####
3322 #      hors-échantillon seklecka      ##
3323 #####
3324
3325 dataset_test <- dataset %>%
3326   filter(Year>=2017 & Year <=2019)
3327
3328 forecast_sek_case_full <- function(case_id,
3329                                     dataset,
3330                                     dataset_train,
3331                                     dataset_test,
3332                                     cx_tbl,
3333                                     sek_fits,
3334                                     sek_k1_bj,
3335                                     sek_k4_bj,
3336                                     sek_k2_bj,
3337                                     sek_k3_bj) {
3338
3339   res_fit <- sek_fits[[case_id]]
```

```

3340   if (is.null(res_fit)) stop("case_id introuvable dans sek_fits")
3341
3342   sex_i      <- res_fit$meta$sex
3343   period_i <- res_fit$meta$periode
3344   a_i       <- res_fit$meta$a
3345
3346   periode_arg <- if (identical(period_i, "Annuel")) NULL else period_i
3347
3348   # -----
3349   # données complètes 2000-2019
3350   # -----
3351   prep_all <- prepare_data_seklecka_mixed(
3352     dataset,
3353     cx_tbl,
3354     sex_label = sex_i,
3355     periode = periode_arg,
3356     a = a_i
3357   )
3358
3359   df_all <- prep_all$df
3360
3361   # -----
3362   # données test 2017-2019
3363   # -----
3364   prep_test <- prepare_data_seklecka_mixed(
3365     dataset_test,
3366     cx_tbl,
3367     sex_label = sex_i,
3368     periode = periode_arg,
3369     a = a_i
3370   )
3371
3372   df_test <- prep_test$df
3373
3374   # -----
3375   # observé sur toute la période
3376   # -----
3377   obs_all <- df_all %>%
3378     mutate(
3379       mx_obs = ifelse(E > 0, D / E, NA_real_),
3380       Year_num = as.integer(Year),
3381       Week_num = as.integer(substr(t, 6, 7)),
3382       year_t = Year_num + (Week_num - 0.5) / 52
3383     ) %>%
3384     transmute(
3385       Age = age_num,
3386       Year_week = t,
3387       Year = Year_num,
3388       Week = Week_num,
3389       year_t = year_t,
3390       Deaths_obs = D,
3391       Exposure = E,
3392       mx_obs = mx_obs,
3393       cohort = cohort,
3394       bx2 = bx2,
3395       bx3 = bx3,
3396       gx2 = gx2
3397     )
3398   # -----
3399   # horizon test
3400   # -----

```

```

3401 periods_test <- df_test %>%
3402   distinct(t, Year) %>%
3403   arrange(Year, t)
3404
3405 years_test <- df_test %>%
3406   distinct(Year) %>%
3407   arrange(Year)
3408
3409 h_week <- nrow(periods_test)
3410 h_year <- nrow(years_test)
3411
3412 # -----
3413 # projection d'une série
3414 # -----
3415 project_one_param <- function(df_bj, case_id, h) {
3416
3417   i <- which(df_bj$id == case_id)
3418   if (length(i) == 0) stop("case_id introuvable dans l'objet BJ : ", case_id)
3419
3420   transform_used <- df_bj$transform_used[i]
3421   best_row <- df_bj$top2_models[[i]]
3422
3423   if (is.null(best_row) || nrow(best_row) == 0) {
3424     stop("Aucun top2_model disponible pour ", case_id)
3425   }
3426
3427   best_fit <- best_row$fit[[1]]
3428
3429   if (is.null(best_fit)) {
3430     stop("Le best_fit est NULL pour ", case_id)
3431   }
3432
3433   # IMPORTANT : utiliser predict() et non forecast()
3434   pred <- predict(best_fit, n.ahead = h)
3435   x_fc <- as.numeric(pred$pred)
3436
3437   if (transform_used == "log") {
3438     x_fc <- exp(x_fc)
3439   }
3440
3441   return(x_fc)
3442 }
3443 # -----
3444 # projection des 4 kappas
3445 # -----
3446 k1_fc <- project_one_param(sek_k1_bj, case_id, h_week)
3447 k4_fc <- project_one_param(sek_k4_bj, case_id, h_week)
3448 k2_fc <- project_one_param(sek_k2_bj, case_id, h_year)
3449 k3_fc <- project_one_param(sek_k3_bj, case_id, h_year)
3450
3451 k1_proj_tbl <- periods_test %>% mutate(k1 = k1_fc)
3452 k4_proj_tbl <- periods_test %>% mutate(k4 = k4_fc)
3453 k2_proj_tbl <- years_test %>% mutate(k2 = k2_fc)
3454 k3_proj_tbl <- years_test %>% mutate(k3 = k3_fc)
3455
3456 alpha_tbl <- res_fit$alpha
3457 gamma_tbl <- res_fit$gamma
3458
3459 # -----
3460 # projection sur 2017-2019
3461 # -----

```

```

3462 proj_test <- df_test %>%
3463   left_join(alpha_tbl, by = c("age_num")) %>%
3464   left_join(gamma_tbl, by = c("cohort")) %>%
3465   left_join(k1_proj_tbl, by = c("t", "Year")) %>%
3466   left_join(k4_proj_tbl, by = c("t", "Year")) %>%
3467   left_join(k2_proj_tbl, by = c("Year")) %>%
3468   left_join(k3_proj_tbl, by = c("Year")) %>%
3469   mutate(
3470     alpha_x = ifelse(is.na(alpha_x), 0, alpha_x),
3471     gamma_c = ifelse(is.na(gamma_c), 0, gamma_c),
3472     eta_proj = alpha_x + gamma_c + k1 + bx2 * k2 + bx3 * k3 + gx2 * k4,
3473     mx_proj = exp(eta_proj),
3474     D_proj = E * mx_proj
3475   ) %>%
3476   transmute(
3477     Age = age_num,
3478     Year_week = t,
3479     mx_proj = mx_proj,
3480     D_proj = D_proj
3481   )
3482
3483   # -----
3484   # fusion finale
3485   # -----
3486   final_df <- obs_all %>%
3487     left_join(proj_test, by = c("Age", "Year_week")) %>%
3488     mutate(
3489       Sex = sex_i,
3490       Period = period_i
3491     ) %>%
3492     arrange(Age, Year, Week)
3493
3494   final_df
3495 }
3496
3497 plot_sek_projection_full <- function(proj_df,
3498                                     ages_to_plot = c(25, 45, 65, 85),
3499                                     title_prefix = "") {
3500
3501   df_plot <- proj_df %>%
3502     filter(Age %in% ages_to_plot)
3503
3504   ggplot(df_plot, aes(x = year_t)) +
3505     geom_line(aes(y = mx_obs, colour = "Observed"), linewidth = 0.9, na.rm = TRUE) +
3506     geom_line(aes(y = mx_proj, colour = "Projected"), linewidth = 0.9,
3507               linetype = "dashed", na.rm = TRUE) +
3508     facet_wrap(~ Age, scales = "free_y") +
3509     theme_bw() +
3510     labs(
3511       title = title_prefix,
3512       x = "Temps",
3513       y = "Taux de mortalité",
3514       colour = ""
3515     )
3516 }
3517
3518 plot_sek_projection_one_age <- function(proj_df, age_x, title_prefix = "") {
3519
3520   df_age <- proj_df %>%
3521     filter(Age == age_x) %>%
3522     arrange(year_t)

```

```
3523
3524   ggplot(df_age, aes(x = year_t)) +
3525     geom_line(aes(y = mx_obs, colour = "Observé"), linewidth = 1.0, na.rm = TRUE) +
3526     geom_line(aes(y = mx_proj, colour = "Projeté"), linewidth = 1.0,
3527               linetype = "solid", na.rm = TRUE) +
3528     scale_color_manual(values = c("Observé"="blue", "Projeté"="red"))+
3529     theme_bw() +
3530     labs(
3531       title = paste0(title_prefix, " Age ", age_x),
3532       x = "Temps",
3533       y = "Taux de mortalité",
3534       colour = ""
3535     )
3536 }
3537
3538
3539 proj_sek_f_ann <- forecast_sek_case_full(
3540   case_id = "f_annuel",
3541   dataset = dataset,
3542   dataset_train = dataset_train,
3543   dataset_test = dataset_test,
3544   cx_tbl = cx_tbl,
3545   sek_fits = sek_fits,
3546   sek_k1_bj = sek_k1_bj,
3547   sek_k4_bj = sek_k4_bj,
3548   sek_k2_bj = sek_k2_bj,
3549   sek_k3_bj = sek_k3_bj
3550 )
3551
3552
3553 plot_sek_projection_one_age(proj_sek_f_ann, 25, NULL)
3554 plot_sek_projection_one_age(proj_sek_f_ann, 45, NULL)
3555 plot_sek_projection_one_age(proj_sek_f_ann, 65, NULL)
3556 plot_sek_projection_one_age(proj_sek_f_ann, 85, NULL)
3557
3558
3559
3560 # -----
3561 # RANGS CHOISIS (modifier si besoin)
3562 # -----
3563 chosen_rank_k1 <- tibble(
3564   id = c("m_annuel"),
3565   chosen_rank = c(1)
3566 )
3567
3568 chosen_rank_k2 <- NULL
3569 chosen_rank_k3 <- NULL
3570 chosen_rank_k4 <- NULL
3571
3572 # -----
3573 # FONCTION D'EXTRACTION
3574 # -----
3575 extract_best_models_one_param <- function(df_bj, param_name, chosen_rank_tbl = NULL) {
3576
3577   if (is.null(chosen_rank_tbl)) {
3578     chosen_rank_tbl <- tibble(
3579       id = df_bj$id,
3580       chosen_rank = 1
3581     )
3582   }
3583 }
```



```

3584 df_bj %>%
3585   select(id, transform_used, top2_models) %>%
3586   left_join(chosen_rank_tbl, by = "id") %>%
3587   mutate(
3588     chosen_rank = ifelse(is.na(chosen_rank), 1, chosen_rank),
3589     chosen_model = map2(top2_models, chosen_rank, function(tab, r) {
3590       if (is.null(tab) || nrow(tab) < r) {
3591         return(NULL)
3592       }
3593       tab[r, , drop = FALSE]
3594     })
3595   ) %>%
3596   transmute(
3597     id = id,
3598     param = param_name,
3599     transform_used = transform_used,
3600     chosen_rank = chosen_rank,
3601     chosen_model
3602   ) %>%
3603   unnest(chosen_model) %>%
3604   mutate(
3605     model = paste0(
3606       "(", p, ",", d, ",", q, ") x (",
3607       P, ",", D, ",", Q, ")[", s, "]"
3608     )
3609   ) %>%
3610   select(
3611     id, param,
3612     model, ncoef,
3613     AIC, lb_pvalue, all_coef_sig
3614   ) %>%
3615   arrange(id)
3616 }
3617
3618 # -----
3619 # EXTRACTION DES MEILLEURS MODELES
3620 # -----
3621 best_k1 <- extract_best_models_one_param(sek_k1_bj, "k1", chosen_rank_k1)
3622 best_k2 <- extract_best_models_one_param(sek_k2_bj, "k2", chosen_rank_k2)
3623 best_k3 <- extract_best_models_one_param(sek_k3_bj, "k3", chosen_rank_k3)
3624 best_k4 <- extract_best_models_one_param(sek_k4_bj, "k4", chosen_rank_k4)
3625
3626 # -----
3627 # TABLEAU FINAL LONG
3628 # -----
3629 best_models_sek_report <- bind_rows(
3630   best_k1,
3631   best_k2,
3632   best_k3,
3633   best_k4
3634 ) %>%
3635   arrange(id, param)
3636
3637 print(best_models_sek_report)
3638
3639 kable(best_models_sek_report, format = "latex", caption = "Tableau récapitulatif des meilleurs modèles")
3640 # -----
3641 # TABLEAU FINAL COURT POUR LE RAPPORT
3642 # -----
3643 best_models_sek_report_clean <- best_models_sek_report %>%
3644   select(id, param, model, AIC, lb_pvalue, all_coef_sig) %>%

```

```
3645     arrange(id, param)
3646
3647 print(best_models_sek_report_clean)
3648
3649 # -----
3650 # VERSION LARGE : 1 LIGNE PAR CAS
3651 # -----
3652 best_models_sek_wide <- best_models_sek_report %>%
3653   select(id, param, model) %>%
3654   pivot_wider(
3655     names_from = param,
3656     values_from = model
3657   ) %>%
3658   arrange(id)
3659
3660 print(best_models_sek_wide)
3661
3662
3663 proj_sek_f_ann %>%
3664   filter(!is.na(mx_proj), mx_obs > 0) %>%
3665   summarise(
3666     RMSE = sqrt(mean((mx_obs - mx_proj)^2)),
3667     MAE = mean(abs(mx_obs - mx_proj)),
3668     MAPE = mean(abs((mx_obs - mx_proj)/mx_obs))
3669   )
3670
3671
3672 case_ids <- names(sek_fits)
3673
3674 compute_errors_one_case <- function(case_id) {
3675
3676   proj <- forecast_sek_case_full(
3677     case_id = case_id,
3678     dataset = dataset,
3679     dataset_train = dataset_train,
3680     dataset_test = dataset_test,
3681     cx_tbl = cx_tbl,
3682     sek_fits = sek_fits,
3683     sek_k1_bj = sek_k1_bj,
3684     sek_k4_bj = sek_k4_bj,
3685     sek_k2_bj = sek_k2_bj,
3686     sek_k3_bj = sek_k3_bj
3687   )
3688
3689   res <- proj %>%
3690     filter(!is.na(mx_proj), mx_obs > 0) %>%
3691     summarise(
3692       RMSE = sqrt(mean((mx_obs - mx_proj)^2)),
3693       MAE = mean(abs(mx_obs - mx_proj)),
3694       MAD = safe_mad(mx_obs, mx_proj),
3695       MAPE = mean(abs((mx_obs - mx_proj)/mx_obs))
3696     )
3697
3698   res$case_id <- case_id
3699
3700   res
3701 }
3702
3703 results_sek_errors <- lapply(case_ids, compute_errors_one_case) %>%
3704   bind_rows() %>%
3705   mutate(
```

```

3706     Sex = substr(case_id, 1, 1),
3707     Period = sub("[mf]_", "", case_id)
3708   ) %>%
3709   select(Sex, Period, RMSE, MAE, MAD, MAPE) %>%
3710   arrange(Sex, Period) %>%
3711   mutate(
3712     RMSE = round(RMSE, 6),
3713     MAE = round(MAE, 6),
3714     MAD = round(MAD, 6),
3715     MAPE = round(MAPE, 6)
3716   )
3717
3718   print(results_sek_errors)
3719   kable(results_sek_errors, format = "latex")
3720
3721   #####
3722   #   Definition of the dataset for pricing   #
3723   #####
3724
3725   #### RAW DATA
3726
3727   price_raw <- function(data_raw,
3728                         sex_choice,
3729                         age_start,
3730                         start_date,
3731                         horizon_years,
3732                         i = 0.02) {
3733
3734
3735     # 1. Transformer la date de départ en vrai format Date
3736     start_date <- as.Date(start_date)
3737
3738     # 2. Filtrer les données utiles
3739     data_case <- data_raw %>%
3740       filter(
3741         Sex == sex_choice,
3742         date >= start_date,
3743         date <= start_date %m+% years(horizon_years)
3744       ) %>%
3745       arrange(date, Age)
3746
3747     # 3. Construire la trajectoire des semaines
3748     trajectory <- data.frame(date = sort(unique(data_case$date)))
3749
3750     # 4. Calculer le nombre d'années écoulées depuis la date de départ
3751     trajectory$years_elapsed <- as.numeric(
3752       difftime(trajectory$date, start_date, units = "days")
3753     ) / 365.25
3754
3755     # 5. Déterminer la classe d'âge atteinte à chaque semaine
3756     trajectory$Age <- get_age_class(age_start, trajectory$years_elapsed)
3757
3758     # 6. Récupérer le taux brut correspondant à chaque date et âge
3759     trajectory <- trajectory %>%
3760       left_join(
3761         data_case %>% select(date, Age, Deaths_rate),
3762         by = c("date", "Age")
3763       )
3764
3765     # 7. Vérification : s'il manque des taux, on arrête
3766     if (any(is.na(trajectory$Deaths_rate))) {

```

```

3767     stop("Il manque des valeurs de Deaths_rate dans la trajectoire brute.")
3768   }
3769
3770   # 8. Calculer les probabilités hebdomadaires de survie
3771   trajectory <- trajectory %>%
3772     mutate(
3773       p_week = exp(-Deaths_rate),
3774       surv_cum = cumprod(p_week)
3775     )
3776
3777   # 9. Construire les dates mensuelles de paiement
3778   payment_dates <- seq.Date(
3779     from = start_date %m+% months(1),
3780     to   = start_date %m+% years(horizon_years),
3781     by   = "month"
3782   )
3783
3784   # 10. Construire le tableau des paiements
3785   payment_table <- data.frame(payment_date = payment_dates)
3786
3787   # 11. Associer à chaque paiement la dernière semaine observée disponible
3788   payment_table$obs_date <- sapply(payment_table$payment_date, function(d) {
3789     max(trajectory$date[trajectory$date <= d])
3790   })
3791   payment_table$obs_date <- as.Date(payment_table$obs_date)
3792   # 12. Récupérer la survie cumulée à chaque date de paiement
3793   payment_table <- payment_table %>%
3794     left_join(
3795       trajectory %>% select(date, surv_cum),
3796       by = c("obs_date" = "date")
3797     )
3798
3799   # 13. Calculer le temps écoulé jusqu'à chaque paiement
3800   payment_table$time_in_years <- as.numeric(
3801     difftime(payment_table$payment_date, start_date, units = "days")
3802   ) / 365.25
3803
3804   # 14. Calculer le facteur d'actualisation
3805   payment_table$discount <- (1 + i)^(-payment_table$time_in_years)
3806
3807   # 15. Calculer la valeur actuelle de chaque paiement
3808   payment_table$pv_term <- payment_table$discount * payment_table$surv_cum
3809
3810   # 16. Calculer le prix total
3811   pv <- sum(payment_table$pv_term)
3812
3813   # 17. Retourner les résultats
3814   return(list(
3815     pv = pv,
3816     trajectory = trajectory,
3817     payment_table = payment_table
3818   ))
3819 }
3820
3821 res_raw <- price_raw(
3822   data_raw = dataset_train,
3823   sex_choice = "m",
3824   age_start = 60,
3825   start_date = "2000-01-03",
3826   horizon_years = 5,
3827   i = 0.02

```

```
3828 )
3829
3830 res_raw$pv
3831
3832
3833 #####
3834 ##### LC DATA#####
3835 #####
3836
3837 price_lc <- function(data_lc,
3838                       sex_choice,
3839                       age_start,
3840                       start_date,
3841                       horizon_years,
3842                       i = 0.02,
3843                       period_choice = "Annuel") {
3844
3845   start_date <- as.Date(start_date)
3846
3847   lc_case <- data_lc %>%
3848     filter(
3849       Sex == sex_choice,
3850       Period == period_choice,
3851       date >= start_date,
3852       date <= start_date %m+% years(horizon_years)
3853     ) %>%
3854     arrange(date, Age)
3855
3856   trajectory <- data.frame(date = sort(unique(lc_case$date)))
3857
3858   trajectory$years_elapsed <- as.numeric(
3859     difftime(trajectory$date, start_date, units = "days")
3860   ) / 365.25
3861
3862   trajectory$Age <- get_age_class(age_start, trajectory$years_elapsed)
3863
3864   trajectory <- trajectory %>%
3865     left_join(
3866       lc_case %>% select(date, Age, mx_hat),
3867       by = c("date", "Age")
3868     )
3869
3870   if (any(is.na(trajectory$mx_hat))) {
3871     stop("Il manque des valeurs de mx_hat dans la trajectoire LC.")
3872   }
3873
3874   trajectory <- trajectory %>%
3875     mutate(
3876       p_week = exp(-mx_hat),
3877       surv_cum = cumprod(p_week)
3878     )
3879
3880   payment_dates <- seq.Date(
3881     from = start_date %m+% months(1),
3882     to   = start_date %m+% years(horizon_years),
3883     by   = "month"
3884   )
3885
3886   payment_table <- data.frame(payment_date = payment_dates)
3887
3888   payment_table$obs_date <- sapply(payment_table$payment_date, function(d) {
```

```

3889     max(trajjectory$date[trajjectory$date <= d])
3890   })
3891   payment_table$obs_date <- as.Date(payment_table$obs_date)
3892   payment_table <- payment_table %>%
3893     left_join(
3894       trajjectory %>% select(date, surv_cum),
3895       by = c("obs_date" = "date")
3896     )
3897
3898   payment_table$time_in_years <- as.numeric(
3899     difftime(payment_table$payment_date, start_date, units = "days")
3900   ) / 365.25
3901
3902   payment_table$discount <- (1 + i)^(-payment_table$time_in_years)
3903   payment_table$pv_term <- payment_table$discount * payment_table$surv_cum
3904
3905   pv <- sum(payment_table$pv_term)
3906
3907   return(list(
3908     pv = pv,
3909     trajjectory = trajjectory,
3910     payment_table = payment_table
3911   ))
3912 }
3913
3914 res_lc <- price_lc(
3915   data_lc = lc_predictions_table,
3916   sex_choice = "m",
3917   age_start = 60,
3918   start_date = "2000-01-03",
3919   horizon_years = 5,
3920   i = 0.02,
3921   period_choice = "Annuel"
3922 )
3923
3924 res_lc$pv
3925
3926
3927 #####
3928 ####   SPO DATA   #####
3929 #####
3930
3931
3932 price_sek <- function(data_sek,
3933                       sex_choice,
3934                       age_start,
3935                       start_date,
3936                       horizon_years,
3937                       i = 0.02,
3938                       period_choice = "Annuel") {
3939
3940   start_date <- as.Date(start_date)
3941
3942   sek_case <- data_sek %>%
3943     filter(
3944       Sex == sex_choice,
3945       Period == period_choice,
3946       date >= start_date,
3947       date <= start_date %m+% years(horizon_years)
3948     ) %>%
3949     arrange(date, Age)

```

```

3950 trajectory <- data.frame(date = sort(unique(sek_case$date)))
3951
3952
3953 trajectory$years_elapsed <- as.numeric(
3954   difftime(trajectory$date, start_date, units = "days")
3955 ) / 365.25
3956
3957 trajectory$Age <- get_age_class(age_start, trajectory$years_elapsed)
3958
3959 trajectory <- trajectory %>%
3960   left_join(
3961     sek_case %>% select(date, Age, mx_hat),
3962     by = c("date", "Age")
3963   )
3964
3965 if (any(is.na(trajectory$mx_hat))) {
3966   stop("Il manque des valeurs de mx_hat dans la trajectoire SP0.")
3967 }
3968
3969 trajectory <- trajectory %>%
3970   mutate(
3971     p_week = exp(-mx_hat),
3972     surv_cum = cumprod(p_week)
3973   )
3974
3975 payment_dates <- seq.Date(
3976   from = start_date %m+% months(1),
3977   to   = start_date %m+% years(horizon_years),
3978   by   = "month"
3979 )
3980
3981 payment_table <- data.frame(payment_date = payment_dates)
3982
3983 payment_table$obs_date <- sapply(payment_table$payment_date, function(d) {
3984   max(trajectory$date[trajectory$date <= d])
3985 })
3986
3987 payment_table$obs_date <- as.Date(payment_table$obs_date)
3988 payment_table <- payment_table %>%
3989   left_join(
3990     trajectory %>% select(date, surv_cum),
3991     by = c("obs_date" = "date")
3992   )
3993
3994 payment_table$time_in_years <- as.numeric(
3995   difftime(payment_table$payment_date, start_date, units = "days")
3996 ) / 365.25
3997
3998 payment_table$discount <- (1 + i)^(-payment_table$time_in_years)
3999 payment_table$pv_term <- payment_table$discount * payment_table$surv_cum
4000
4001 pv <- sum(payment_table$pv_term)
4002
4003 return(list(
4004   pv = pv,
4005   trajectory = trajectory,
4006   payment_table = payment_table
4007 ))
4008 }
4009
4010 res_sek <- price_sek(

```

```
4011 data_sek = sek_predictions_table,
4012 sex_choice = "m",
4013 age_start = 60,
4014 start_date = "2000-01-03",
4015 horizon_years = 5,
4016 i = 0.02,
4017 period_choice = "Annuel"
4018 )
4019
4020 res_sek$pv
4021
4022
4023 #####
4024 ### Comparaison #####
4025 #####
4026
4027
4028 compare_pricing_models <- function(data_raw,
4029                                   data_lc,
4030                                   data_sek,
4031                                   sex_choice,
4032                                   age_start,
4033                                   start_date,
4034                                   horizon_years,
4035                                   i = 0.02,
4036                                   period_choice = "Annuel") {
4037
4038   res_raw <- price_raw(
4039     data_raw = data_raw,
4040     sex_choice = sex_choice,
4041     age_start = age_start,
4042     start_date = start_date,
4043     horizon_years = horizon_years,
4044     i = i
4045   )
4046
4047   res_lc <- price_lc(
4048     data_lc = data_lc,
4049     sex_choice = sex_choice,
4050     age_start = age_start,
4051     start_date = start_date,
4052     horizon_years = horizon_years,
4053     i = i,
4054     period_choice = period_choice
4055   )
4056
4057   res_sek <- price_sek(
4058     data_sek = data_sek,
4059     sex_choice = sex_choice,
4060     age_start = age_start,
4061     start_date = start_date,
4062     horizon_years = horizon_years,
4063     i = i,
4064     period_choice = period_choice
4065   )
4066
4067   PV_raw <- res_raw$pv
4068   PV_lc <- res_lc$pv
4069   PV_sek <- res_sek$pv
4070
4071   err_lc <- (PV_lc - PV_raw) / PV_raw * 100
```



```

4072     err_sek <- (PV_sek - PV_raw) / PV_raw * 100
4073
4074     summary_table <- data.frame(
4075       Sex = sex_choice,
4076       Age_start = age_start,
4077       Start_date = as.Date(start_date),
4078       Horizon_years = horizon_years,
4079       Rate = i,
4080       PV_raw = PV_raw,
4081       PV_LC = PV_lc,
4082       PV_SPO = PV_sek,
4083       Error_LC_pct = err_lc,
4084       Error_SPO_pct = err_sek
4085     )
4086
4087     return(list(
4088       summary = summary_table,
4089       raw = res_raw,
4090       lc = res_lc,
4091       sek = res_sek
4092     ))
4093   }
4094
4095
4096   comparison_one_case <- compare_pricing_models(
4097     data_raw = dataset_train,
4098     data_lc = lc_predictions_table,
4099     data_sek = sek_predictions_table,
4100     sex_choice = "m",
4101     age_start = 30,
4102     start_date = "2000-01-03",
4103     horizon_years = 5,
4104     i = 0.02,
4105     period_choice = "Annuel"
4106   )
4107
4108   comparison_one_case$summary
4109
4110
4111
4112   run_pricing_grid <- function(data_raw,
4113                                data_lc,
4114                                data_sek,
4115                                sexes,
4116                                ages,
4117                                start_dates,
4118                                horizons,
4119                                i = 0.02,
4120                                period_choice = "Annuel") {
4121
4122     # 1. Transformer les dates de départ en format Date
4123     start_dates <- as.Date(start_dates)
4124
4125     # 2. Trouver la dernière date disponible dans les données brutes
4126     max_date <- max(data_raw$date, na.rm = TRUE)
4127
4128     # 3. Construire toutes les combinaisons possibles
4129     grid <- expand.grid(
4130       Sex = sexes,
4131       Age_start = ages,
4132       Start_date = start_dates,

```

```

4133     Horizon_years = horizons,
4134     stringsAsFactors = FALSE
4135   )
4136
4137   # 4. Transformer en data.frame propre
4138   grid <- as.data.frame(grid)
4139
4140   # 5. Ajouter la date de fin théorique
4141   grid$End_date <- as.Date(grid$Start_date) %m+% years(grid$Horizon_years)
4142
4143   # 6. Garder seulement les cas compatibles avec les données disponibles
4144   grid_valid <- grid %>%
4145     filter(End_date <= max_date)
4146
4147   # 7. Vérification : si aucun cas valide, arrêter
4148   if (nrow(grid_valid) == 0) {
4149     stop("Aucune combinaison valide : l'horizon dépasse la dernière date disponible.")
4150   }
4151
4152   # 8. Fonction interne pour calculer un cas
4153   compute_one_case <- function(sex_choice, age_start, start_date, horizon_years) {
4154
4155     res <- compare_pricing_models(
4156       data_raw = data_raw,
4157       data_lc = data_lc,
4158       data_sek = data_sek,
4159       sex_choice = sex_choice,
4160       age_start = age_start,
4161       start_date = start_date,
4162       horizon_years = horizon_years,
4163       i = i,
4164       period_choice = period_choice
4165     )
4166
4167     return(res$summary)
4168   }
4169
4170   # 9. Boucle sur tous les cas valides
4171   results_list <- vector("list", nrow(grid_valid))
4172
4173   for (k in seq_len(nrow(grid_valid))) {
4174     results_list[[k]] <- compute_one_case(
4175       sex_choice = grid_valid$Sex[k],
4176       age_start = grid_valid$Age_start[k],
4177       start_date = grid_valid$Start_date[k],
4178       horizon_years = grid_valid$Horizon_years[k]
4179     )
4180   }
4181
4182   # 10. Fusionner tous les résultats
4183   results_table <- dplyr::bind_rows(results_list)
4184
4185   # 11. Trier proprement le tableau final
4186   results_table <- results_table %>%
4187     arrange(Sex, Age_start, Start_date, Horizon_years)
4188
4189   # 12. Retourner les objets utiles
4190   return(list(
4191     grid_valid = grid_valid,
4192     results_table = results_table
4193   ))

```

```

4194 }
4195
4196
4197
4198
4199 start_dates_example <- "2000-01-03"
4200 pricing_results <- run_pricing_grid(
4201   data_raw = dataset_train,
4202   data_lc = lc_predictions_table,
4203   data_sek = sek_predictions_table,
4204   sexes = c("m", "f"),
4205   ages = c(30, 40, 50, 60),
4206   start_dates = start_dates_example,
4207   horizons = 1:15,
4208   i = 0.02,
4209   period_choice = "Annuel"
4210 )
4211
4212 result <- as.data.frame(pricing_results$results_table )
4213
4214
4215
4216
4217 plot_errors <- pricing_results$results_table %>%
4218   select(Sex, Age_start, Horizon_years, Error_LC_pct, Error_SPO_pct, PV_raw, PV_LC, PV_SPO) %>%
4219   pivot_longer(
4220     cols = c(Error_LC_pct, Error_SPO_pct),
4221     names_to = "Model",
4222     values_to = "Error_pct"
4223   ) %>%
4224   mutate(
4225     Model = recode(
4226       Model,
4227       Error_LC_pct = "Lee-Carter",
4228       Error_SPO_pct = "SPO"
4229     ),
4230     Sex = recode(
4231       Sex,
4232       "m" = "Homme",
4233       "f" = "Femme"
4234     )
4235   )
4236
4237 plot_one_case <- function(data, sex_value, age_value) {
4238
4239   df_plot <- data %>%
4240     filter(Sex == sex_value, Age_start == age_value)
4241
4242   ggplot(df_plot, aes(x = Horizon_years, y = Error_pct, color = Model, group = Model)) +
4243
4244     # Ligne horizontale à 0
4245     geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
4246
4247     # Courbes
4248     geom_line(linewidth = 1) +
4249     geom_point(size = 2) +
4250
4251     # PLUS DE CHIFFRES SUR LES AXES
4252     scale_x_continuous(
4253       breaks = seq(min(df_plot$Horizon_years), max(df_plot$Horizon_years), by = 1)
4254     ) +

```

```

4255
4256     scale_y_continuous(
4257       breaks = seq(
4258         floor(min(df_plot$Error_pct)),
4259         ceiling(max(df_plot$Error_pct)),
4260         by = 0.5
4261       )
4262     ) +
4263
4264     labs(
4265       title = paste("Erreur relative du pricing -", sex_value, "- âge", age_value),
4266       x = "Horizon (années)",
4267       y = "Erreur relative (%)",
4268       color = "Modèle"
4269     ) +
4270
4271     # CADRE COMPLET
4272     theme_minimal() +
4273     theme(panel.grid = element_line(color = "grey80"))+
4274
4275     theme(
4276       plot.title = element_text(face = "bold", hjust = 0.5),
4277
4278       # CADRE AUTOUR
4279       panel.border = element_rect(color = "black", fill = NA, linewidth = 1),
4280
4281       # AXES PLUS VISIBLES
4282       axis.line = element_line(color = "black"),
4283
4284       # PLUS DE LISIBILITÉ
4285       axis.text = element_text(size = 11),
4286       axis.title = element_text(size = 12)
4287     )
4288 }
4289
4290 plot_one_case(plot_errors, "Homme", 60)
4291
4292
4293 plot_price <- pricing_results$results_table %>%
4294   select(Sex, Age_start, Horizon_years, PV_raw, PV_LC, PV_SPO) %>%
4295   pivot_longer(
4296     cols = c(PV_raw, PV_LC, PV_SPO),
4297     names_to = "Model",
4298     values_to = "Price"
4299   ) %>%
4300   mutate(
4301     Model = recode(
4302       Model,
4303       PV_raw = "Base",
4304       PV_LC = "Lee-Carter",
4305       PV_SPO = "SPO"
4306     ),
4307     Sex = recode(
4308       Sex,
4309       "m" = "Homme",
4310       "f" = "Femme"
4311     )
4312   )
4313
4314 plot_one_case_p <- function(data, sex_value, age_value) {
4315

```

```
4316 df_plot <- data %>%
4317   filter(Sex == sex_value, Age_start == age_value)
4318
4319 ggplot(df_plot, aes(x = Horizon_years, y = Price, color = Model, group = Model)) +
4320   geom_line(linewidth = 1) +
4321   geom_point(size = 2) +
4322   scale_x_continuous(
4323     breaks = seq(min(df_plot$Horizon_years), max(df_plot$Horizon_years), by = 1)
4324   ) +
4325   scale_y_continuous(
4326     breaks = pretty(df_plot$Price, n = 8)
4327   ) +
4328   labs(
4329     title = paste("Prix de la rente -", sex_value, "- âge", age_value),
4330     x = "Horizon (années)",
4331     y = "Prix",
4332     color = "Modèle"
4333   ) +
4334   theme_minimal() +
4335   theme(
4336     panel.grid = element_line(color = "grey80"),
4337     plot.title = element_text(face = "bold", hjust = 0.5),
4338     panel.border = element_rect(color = "black", fill = NA, linewidth = 1),
4339     axis.line = element_line(color = "black"),
4340     axis.text = element_text(size = 11),
4341     axis.title = element_text(size = 12)
4342   )
4343 }
4344
4345 plot_one_case_p(plot_price, "Femme", 50)
```