Université Paris 1 Panthéon-Sorbonne

Rédigé par : Pierre Monnin

5 septembre 2025

Table des matières

Préambule **2**

I/ Cadre théorique des SGWT : **3**
A) La transformée de Fourier 3
B) Les ondelettes 5
C) Les Spectral Graph Wavelets 8

II/ Application : Étude du taux de mortalité à partir des SGWT **12**
A) Préparation des données : 12
B) Analyse des tendances structurelles à partir des SGWT : 13
C) Modélisation du taux de mortalité en France en fonction de la région et du temps : 17

Conclusion **21**

Préambule

L'objectif de ce projet est de présenter les Spectral Graph Wavelets, généralisation du concept d'ondelette au cadre des graphes, et un exemple de leur utilisation dans un cadre actuariel. Dans un premier temps, nous allons présenter la notion de façon théorique, puis nous allons les mettre en application sur les données de mortalité régionales en France. L'objectif sera de définir un modèle prédictif du taux de mortalité à partir des coefficients des SGWT. Cette modélisation du taux de mortalité peut être utilisée dans un cadre de tarification de produits d'assurance-vie, dans un objectif de provisionnement de rentes viagères ou dans un cadre réglementaire (Solvabilité II / IFRS 17).

I/ Cadre théorique des SGWT :

Dans cette partie, nous allons reprendre d'un point de vue théorique les bases des SGWT. Nous présenterons aussi d'autres notions nécessaires à leur compréhension :

- La transformée de Fourier
- Les ondelettes

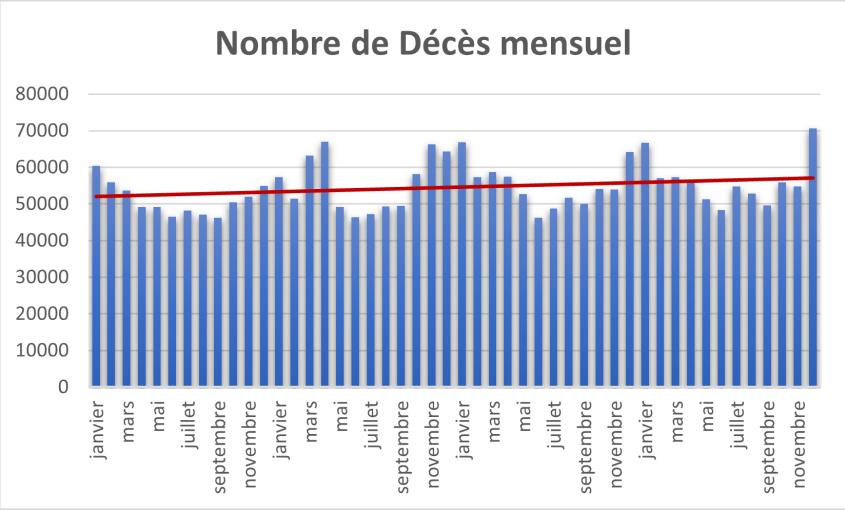
Les graphiques présents sur ce fichier sont le résultat de manipulations sur le fichier Excel : `Donnée_Illustration_théorique.xlsx`

A) La transformée de Fourier

Prenons une série temporelle, celle-ci se caractérise par :

- l'évolution des observations de notre série au fil du temps, que l'on peut visualiser à partir d'une courbe ;
- son évolution fréquentielle, représentée par une série de pics, permettant de connaître la composition structurelle de notre série en détectant par exemple un cycle mensuel, trimestriel, annuel... ou bien des pics extrêmes, représentation d'événements majeurs influant fortement sur le signal de notre série.

Prenons comme série temporelle la mortalité en France à l'échelle nationale.
L'évolution de cette série dans le temps s'illustre au travers du graphique ci-dessous :



On peut y observer une tendance légèrement croissante dans le temps, des cycles périodiques, hauts en hiver, bas en été.

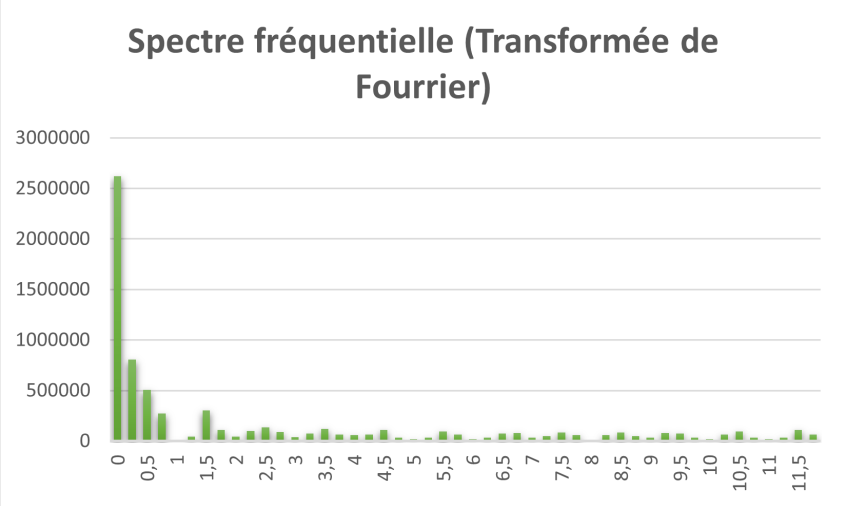
Cette composition de cycles peut être observée via notre graphique, mais pour la confirmer nous devons observer en détail le spectre des fréquences.
Ce spectre s'obtient en appliquant la transformée de Fourier sur les données de notre série temporelle.
Voici, dans un cadre discret, la formule utilisée :

$$X(f) = \sum_{t=0}^{N-1} x_t \, e^{-2i\pi ft/N}$$

Avec :

- x_t : la valeur de la série au temps t
- f : l'indice fréquentiel

Ci-dessous, le spectre obtenu suite à l'application de la transformée de Fourier sur nos données de mortalité nationale :



Le premier pic, considéré comme le pic numéro zéro, n'est pas à interpréter.
Notre analyse commence donc à partir du second pic.
Ici, celui-ci est plutôt élevé, signe d'une saisonnalité annuelle de notre série.
On observe un second pic encore significatif, ce qui signifie que notre signal a une saisonnalité semestrielle.
Après celui-ci, les pics sont tous très bas, il n'y a donc pas d'autres structures au sein de nos données.
Cela ne représente que du bruit.
Le spectre obtenu vient donc confirmer notre analyse sur une saisonnalité annuelle ; en revanche, il vient compléter l'information en nous informant d'une saisonnalité semestrielle.
Celle-ci pourrait représenter une hausse de mortalité au cours de la période hivernale avec une épidémie de grippe récurrente ou bien une hausse en été liée aux fortes périodes de canicule chaque année, voire les deux.
Dans ce cas précis, nous faisons face à la limite de ce que permet de connaître la transformée de Fourier.
On peut connaître la composition de notre signal et effectuer des suppositions à partir

de la connaissance de nos données, mais il est impossible d'affirmer statistiquement la localisation dans le temps de cette composition.
Afin de pallier ce problème, c'est là qu'intervient notre deuxième notion, les ondelettes.

B) Les ondelettes

Pour commencer, les ondelettes sont des fonctions qui valident les propriétés suivantes :

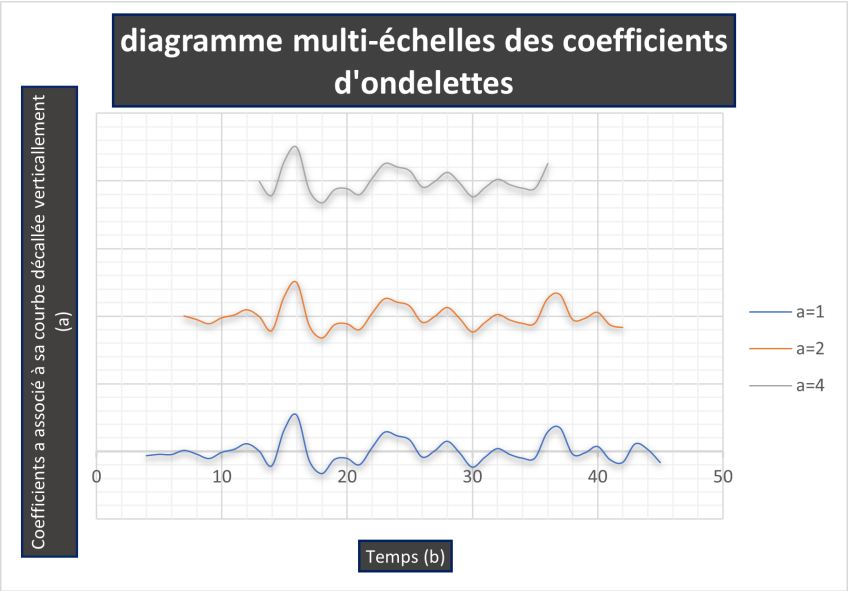
- Une moyenne nulle :
$$\int_R \psi_{a,b}(t)\,dt = 0$$
Ce qui permet de s'assurer que l'ondelette détecte bien le signal local et non le niveau moyen correspondant à la tendance de celui-ci.

- Localisation dans le temps :
L'ondelette est localisée dans le temps par rapport au point b, ici qui correspondrait à un mois fixé. L'abscisse t correspond au déplacement dans le temps par rapport au mois b.
- Capter une certaine fréquence :
Déterminée par le paramètre a. Plus a est petit, plus l'ondelette *zoome* autour du mois et détectera plus facilement les événements et tendances mensuels ; plus a est grand, moins le zoom sera fort et plus celle-ci détectera les événements et tendances trimestrielles et annuelles.

À partir de fonctions de ce type, on peut obtenir une carte échelle-temps prenant a pour paramètre d'échelle et b comme paramètre de temps (autour de quel moment centre-t-on notre observation) à partir de notre signal $x(t)$ et de nos ondelettes $\psi_{a,b}(t)$:

$$W(a,b) = \int_R x(t)\,\psi_{a,b}(t)\,dt$$

Ici, comme nous sommes dans un cas discret, nous effectuons un produit scalaire entre le vecteur des $x(t)$ et celui des $\psi_{a,b}(t)$.
En appliquant à nos données, on obtient le graphique ci-dessous :



Pour les ondelettes, nous avons choisi la famille des ondelettes de Ricker, aussi appelée *Mexican Hat* :

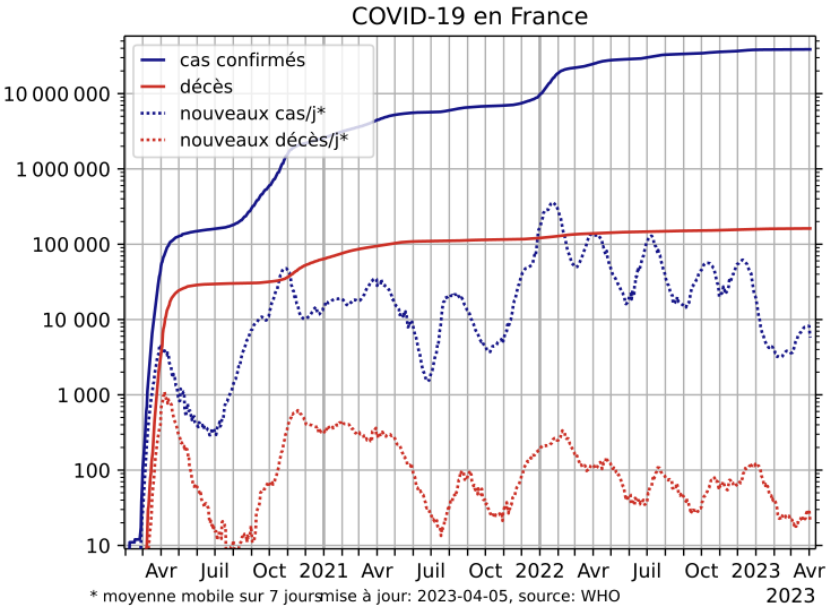
$$\psi_a(t) = \left(1 - \left(\frac{t}{a}\right)^2\right) e^{-\frac{t^2}{2a^2}}$$

Nous avons utilisé la formule généralisée en fonction de a utilisée dans les travaux d'Hammond.
Afin d'améliorer la lecture et l'interprétation des résultats, un coefficient de normalisation a été ajouté. Sans ce coefficient, les résultats auraient explosé.
Nos données vont de l'année 2019 à l'année 2022 incluse.
On considère comme intervalle pour t : $[-3a\,;\,3a]$, choisi car nos observations sont mensuelles.
Cela implique :

- Pour $a = 1$:
  le premier point b se situe en **avril 2019** (janvier 2019 + 4), et le dernier en **septembre 2022** (décembre 2022 – 4).
  ⇒ Correspond à une **fréquence semestrielle** (6 mois d'amplitude autour de b).
- Pour $a = 2$:
  le premier point b se situe en **juillet 2019** (janvier 2019 + 7), et le dernier en **juin 2022** (décembre 2022 – 7).
  ⇒ Correspond à une **fréquence annuelle** (12 mois d'amplitude autour de b).
- Pour $a = 4$:
  le premier point b se situera à janvier 2019 + 13 donc à janvier 2020 et le dernier à décembre 2022 - 13 donc à décembre 2021.
  ⇒ Correspond à une fréquence de 2 ans (24 mois d'amplitude autour de b).

Sur notre graphique, les courbes des différentes échelles ont l'air très similaires car l'écart est biaisé par notre double normalisation (dans la formule de Ricker pour ne pas avoir des résultats qui explosent, puis dans les coefficients afin de pouvoir superposer les graphes).

Sans cela, les variations auraient été plus prononcées sur les courbes où a était petit et plus lissées sur les courbes où a était plus grand.
Concernant l'interprétation de notre graphique maintenant.
Chacune des courbes nous permet de détecter différents cycles comme précisé plus haut, ainsi que de les localiser.
Cela nous permet donc de répondre à la question posée au moment de l'interprétation du spectre obtenu grâce à la transformée de Fourier sur le cycle semestriel.
Lorsque l'on observe la courbe $a = 1$, spécifique aux variations semestrielles, on constate une alternance marquée entre hausse et baisse du signal, tous les 6 mois sur les années 2020, 2021.
Celle-ci s'atténue fortement en 2022 et ne semble pas présente en 2019.
On observe aussi des pics significatifs communs aux 3 courbes, notamment de mars 2020 à avril 2020 et un autre de novembre 2021 à décembre 2021.
La présence de pics communs aux 3 échelles signifie la présence d'un événement de forte ampleur qui, quelle que soit sa durée, a quand même été détecté par nos 3 échelles.
À l'inverse, si on observe un pic présent uniquement pour une petite échelle, celui-ci s'apparente sûrement à du bruit.
En observant sur le graphe ci-dessous extrait de Wikipédia, illustrant les morts du Covid en fonction du mois, la courbe présentant le nombre de nouveaux décès, on constate un premier pic correspondant à la première vague de mortalité de l'épidémie de COVID, de mars à avril 2020.
On observe une autre montée importante sur la période de novembre 2021 à décembre 2021.
En effectuant le parallèle, on peut ainsi interpréter l'origine de nos deux pics multiscalaires de notre carte temps-échelle.



Nos ondelettes nous permettent donc de dater les éléments que l'on distingue au sein de notre composition de fréquences, nette amélioration comparée à la transformée de Fourier.

Néanmoins, les ondelettes présentent aussi leurs limites.
La première s'illustre sur notre graphique.
Plus notre échelle a est grande, plus il nous faut un nombre important de données avant et après afin de représenter la fréquence autour du point b choisi (causé par l'intervalle $[-3a\,;\,3a]$ mentionné plus haut).
La seconde limite est la raison pour laquelle les SGWT ont été développées. Si désormais on veut observer notre fréquence sur une échelle autre que le temps dont la relation entre le premier et le second point n'est pas linéaire, nos ondelettes ne sont plus du tout performantes.
On parle de relation linéaire pour le temps car on peut modéliser chacun de ces points sur une droite.
En revanche, il y a d'autres échelles, comme l'échelle géographique, dans notre étude de cas, qui ne sont pas linéaires.
Les points ne peuvent pas être représentés sur une droite mais sur une carte.

C) Les Spectral Graph Wavelets

Notre carte est représentée par un graphe que l'on notera G tel que :
$G = (V, E)$ où :

- V représente l'ensemble des sommets du graphe, dans l'exemple nos régions.
- E représente l'ensemble des arêtes (arc si graphe orienté), dans notre exemple, arêtes entre régions voisines.

Ici, nous allons considérer que notre graphe est non orienté, c'est-à-dire que le poids des arêtes est le même d'un sommet à l'autre.

1. Le Laplacien du graphe :

On note $W = [w_{i,j}]$ matrice de pondération du graphe, où $w_{i,j}$ représente le poids de la relation reliant le sommet i au sommet j.
Le graphe étant non orienté ($w_{i,j} = w_{j,i}$), la matrice W est donc **symétrique**.
Dans le cadre où $w_{i,j}$ sont binaires, on parle de matrice d'adjacence (lien entre sommet i et sommet j ou non).
Enfin, on note L, Laplacien de notre graphe, matrice symétrique telle que :

$$L = D - W, \qquad D_{ii} = \sum_j w_{ij}$$

Ici, la matrice D, aussi appelée matrice des degrés, est une matrice diagonale dont chaque coefficient D_{ii} correspond au nombre d'arêtes incidentes au sommet i.

2. Les ondelettes spectrales :

On note ensuite le spectre du Laplacien :

$$\mathrm{Spec}(L) = \{\lambda_1, \ldots, \lambda_n\},$$

l'ensemble des valeurs propres de L.

Comme L est symétrique et semi-défini positif, il est diagonalisable, tel que :

$$L = U\Delta U^\top,$$

où :

- Δ est une matrice diagonale composée des valeurs propres de L,
- U est la matrice dont les colonnes sont les vecteurs propres associés, de sorte que $\{u_1, \ldots, u_n\}$ forme une base orthogonale.

On prend le spectre de L comme *base de Fourier généralisée*, tel que :

- les vecteurs propres sont associés aux modes "fréquentiels" du graphe,
- les valeurs propres correspondent aux fréquences associées.

On définit enfin une fonction notée g, appelée *filtre spectral*, prenant en paramètres :

- la fréquence du graphe λ (valeur propre, définie par le graphe),
- l'échelle a (paramètre choisi en fonction de ce que l'on souhaite observer dans le signal).

On ajoutera comme condition que g appartient à l'espace L^2.
On a donc :

$$g(aL) = U \operatorname{diag}\Big(g(a\lambda_1), \ldots, g(a\lambda_n)\Big)\, U^\top.$$

Si l'on suppose que g est décroissante :

- **Cas** $a > 1$ **:**
  Plus a est grand, plus on sera sensible aux basses fréquences. En effet, g, décroissante, diminue plus rapidement et détecte surtout les petites valeurs propres.
  ⇒ On pourra analyser les tendances **globales**.

- **Cas** $0 < a < 1$ **:**
  Plus sensible aux hautes fréquences, car g décroît plus lentement. Ainsi, plus a est petit, plus on détectera les grandes valeurs propres.
  ⇒ On pourra analyser des détails de plus en plus précis (effet *zoom*).

On définit alors une fonction g qui oscille, de manière dépendante du choix de a, afin d'isoler les fréquences par ondelette spectrale ϕ_a tel que :

$$\psi_a = g(aL)f$$

Ici, f est le signal du graphe, c'est-à-dire une fonction qui, à chaque sommet du graphe, associe un scalaire.

Ainsi :

- les ondelettes spectrales définies pour des petites valeurs de a permettent d'observer des **détails locaux**,

- celles définies pour de grandes valeurs de a révèlent les **structures globales**.

Enfin, le calcul exact de $g(aL)$ peut s'avérer coûteux. Afin de gagner en efficacité, on peut choisir de l'approximer par un polynôme de Tchebyshev.

3. Approximation polynomiale (Tchebyshev) :

On peut définir les polynômes de Tchebyshev de deux façons :

- **Polynômes de première espèce** :
  il s'agit de toute famille de polynômes T_n validant la relation

$$T_n(\cos\theta) = \cos(n\theta).$$

- **Polynômes de seconde espèce** :
  il s'agit de toute famille de polynômes U_n validant la relation

$$U_n(\cos\theta) = \frac{\sin\Big((n+1)\theta\Big)}{\sin(\theta)}.$$

Ces familles de polynômes possèdent trois grandes propriétés :

- **Orthogonalité** sur l'intervalle $[-1, 1]$, à condition d'appliquer un poids au produit scalaire :

$$\int_{-1}^{1} \frac{T_m(x)\,T_n(x)}{\sqrt{1-x^2}}\,dx = \begin{cases} 0, & m \neq n, \\ \pi, & m = n = 0, \\ \frac{\pi}{2}, & m = n \neq 0. \end{cases}$$

- **Relation de récurrence** :

$$T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x), \qquad T_0(x) = 1,\ T_1(x) = x.$$

- **Racines** données par la formule :

$$x_k = \cos\left(\frac{2k-1}{2n}\pi\right), \qquad k = 1, \ldots, n.$$

À partir de la propriété d'orthogonalité, on peut en déduire que :

Avec le poids $w(x) = \frac{1}{\sqrt{1-x^2}}$, les polynômes de Tchebyshev $\{T_n\}$ forment une **famille orthogonale complète** dans $L^2([-1, 1], w)$.

En effet, d'après le théorème de Weierstrass, les polynômes sont denses dans l'espace des fonctions continues muni de la norme uniforme.
Ainsi, toute fonction continue peut être approchée par une combinaison linéaire de polynômes.

On peut étendre ce résultat aux ensembles plus larges :
$L^2([-1, 1], w(x)\,dx)$, pour tout poids $w(x) > 0$
les polynômes de Tchebyshev sont donc **denses dans** L^2.
De plus, comme $g \in L^2$, on peut approximer g par une combinaison linéaire de polynômes de Tchebyshev :

$$g(x) \approx \sum_{k=0}^{K} c_k\,T_k(x),$$

où c_k désignent les coefficients de projection de g sur la base des polynômes de Tchebyshev que l'on calculera de cette façon :

$$c_k = \frac{\langle g, T_k \rangle}{\langle T_k, T_k \rangle}, \qquad \text{(possible seulement grâce aux résultats d'orthogonalité sur l'intervalle } [-1, 1]).$$

Il est nécessaire d'avoir notre spectre dans l'intervalle $[-1, 1]$, pour s'en assurer on peut utiliser un Laplacien normalisé.
Ce qui nous permet d'obtenir la formule :

$$c_k = \frac{2}{\pi} \int_{-1}^{1} \frac{g(x)\, T_k(x)}{\sqrt{1 - x^2}}\, dx \qquad \text{(avec ajustement pour } k = 0).$$

On peut appliquer ces résultats à αL.
La deuxième propriété (récurrence) nous permet de calculer facilement les T_k sans avoir à élever L (matrice) à des puissances très élevées, ce qui réduit considérablement le coût de calcul.

Il ne reste plus qu'à tracer les $w(a, i)$, un par sommet.
On effectue ce traitement pour plusieurs valeurs de a, ce qui permet aux *Spectral Graph Wavelet Transform* (SGWT) de capter le contenu, même non linéaire, au sein de notre jeu de données.

Bien que développées en 2011, les *Spectral Graph Wavelets* (SGW) sont toujours d'actualité, car elles constituent la base de nombreuses méthodes modernes (Graph Neural Networks, deep learning sur graphes).
Elles peuvent donc parfaitement s'adapter à des problématiques contemporaines, notamment en **Actuariat Vie**, comme nous allons l'illustrer dans l'exemple pratique ci-dessous.

II/ Application : Étude du taux de mortalité à partir des SGWT

L'objectif est, d'une part, de capter les variations locales (épidémies, canicules, chocs sanitaires) et, d'autre part, de mettre en évidence les tendances structurelles partagées entre régions.
Enfin, nous tenterons de construire un modèle de prédiction à partir des coefficients, et nous comparerons les résultats aux modèles couramment utilisés dans ce type de cadre.

D'un point de vue actuariel, l'analyse multi-échelle de la mortalité présente un double intérêt. À petite échelle, elle permet de détecter les variations locales, souvent associées à des événements extrêmes (canicule, épidémie), cruciales pour le suivi de la sinistralité à court terme. À grande échelle, elle met en évidence les tendances structurelles et régionales de mortalité, indispensables pour la tarification et le provisionnement, la construction de tables régionales et la projection de la longévité dans un cadre réglementaire.

Nous réaliserons cette application en Python.
Fichier d'outils : `Tools.py`.
Pour cela, nous suivrons un protocole détaillé (détails dans les notebooks).

A) Préparation des données :

Nous avons construit notre base de données à partir :

- des données de mortalité par département / jour de 2018 à 2023.
  Source : *Téléchargement des fichiers des décès quotidiens — Nombre de décès quotidiens* | Insee
- des données de population par années / région.
  Source : *Estimation de la population au 1er janvier 2025* | Insee

La base de données finale comporte une ligne par année / mois / région, avec l'information sur ces trois variables ainsi que sur le taux de mortalité.

Pour obtenir le taux de mortalité, nous avons pris :

$$\text{Taux de mortalité}_{\text{année, mois, région}} = \frac{\text{nombre de morts}_{\text{année, mois, région}}}{\text{population}_{\text{année, région}}}.$$

Nous avons divisé par la **population annuelle** (et non par la population année / mois), car l'évolution mensuelle de la population est trop faible pour être significative ; complexifier la base « population » ne nous aurait donc fait gagner que très peu en précision.

Lien Notebook pour cette partie : `1_Construction_base_de_données.ipynb`
input :
`DC_2018_det.csv` jusqu'à `DC_2025_det.csv`
`Données_Population.csv`
output :
`Base_de_données.csv`
`Données_Mortalité_20242025.csv`
Pour les autres parties on utilisera :
`2_SGWT_Application&Modélisation.ipynb`

B) Analyse des tendances structurelles à partir des SGWT :

Les inputs utilisés pour cette partie sont les deux outputs de la première.
On n'exporte aucun fichier dans cette partie.

1) Construction du graphe :

Dans un premier temps, nous avons choisi d'exclure la Corse et l'Outre-mer des sommets d'étude.
En effet, notre approche au niveau des sommets et de leurs voisins est géographique.
En ce sens, ces deux régions, isolées par la mer, n'ont pas de voisins dans le graphe métropolitain ; l'analyse des dépendances géographiques y est donc peu pertinente.

Nous avons ensuite défini visuellement un dictionnaire des voisinages, regroupant nos régions par proximité géographique. Nous avons attribué un poids de 1 à chaque arête entre régions d'un même voisinage.
Nous avons aussi créé un dictionnaire contenant les coordonnées de chaque région à partir de la longitude et de la latitude du centre de chacune d'elles (objectif : améliorer la lisibilité de nos sorties graphiques).

Il aurait été possible de gagner en précision en utilisant des algorithmes comme *k-NN* que l'on aurait appliqués sur les coordonnées géographiques des régions.
Nous aurions aussi pu nuancer les poids en ne les fixant pas tous à 1, mais en les différenciant selon la proximité géographique au sein d'un même voisinage.
Ces choix n'ont pas été retenus, car l'optimisation qu'ils permettent ne semble pas essentielle au regard de la complexité ajoutée.

Pour finir, nous avons calculé le Laplacien normalisé. Ce choix est motivé par la différence potentielle du nombre de voisins entre les régions, différence fortement réduite par la normalisation du Laplacien.

2) Définition des SGWT :

Dans cette section, nous avons refait les calculs précédents, cette fois-ci avec `PyGSP`, afin de gagner en temps de calcul en évitant le calcul des valeurs propres et la diagonalisation du Laplacien (coûteux pour les grands graphes).
À la place, `PyGSP` estime la valeur propre maximale $\lambda_{\max}$ pour normaliser le spectre de L dans $[-1, 1]$ et approcher $g(L)$ sans avoir à diagonaliser le Laplacien.
L'objectif est d'appliquer un processus général et réutilisable, quelle que soit la taille de nos données.

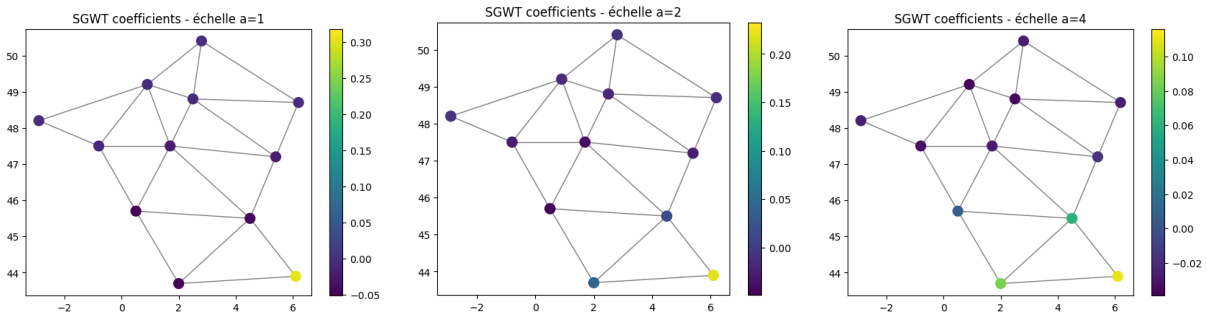
Comme mentionné dans la partie théorique, nous avons paramétré `PyGSP` afin d'approximer notre fonction spectrale à l'aide des polynômes de Tchebyshev.
Nous calculons aussi L_{sym} (Laplacien normalisé), que l'on recale ensuite pour s'assurer que son spectre tombe dans $[-1, 1]$, ce qui permet d'utiliser les polynômes de Tchebyshev, à partir de la valeur propre maximale estimée.

On choisit des filtres bande passante définis pour chaque échelle a ainsi :

$$g_a(\lambda) \;=\; (a\,\lambda)\, e^{-a\lambda}, \qquad \lambda \geq 0.$$

Ce choix vérifie $g_a(0) = 0$ (moyenne nulle) et est centré autour de $\lambda^\star = \frac{1}{a}$ (bande passante).
On choisit d'appliquer $a \in \{1, 2, 4\}$, échelles nous permettant de couvrir le spectre sans multiplier inutilement le nombre de courbes.

Nous appliquons ensuite nos filtres à un signal défini comme un Dirac au nœud choisi aléatoirement ; le choix du nœud importe peu, car il n'impacte pas les résultats finaux.
L'objectif est de visualiser le comportement du filtre ainsi que l'approximation de Tchebyshev sur les coefficients d'ondelettes au travers des trois graphiques ci-dessous :



Voici donc le résultat de la cartographie de nos coefficients calculés pour une échelle donnée. On rappelle que l'impulsion de Dirac choisie arbitrairement pointe ici sur le sommet représentant la région *Provence-Alpes-Côte d'Azur*. On rappelle que chaque point porte les coordonnées géographiques du centre des différentes régions, ce qui explique l'aspect déformé par rapport à la forme réelle de la France. La structure des graphes (arêtes, sommets, position, etc.) représente bien la définition que l'on a voulu donner.

Concernant l'application de SGWT et le calcul des coefficients : on observe que plus l'échelle est petite, plus la valeur des coefficients est forte pour les sommets situés près

du sommet choisi, et inversement si les sommets sont éloignés — logique, car plus a est petit, plus l'échelle est locale. Plus a est grand, plus l'écart de coefficient entre les points voisins du sommet choisi et ceux qui en sont éloignés est faible — logique, car plus a est grand, moins l'échelle d'observation est locale. Pour $a = 1$, on observe que le seul coefficient réellement détecté est celui de la région pointée, signe que l'échelle régionale est bien définie en $a = 1$ (pareil pour les autres). Notre graphe et nos ondelettes sont donc définis de façon cohérente, on peut passer à la suite de l'application.

3) Visualisation et interprétation des résultats de l'application des SGWT au taux de mortalité

Afin de visualiser l'évolution de nos structures spatiales du taux de mortalité dans le temps, nous allons tracer un *scalogramme* (temps × échelles) dont le principe est le suivant :
À chaque date t, on considère le vecteur de signal $f_t \in R^N$ (par ex. mortalité par région). On applique la banque de filtres $\{g_a(\lambda)\}_{a\in\mathcal{A}}$ (échelles $a_1, \ldots, a_S$) sur le spectre de L :

$$\forall a \in \mathcal{A}, \qquad c_{t,a} \;=\; g_a(L)\, f_t \;\in\; R^N.$$

En empilant les colonnes, on obtient la matrice des coefficients

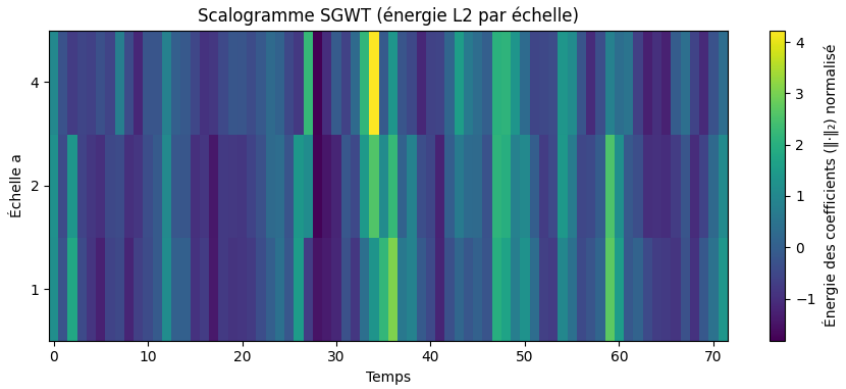
$$C_t \;=\; \left[\, c_{t,a_1} \;\; c_{t,a_2} \;\; \ldots \;\; c_{t,a_S} \,\right] \;\in\; R^{N\times S} \qquad (N \text{ sommets},\; S \text{ échelles}).$$

Pour tracer un *scalogramme*, on agrège par échelle (sur les sommets) une énergie des coefficients, par exemple la norme L^2 :

$$E(t,a) \;=\; \left\| c_{t,a} \right\|_2 \;=\; \left\| C_t(:,a) \right\|_2.$$

On affiche ensuite la **carte 2D** : axe x = temps t, axe y = échelles a, couleur = énergie $E(t,a)$.

Nous avons utilisé la norme L^2 plutôt qu'une autre norme comme L^1 ou L^∞ pour agréger nos coefficients régionaux par échelle, car elle conserve au mieux l'énergie des coefficients et est donc la plus adaptée à la comparaison visuelle.
Ensuite, nous normalisons le vecteur d'énergie par échelle, ce qui permet la comparaison visuelle.



Sur notre scalogramme, on observe que :

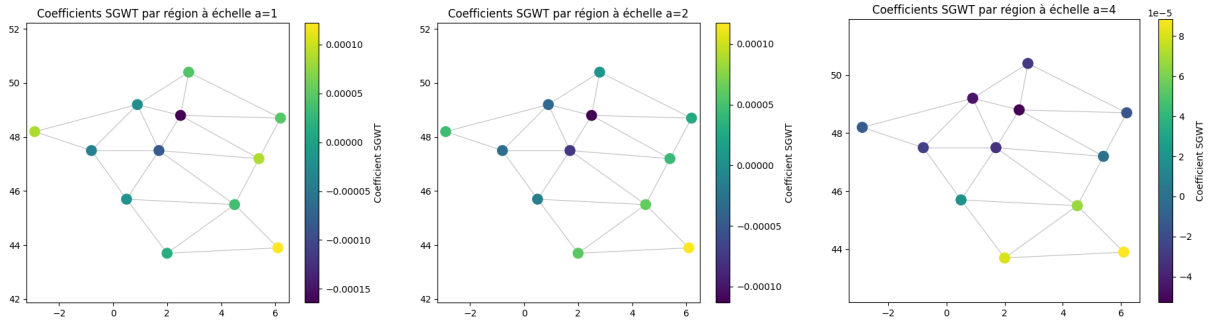
– il est plus clair sur la période 2020–2023, ce qui correspond à un fort taux de mortalité lié à la période COVID ;

– il est plutôt homogène entre les échelles, ce qui signifie que sur cette période il y a eu peu d'événements strictement locaux justifiant une hausse de mortalité, mais davantage d'événements globaux ;

– il affiche des pics plus importants sur les périodes novembre/décembre 2020 et janvier/mars 2022, correspondant aux pics COVID.

En résumé, ce qui ressort principalement de ce graphique est une hausse du taux de mortalité liée au COVID, avec des mois plus élevés que d'autres.
Nous pouvons maintenant zoomer sur les pics les plus importants pour voir quelles régions ont été les plus vulnérables.
Afin de mieux interpréter nos résultats, nous affichons ci-dessous une carte des régions françaises :

Ici, nous le faisons pour $t = 34$; voici les graphiques obtenus :



À une échelle plus locale, à l'ouest, la Bretagne semble avoir été la plus touchée ; au nord, le Nord–Pas-de-Calais Picardie. À l'échelle globale, le Sud apparaît comme la zone la plus touchée, avec dans le top 3 national les trois régions sudistes (coloration jaune-verte).

L'Île-de-France, quant à elle, semble faire partie des régions les moins touchées ; cela s'explique par une population très élevée (normalisation par la population) et une population plus jeune (donc moins vulnérable au COVID).
Dans un contexte actuariel de provisionnement ou de tarification, pour savoir où nos risques sont les plus importants, on pourrait multiplier les coefficients par une part de répartition régionale de notre portefeuille, afin d'obtenir une cartographie des risques propre à la répartition de nos sociétaires.

C) Modélisation du taux de mortalité en France en fonction de la région et du temps :

L'objectif est de voir comment les SGWT peuvent nous aider à établir un modèle de prédiction de notre taux de mortalité (2024–2025) et de le comparer aux modèles classiques de prédiction de série temporelle utilisés.

1) Modélisation classique

Dans un premier temps, nous avons décidé d'établir plusieurs modèles de prédiction classiques du taux de mortalité, afin de comparer le meilleur d'entre eux au modèle de prédiction élaboré à partir des coefficients des SGWT.
Nous avons choisi comme premier modèle, pour chaque région, un `SARIMA(1,1,1)(1,1,1)` afin d'initier une base de prédiction, sans aucune optimisation des paramètres.
En second modèle, nous avons appliqué la fonction `auto.arima()` afin qu'elle sélectionne le SARIMA avec les paramètres optimaux pour chacune des régions.
Enfin, en dernier modèle, nous avons utilisé la bibliothèque `Prophet`, qui permet d'optimiser rapidement un modèle prédictif pour une série temporelle.
Le projet n'étant pas centré sur ce type de modèles, nous passons les détails de la modélisation, détails que vous trouverez dans le notebook mentionné plus haut.

Afin de sélectionner le meilleur modèle, nous avons d'abord visualisé les métriques de test moyennes sur l'ensemble des régions (MAE, RMSE, MAPE).

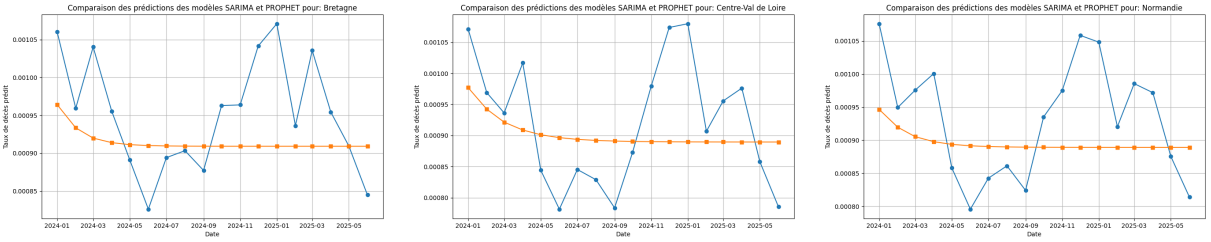
Voici le tableau comparatif des métriques pour les trois modèles :

	Modèle	MAE	RMSE	MAPE
0	Modèle 1	0.000083	0.000100	9.579298
1	Modèle 2	0.000060	0.000080	7.002441
2	Modèle 3	0.000057	0.000075	7.129032

La modélisation `Prophet` apparaît donc, en moyenne, comme plus efficace. Pour confirmer ce choix face à `auto.arima()` (le modèle initial étant d'office hors-jeu), nous avons ensuite visualisé, région par région, les courbes de prédictions superposées ; tous les graphiques sont présentés dans le notebook.
Il est aussi important de visualiser, car notre taux de mortalité étant très faible (d'échelle 10^{-4}), un écart apparemment modeste entre RMSE ou MAE peut, en réalité, être significatif.
Pour certaines régions, les courbes sont très similaires ; pour d'autres, on observe qu'`auto.arima()`

ne capte pas du tout la tendance.
Cela est certainement lié, pour certaines régions, à la non-validité des hypothèses d'indépendance et de normalité des résidus, indispensables à la pertinence et à la convergence des processus SARIMA.

En voici trois exemples ci-dessous :



On en conclut que le modèle `Prophet` modélise mieux les taux de mortalité dans le temps : il capte mieux les tendances et les variations, même s'il a tendance à les atténuer légèrement.
Comme comparaison à notre modèle SGWT, nous retiendrons donc le modèle `Prophet`.

2) Modèle de prédiction à partir des SGWT

L'objectif ici est de développer un modèle de prédiction à partir des données de l'année n pour prévoir les années $n+1$ et $n+2$.
Nous avons d'abord calculé les *features* notées $w(a,i,t)$ ainsi que les $w(a,i,t-1)$, avec un décalage de 1 jusqu'à $w(a,i,t-12)$ (décalage de 12 mois).
On obtient donc 36 coefficients par couple (date, région). Le tout nous donne un tableau noté X.
Nous retirons de X le *lag* 0 pour les trois échelles, car il comporte dans son calcul l'information sur la valeur réelle du taux de mortalité, ce qui viendrait biaiser complètement la prédiction.
Dans un autre tableau noté y, nous avons reporté le taux de mortalité associé à chaque couple (date, région), que nous avons pris soin d'aligner avec le tableau X.
Ensuite, nous retirons de X et y les 148 premières lignes, associées à la première année utilisée, qui seront logiquement en `NaN` (il faut un passif d'un an pour calculer l'ensemble des coefficients).
Après cela, nous disposons d'un jeu de données (*dataset*) propre, correspondant à notre objectif, à partir duquel on peut entraîner nos modèles de prédiction.

Une fois le modèle entraîné, nous prédisons, à partir des coefficients $w(a,i,t-1)$ correspondant à décembre 2023 jusqu'à $w(a,i,t-12)$ correspondant à décembre 2022, le taux de mortalité de janvier 2024. Puis, pour février 2024, nous utilisons $w(a,i,t-1)$ prédit et les 11 autres réellement observés, et ainsi de suite jusqu'à la dernière valeur à prédire. Ainsi, nous aurons effectivement modélisé 2024 et 2025 uniquement à partir des données de 2023.

Nous avons décidé d'entraîner trois modèles, dont les paramètres ont tous été optimisés via une *GridSearch* en fonction du critère de RMSE moyen sur l'ensemble des régions :

– régression LASSO ;

– Random Forest (forêts aléatoires) ;
– Gradient Boosting.

Nos *GridSearch* nous ont permis d'obtenir les résultats suivants :

```
'Lasso best alpha:'

1.8329807108324375e-06

'Best CV RMSE:'

0.00013367660502103446

'R² train (meilleur Lasso):'

0.35753025250231096

'R² test  (meilleur Lasso):'

0.4335892376523127
```

```
'RF best params:'

{'max_depth': 20,
 'max_features': 'sqrt',
 'min_samples_leaf': 2,
 'min_samples_split': 2,
 'n_estimators': 300}

'Best CV RMSE:'

0.0001141771085860455

'R² train (meilleur Lasso):'

0.8992795506640167

'R² test  (meilleur Lasso):'

0.7331650772755832
```

```
'GB best params:'

{'l2_regularization': 0.1,
 'learning_rate': 0.05,
 'max_bins': 255,
 'max_iter': 800,
 'max_leaf_nodes': 31,
 'min_samples_leaf': 50}

'Best CV RMSE:'

0.00013586979888624142

'R² train (meilleur Lasso):'

0.531487958700791

'R² test  (meilleur Lasso):'

0.5791711368704688
```

On observe donc que, sur absolument tous les critères (RMSE minimisé, R^2 test maximisé), notre modèle de Random Forest est celui qui semble être le plus efficace.
Attention tout de même : le fort écart entre R^2_{test} et R^2_{train} témoigne d'un fort surapprentissage de notre modèle.

3) Comparaison des modèles

On applique donc notre modèle *Random Forest* afin de prédire le taux de mortalité 2024 et 2025, comme indiqué plus haut, et l'on obtient les résultats suivants, comparés aux résultats de nos baselines rappelées à droite :

	Modèle	MAE	RMSE	MAPE
0	RandomForest	0.000071	0.000095	8.242904

	Modèle	MAE	RMSE	MAPE
0	Modèle 1	0.000083	0.000100	9.579298
1	Modèle 2	0.000060	0.000080	7.002441
2	Modèle 3	0.000057	0.000075	7.129032

On observe clairement qu'en moyenne, notre modèle prédictif créé à partir des coefficients des SGWT est moins bon que les modèles classiques utilisés (`auto.arima()` et `Prophet`).
En zoomant de plus près, région par région, on observe que, pour certaines régions, le modèle obtenu à partir des SGWT offre tout de même des prédictions plus précises que `Prophet`.

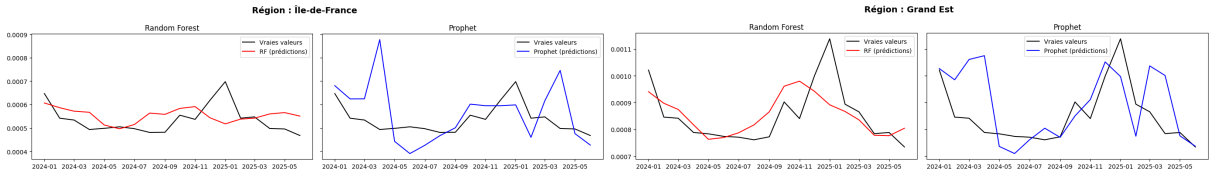
Sur le plan statistique :

	Modèle	Région	MAE	RMSE	MAPE
0	RandomForest	Auvergne-Rhône-Alpes	0.000072	0.000091	9.746571
1	RandomForest	Bourgogne-Franche-Comté	0.000120	0.000159	11.843997
2	RandomForest	Bretagne	0.000056	0.000083	5.848674
3	RandomForest	Centre-Val de Loire	0.000064	0.000101	6.372781
4	RandomForest	Grand Est	0.000057	0.000081	6.388311
5	RandomForest	Hauts-de-France	0.000069	0.000081	8.444486
6	RandomForest	Normandie	0.000070	0.000100	7.270891
7	RandomForest	Nouvelle-Aquitaine	0.000063	0.000088	6.352232
8	RandomForest	Occitanie	0.000080	0.000103	9.103935
9	RandomForest	Pays de la Loire	0.000059	0.000076	7.238667
10	RandomForest	Provence-Alpes-Côte d'Azur	0.000089	0.000112	10.471676
11	RandomForest	Île-de-France	0.000053	0.000067	9.832622

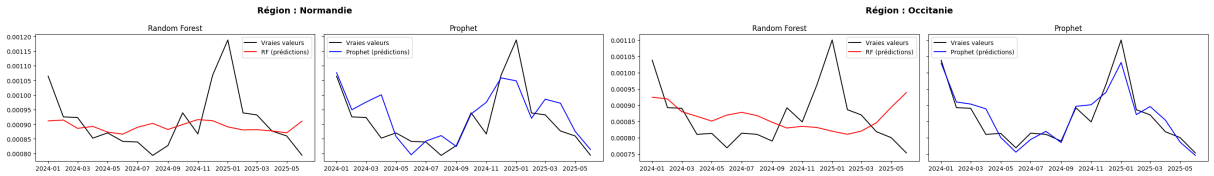
	Modèle	Région	MAE	RMSE	MAPE
24	Modèle 3	Auvergne-Rhône-Alpes	0.000071	0.000095	9.493740
25	Modèle 3	Bourgogne-Franche-Comté	0.000090	0.000113	9.370209
26	Modèle 3	Bretagne	0.000055	0.000064	6.089327
27	Modèle 3	Centre-Val de Loire	0.000057	0.000077	6.030987
28	Modèle 3	Grand Est	0.000092	0.000125	10.875018
29	Modèle 3	Hauts-de-France	0.000059	0.000077	7.383700
30	Modèle 3	Normandie	0.000046	0.000065	5.071508
31	Modèle 3	Nouvelle-Aquitaine	0.000029	0.000036	2.959657
32	Modèle 3	Occitanie	0.000024	0.000032	2.699300
33	Modèle 3	Pays de la Loire	0.000038	0.000052	4.700375
34	Modèle 3	Provence-Alpes-Côte d'Azur	0.000037	0.000046	4.339397
35	Modèle 3	Île-de-France	0.000086	0.000124	16.535173

On observe que nos prédictions sont meilleures pour les régions Grand Est et Île-de-France.

Sur le plan visuel, cela se confirme :



Néanmoins, on observe aussi que, sur d'autres régions, l'écart est flagrant en faveur de `Prophet` :



Conclusion

On peut donc conclure qu'à l'échelle nationale, notre modélisation `Prophet` est plus précise que la *Random Forest* entraînée à partir des coefficients de notre SGWT.
Néanmoins, cette méthode reste précise et les résultats ne sont pas si éloignés.
Les SGWT offrent donc une alternative de modélisation très intéressante aux techniques habituellement utilisées pour la prédiction de séries temporelles.
En actuariat, il est important de disposer de plusieurs méthodologies afin de mieux interpréter les résultats.
On pourrait d'ailleurs choisir une approche par crédibilité en pondérant `Prophet` et *Random Forest* afin de réaliser un mix des deux.
De plus, sur certaines régions comme l'Île-de-France, nos *Random Forests* offrent des prédictions plus performantes que `Prophet`. En fonction de la distribution géographique de notre portefeuille de sociétaires, cette approche peut donc s'avérer plus intéressante (par exemple en cas de forte concentration en Île-de-France ou dans la région Grand Est), et inversement.
Enfin, au-delà de la modélisation, les SGWT apportent des informations précieuses sur les structures internes des données.
On observe par exemple une saisonnalité hiver/été marquée, des pics associés à des événements majeurs tels que l'apogée de la crise COVID, ainsi qu'une différence de réaction selon l'emplacement géographique (le sud plus touché que le nord).
Ces informations peuvent également justifier des différences de tarification ou de provisionnement selon les régions ou les périodes, adaptées à un objectif précis et orienté métier.