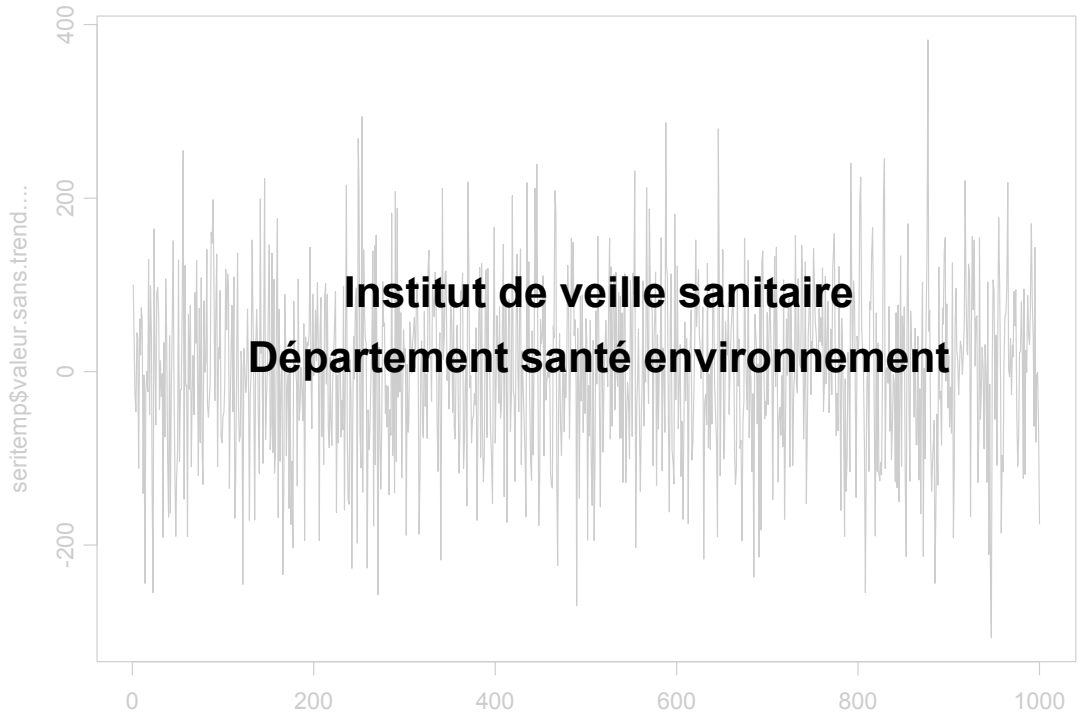
Séries temporelles et modèles de régression

Application à l'analyse des associations à court terme entre la pollution atmosphérique et la santé



Daniel Eilstein[(1)], Alain Le Tertre[(1)], Abdelkrim Zeghnoun[(1)], Sylvie Cassadou[(1)], Laurent Filleul[(1)], Laurence Pascal[(1)], Hélène Prouvost[(2)], Christophe Declercq[(2)], Philippe Saviuc[(1)], Agnès Lefranc[(3)], Catherine Nunes[(3)], Benoît Chardon[(3)], Jean-François Jusot[(1)], Myriam D'Helf[(1)], Pascal Fabre[(1)], Sylvia Médina[(1)], Philippe Quénel[(1)].

(1) Département santé-environnement, Institut de veille sanitaire, 12, rue du Val d'Osne 94415 Saint-Maurice Cedex, France.

(2) Observatoire régional de santé Nord-Pas-de-Calais. 13, rue Faidherbe 59046 Lille Cedex, France.

(3) Observatoire régional de santé d'Ile-de-France. 21-23, rue Miollis 75732 Paris Cedex, France.

Liste des acronymes, sigles, symboles et notations utilisés

Acronymes

AIC : critère d'Akaike (*Akaike-information criterion*)

Aphea : *Air Pollution and Health, a European Approach*

AR, MA, ARMA : modèles autorégressif (*autoregressive*), moyenne mobile (*moving average*), autorégressif et moyenne mobile

DSE : Département santé-environnement

GAM : modèle additif généralisé

GLM : modèle linéaire généralisé

GPS : *Global positioning system*

IID : *independently and identically distributed*

Insee : Institut national de la statistique et des études économiques

InVS : Institut de veille sanitaire

MCMC : Monte Carlo par chaîne de Markov (méthodes de)

PA : pollution atmosphérique

Psas-9 : Programme de surveillance Air & Santé-9 villes

Sigles

NO : monoxyde d'azote

NO_2 : dioxyde d'azote

SO_2 : dioxyde de soufre

Notations

De façon générale :

Une lettre minuscule « non grasse » désigne un scalaire (x),

Une lettre minuscule « grasse » désigne un vecteur (**x**),

Une lettre majuscule « grasse » désigne une matrice (**X**),

Une lettre majuscule « non grasse » désigne une variable aléatoire (X),

Une lettre majuscule, grasse et italique désigne un vecteur de variables aléatoires (***X***).

En particulier

Scalaires

$x, y, \mu, \eta, \ldots$

$x_i, y_i, \mu_i, \eta_i, \ldots$

Vecteurs

$\mathbf{x}, \mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\eta}, \ldots$
$\mathbf{x_i}, \mathbf{y_i}, \boldsymbol{\mu_i}, \boldsymbol{\eta_i}, \ldots$
a' désigne le vecteur transposé de **a**

Vecteurs colonnes

$\mathbf{y} = (y_1,y_2,\ldots,y_i,\ldots,y_n)'$

$\boldsymbol{\mu} = (\mu_1,\mu_2,\ldots,\mu_i,\ldots,\mu_n)'$

$\boldsymbol{\eta} = (\eta_1,\eta_2,\ldots,\eta_i,\ldots,\eta_n)'$

$\mathbf{x_1} = (x_{11},x_{21},\ldots,x_{i1},\ldots,x_{n1})'$

$\mathbf{x_2} = (x_{12},x_{22},\ldots,x_{i2},\ldots,x_{n2})'$

…

$\mathbf{x_j} = (x_{1j},x_{2j},\ldots,x_{ij},\ldots,x_{nj})'$

…

$\mathbf{x_p} = (x_{1p},x_{2p},\ldots,x_{ip},\ldots x_{np})'$

$\boldsymbol{\beta} = (\beta_1,\beta_2,\ldots,\beta_j,\ldots,\beta_p)'$

Vecteurs lignes

$\mathbf{x_{[1]}} = (x_{11}, x_{12},\ldots, x_{1j},\ldots x_{1p})$

$\mathbf{x_{[2]}} = (x_{21}, x_{22},\ldots, x_{2j},\ldots x_{2p})$

…

$\mathbf{x_{[i]}} = (x_{i1}, x_{i2},\ldots, x_{ij},\ldots x_{ip})$

…

$\mathbf{x_{[n]}} = (x_{n1}, x_{n2},\ldots, x_{nj},\ldots x_{np})$

Matrices

X et (x_{ij}) désignent des matrices : $\mathbf{X} = (x_{ij})$ $((i,j) \in I \times J)$ avec $I = [1,n]$, $J = [1,p]$

$$\mathbf{X} = \begin{bmatrix} x_{11} & \ldots & x_{1j} & \ldots & x_{1p} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{i1} & \ldots & x_{ij} & \ldots & x_{ip} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{n1} & \ldots & x_{nj} & \ldots & x_{np} \end{bmatrix}$$

Variables aléatoires

Variables aléatoires individuelles

X, Y, Z, …

X_i, Y_i, Z_i, …

Vecteurs de variables aléatoires

X, ***Y***, ***Z***

IN : ensemble des nombres naturels (entiers positifs ou nuls)

*Z*I : ensemble des nombres entiers

IR : ensemble des nombres réels

Autres notations

$x \to a$ quand $t \to b$: la variable x tend vers la limite a, quand la variable t tend vers la limite b. Les symboles a et b peuvent être des nombres finis ou infinis.

Repères

Il n'est pas indispensable de tout lire dans ce guide. Il est des parties plus ou moins théoriques, des moments plus ou moins faciles. Le lecteur intéressé pourra s'arrêter aux passages contenant les développements méthodologiques de certaines notions. Les autres pourront passer (ces passages). Pour faciliter la lecture, ils seront repérés par deux images : la première (des clefs) annoncera le début du passage méthodologique, la seconde (une ambulance !) sa fin.

La présence d'une astérisque (*) renvoie à une entrée dans le glossaire en fin de document.

Sommaire

Préface

C'est un véritable plaisir pour moi d'écrire quelques mots de préface à ce document original à plus d'un titre dans les productions de l'Institut de veille sanitaire.

Sur le fond, tout d'abord, le titre indique qu'il va être question de séries temporelles et de modèles de régression. Tel que, pour informatif et précis qu'il soit, il présage d'une certaine aridité. Heureusement, le sous-titre lève un peu plus le voile et, dès lors, la curiosité du lecteur commence à s'aiguiser. Ce document traite donc de l'analyse de séries temporelles aux fins d'étudier les relations entre un ou plusieurs déterminants de l'environnement et la santé humaine.

L'étude des relations entre les facteurs environnementaux et l'état de santé de la population soulève d'importants problèmes méthodologiques. Les spécificités avec lesquelles s'expriment les relations entre les déterminants environnementaux et les pathologies (faibles niveaux d'exposition, pathologies non spécifiques se manifestant souvent à long terme, exposition multifactorielle), nécessitent une réflexion permanente sur les outils méthodologiques à mettre en œuvre et leur amélioration.

En ce qui concerne les outils épidémiologiques, la mise en place des études écologiques et, notamment, des études écologiques temporelles, a permis de s'affranchir d'un certain nombre de difficultés méthodologiques (mais en en rajoutant d'autres) en recourant à des données agrégées, tant pour qualifier l'exposition que pour décrire l'état de santé d'une population donnée au sein d'une même unité géographique et au cours du temps. Il s'agit ainsi d'étudier, ici, l'association au jour le jour (le plus souvent) entre des indicateurs de pollution idoines pour caractériser au mieux l'exposition de la population d'étude et des indicateurs sanitaires relevés dans cette même population (comptes journaliers de mortalité, d'hospitalisation, de consommation médicamenteuse…). Ces données peuvent être aisément disponibles, car enregistrées en routine, par le biais de systèmes d'information existants (réseaux de surveillance de la qualité des milieux, systèmes de surveillance épidémiologique ou bases de données médico-administratives) et permettent ainsi la conduite d'études rétrospectives. L'étude de larges populations rend possible la mise en évidence des risques dits faibles grâce à une puissance convenable. Utilisant de longues séries de données et/ou travaillant sur plusieurs zones géographiques contrastées, elles permettent également d'enrichir l'information par un large spectre de niveaux d'exposition.

Le domaine des effets sanitaires à court terme de la pollution atmosphérique a vu depuis le début des années 90 se développer des études mettant en œuvre des méthodes d'analyse de séries temporelles innovantes car confrontées à de multiples contraintes. En effet, tout comme dans les études individuelles, il est nécessaire de prendre en compte dans ces analyses d'éventuels facteurs de confusion. Si le *design* écologique permet de s'affranchir des facteurs individuels comme l'âge, le tabagisme ou l'exposition professionnelle, les tiers facteurs liés temporellement à la fois à l'exposition et à l'état de santé doivent être pris en compte (comme l'activité du virus grippal ou la température) ainsi que des phénomènes d'essence temporelle qui entraînent simultanément des variations des indicateurs d'exposition et sanitaires. Il s'agit des phénomènes de tendance (on sait par exemple que les niveaux de pollution et les niveaux de mortalité diminuent au cours du temps) ou des phénomènes cycliques qui peuvent aussi affecter les différents indicateurs (on sait également que les niveaux de mortalité et de pollution atmosphérique varient en fonction de la saison).

D'autres contraintes plus spécifiques s'ajoutent encore pour rendre plus complexes les étapes de modélisation et d'analyse. Ainsi, les séries temporelles peuvent présenter des variations non cycliques (comme celles entraînées par la fermeture de lits sur l'activité hospitalière). De plus, il convient de prendre en compte la dépendance qui existe dans les séries entre des observations consécutives (autocorrélation des variables). Enfin, il s'avère souvent indispensable d'étudier la forme de la relation existant entre les séries sans *a priori.*

D'aucun pourrait se demander s'il n'est pas dans l'ambition du rédacteur de la préface de réécrire le document. Il n'en est rien ! Cette synthèse des quelques points méthodologiques clés n'est là que pour aiguillonner la curiosité du lecteur qui, à ce stade, doit se dire : « Les données utilisées sont

peut-être d'un accès facile mais une fois toutes les séries chronologiques en main, comment faire ? Quelle stratégie de modélisation ? Avec quels outils ? ». Eh bien, soyez heureux, lecteur, à l'appétit aiguisé ! Ce document apportera des réponses à toutes les questions que vous vous posez et fera même naître en vous les questions que vous ne vous posez pas encore et y répondra derechef. Car, et j'en arrive à la forme pour en finir, c'est un excellent guide pédagogique qui vous prend par la main pour vous faire traverser sans encombre les différents cercles initiatiques de la modélisation des séries temporelles. Alors, n'hésitez pas ! Plongez !

Martine Ledrans

Responsable du Département santé environnement

Institut de veille sanitaire

Préambule

En 1997, l'Institut de veille sanitaire (InVS), alors Réseau national de santé publique, a mis en place le programme Psas-9 (Programme de surveillance air et santé – 9 villes) dans neuf villes françaises (Bordeaux, Le Havre, Lille, Lyon, Marseille, Paris, Rouen, Strasbourg, Toulouse) afin de fournir des outils méthodologiques pour l'application des recommandations de la loi sur l'air et l'utilisation rationnelle de l'énergie de décembre 1996. Les deux premières phases du programme (1997-1999 et 2000-2002) ont permis d'estimer l'effet à court terme [1] de la pollution atmosphérique (PA) urbaine sur la mortalité totale, cardiovasculaire et respiratoire et les admissions en milieu hospitalier pour des motifs cardiovasculaires ou respiratoires. Au terme de cette étude, une évaluation de l'impact sanitaire (EIS) de l'exposition à court terme à la pollution atmosphérique a estimé le nombre d'événements sanitaires (décès anticipés, hospitalisations) attribuable à une augmentation du niveau de la pollution atmosphérique. La troisième phase a débuté en 2003 et a abordé un ensemble d'autres thèmes relatifs aux relations air et santé (autres indicateurs, effet à long terme, effets respectifs de l'ozone et de la température durant la période caniculaire de 2003, etc.).

L'équipe, initialement, non ou peu formée à la problématique air – santé, a découvert, outre l'énorme plaisir d'un travail en commun coordonné dans sa première phase par Philippe Quénel, la joie non dissimulée d'apprendre les méthodes appropriées sous la férule d'Alain Le Tertre, statisticien au Département santé environnement.

Les méthodes dont il a été question au cours des réflexions du Psas-9 et dont il sera fait état dans ce manuel sont dédiées à l'analyse des séries temporelles. Elles ont pour finalité d'analyser les relations entre la qualité de l'air et la santé humaine, dans le court terme et ont fait l'objet de nombreux développements au cours des quinze dernières années.

La compréhension et l'appropriation de ces méthodes que le groupe a commencé à maîtriser – non sans souffrances, non sans désespoirs – bénéficient autant de la théorie que de la pratique. Ainsi, il a semblé judicieux de faire un retour sur l'une et l'autre, *logos* et *praxis*, unies pour l'éternité… Et, la générosité légendaire des membres du Psas-9 n'ayant pas de limite, ceux-ci ont décidé de faire partager les connaissances qu'ils ont laborieusement récoltées (*to harvest* en anglais, autre évocation douloureuse…) en ce domaine.

Une remarque, enfin, pour clore ce préambule : les notions et les méthodes présentées, décrites et expliquées ici sont illustrées par des exemples extraits de la problématique air-santé mais sont appliquables, bien sûr, sans grande difficulté, à d'autres thématiques.

Daniel Eilstein

[1] L'effet à court terme de la pollution atmosphérique sur la santé est la responsabilité, relative à l'apparition d'un événement de santé (décès, manifestation d'un symptôme, etc.), des concentrations en polluants du jour et des cinq jours précédents.

Remerciements

Merci à **Pascal Beaudeau**, **Jean Donadieu**, **Martine Ledrans**, **Jérôme Pouey**, **Coralie Ravault**, **Dominique Salamanca**, **Daouda Sissoko** et **Stéphanie Vandentorren**, pour leur participation active à la formation à la modélisation des séries temporelles des 12, 13, 14 et 15 janvier 2004. Leurs interventions et leurs conseils avisés ont permis d'activer la réflexion sur le sujet et… par réflexion, d'enrichir ce document.

Merci aussi à **Pascale Rouaud**, statisticienne qui a relu ce manuel, a corrigé les erreurs probabilistes et a remédié aux incertitudes statistiques.

Merci encore à **François Belanger** qui a aussi relu ce manuel, a corrigé les erreurs de bons sens et a bien voulu jouer le difficile rôle du candide.

*Les auteurs tiennent, enfin, à remercier chaleureusement **Linda Boyeaux** et **François Belanger** du* Département formation documentation *de l'Institut de veille sanitaire pour leur investissement dans l'organisation de la formation à la modélisation des séries temporelles, laquelle a donné prétexte et matière à ce manuel.*

1. Introduction

La recherche et la mise en évidence d'associations statistiques entre des indicateurs de pollution atmosphérique et des indicateurs de l'état de santé d'une population ont fait l'objet de nombreux travaux dans le monde depuis une cinquantaine d'années.

La multitude des équipes de recherche impliquées dans ce domaine et la richesse des problèmes abordés (effets à long terme, effets à court terme, nature des polluants considérés, facteurs de confusion potentiels), ont mené à des approches très diverses (approches individuelles, études en population générale, au sein de populations fragiles), à des schémas d'étude (études cas-témoins, études de cohortes) et à des méthodes (études de séries temporelles) fort différents.

Il semble, cependant, qu'un consensus s'établisse depuis une dizaine d'année autour d'un certain type d'analyse, dédiée plus particulièrement au traitement statistique des effets sanitaires *à court terme* de la pollution atmosphérique ambiante urbaine : la méthode de choix, à ce jour, est l'analyse de séries temporelles. Dans la suite du texte (§ 2.6), nous examinerons plus en détail l'intérêt du recours à l'analyse des séries temporelles mais nous pouvons d'ores et déjà pointer quelques unes des caractéristiques principales de cette approche : celle-ci permet, en effet, de décrire une série de données, de donner un éclairage sur les mécanismes sous-jacents et de prédire le devenir de la série.

De plus, cette méthode, reposant sur des données relativement faciles à recueillir en routine, se plie facilement aux exigences méthodologiques de la *recherche* mais répond également, tout particulièrement, aux impératifs de santé publique pour la *surveillance* des risques à *court terme* de la pollution atmosphérique ambiante sur la santé d'une *population*..

Cette surveillance était la mission assignée au Programme de surveillance Air & Santé-9 villes (Psas-9), mis en place par le Département santé environnement (DSE) de l'Institut de veille sanitaire (InVS) en 1997. Elle a pour objectif de fournir les outils épidémiologiques nécessaires (risques relatifs) à la quantification de l'impact sanitaire de la pollution atmosphérique urbaine. Par là, elle vise à fournir des éléments d'information sanitaire utiles à la prise de décision dans le domaine de la gestion de la qualité de l'air en France.

Au cours des dix dernières années, dans le domaine de la pollution atmosphérique, l'analyse de séries temporelles a bénéficié de nombreux développements méthodologiques (protocole Aphea[2] [1,2]) et statistiques (modèle additif généralisé [3-5]). Ainsi, par exemple, en France, l'étude Erpurs (Évaluation des risques de la pollution urbaine pour la santé *en Île-de-France*) [6], parue en 1994, avait mis en évidence des associations significatives entre la pollution et un ensemble d'indicateurs de santé (mortalités et hospitalisations cardio-vasculaires et respiratoires, visites de SOS-médecins, arrêts de travail) sur la base de données enregistrées au cours de la période 1987 à 1992. L'approche utilisée était une analyse de séries temporelles. Le programme Psas-9 a utilisé ces méthodes pour l'analyse des relations entre les variations journalières des niveaux d'indicateurs de pollution atmosphérique et les variations journalières d'un compte journalier d'événements sanitaires (nombre de décès, nombre d'hospitalisations). Dans le cadre de cette mission, le Psas-9 a été confronté à des problèmes méthodologiques et statistiques liés aux différentes étapes de la modélisation.

De ces difficultés est née l'idée de structurer l'analyse et de donner à l'approche, parfois intuitive, une consistance méthodologique, garante de la reproductibilité des calculs et de la fiabilité des résultats. À partir de cette intention, la méthode qui s'apparente à une évaluation des pratiques, a consisté en une réflexion collégialement menée sur la base d'une trame constituée des étapes consensuelles. Ce projet a donné lieu à un ensemble d'échanges, de débats contradictoires, d'enrichissements, de réorganisation de l'architecture de la programmation. Le retour à la théorie a permis de justifier certains choix et d'en déterminer d'autres.

L'objectif de ce manuel est d'expliciter les bases statistiques sur lesquelles est fondée l'analyse puis d'en décrire les différentes étapes en l'illustrant, pas à pas, par des exemples tirés du domaine de l'étude des effets de la qualité de l'air sur la santé (répétons que les notions et méthodes qui seront

[2] Aphea : *Air Pollution and Health, a European Approach*.

exposées ici, sont appliquables à d'autres domaines). De fait, ce travail est issu de l'expérience partagée des épidémiologistes du Psas-9. Même partagée, cette expérience ne prétend pas valoir pour référence absolue ni avoir exploré toutes les subtilités de l'analyse mais son ambition est de fournir (les) quelques clefs nécessaires aux orientations et aux choix adéquats appelant décisions à chacun des nœuds de la démarche analytique.

Par ailleurs, le manuel a accompagné, dans sa version initiale, la formation organisée par l'Institut de veille sanitaire. Il s'est ainsi enrichi au cours et au terme de celle-ci de et par les questions et réponses suscitées au fur et à mesure de son déroulement.

Ce guide abordera d'abord la définition de série temporelle ainsi que l'analyse de ses différentes composantes à partir d'exemples (chapitre 2). Dans la suite de ce chapitre, nous exposerons la notion de processus, substrat théorique nécessaire à la formalisation des séries temporelles. Suivront quelques exemples de processus et nous terminerons par l'intérêt du recours aux séries temporelles. Le chapitre 3 sera dédié aux modèles linéaires et additifs généralisés, outils éminemment utiles dans de nombreux secteurs des statistiques et de l'épidémiologie. Les méthodes d'ajustement dédiées à ces modèles complèteront ce chapitre. Les principes généraux de la (des) modélisation(s) des séries temporelles feront l'objet du chapitre 4, tout en évoquant les qualités et les défauts des modèles proposés. Le chapitre suivant (chapitre 5) présentera les caractéristiques de base du logiciel S-PLUS® (commandes, manipulation de données) et survolera les instructions propres à la démarche de la modélisation. Partant d'exemples, nous détaillerons, dans le sixième et dernier chapitre, les différentes étapes de la modélisation en motivant les choix à chaque étape et en décrivant les commandes *ad hoc* sur S-PLUS®.

Remarque : ce manuel n'a pas la prétention d'être exhaustif. Il lui sera peut-être fait reproche de ne décrire que (ou en tout cas de privilégier) certains des outils dédiés à l'analyse des séries temporelles – les modèles de régressions – mais il a semblé plus judicieux de conserver à ce travail une dimension essentiellement pratique. Ainsi, dans ce manuel, vous ne trouverez pas : la modélisation de Box-jenkins, l'analyse spectrale, la notion de filtre de Kalman, etc.

2. Séries temporelles

2.1. Exemples de séries temporelles

Quelques exemples

Les séries temporelles, appelées aussi séries chronologiques (ou même chroniques), occupent une place importante dans tous les domaines de l'observation ou de la collection de données. Avant d'aborder la définition d'une série temporelle, nous passerons en revue un certain nombre d'exemples pouvant être rencontrés dans la vie courante.

Ainsi, la mesure du niveau de la mer réalisée jour après jour à Saint-Malo à 3 heures du matin, en 1998, par exemple, donne une série de valeurs exprimées en mètres, chacune d'entre elles correspondant à un jour de cette année.

S'il venait à l'esprit de comptabiliser le nombre de crabes cachés derrière les rochers de Saint-Malo durant un mois et ceci pendant une vingtaine d'années, il serait possible d'établir une série de nombre entiers (des comptes), dont chacun serait affecté à un mois déterminé, représenté par son numéro d'ordre.

Une autre observation pourrait être réalisée s'il s'agissait de faire le bilan du flux des gens qui entrent dans un grand magasin et qui en sortent, toutes les demi-heures, par exemple. Il serait possible d'obtenir une série du type : 1, 0, 4, -3, 7, 1, -4, etc.

Si l'intérêt se portait sur la présence ou non d'un congrès d'épidémiologie dans une région donnée au courant de l'année, et que 0 soit affecté à une année sans congrès et 1 à une année avec congrès, 100 ans durant, une série de 0 et de 1 pourrait ainsi être mise en relation avec la suite des 100 années étudiées.

Si l'on fait l'hypothèse qu'un congrès d'épidémiologie au moins a lieu tous les ans en France, une série pourrait être constituée par le titre de la première présentation orale du premier congrès de l'année, mis en relation avec le millésime, durant 20 années successives.

Si, non content de comptabiliser les crabes de Saint-Malo, il devenait intéressant de les comptabiliser en un autre lieu, à Dinan par exemple, une série pourrait être constituée à partir du nombre de crabes cachés derrière les rochers de Saint-Malo et du nombre de crabes cachés derrière les rochers des plages de Dinan, mis en relation avec le numéro du mois.

D'ailleurs, pour ce qui est de notre congrès annuel d'épidémiologie, pourquoi se restreindre au nom de la première présentation ? Il serait possible, en effet, de retenir les titres de tous les posters.

Reprenons, pour finir, l'exemple de la mesure du niveau de la mer. Nous avons supposé qu'elle se faisait à une heure précise, chaque jour (3 heures du matin). Une machine, cependant, pourrait réaliser un enregistrement du niveau en continu et nous pourrions ainsi disposer d'informations (infiniment) plus nombreuses.

Quelques remarques

Dans les exemples qui précèdent, le terme *série* est employé pour évoquer que des objets – ici des nombres ou des mots – sont classés dans un certain ordre. Dans le cas contraire, on aurait parlé de tas, de groupe, de réunion, etc. L'ordre dont il est fait état utilise comme outil le temps [3] et, plus précisément, une mesure du temps, exprimée en mois, années, minutes, etc. (dans cet « etc. », est comprise une mesure continue du temps). En résumé, ces séries associent des objets divers à des marques temporelles successives plus ou moins séparées, plus ou moins équidistantes, c'est-à-dire

[3] Le temps n'est pas le seul outil de classement possible, bien sûr. Il serait possible de classer les crabes selon leur taille, leur goût, l'endroit où ils ont « été » trouvés (devant les, au dessus des, en dessous des, à droite des, à gauche des, derrière les rochers, dans les rochers, etc.)

séparées par la même durée (pour les secondes, les minutes ou les heures, c'est rigoureusement vrai, pour les mois et années, cela l'est un peu moins). D'ailleurs, l'équidistance de deux marques temporelles successives n'est pas obligatoire pour parler de série temporelle. La série est dite temporelle, nous le verrons plus tard, parce qu'elle indice (ou indexe) l'objet enregistré (compte, mesure, couleur, etc.) par le temps. Ceci dit, dans la suite, les enregistrements seront supposés réalisés à des moments successifs équidistants.

Ce qui vient d'être présenté amène à une première remarque. Nous avons vu que la nature des objets faisant partie de la série pouvait être numérique (le niveau de la mer, le compte mensuel des crabes, le flux des clients du grand magasin, les 0 et les 1, témoins de la présence ou de l'absence d'un congrès) mais aussi plus « qualitative » telle celle du programme du congrès. Dans ce dernier cas, pour des raisons de formalisation, il sera indispensable de prévoir une transformation numérique de notre série. C'est à dire qu'il faudra associer aux objets non numériques, un nombre. Par exemple, les titres des interventions peuvent être classées en différentes catégories selon des critères définis (nature du sujet, intérêt, etc.) et chaque catégorie peut être associée à un nombre.

Une autre remarque s'impose au vu des exemples figurant ci-dessus. Les nombres témoignant des mesures réalisées sont de types divers. Ainsi, dans le premier et le dernier exemple (le niveau de la mer), les nombres sont réels, c'est-à-dire appartiennent à l'ensemble IR des nombres réels (plus précisément : positifs) ; les exemples concernant les crabes font intervenir des nombres entiers positifs ou nuls (IN) ; l'exemple du grand magasin fait intervenir des nombres entiers positifs, nuls ou négatifs (*Z*) ; la présence ou non de congrès dans une région donnée mènerait à une suite de 0 et de 1.

L'exemple des crabes à Saint-Malo et à Dinan, d'une part, et l'exemple des titres de posters, d'autre part, montrent qu'il est possible d'indicer par le temps, non pas un seul mais deux ou plusieurs nombres à la fois.

Enfin, le dernier exemple (celui du niveau de la mer, mesuré en continu) montre que le temps peut prendre toutes les valeurs réelles comprises entre deux instants. Ce type de mesure mène à des calculs assez compliqués, raison pour laquelle, comme la plupart des manuels qui parlent des séries temporelles et, comme il a été signalé plus haut, nous considérerons le temps ici comme un phénomène discret (discontinu) ou, tout au moins, nous ne retiendrons que des marques du temps séparées et équidistantes.

Quelques compléments

Ces enregistrements peuvent faire, bien sûr, l'objet d'une représentation graphique qui placera les marques temporelles en abscisse et les nombres mesurés en ordonnée ou selon plusieurs axes si besoin (c'est le cas où plusieurs variables sont représentées).

Comme nous le verrons plus bas, une autre façon de représenter ces enregistrements dépendant du temps quand plusieurs séries de nombres sont étudiées simultanément (comme dans l'exemple Saint-Malo-Dinan) est de faire correspondre à chaque série un axe mais sans que le temps soit représenté directement (il n'y a pas d'axe lui correspondant). Le graphe obtenu est une sorte de trajectoire dans l'espace des valeurs possibles, chaque point correspondant à un moment donné. Chaque point de ce graphe a, pour coordonnées, les valeurs prises par les séries de nombres correspondant au même instant. Ces trajectoires sont utilisées habituellement pour représenter les systèmes dynamiques* [4].

Quelques exemples de séries et leur représentation graphique

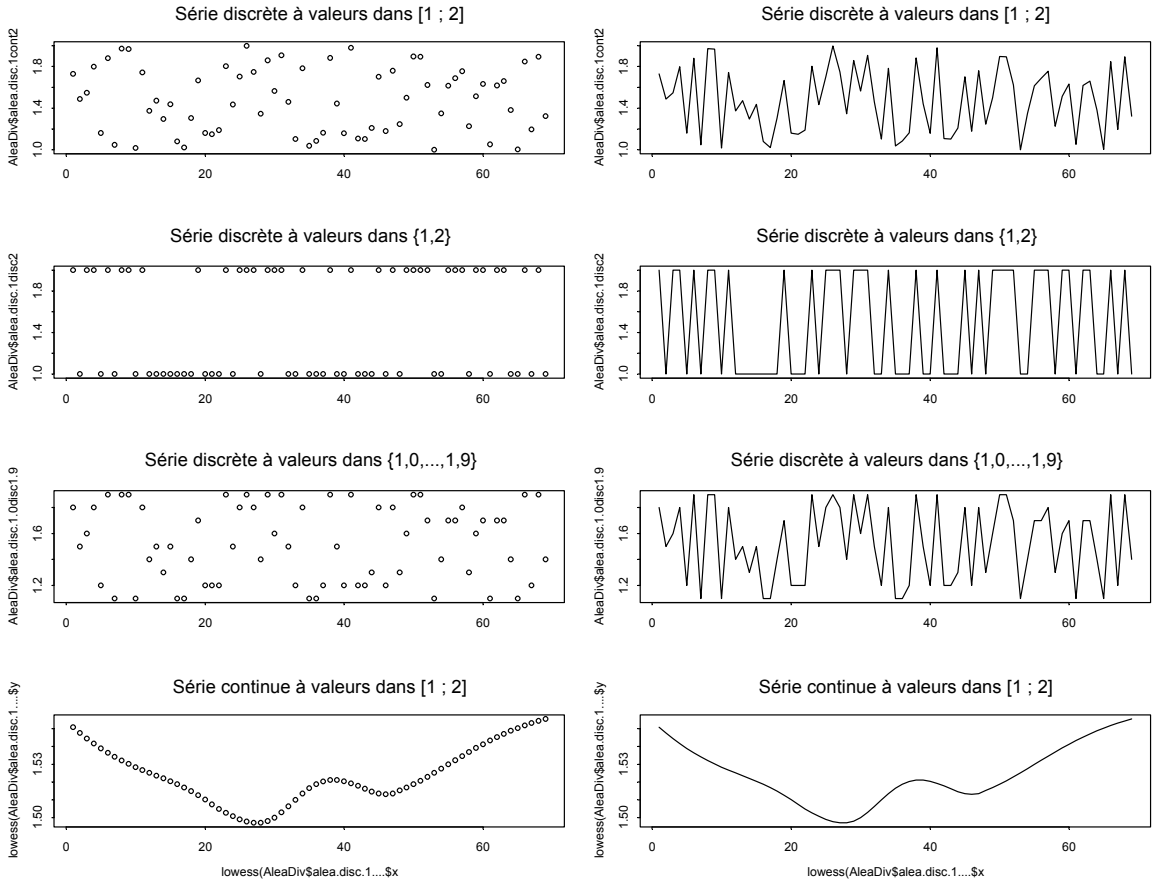
Voici quelques exemples de séries temporelles représentées graphiquement (figure 1). Les graphes de droite et de gauche représentent la même série, les uns avec des points, les autres avec des lignes continues. Ces exemples sont plutôt théoriques mais il est facile d'imaginer des exemples de phénomènes réels, leur correspondant. Les trois premières séries sont discrètes et prennent,

[4] La présence d'une astérisque (*) renvoie à une entrée dans le glossaire en fin de document.

respectivement, leurs valeurs dans l'intervalle, continu sur IR, [1 ; 2] (un exemple : la mesure du pH d'un ensemble de mélanges de deux solutions acides dont les pH sont égaux, respectivement, à 1 et 2), dans l'ensemble constitué des deux valeurs 1 et 2 (noté {1, 2}) – ce pourrait-être le premier chiffre du numéro de sécurité sociale des patients se présentant en consultation dans un hôpital – et dans l'ensemble de valeurs {1,0, 1,2, …, 1,9} [5] (une mesure réalisée à l'aide d'un appareil numérique, par exemple). La dernière est une série continue et prend ses valeurs dans [1 ; 2] (exemple : une mesure de glycémie en continu).

Remarque. Pour créer ces séries, nous avons d'abord réalisé une série avec la fonction ALEA() d'Excel® (graphes du haut) puis nous avons découpé les sorties de façon dichotomique 1-2, puis selon 10 valeurs, enfin nous avons réalisé un lissage (graphes inférieurs).

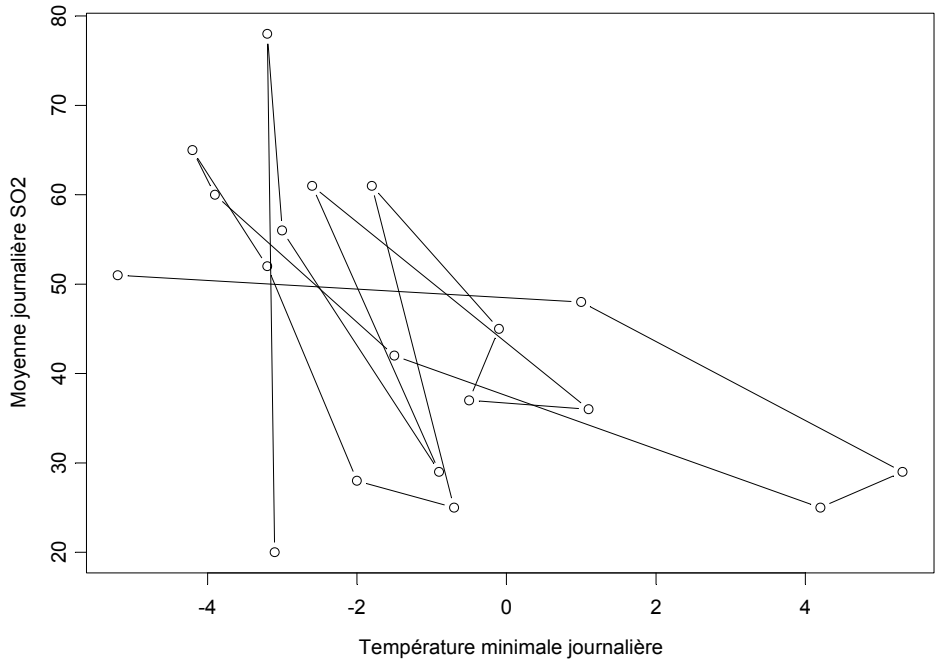
Figure 1. Exemples de séries discrètes et de série continue



Dans la figure qui suit (figure 2), deux variables (température minimale journalière et concentration moyenne journalière en dioxyde de soufre) sont représentées l'une par rapport à l'autre, chaque point correspondant à un moment. Il s'agit d'une trajectoire dans un espace de phases à deux dimensions (les segments de droites n'ont pas de valeur réelles mais matérialisent le lien chronologique entre les mesures successives).

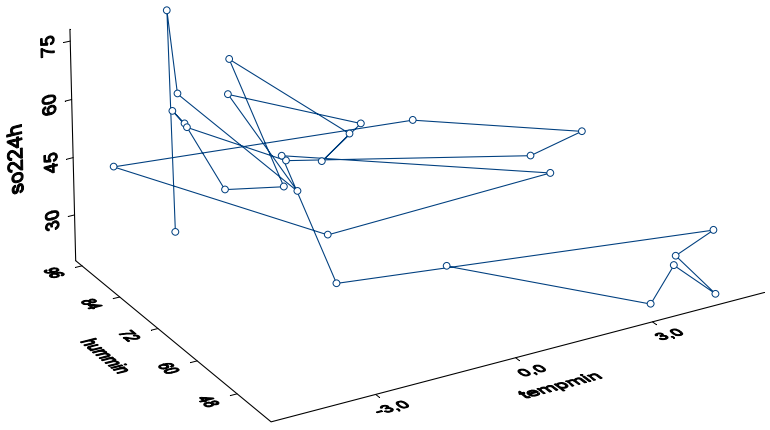
[5] En notation anglo-saxone, plus claire, ceci s'écrirait « {1.0, 1.2, …, 1.9} ».

Figure 2. Exemple de trajectoire dans un espace de phase à 2 dimensions



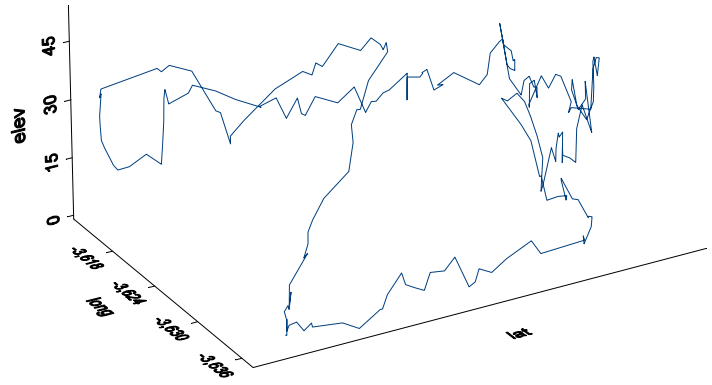
La figure 3 montre un exemple de trajectoire dans un espace de phase à trois dimensions. Trois variables dépendant du temps y sont représentées (température minimale journalière, humidité minimale journalière, concentration moyenne journalière en dioxyde de soufre).

Figure 3. Exemple de trajectoire dans un espace de phase à 3 dimensions



Voici un autre exemple de trajectoire « en 3D ». Il s'agit de la représentation graphique d'une ballade d'une dizaine de kilomètres dans les Côtes d'Armor en juillet 2004 : latitute, longitude et élévation enregistrées par un GPS [6] (figure 4).

Figure 4. Latitude, longitude et élévation enregistrées au cours d'une ballade dans la presqu'île de l'Armorique (Plestin-les-Grèves, Côtes d'Armor)



2.2. Définitions

Deux définitions de base

Stricto sensu, une **série temporelle** est une série d'observations numériques (mesures) indicées par le temps. Ces observations seront représentées par :

$$y_1, y_2, \ldots, y_n$$ [7]

1, 2, …, n représentent les marques temporelles, y_i est la valeur de la mesure réalisée au temps i.

L'étude des séries temporelles a bénéficié de nombreux apports théoriques fondés, essentiellement, sur les progrès de leur modélisation. Il est ainsi d'usage, aujourd'hui, d'aborder ce domaine en évoquant la notion de *processus* stochastique (ou aléatoire).

Il n'est pas du propos de ce manuel de définir rigoureusement ce qu'est un processus (il est possible de se reporter pour cela à des ouvrages spécialisés, le plus souvent du domaine de l'économétrie [7-12]). Notre intention est de rester au plus près des considérations pratiques. Si l'on veut *approcher*, cependant, la définition précise et « opératoire » de la notion de série temporelle, il est nécessaire de faire référence à la notion de processus aléatoire comme phénomène sous-jacent à ce qu'on observe.

---

[6] Les données ont été enregistrées sur le GPS (acronyme de *Global positioning system*) Garmin eTrex Summit® et récupérées sur le logiciel CartoExploreur 3®.

[7] Dans ce manuel, la série de valeurs expliquées sera représentée en général par $y_1, y_2, \ldots, y_n$, les notations $x_1, x_2, \ldots, x_n$ ou $z_1, z_2, \ldots, z_n$ seront plutôt réservées aux séries de covariables.

Pour annoncer la suite, nous dirons qu'**un processus aléatoire est une suite de variables aléatoires indicées par le temps (et définies sur un espace des états de la nature)**. Le processus peut être représenté par une notation du type :

$$Y_t$$

Avec t, une mesure du temps. Il y a autant de variables aléatoires que de valeurs de t. Nous verrons cette notation plus en détail dans la suite du texte.

Ajoutons que ces variables ne sont pas choisies au hasard et qu'elles ont un ensemble de caractéristiques communes.

Quelques questions bien légitimes et justification de ces définitions

Cette définition sera explicitée plus tard. Cependant, une question s'impose d'emblée : pourquoi a-t-on besoin de faire référence aux processus aléatoires et, par là même, aux variables aléatoires pour traiter des séries chronologiques ? Il semble, en effet, qu'avec les exemples précités, il a déjà été donné une notion précise de ce qu'est une série temporelle. En fait, il n'en est rien. On peut dire, pour l'instant, que ces exemples répondent bien à la définition de série temporelle [8] mais pour pouvoir les traiter statistiquement, il faut supposer que la ou les séries de nombres vues plus haut ne sont que des *réalisations* particulières de processus.

Revenons, pour cela, à la notion de variable aléatoire. Supposons que l'on s'intéresse à la concentration en sel de l'eau de mer. L'expérience que nous envisageons pour répondre à cette question consiste à prélever 20 échantillons de 10 cm^3 d'eau de mer, de les faire analyser et de retenir la concentration en chlorure de sodium (NaCl) de chacun d'eux. Cette concentration est exprimée en g/l. Nous obtenons 20 nombres (y_1, y_2,…, y_{20}) réels positifs et en général différents les uns des autres.

Une première question : « que faisons-nous de ces 20 nombres ? » La moyenne, bien sûr, puisque nous avons appris que nous avions plus de chance d'approcher la vraie concentration du NaCl avec celle-ci qu'avec l'un (choisi au hasard) des 20 résultats initiaux.

La deuxième question est « mais, finalement, pourquoi approche-t-on mieux la vraie concentration en NaCl de cette façon ? » La réponse tient justement en ce fait que, dans notre expérience, le nombre obtenu pour un prélèvement donné (disons le $9^{ème}$) est *une réalisation particulière* (y_9) parmi l'ensemble des résultats qu'on aurait pu obtenir pour ce $9^{ème}$ prélèvement si on avait pu le répéter une infinité de fois. L'ensemble de ces résultats est l'ensemble des valeurs que peut prendre la variable aléatoire (appelons-là Y_9) associée à ce $9^{ème}$ prélèvement. Nous savons que chacune des réalisations n'a pas la même chance d'être observée et c'est pourquoi nous attribuons à chacune des réalisations une probabilité (ici, plus rigoureusement une valeur de densité de probabilité puisque la variable est continue) et une loi de probabilité à la variable Y_9. Nous savons aussi que la vraie valeur recherchée (ici la concentration de NaCl) est estimée par la moyenne de ces réalisations, dite espérance de Y_9. Donc il faudrait réaliser plusieurs fois ce même $9^{ème}$ prélèvement (une infinité de fois serait l'idéal mais 10^{10} fois serait déjà pas si mal) mais ça n'est pas possible puisque tout n'est jamais pareil, une fois l'action faite ! Aussi pour remplacer la *réitération* d'un prélèvement donné, 19 fois (1+19=20), on fait 19 autres prélèvements représentés chacun par une variable aléatoire différente, Y_1, Y_2, …,Y_{18}, Y_{20}. Puis on suppose, sous le couvert d'un certain nombre d'hypothèses (les variables aléatoires représentant les prélèvements sont indépendantes, de même espérance, de même variance, etc.), que « faire 20 prélèvements, 1 fois » revient à « en faire 1, 20 fois ».

De plus, la précision liée à la moyenne est plus grande que celle qu'on peut accorder à chacune des mesures. En d'autres termes, la variance de l'espérance est inférieure à celle des observations.

Revenons au processus aléatoire dont nous avons dit qu'il était composé d'une suite de variables aléatoires indicées (indexées) par le temps. Pour comprendre un peu mieux la notion de variables indicées par le temps, reprenons l'exemple précédent en le modifiant : aussi, supposons à présent

---

[8] Les séries chronologiques étant composées de nombres, on suppose bien sûr que les séries d'objets qualitatifs ont été l'objet d'une transformation numérique.

qu'il s'agit de mesurer la concentration en NaCl une fois par jour, pendant 20 jours successifs. Les variables aléatoires correspondant à ces mesures sont appelées, comme précédemment, Y_1, Y_2, …,Y_{18}, Y_{20} mais, en apparence identique, cette notation n'a pas la même signification : ici, les indices représentent le temps et les variables ne représentent pas le même phénomène exactement mais un phénomène qui dépend du moment. Les indices peuvent correspondrent à des jours, des mois successifs, des années, des minutes, des secondes successives. La succession de ces variables aléatoires constitue un processus aléatoire. Ces variables n'ont pas forcément la même espérance ni forcément la même variance mais elles doivent souscrire à certaines conditions comme l'identité de la nature de la loi de probabilité par exemple.

Nouveaux exemples

Les exemples de processus aléatoires sont nombreux. Certains ont été vus plus haut. En voici d'autres qui permettront de décrire plus précisément les différents types de séries temporelles pouvant être rencontrés dans la vie de tous les jours.

Le coefficient de marée, exprimé deux fois par jour pour chaque jour de l'année, est une série de valeurs donnée, année après année ; chaque valeur de coefficient de marée est un nombre entier compris entre 20 (marée de morte eau exceptionnelle) et 120 (marée de vive eau exceptionnelle) inscrit à l'avance dans un almanach et établi à partir d'un ensemble de calculs. Ce coefficient sert à prévoir la hauteur de la mer dans un endroit quelconque ou plus précisément le marnage qui est la dénivellation entre une Pleine Mer et une Basse Mer consécutives. En fait, le coefficient de marée est un contre-exemple de processus puisqu'il est donné sans notion de probabilité ni notion de mesure pouvant être une réalisation de celui-ci. Par contre le marnage peut être considéré comme une série temporelle (2 valeurs par jour), chacune des valeurs mesurées étant une réalisation d'une variable aléatoire dont l'espérance est donnée par le calcul M = 2 U C, avec M, l'espérance du marnage, C, le coefficient de marée, U, l'unité de hauteur dépendant du lieu. La loi de distribution pourrait être une loi gamma, par exemple.

Le nombre journalier de décès dans une zone donnée est une série temporelle. À chaque jour correspond une variable aléatoire dont le domaine des valeurs est constitué par l'ensemble IN des nombres entiers supérieurs ou égaux à 0. La loi de probabilité de chacune de ces variables est, conventionnellement, une loi de Poisson $P(\lambda)$ dont le paramètre dépend d'un ensemble de facteurs extrinsèques ou intrinsèques.

L'incidence annuelle du cancer du sein en France est une série temporelle. L'incidence est, à une constante près, le rapport du nombre annuel de nouveaux cas de cancer au nombre de personnes-années, c'est à dire, en gros, à l'effectif de la population calculé au milieu de l'année. Si l'on suppose qu'il n'y a pas d'incertitude sur l'effectif de la population (ce qui est faux puisque l'Insee réalise en fait des estimations de populations à partir des recensements), toute la variabilité de l'incidence provient du numérateur (variable *nombre de cas incidents*) auquel on attribue, comme précédemment, une loi de probabilité de Poisson. Chacune des variables aléatoires *nombre de cas incidents* est ainsi supposée suivre une loi de Poisson, différente *a priori*, d'une année à l'autre.

L'effectif de la population d'une zone déterminée, estimé au 1er janvier de chaque année constitue une série temporelle. Chaque opération d'estimation peut être représentée par une variable aléatoire annuelle prenant ses valeurs dans IN et chaque estimation par une réalisation de cette variable aléatoire. Ainsi, une des façons de modéliser l'effectif de la population au temps t (ici, l'année), noté N(t), est de supposer qu'il suit une loi de Poisson (encore) de paramètre λt [13]:

$$P(N(t)=n)=\frac{e^{-\lambda t}(\lambda t)^n}{n!}$$

D'autres modèles sont utilisés (modèle de Yule, modèles avec prise en compte de l'immigration, etc.). En tous cas, tous aboutissent à un processus.

La température moyenne journalière mesurée sur la banquise tous les jours peut être considérée comme la réalisation d'une variable aléatoire journalière prenant ses valeurs dans IR. La loi de probabilité pourrait être une loi normale mais tronquée puisque la température a une limite inférieure

certaine (théorique, tout au moins), fixée à 0° Kelvin (-273°C environ ou -459,4°F) et une limite supérieure vraisemblable (la banquise…). Chaque jour, en tous cas, est caractérisé par une loi de probabilité.

Groupons les exemples de séries selon le type de variables

Temps discret et variable discrète

- Bruit et marche aléatoire poissonienne (voir plus bas § 2.5.1 et § 2.5.2)
- 0/1 : été/hiver, vacances, congés
- 1-7 : jour de la semaine
- N : Nombre d'événements journaliers ou sur un autre pas de temps (décès, hospitalisations, cas incidents)
- Nombre hebdomadaire de cas de grippe
- Comptes polliniques journaliers

Temps discret et variable continue

- Bruits, marches aléatoires
- T°min journalière, T°max journalière
- Humidité minimale journalière
- Concentration horaire en SO_2

Temps continu et variable continue

- Température
- Longueur parcourue
- Vitesse instantanée
- « Prévalence instantanée »

Temps continu et variable discrète

Plus difficile à concevoir ce type de série est représentatif de tout phénomène prenant des « valeurs » discrètes et observé continûment. Il faut admettre la réalité de « sauts » instantanés d'une valeur à une autre valeur non contiguë.

- Nombre de personnes dans un lieu
- Nombre d'événements instantané (nombre de cas incidents instantané)
- Nombre de cas présents instantané (nombre de cas prévalents instantané)

Et en résumant :

		Temps	
		Discret	Continu
Variable	Discrète	Compte 0/1 ♠, ♥, ♦, ♣	Comptes instantanés
	Continue	Pression moyenne journalière Incidence, prévalence Marche aléatoire	Distance parcourue Température

Quant aux définitions de séries temporelles et processus, si on se résume et si on précise…

Ce qu'on sait, à présent :

- Une série temporelle est une série de nombres indicés par le temps, chacun d'entre eux étant **une** réalisation d'une variable aléatoire (figure 5). Ceci peut s'exprimer aussi en disant que la série temporelle est **une** réalisation d'une famille de variables aléatoires indicées par le temps (figure 6). Cette famille de variables aléatoires s'appelle un processus aléatoire.
- Un processus aléatoire, quant à lui, est une famille de variables aléatoires (figure 7).

Figure 5. Exemple de série temporelle

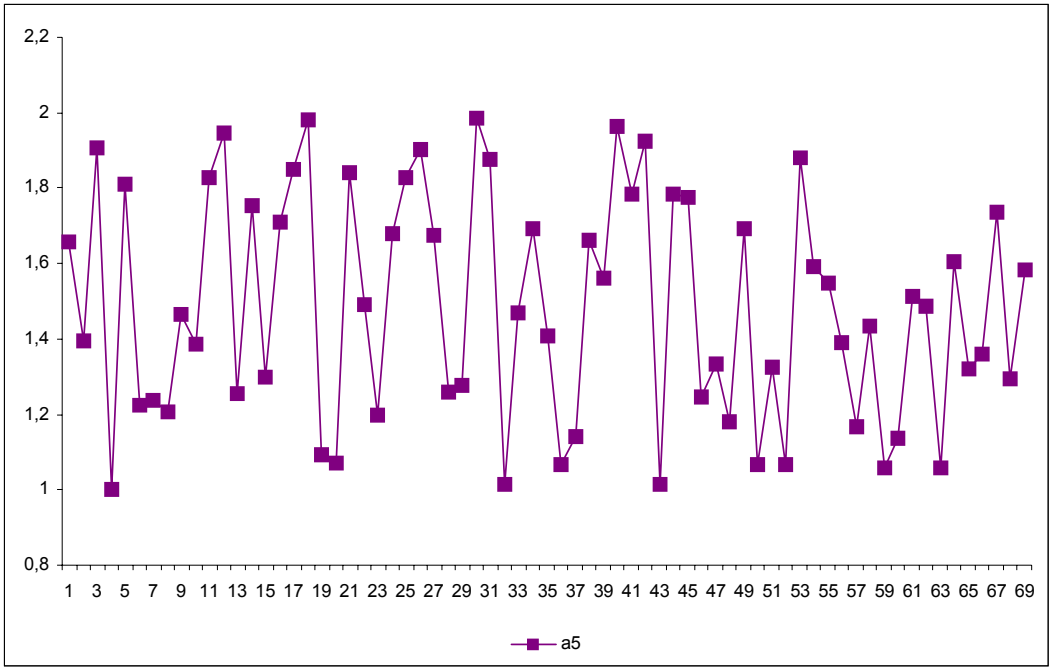


Figure 6. Exemple de série temporelle au sein d'un processus

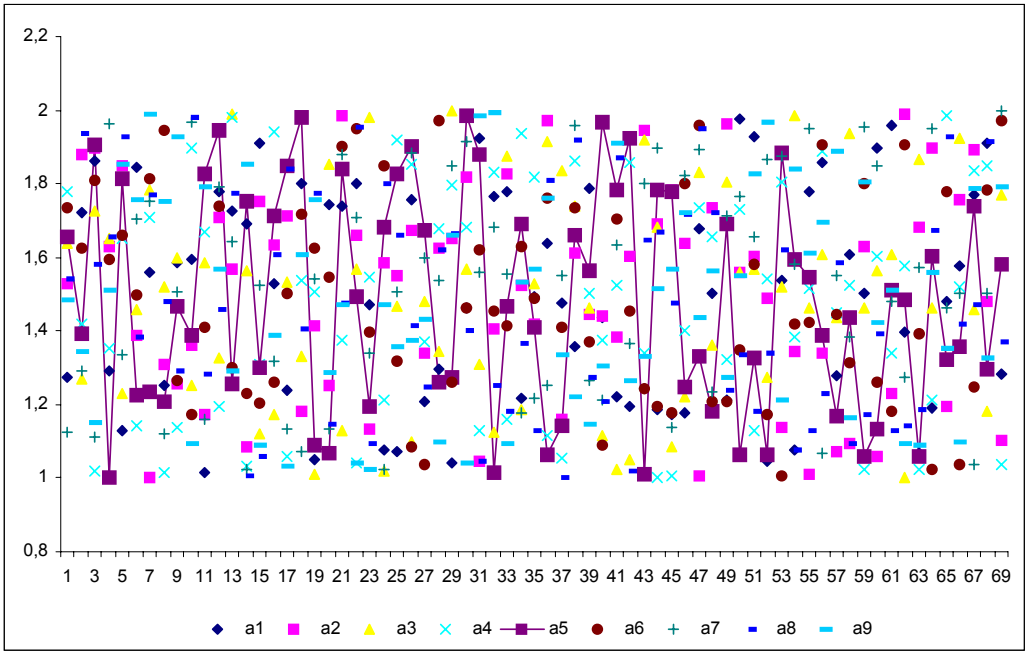
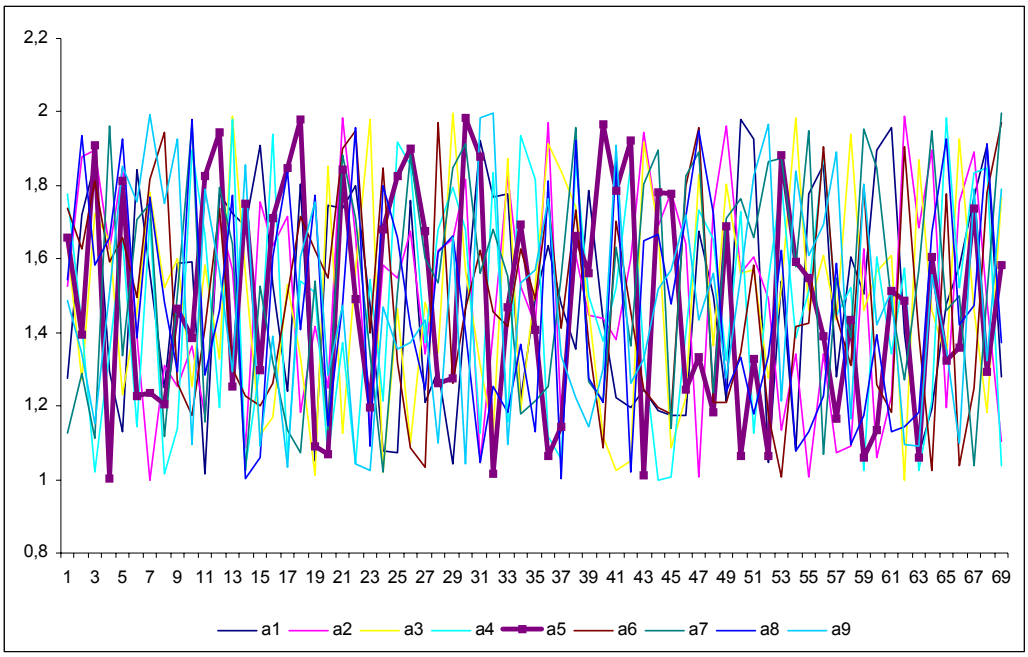


Figure 7. Exemple de processus



On trouvera les notions évoquées plus haut, dans les deux encarts ci-dessous.

- Espace probabilisé : (Ω, F, P)
  - Expérience : $\omega \in \Omega$
  - Événement : $A \subset \Omega$
  - P(A)
- Variable aléatoire : Y
  - $y = Y(\omega)$, $A' = Y(A)$
  - $\omega \in Y^{-1}(y)$, $A \subset Y^{-1}(A')$
- Probabilité sur R
  - $P_Y(A') = P(Y^{-1}(A')) \geq P(A)$

$\Omega \xrightarrow{Y} R$
$\omega \xrightarrow{Y} y$
$A \xrightarrow{Y} A'$

- Espace probabilisé : (Ω, F, P)
  - Expérience : $\omega \in \Omega$
  - Événement : $A \subset \Omega$
  - P(A)
- Variables aléatoires : Y_t
  - $y_t = Y_t(\omega)$, $A_t' = Y_t(A)$
  - $\omega \in Y_t^{-1}(y_t)$, $A \subset Y_t^{-1}(A_t')$
- Probabilité sur R
  - $P_{Yt}(A_t') = P(Y_t^{-1}(A_t')) \geq P(A)$

$\Omega \xrightarrow{Y_t} R$
$\omega \xrightarrow{Y_t} y_t$
$A \xrightarrow{Y_t} A_t'$

Il est temps, à présent, d'aller un peu plus loin dans la définition du processus et, en tous cas, de savoir ce qui fait qu'une série de variables aléatoires est un processus.

Retour sur les définitions de variables et de processus aléatoires

Si nous regardons dans la littérature [7], nous lisons qu'un processus aléatoire est une famille de variables aléatoires { Yt, t ∈ T } [9], définies sur un espace probabilisé (Ω,*F*,P). Nous n'approfondirons pas, bien sûr, cette formalisation de la théorie des probabilités mais si nous nous penchons quelques instants sur sa signification, ceci peut nous éclairer sur ce que l'on impose aux variables aléatoires pour qu'elles constituent un processus aléatoire.

Pour expliciter la notion d'espace probabilisé (Ω,*F*,P), nous nous servirons de l'exemple éculé des deux dés que l'on jette et dont on attend qu'ils montrent chacun l'une de leur face avec un certain nombre de points compris entre 1 et 6. Ω est l'ensemble des résultats possibles de nos jets de dés. Nous déciderons de façon arbitraire – là non plus, aucune originalité – que l'ensemble des résultats est l'ensemble des couples de nombres affichés par les dés (on aurait pu choisir un autre ensemble tel que l'ensemble des sommes des affichages possibles, etc.), soit { (1,1), (1,2), …, (6,6) }. Ω est cet ensemble.

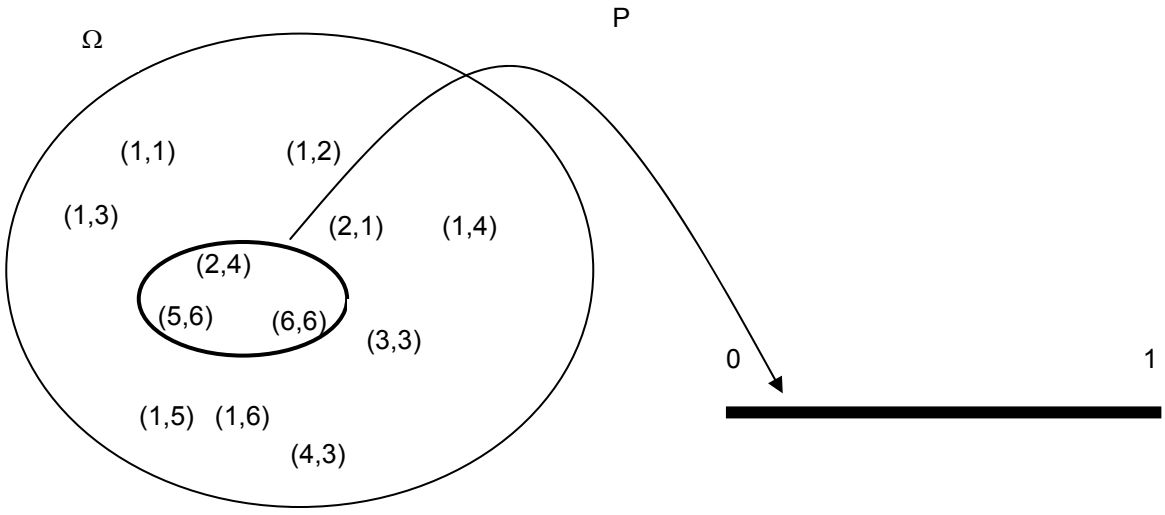
$$\Omega = \{ (1,1), (1,2), \ldots, (6,6) \}.$$

Ω contient 36 éléments c'est-à-dire 36 couples de nombres.

F est un morceau (on dit un sous-ensemble) de l'ensemble des parties de Ω (figure 8). Une partie de *F* (et donc de Ω) est un sous ensemble du type { (2,4), (5,6), (6,6) }, par exemple. *F* répond au nom de tribu lorsqu'il vérifie un certain nombre de propriétés que nous ne détaillerons pas ici. Ce qu'il faut retenir c'est que toutes ces parties de Ω sont des événements et qu'en gros, la tribu *F* est l'ensemble de ces événements.

Nous arrivons à P, dite loi de probabilité qui associe à chacun des événements de *F* un nombre compris entre 0 et 1 (0 et 1, inclus, bien sûr). Si l'événement est { (2,4), (5,6), (6,6) }, comme plus haut dans notre exemple du jet de dés, P lui attribue le nombre 3/36 soit 1/12 [10].

Figure 8. Ensemble probabilisé des résultats du lancer de deux dés.



Une variable aléatoire (réelle), Y, est une application de Ω dans IR, c'est à dire que c'est une fonction Y qui associe à tout élément ω de Ω un nombre réel, que l'on écrit Y(ω). L'exemple, tout aussi classique que les précédents est : Y associe à chaque résultat du lancer des deux dés, la somme des affichages des deux dés. Par exemple, si le lancer affiche (4,3), Y lui associera 7. Il en sera de même pour (2,5), etc.

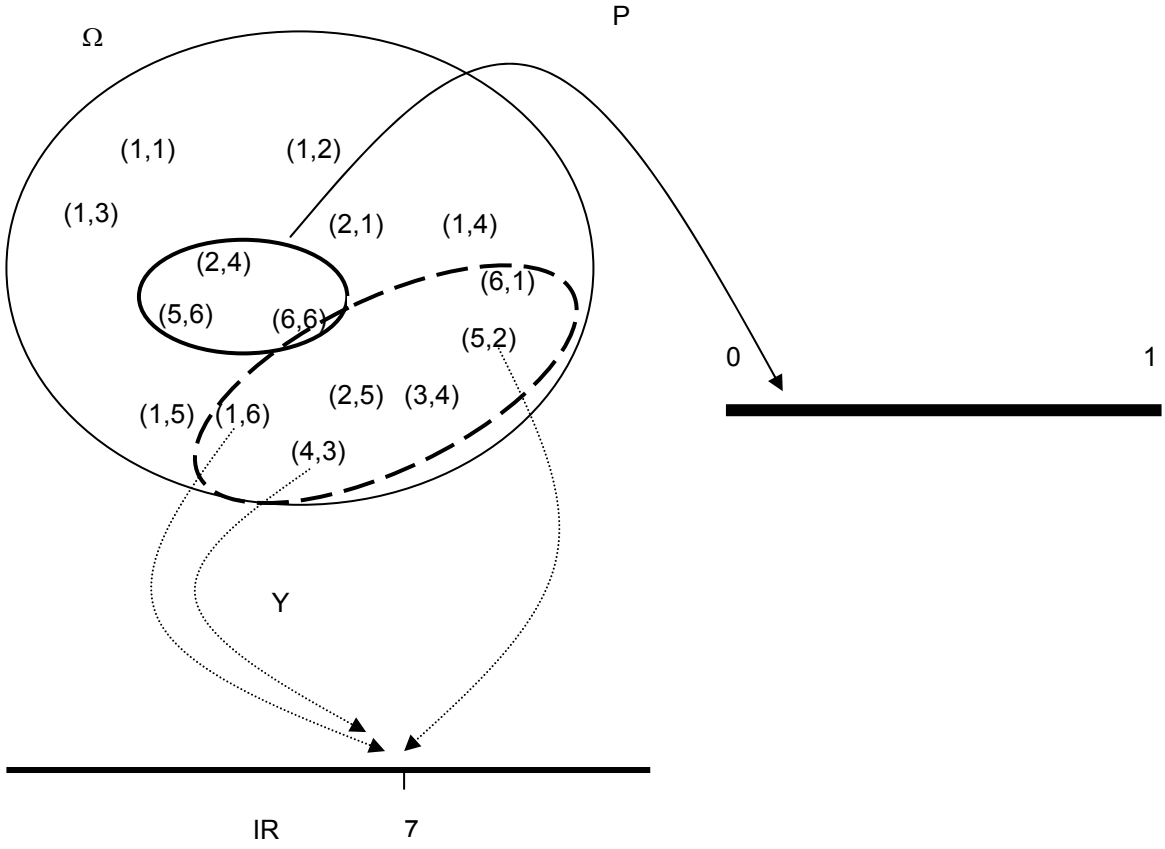
---

[9] T est un ensemble de valeurs temporelles, un ensemble de moments, plus simplement.

[10] On suppose que les dés sont non pipés, etc.

Nous voyons qu'à un nombre de IR, il correspond par l'intermédiaire de la fonction Y, un ensemble de résultats de Ω (*ie.* un certain nombre d'événements). Ainsi 7 correspond à l'ensemble { (1,6), (2,5), (3,4), (4,3), (5,2), (6,1) } (figure 9). De même, à un sous-ensemble de IR, il est possible de faire correspondre un sous-ensemble de Ω : ainsi au sous-ensemble { 3, 7, 2 } de IR, il correspond { (1,2), (2,1), (1,6), (2,5), (3,4), (4,3), (5,2), (6,1), (1,1) } de Ω.

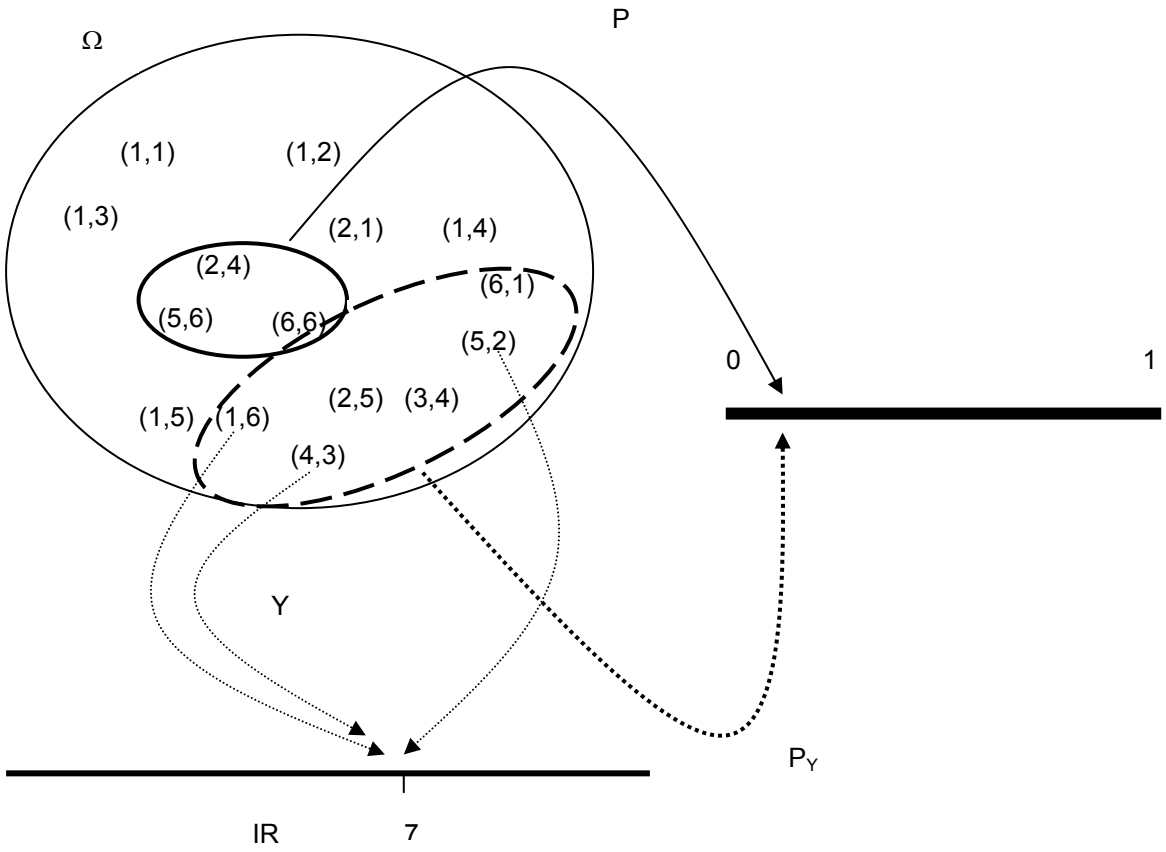
Figure 9. Ensemble probabilisé des résultats du lancer de deux dés et variable aléatoire « Somme des affichages »



Il est tout à fait possible, à présent, de définir une loi de probabilité P' dans IR, correspondant à la loi P déjà définie dans Ω. Décidons pour cela que, pour toute partie A de IR (il faut, là aussi, une tribu, ensemble de sous-ensembles de IR), P'(A) est égal à la probabilité P de la réunion de tous les sous-ensembles de Ω qui ont A pour image par Y (*ie.* A leur correspond par l'application Y).

Dans notre exemple P'({7}) = P({ (1,6), (2,5), (3,4), (4,3), (5,2), (6,1) }) = 1/6 et P'({ 3, 7, 2 }) = P({ (1,2), (2,1), (1,6), (2,5), (3,4), (4,3), (5,2), (6,1), (1,1) }) = 9/36 = 1/4. P' est souvent notée P_Y pour rappeler que c'est la loi de probabilité image de P par Y. Elle est appelée loi de probabilité de Y (figure 10).

Figure 10. Ensemble probabilisé des résultats du lancer de deux dés, variable aléatoire « Somme des affichages » et loi de probabilité de la variable aléatoire

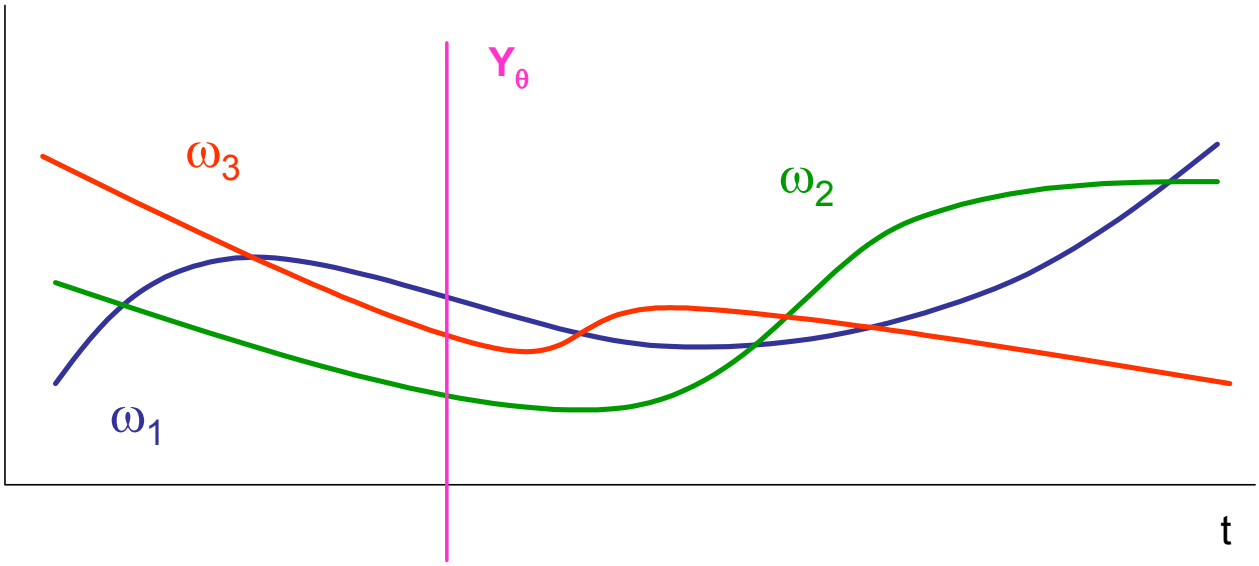


Nous voyons ainsi que les variables aléatoires du processus Y_t qui sont définies sur le même espace (Ω) – lequel est structuré de la même façon (*F*, identique) pour toutes les variables et muni de la même loi de probabilité (P) – ne font que transformer, « déformer » quelque peu la loi P mais sans la modifier structurellement. Ainsi chaque observation peut être une réalisation d'une variable aléatoire indépendante des autres mais assortie de la même loi *ou presque* que les autres. Un exemple classique est le processus de Poisson, constitué d'une famille de variables aléatoires Y_t, dont les lois de distribution sont des lois de Poisson de paramètre $\lambda(t)$ ou λ_t. À chaque date t, correspond une variable aléatoire suivant une loi de Poisson dont le paramètre (ou l'espérance, dans ce cas) dépend du temps.

À partir de la théorie, il est possible de regarder un processus de deux façons. Si l'on considère l'ensemble des valeurs prises par la série correspondant à un élément ω de Ω, il s'agit d'une trajectoire. Si l'on considère l'ensemble des valeurs possibles à un moment t, alors il s'agit de l'ensemble des valeurs pouvant être prises par la variable aléatoire (figure 11) [(11)].

[11] Ceci peut être rapproché de la dualité des conceptions ondulatoire et corpusculaire de la lumière. Les deux approches sont possibles et on choisit celle des deux qui nous rend le plus de service sur le moment.

Figure 11. Un processus est à la fois un ensemble de trajectoires et une suite de variables aléatoires



Remarque. La définition de série temporelle fait souvent l'objet d'une confusion entre la série d'observations et le processus à l'origine de la série.

2.3. Caractéristiques et propriétés des séries temporelles

Les séries temporelles et les mécanismes générateurs que sont les processus, ont une structure. En d'autres termes les variables aléatoires composant le processus ne sont pas forcément indépendantes les unes des autres mais établissent certaines relations qui donnent à l'ensemble une sorte de structure.

Avant d'aller plus avant dans l'exploration de cette structure, il convient de se pencher sur une notion importante, la fonction d'autocovariance qui est aux séries temporelles ce que la matrice variance-covariance est aux variables aléatoires classiques.

2.3.1. Autocovariance

Soit Y_t (notation simplifiée pour { Yt, t ∈ T }), un processus, r et s deux instants.

L'autocovariance (ou fonction d'autocovariance) de Y_t pour les deux instants r et s est, par définition, la covariance des variables Y_r et Y_s [(12)].

L'autocovariance s'écrit γ_Y et :

$$\gamma_Y (r , s) = Cov (Y_r , Y_s)$$

2.3.2. Autocorrélation

La notion d'autocorrélation découle de la notion d'autocovariance comme la corrélation de la covariance.

[12] Les variances des variables composant le processus sont supposées finies.

Par définition l'autocorrélation (ou le coefficient d'autocorrélation) de la série Y_t est :

$$\rho_Y(h) = \frac{\gamma_Y(h)}{\gamma_Y(0)} = Cor(Y_{t+h}, Y_t) \text{ pour tout t et pour tout h.}$$

Cette formule utilise l'une des composantes de la stationnarité d'une série, propriété que nous verrons plus loin.

2.3.3. Autocorrélation partielle

L'autocorrélation (ou le coefficient d'autocorrélation) partielle d'ordre K de la série Y_t est égale au coefficient de corrélation entre :

$$Y_t - E(Y_t / Y_{t-1}, Y_{t-2}, \ldots, Y_{t-K+1}) \quad \text{et} \quad Y_{t-K} - E(Y_{t-K} / Y_{t-1}, Y_{t-2}, \ldots, Y_{t-K+1})$$

Ce coefficient, noté r(K) mesure la corrélation entre Y_t et Y_{t-K}, lorsqu'on a éliminé les parties de Y_t et Y_{t-K}, expliquées par les variables intermédiaires.

2.3.4. Stationnarité

Commençons par une définition.

Un processus discret Y_t est dit stationnaire (du second ordre) si :

- $E(|Y_t|^2)$ est fini pour tout t entier ;
- $E(Y_t) = \mu$ quel que soit t ;
- $\gamma_Y (r + u , s + u) = \gamma_Y (r , s)$ quels que soient r, s et u, entiers.

La première condition dit, en gros, que les moments du second ordre ($E(|Y_t|^2)$) – et donc que les variables Y_t – ne doivent pas prendre des valeurs infiniment grandes.

La seconde dit que les espérances (les moyennes) des variables Y_t sont égales.

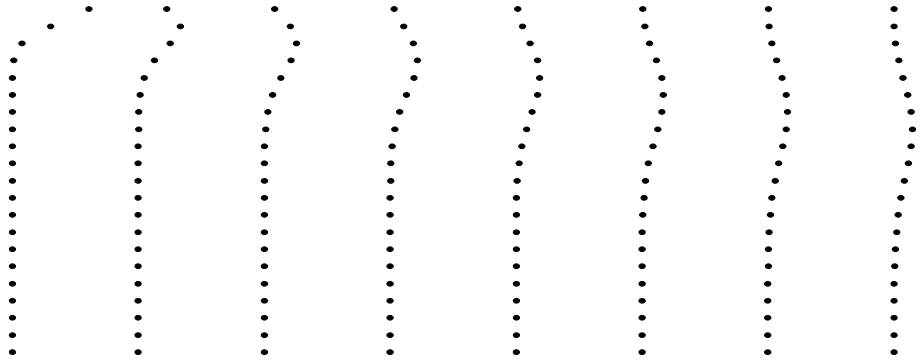
La troisième dit que la corrélation entre deux variables ne dépend que de l'écart entre les instants respectifs de ces variables. Ceci veut dire aussi que lorsqu'on se déplace sur l'axe du temps, la corrélation entre les variables séparées d'un certain délai est toujours la même.

Cette condition ne dépend donc que du délai et peut s'écrire en fonction de ce dernier :

$$\gamma_Y (h) = Cov (Y_t , Y_{t+h}), \text{ quels que soient t et h.}$$

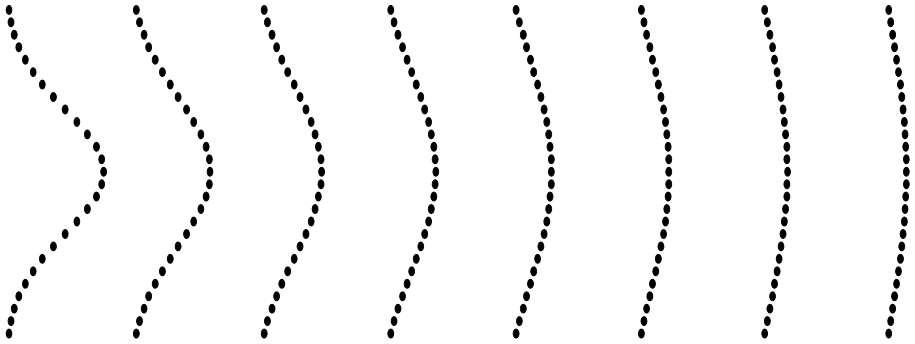
Voici trois exemples de processus (figures 12, 13 et 14).

Figure 12. Processus de Poisson non stationnaire



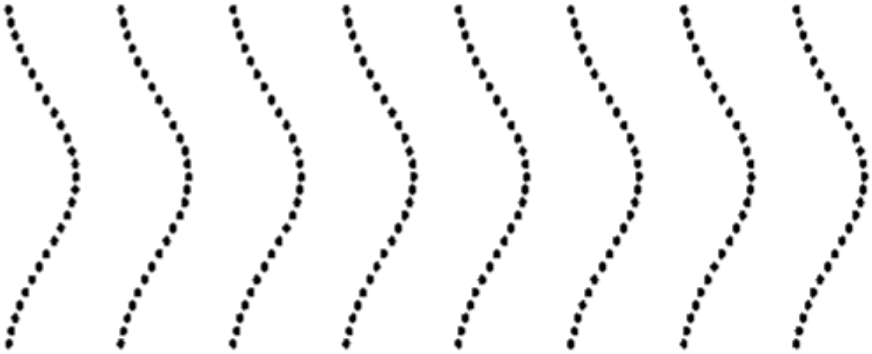
Succession de variables aléatoires suivant des lois de Poisson dont les paramètres sont respectivement égaux à 0,5, 1,5, 2,5, 3,5, 4,5, 5,5, 6,5 et 7,5.

Figure 13. Processus de lois normales, non stationnaire



Succession de variables aléatoires suivant des lois normales dont les espérances sont identiques et égales à 1 et dont les écart-types sont égaux à 0,5, 0,6, 0,7, 0,8, 0,9, 1,0, 1,1 et 1,2.

Figure 14. Processus de lois normales, stationnaire



Succession de variables aléatoires suivant des lois normales dont les espérances sont identiques et égales à 1 et dont les écart-types sont identiques et égaux à 0,6.

La stationnarité dont nous venons de faire état est dite du second ordre. Il en existe une autre dite stricte. Celle-ci munit le p-uplet $\left(Y_{t_1}, Y_{t_2}, ..., Y_{t_p}\right)$ d'une distribution conjointe constante pour tout décalage h.

On a :

$$\text{distrib}\left(Y_{t_1+h}, Y_{t_2+h}, ..., Y_{t_p+h}\right) = \text{distrib}\left(Y_{t_1}, Y_{t_2}, ..., Y_{t_p}\right)$$

En prenant p égal à 1 cette relation signifie que la distribution de Y_t est la même, quelque soit t.

La stricte stationnarité implique la stationnarité du second ordre à la condition que les moments de second ordre soient finis. La réciproque est fausse.

Les processus stationnaires occupent une place primordiale au sein des analyses de séries temporelles. Cependant, en général, les processus ne sont pas stationnaires. Aussi, pour étudier un processus, il faut en général, le « stationnariser ».

2.3.5. Ergodicité

Cette notion fondamentale vient de la considération suivante : nous savons à présent qu'une série temporelle est **une** réalisation particulière d'un processus et donc que chaque observation est l'**une** des réalisations de la variable aléatoire indicée correspondante. Comment, alors, calculer l'espérance, la variance, la fonction d'autocorrélation du processus alors que nous savons qu'il nous faut connaître beaucoup plus d'un point par variable aléatoire ? En d'autres termes, comme on ne peut faire de calcul statistique sur un cas, il faut trouver un autre moyen.

On dira, alors, qu'un processus stationnaire est ergodique [13] si l'on peut calculer l'ensemble de ses caractéristiques (moyenne, variance, fonction d'autocorrélation) à partir d'une seule trajectoire du processus, c'est-à-dire à partir d'une observation du processus et, par conséquent, de façon pratique, à partir de la série temporelle observée suffisamment longtemps. En bref, on décide que la série observée est typique du processus.

Ainsi, par exemple, l'espérance du processus est la limite, quand la durée d'observation tend vers l'infini, de la moyenne des valeurs des observations de la série.

Il existe un ensemble de conditions nécessaires pour qu'un processus stationnaire soit ergodique. Ceci est l'objet de la théorie ergodique (cf. la conjecture du chat de Boltzmann [14]).

2.4. Composantes

2.4.1. Nature des composantes d'une série temporelle

L'examen d'une série temporelle (**une** réalisation d'un processus) permet, en général, de lui reconnaître trois types de composantes : une tendance, une composante saisonnière et une variation aléatoire. D'autres caractéristiques évolutives, comme les chocs, peuvent être également observées mais sont moins intimement liées à la structure de la série. Il est alors utile de séparer ces composantes, et ceci pour deux raisons. La première est de répondre à des questions de bon sens comme celle de la croissance ou la décroissance générale du phénomène observé. L'extraction de la tendance et l'analyse de celle-ci répondront à cette question. Il est intéressant aussi de mettre en évidence la présence éventuelle d'une variation périodique grâce à l'analyse de la composante saisonnière. La seconde de ces raisons est de débarrasser le phénomène de sa tendance et de ses variations périodiques pour observer plus aisément le phénomène aléatoire.

---

[13] Du grec *ergon*, travail.

[14] Ludwig Boltzmann met un chat dans une boîte totalement close et l'observe pendant un temps infiniment long (cette expérience est virtuelle, bien sûr). Le chat passe par tous les états possibles, mort puis vivant à nouveau, à l'envers, sous forme d'une souris, etc. Pourquoi ? Parce que le chat est ergodique et que la trajectoire qu'il suit – suffisamment longtemps – est sensée résumer l'ensemble des états possibles que ce chat peut prendre à un moment donné.

Pour illustrer la décomposition d'une série temporelle, nous avons eu recours à un procédé quelque peu artificiel, à savoir la création d'une série de toutes pièces à partir d'une tendance linéairement croissante (y = 0,3 * t), d'une fonction trigonométrique (y = 100 * cos(0,05 * t)) et d'une fonction densité de type gaussien (un bruit [15]) du type y = 100 * z avec z suivant une loi normale de moyenne 0 et d'écart-type égal à 1. Ainsi notre série s'écrit :

y = 0,3 * t + 100 * cos(0,05 * t) + 100 * z, avec z ~ N (0,1)

Ainsi, la figure 15 représente la série brute, de tendance manifestement croissante, affectée d'une variation périodique. La figure 16 représente la série débarrassée de sa tendance. La figure 17, enfin, montre la série débarrassée de sa tendance et de sa composante périodique.

Figure 15. Série temporelle

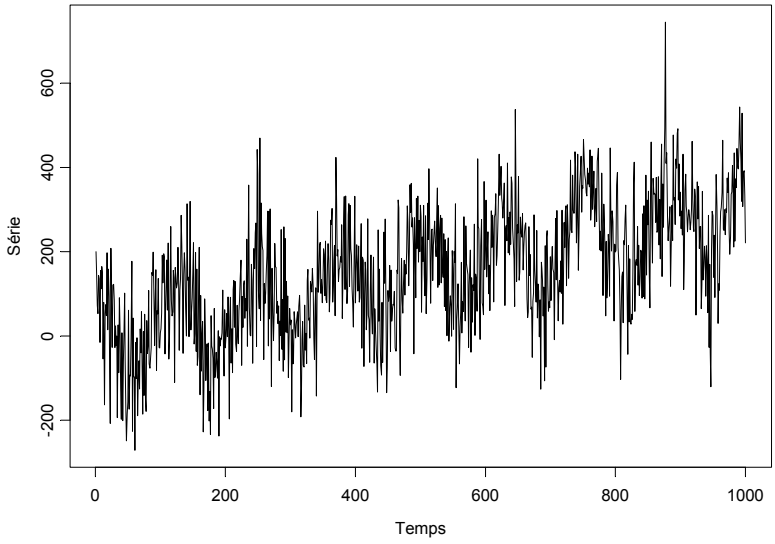
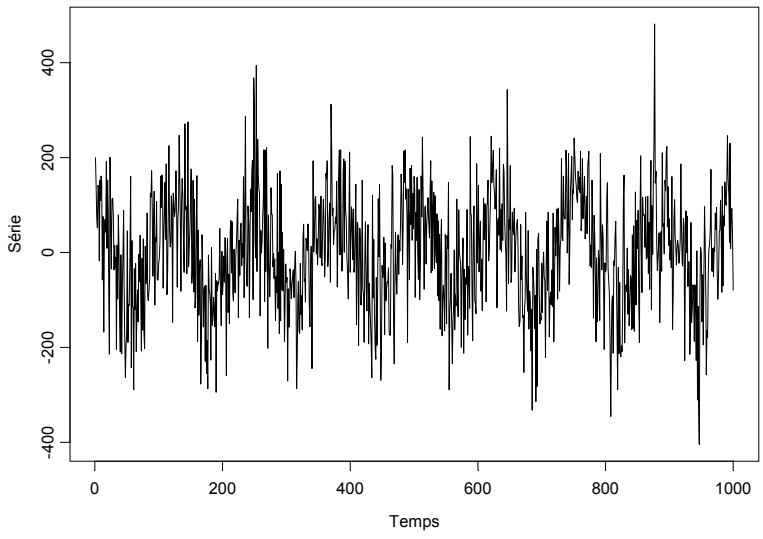
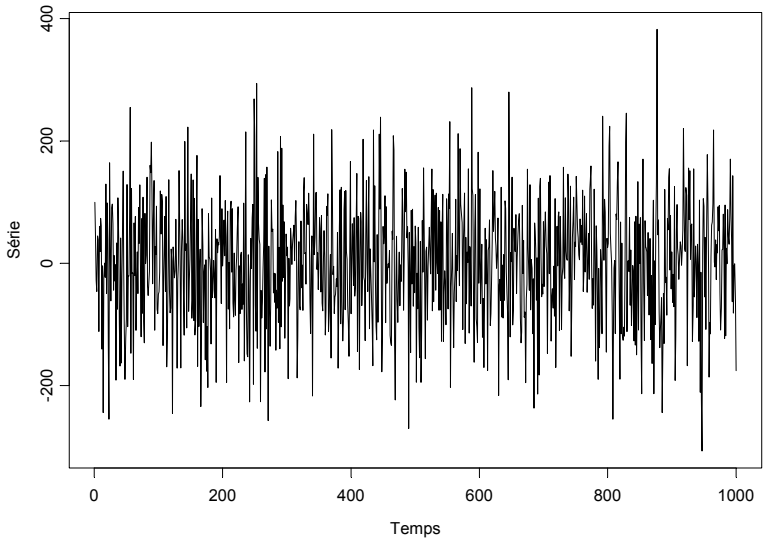


Figure 16. Série temporelle sans tendance



[15] Nous verrons ce qu'est un « bruit » dans la suite du texte.

Figure 17. Série temporelle sans tendance et désaisonnalisée



Ainsi, de façon générale, la série Y_t peut être écrite de la façon suivante :

$$Y_t = m_t + s_t + \varepsilon_t$$

m_t est la tendance indicée par le temps, s_t est la variation périodique, ε_t est le bruit.

2.4.2. Décomposition d'une série temporelle

Dans cette partie, nous verrons un ensemble de techniques et de méthodes empiriques qui permettront de mettre en évidence une tendance ou un phénomène périodique, de l'estimer puis de la ou le supprimer. Ces méthodes font partie de l'approche descriptive de la série et permettent de guider la démarche de modélisation.

Détermination de la tendance

Moyenne mobile

La première de ces méthodes est celle de la moyenne mobile. Cette dernière remplace une valeur y_k de la série par une moyenne de y_k et des valeurs entourant y_k. Le nombre de valeurs concernées (valeurs entourant y_k et y_k lui-même) est appelé ordre de la moyenne mobile. Cette moyenne peut être symétrique (on intègre à la moyenne autant de valeurs de part et d'autre de y_k) ou dissymétrique. Elle peut pondérer certaines valeurs, par exemple en donnant plus de poids aux valeurs entourant y_k de près.

Nous considérerons, ici, la moyenne mobile (d'ordre 2 p + 1) la plus simple et remplacerons y_k par

$$\frac{y_{k-p} + y_{k-p+1} + \ldots + y_{k-1} + y_k + y_{k+1} + \ldots + y_{k+p-1} + y_{k+p}}{2p+1}$$

Pour « extraire » la tendance il faut « filtrer » les variations saisonnières. Pour cela, il convient de choisir p tel que 2 p + 1 soit égal à la période du phénomène périodique [16].

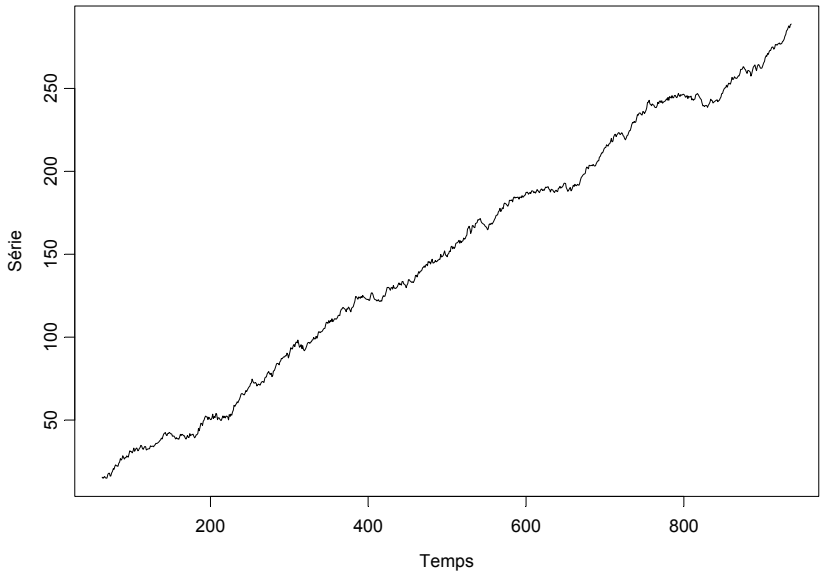
Dans l'exemple précédent, l'observation du graphe et du tableau des valeurs montre que la période est à peu près 125 (ce qui n'est pas surprenant, vu que la série, construite de toutes pièces, contient la fonction périodique cos (0,05 t) de période de $\frac{2\Pi}{0,05}$, soit 125,7 à peu près. On choisit alors p = 62.

La moyenne mobile sera donc, ici :

$$\frac{y_{k-62} + \ldots + y_k + \ldots + y_{k+62}}{125}$$

La figure 18 montre le résultat du filtrage avec mise en évidence de la tendance. Il s'agit d'une tendance linéaire et la pente est proche de 0,3. Nous retrouvons la valeur que nous avions imposée à la tendance du phénomène.

Figure 18. Série temporelle : élimination des variations saisonnières et des variations aléatoires par moyenne mobile d'ordre 125



La moyenne mobile diminue bien sûr le nombre de valeurs de la série.

Régression (ajustement)

Il est aussi possible d'utiliser une régression linéaire pour trouver la tendance. Ici une régression linéaire du type y = a t + b

On trouve pour a : 0,307 avec un écart type (e.t.) égal à 0,01 et pour b : -4,20 (e.t.: 7,66)

[16] Dans le cas où la période du phénomène est paire, la moyenne mobile est un peu plus compliquée mais on y parvient tout de même sans trop de difficulté ([15], page 87).

On peut aussi tenter de représenter la tendance sous toute forme paramétrique, quadratique ($y = a\ t^2 + b\ t + c$) ou autre si le contexte s'y prête.

Élimination de la tendance

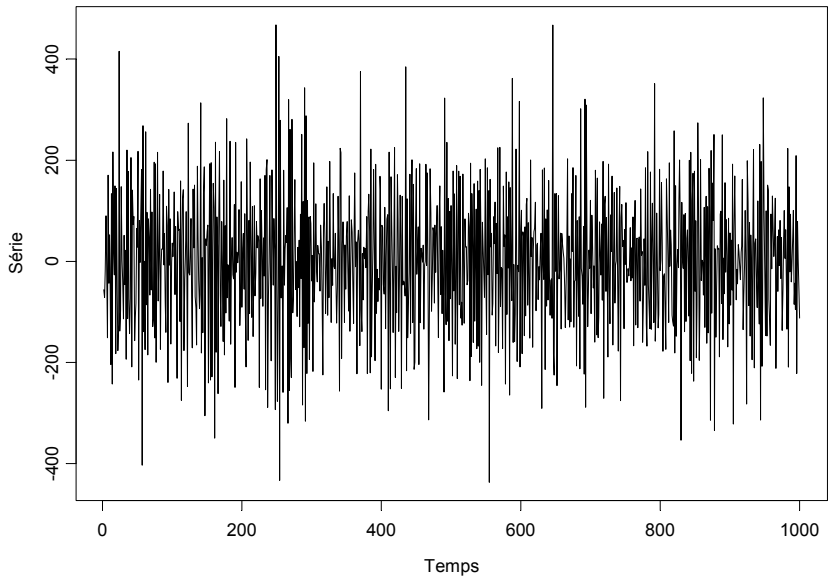
Il peut être intéressant aussi de supprimer la tendance. Une des méthodes simple est, lorsqu'on a déterminé la tendance par un des moyens précédemment exposés, de la soustraire à la série initiale. Ceci peut être représenté par le passage de la figure 15 à la figure 16.

L'autre moyen est de recourir aux différences d'ordre 1 ou d'ordre supérieur. La différence d'ordre 1 (ou première) est :

$$\Delta_t = y_t - y_{t-1}$$

Le graphe de Δ_t est représenté ci-dessous (figure 19) :

Figure 19. Différence d'ordre 1



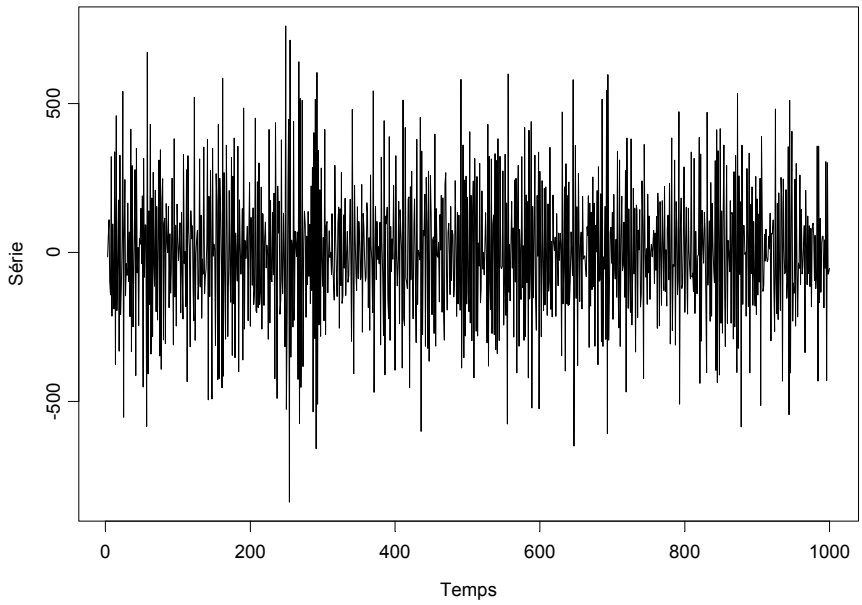
Il est possible d'appliquer à nouveau une différence première à la série des différences Δ_t :

$$\Delta'_t = \Delta_t - \Delta_{t-1} = y_t - y_{t-1} - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2}$$

Δ'_t est appelée différence seconde de la série. Son graphe est représenté sur la figure 20.

Les *différentes différences* diminuent le nombre de points de la série.

Figure 20. Différence seconde.



Détermination de la composante saisonnière

Il existe de nombreuses méthodes, là encore. Ceci dit, avant de déterminer la composante saisonnière, il faut la détecter car sa présence n'est pas toujours évidente. Pour cela, il existe un moyen, basé sur le calcul du coefficient d'autocorrélation exprimé en fonction du décalage entre deux observations (cf. § 2.3.2.).

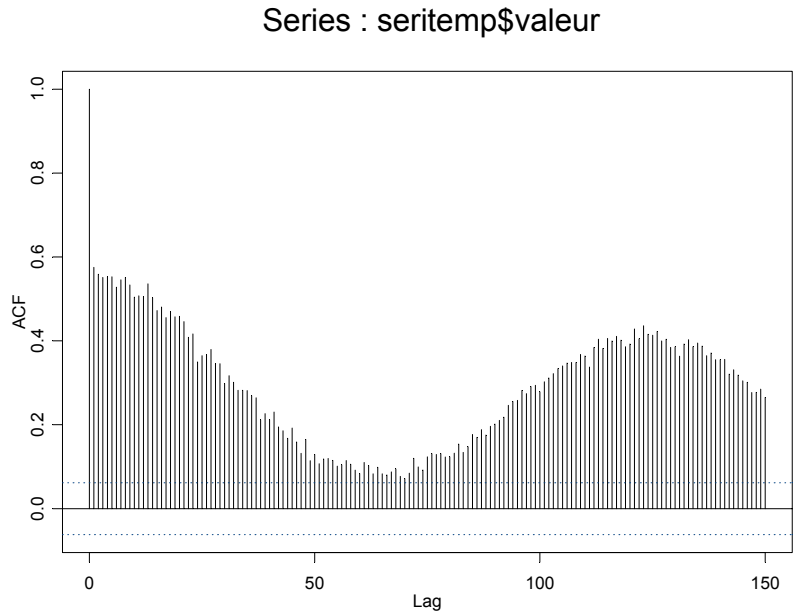
La période correspond, si elle existe, à un maximum d'autocorrélation.

Détection d'une composante périodique éventuelle

Reprenons notre exemple précédent et représentons le coefficient d'autocorrélation de la série en fonction du décalage (cf. § 2.3.2). L'observation de la figure 21 montre que les premiers maxima significatifs du coefficient d'autocorrélation sont observés pour le décalage (*lag*) 0 – ce qui est normal – et pour le décalage 130 environ, ce qui est aussi normal [(17)] (les segments de droites verticaux représentant les valeurs de l'autocorrélation dépassent les lignes en pointillés qui représentent les limites de l'intervalle de confiance du coefficient). Donc la période doit être de 130 jours.

[17] Voir plus haut et voir plus bas.

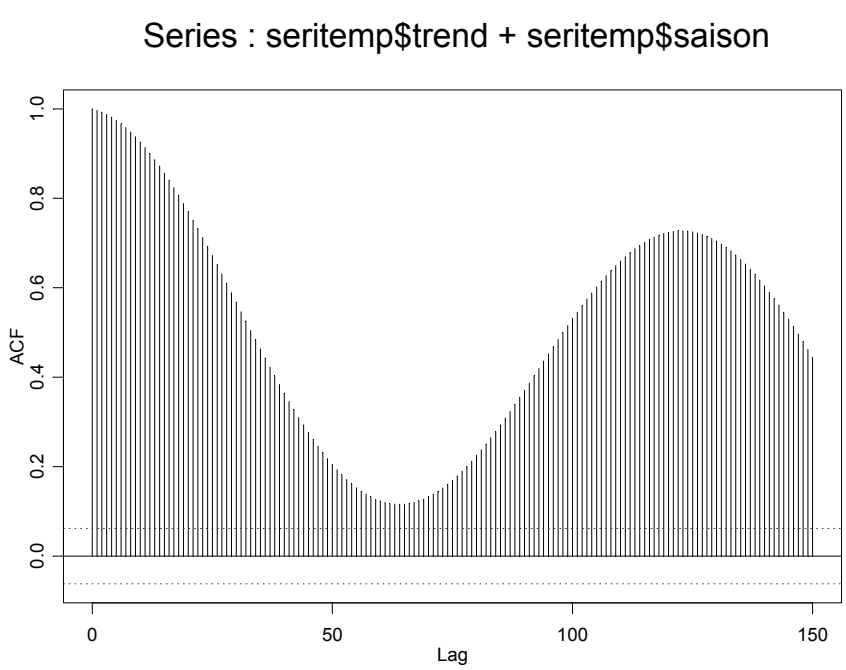
Figure 21. Corrélogramme de la série temporelle



Ceci peut être vérifié plus précisément par les valeurs du coefficient (Annexe 2, tableau 5) : le deuxième maximum du coefficient est obtenu pour le décalage 123.

On peut aussi calculer le coefficient d'autocorrélation pour la série dépourvue de sa composante aléatoire. La figure 22 représente son corrélogramme :

Figure 22. Corrélogramme de la série temporelle sans bruit

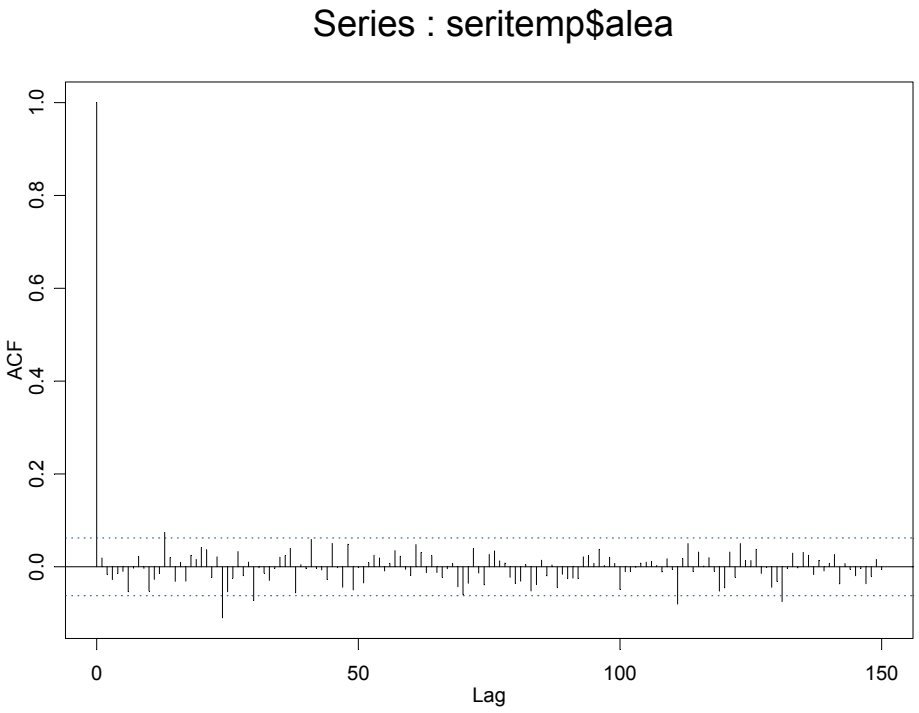


Là encore, on peut vérifier les valeurs du coefficient en annexe 2, tableau 5 : le deuxième maximum du coefficient est obtenu pour le décalage 123, aussi, comme pour la série initiale, bien sur.

123 est donc la valeur de la période du phénomène périodique. Rappelons, à ce propos (cf. § 2.4.1), que la composante périodique de la série avait été générée par la fonction cos(0,05 * t). Nous savons, alors que la période de cette fonction est égale à $\frac{2\Pi}{0,05}$ soit 125,7 [18]. Cette valeur est donc cohérente avec l'estimation de la période par la méthode du coefficient d'autocorrélation.

Terminons en observant le corrélogramme de la série sans tendance et désaisonnalisée (cf. § 2.4.2). Il s'agit de la composante aléatoire (Figure 23. *Figure 23. Corrélogramme de la composante aléatoire.*). Il n'y a quasiment pas d'autocorrélation, dans ce cas.

Figure 23. Corrélogramme de la composante aléatoire.



Estimation des paramètres saisonniers

Considérons la série sous la forme (cf. § 2.4.1.) :

$$Y_t = m_t + s_t + \varepsilon_t$$

Avec, rappelons le, m_t la tendance, s_t la variation périodique, ε_t le bruit.

Supposons que nous soyons fondés à penser que le phénomène périodique s_t, puisse être représenté par quatre paramètres s_1, s_2, s_3, s_4. Ceci pourrait correspondre aux quatre trimestres, par exemple.

Il est possible de supprimer, alors, la tendance m_t par une des méthodes précédentes puis d'estimer les paramètres saisonniers à partir des valeurs de $Y_t - m_t$ grâce à la méthode des moindres carrés, par exemple.

[18] Une fonction périodique du type f ($\omega t + \varphi$) est de période T = $\frac{2\Pi}{\omega}$

Modélisation analytique

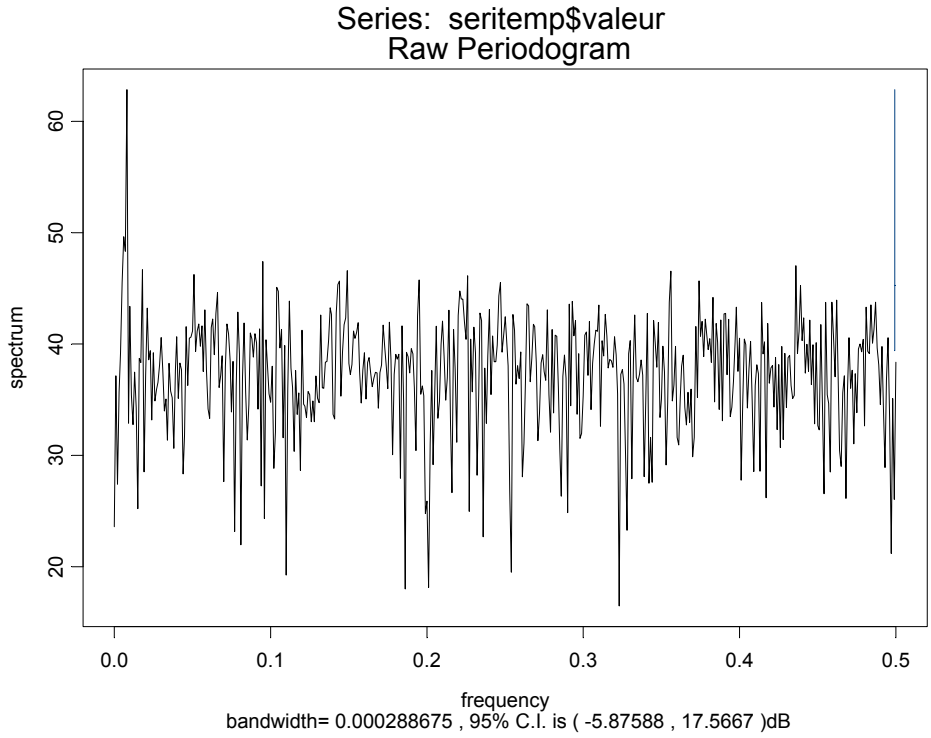
Si l'on a l'impression que le modèle correspond assez bien à une fonction périodique connue, il est possible d'essayer de la modéliser par une fonction de ce type, une fonction trigonométrique, par exemple, comme sin ($\omega t + \varphi$). Il faudra alors estimer ω et φ.

L'analyse spectrale

L'analyse spectrale est la recherche et la détermination du *spectre* de fréquences de la série temporelle. En d'autres termes, elle attribue à chaque fréquence (l'inverse de la période) de la série [(19)] un *poids*. Si une fréquence (ou une période) est fortement représentée dans la série, cette fréquence (ou cette période) se verra attribuer un *poids* important. Au contraire, si une fréquence (ou une période) est absente, elle aura un *poids* nul. La représentation des *poids* en fonction des fréquences est le spectre de la série. Le graphe obtenu est appelé périodogramme.

La figure 24 représente le périodogramme de notre série fabriquée. Celui-ci comporte, en abscisses, les fréquences et, en ordonnées, le spectre c'est-à-dire les poids correspondants aux fréquences.

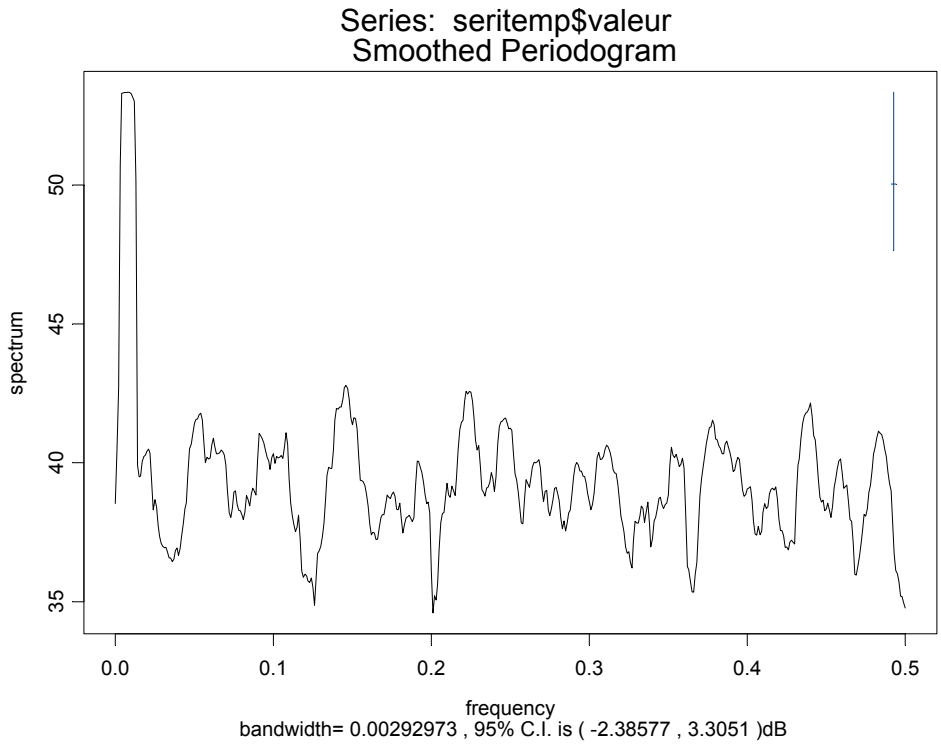
Figure 24. Périodogramme



On peut lisser ce périodogramme pour mieux dicerner les fréquences intéressantes (figure 25).

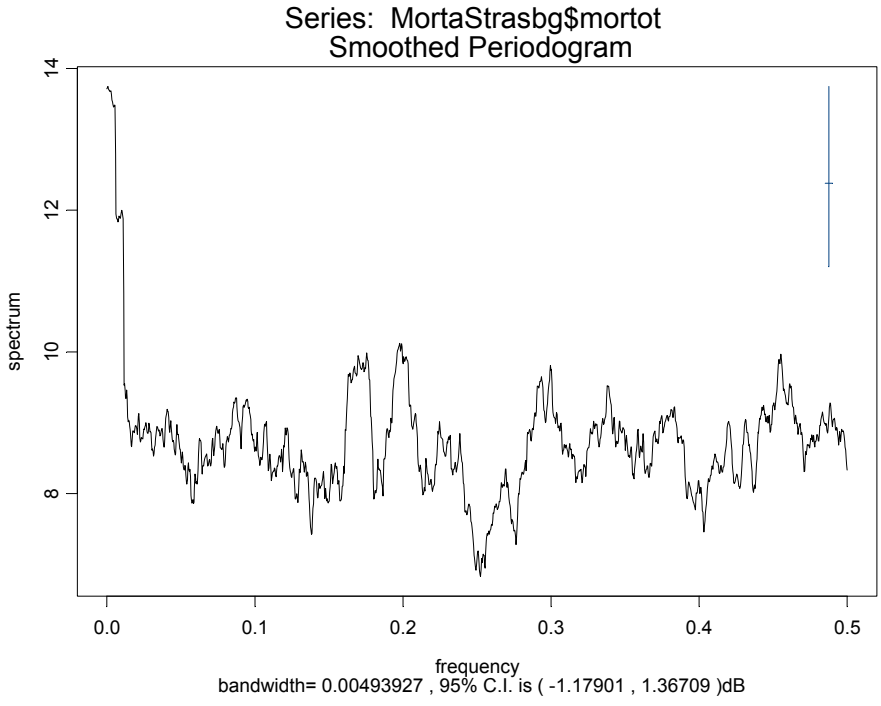
[19] Le principe sur lequel repose l'analyse spectrale est que toute fonction périodique est la somme de fonctions périodiques simples (fonctions sinus et cosinus). Cette somme est appelée décomposition en séries de Fourier.

Figure 25. Périodogramme lissé



La figure ci-dessous (figure 26) montre un exemple de périodogramme lissé, appliqué à une série réellement mesurée (nombre de décès journalier).

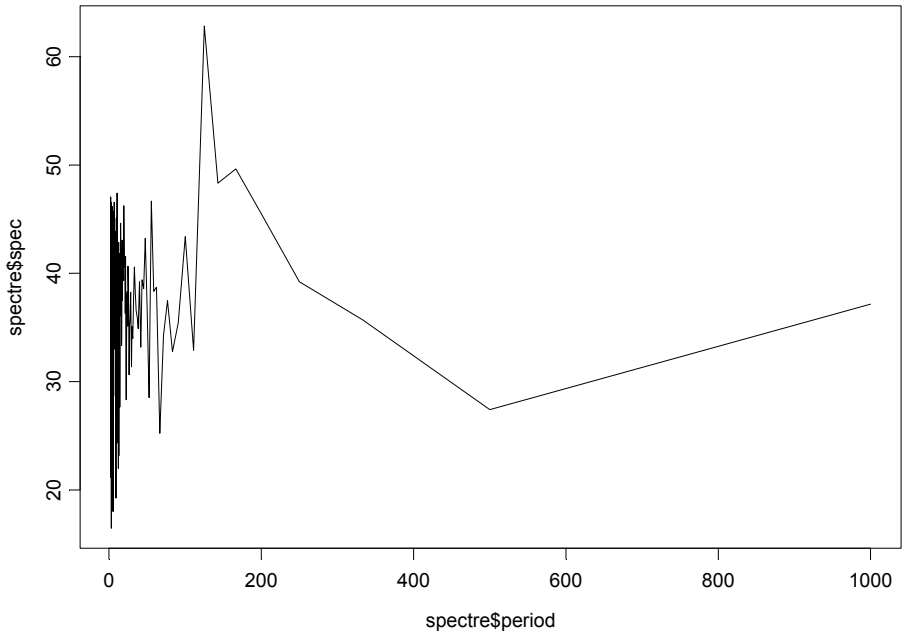
Figure 26. Périodogramme d'une série réellement enregistrée.



Pour connaître les périodes impliquées, il suffit de calculer l'inverse des fréquences (les logiciels statistiques produisent en général un tableau avec les fréquences et les poids correspondants et permettent ainsi de calculer facilement les périodes).

Ainsi, la figure suivante (figure 27) représente le graphe des intensités (en ordonnées) en fonction des périodes (en abscisses).

Figure 27. Périodogramme représentant le spectre des périodes



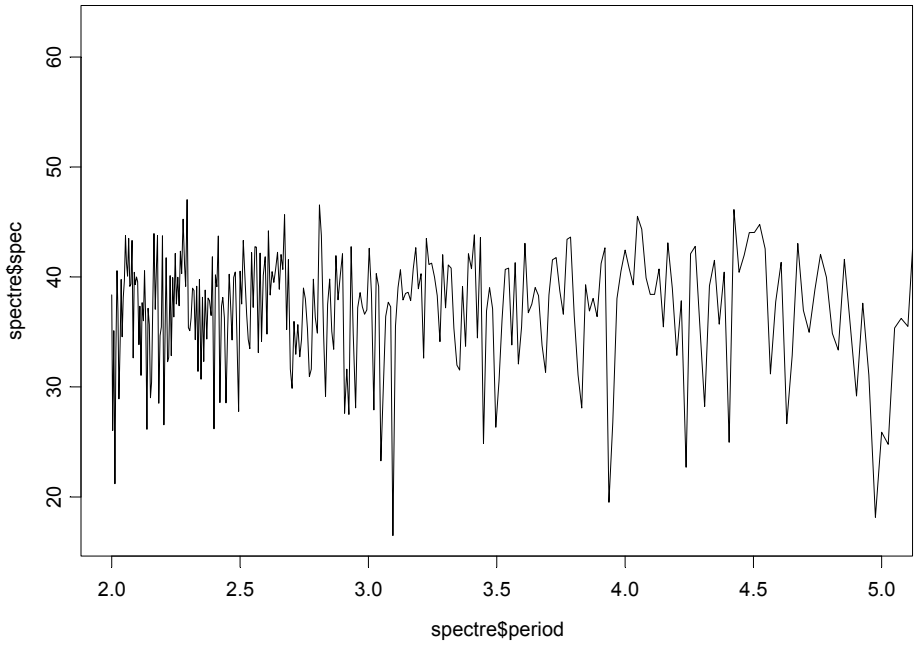
Le pic correspond à la période 125, comme prévu. Il y a d'autres maxima relatifs. Les périodes parmi les plus fortement représentées sont présentées dans le tableau suivant (tableau 1) :

Tableau 1. Périodes les plus représentées dans la série construite.

période	spectre
125,00	62,83
166,67	49,64
142,86	48,32
10,53	47,41
2,29	47,03
55,56	46,67
6,71	46,59
2,81	46,55
19,61	46,25
4,42	46,14
5,13	45,76
2,67	45,66
6,94	45,65
4,05	45,51
200,00	45,50

Si l'on veut regarder ce qui se passe pour les périodes les plus petites, il suffit de « zoomer » (figure 28)

Figure 28. Faibles périodes fortement représentées



Désaisonnalisation

Une méthode élémentaire consiste à déterminer la composante saisonnière selon l'une des méthodes vues précédemment et de la soustraire à la série.

Une autre méthode est basée sur l'application d'un ensemble de moyennes mobiles filtrant la série initiale (figure 18) ou l'application de filtres différences de type $\Delta t = y_t - y_{t-s}$ avec s la période du phénomène.

Ainsi, si l'on veut, à la fois, ôter la tendance et la saisonnalité, il faut appliquer deux filtres *différences* successivement – ici, une différence d'ordre 1 (soit, la différence $y_t - y_{t-1}$) – et une différence d'amplitude égale à la période – ici, une différence du type $y_t - y_{t-s}$ avec s la période.

Remarque. Il ne faut pas confondre l'ordre et l'amplitude d'une différence. L'amplitude est l'écart entre les deux valeurs différenciées ($\Delta t_\delta = y_t - y_{t-\delta}$). L'ordre est le nombre de différences successives réalisées.

2.5. Processus classiques

Les processus, nous l'avons vu, sont au cœur de la notion de série temporelle. Nous passerons en revue rapidement quelques processus souvent cités.

Et tout seigneur tout honneur, nous commencerons notre tour par le bruit blanc, puis nous visiterons le domaine des processus ARMA et famille, des processus de Markov et terminerons dans le chaos.

2.5.1. Bruit blanc

Familier des professionnels de la théorie du signal [14], le bruit blanc [20] est un processus dont

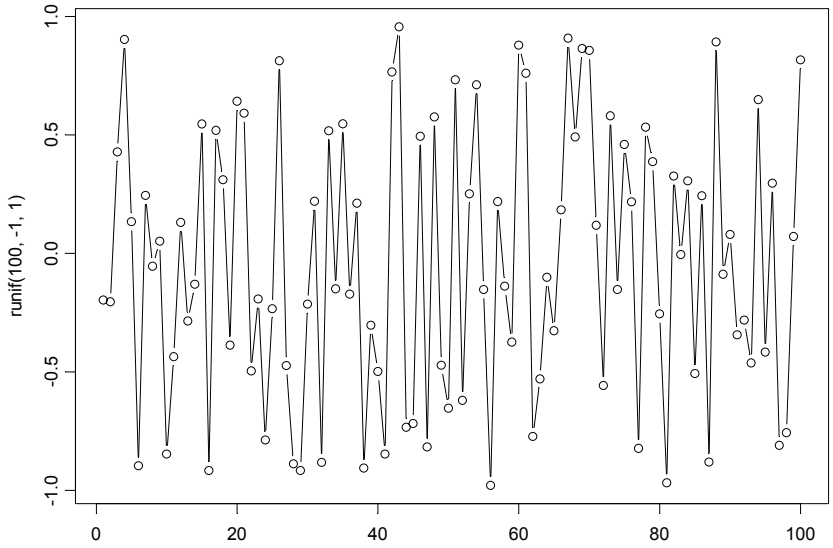
- les espérances (des variables aléatoires constitutives du processus) sont égales à 0,
- les variances sont égales à σ^2, une constante (donc la même pour toutes les variables)
- la fonction d'autocovariance est $\gamma(h)$ avec :
    - o $\gamma(h) = \sigma^2$ si $h = 0$
    - o $\gamma(h) = 0$ si $h \neq 0$

  (ceci veut dire que les variables sont non corrélées)

Quant à la loi de distribution des variables aléatoires, on n'en sait pas grand-chose et même on n'en sait rien. Donc nous sommes plus ou moins libre de rajouter quelques caractéristiques supplémentaires : par exemple, nous pouvons décider que les variables sont iid (*independently and identically distributed*).

Voici, ci-dessous (figure 29), une trajectoire correspondant à un bruit extrait d'une loi uniforme sur [-1,1] :

Figure 29. Série temporelle extraite d'une loi uniforme sur [-1,1].



Voici d'autres exemples de bruits : bruits uniformes (figure 30) ou normaux (figure 31).

Et les bruits de Poisson ? Ce ne sont pas des bruits blancs puisque leur espérance n'est pas nulle (figure 32).

[20] Il existe d'autres types de bruits : bruits roses, bruits blanc faibles, forts, gaussiens, markoviens, etc.. Ces bruits se distinguent par la structure plus ou moins rigide qui les soutient. On se reportera avec profit à des ouvrages dédiés à l'économétrie [**Erreur ! Signet non défini.**] ou à la théorie du signal [**Erreur ! Signet non défini.**]

Figure 30. Bruits uniformes U(-1;1)

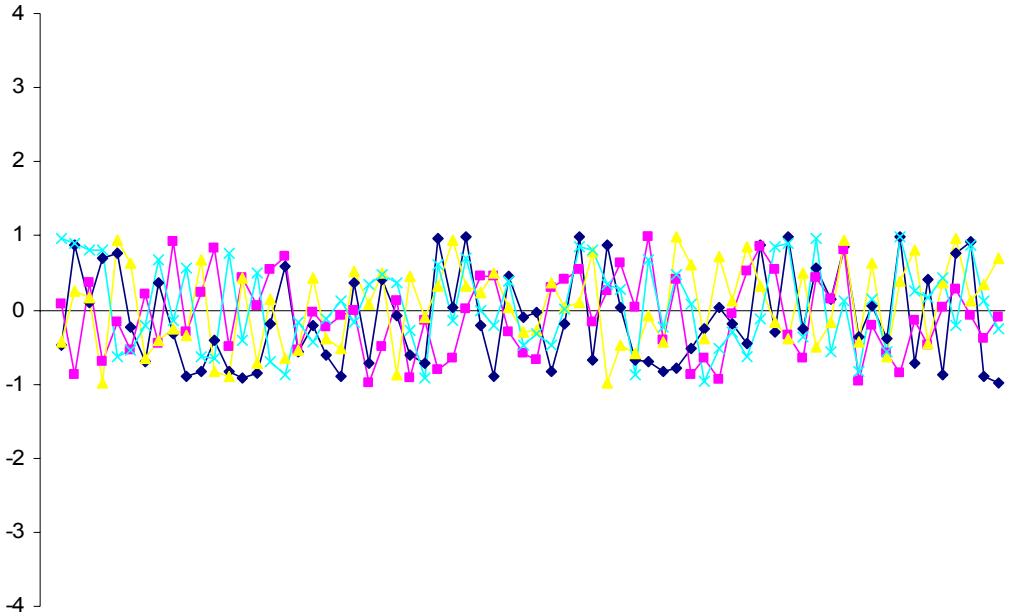


Figure 31. Bruits normaux N(0;1)

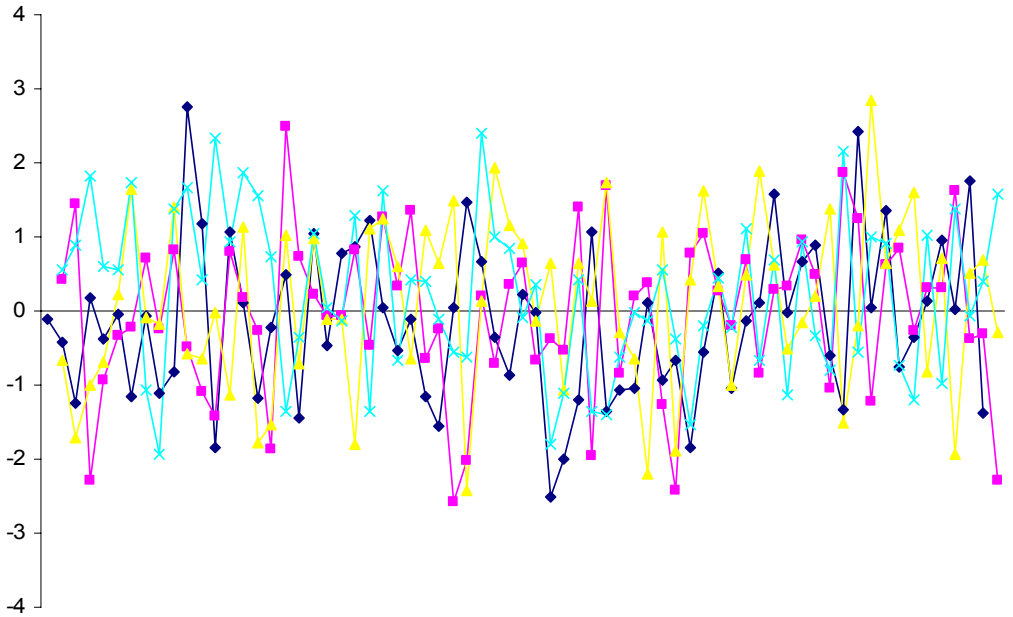
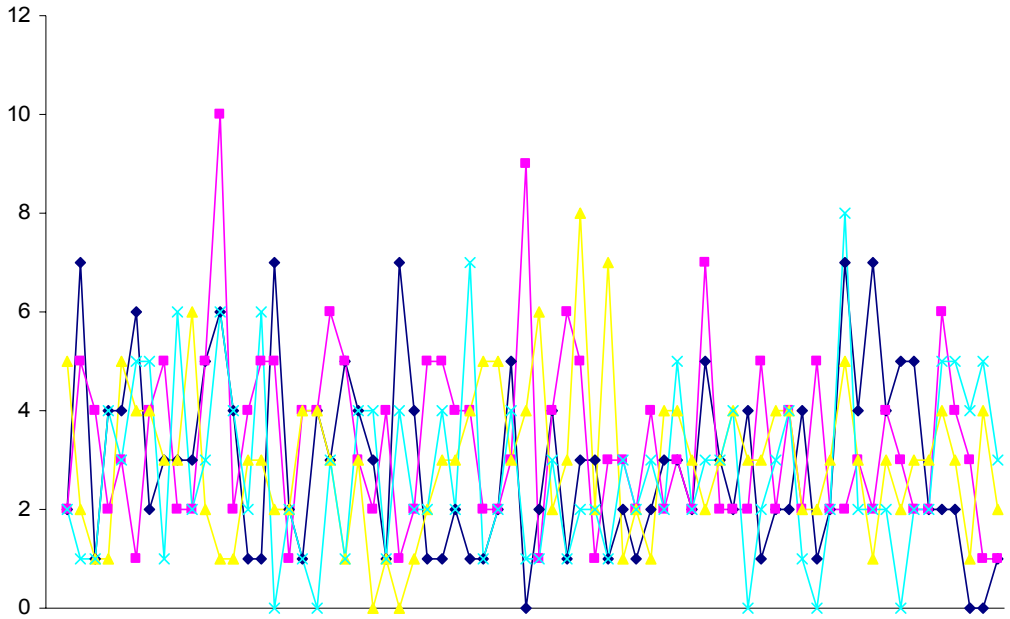


Figure 32. Bruits poissonniens P(3)



2.5.2. Processus ARMA

Que veut dire ARMA ? AR veut dire *autoregressive*, MA veut dire *moving average*.

Un processus ARMA est ainsi composé d'un processus autorégressif et d'un processus de type moyenne mobile. Il nous reste donc à définir ce que sont ces processus. Commençons par le MA de ARMA.

Processus MA (q)

Y_t est un processus MA d'ordre q s'il se présente sous la forme :

$$Y_t = Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} + \ldots + \theta_q Z_{t-q}$$

Avec Z_t, bruit d'espérance 0 et de variance σ^2 et les θ, des paramètres.

En résumé, Y_t est une somme de bruits.

C'est un processus stationnaire.

Remarque. Sous certaines conditions, le processus MA peut s'écrire comme une forme autorégressive infinie (dans laquelle Y_t est exprimé en fonction des autres variables du processus) :

$$\sum_{i=-\infty}^{i=+\infty} \pi_i Y_{t-i} = Z_t$$

Prenons le cas particulier du processus MA(1) : $Y_t = Z_t - \theta Z_{t-1}$

Le coefficient d'autocorrélation a pour valeur :

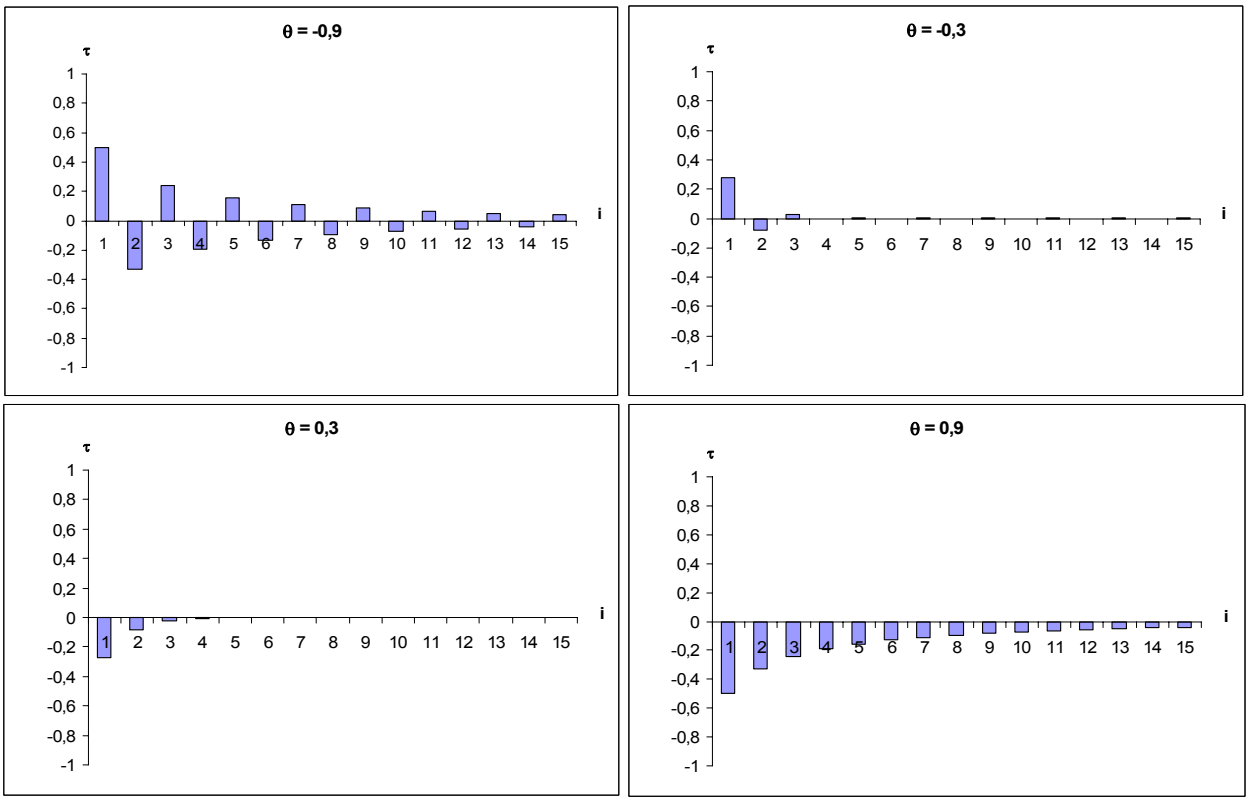
$$\rho_i = \begin{cases} \dfrac{-\theta}{1+\theta^2} & \text{si } i = 1 \\ 0 & \text{si } i > 1 \end{cases}$$

Le coefficient d'autocorrélation partielle est égal à :

$$\tau_i = \frac{-\theta^i\left(1-\theta^2\right)}{1-\theta^{2(i+1)}}$$

Le corrélograme partiel peut ainsi prendre différentes formes selon la valeur de θ (figure 33).

Figure 33. Coefficient d'autocorrélation partielle de MA(1) pour différentes valeurs de θ.



Y_t est un processus AR d'ordre p s'il se présente sous la forme :

$$Y_t = \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \ldots + \varphi_p Y_{t-p} + \varepsilon_t$$

ε_t est un bruit blanc.

Remarque. Sous certaines conditions, le processus AR peut s'écrire comme une somme infinie de bruits blancs :

$$Y_t = \sum_{j=-\infty}^{j=+\infty} h_j \varepsilon_{t-j}$$

Cas particulier du processus AR(1) : $Y_t = \varphi Y_{t-1} + \varepsilon_t$

De plus AR(1) stationnaire $\Leftrightarrow |\varphi| < 1$

Cas particulier de cas particulier : la marche aléatoire

Ce processus particulier s'écrit :

$$Y_t = Y_{t-1} + \varepsilon_t$$

Différentes marches aléatoires existent selon la nature du bruit (figures 34,35 et 36).

Figure 34. Exemples de marches aléatoires normales

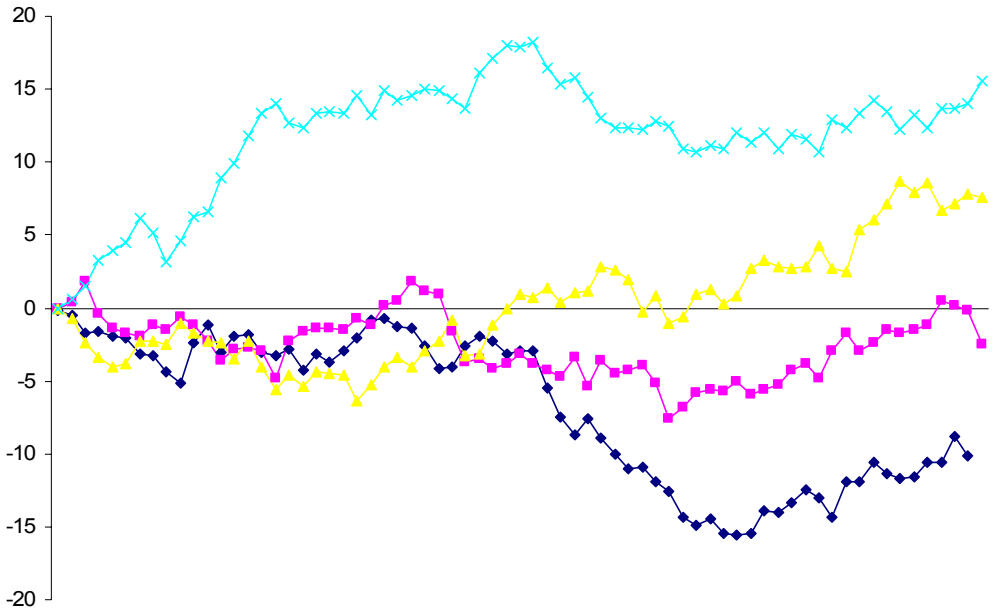


Figure 35. Exemples de marches aléatoires uniformes

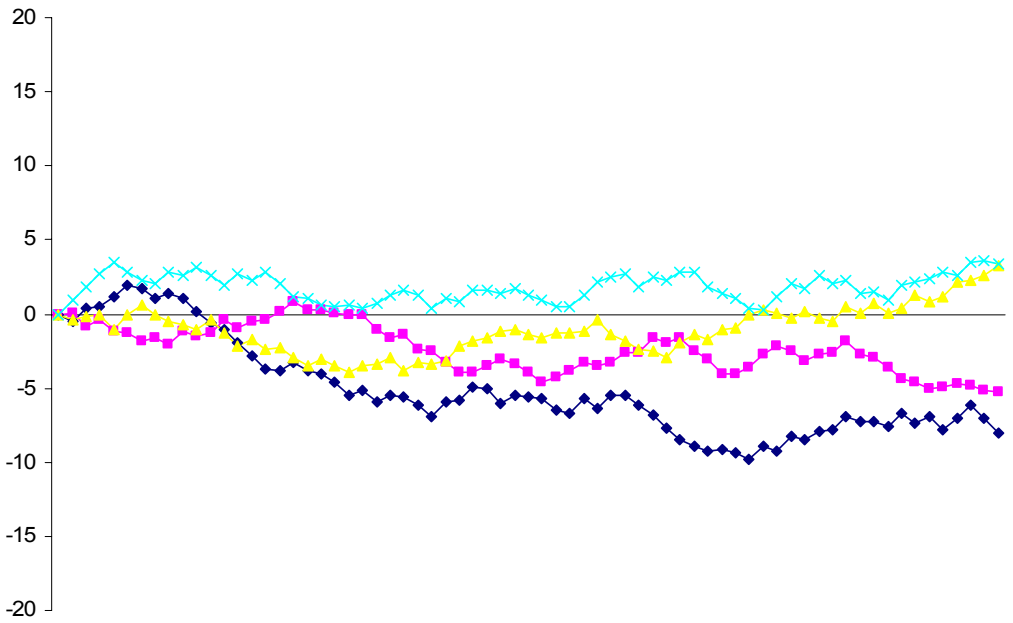
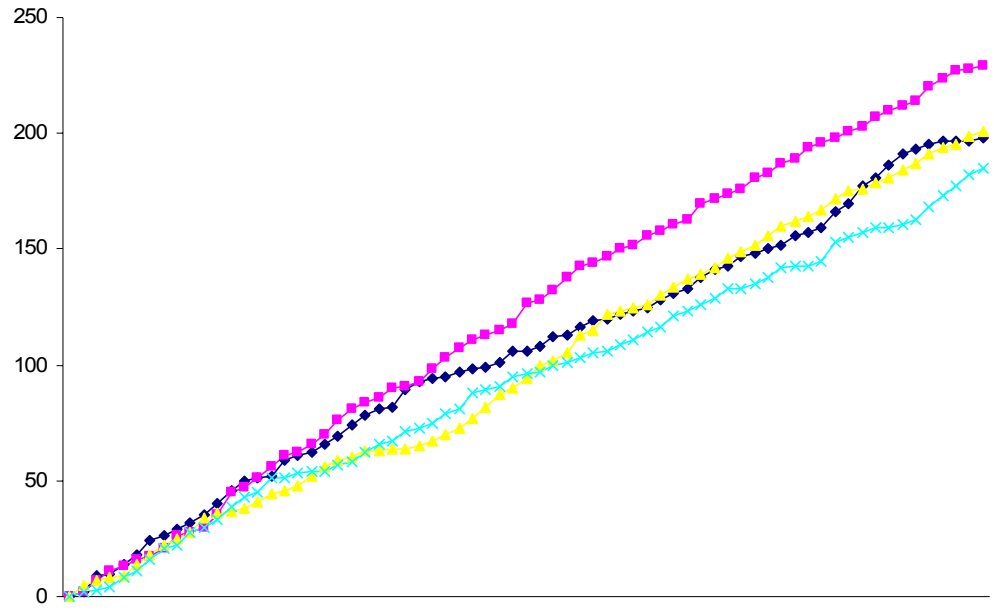


Figure 36. Exemples de marches aléatoires de Poisson



Le coefficient d'autocorrélation du processus AR(1) est :

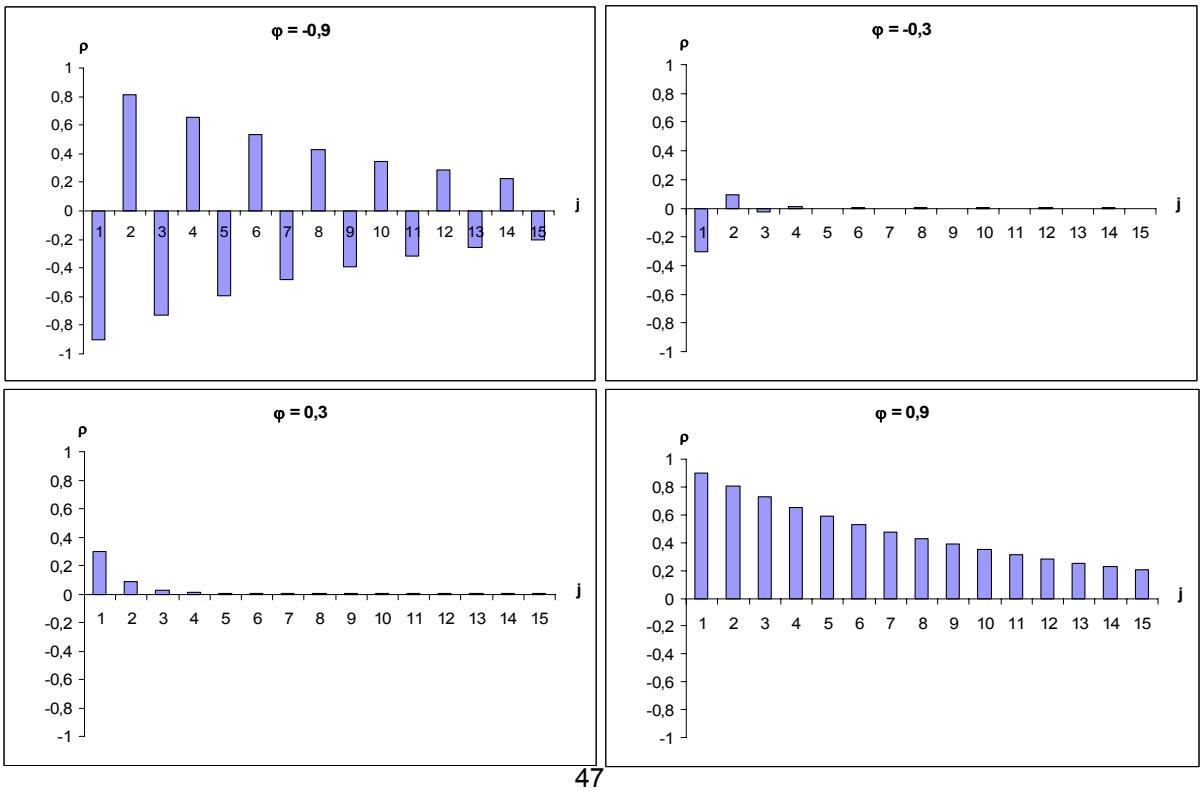
$$\rho_j = \varphi^j$$

Son coefficient d'autocorrélation partielle est :

$$\tau_j = \begin{cases} \varphi \ \text{si} \ j = 1 \\ 0 \ \text{si} \ j > 1 \end{cases}$$

Le corrélogramme peut prendre différentes formes selon la valeur de φ (figure 37).

Figure 37. Coefficient d'autocorrélation de AR(1) pour différentes valeurs de φ



Processus ARMA (p, q)

La notion de processus ARMA réunit celles de « processus autorégressif » et de « processus moyenne mobile » [15]. Dans les modèles ARMA, la valeur prise au temps t par la variable étudiée est une fonction linéaire de ses valeurs passées et des valeurs présentes ou passées d'un bruit blanc.

La forme générale d'un modèle ARMA (p, q) se présente de la façon suivante :

$$Y_t + \varphi_1 Y_{t-1} + \ldots + \varphi_p Y_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

ε_t est un bruit blanc.

Le processus ARMA (p, q) peut être représenté aussi par la symbolique suivante (équivalente de l'écriture précédente) :

$\varphi_p(B)Y_t = \theta_q(B)\varepsilon_t$

B est l'opérateur retard, c'est-à-dire qu'il transforme Y_t en Y_{t-1} : $BY_t = Y_{t-1}$

$\varphi_p(B)$ est le polynôme $1 - \varphi_1 B - \varphi_2 B^2 - \ldots - \varphi_p B^p$

$\theta_q(B)$ est le polynôme $1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^p$

Le modèle ARIMA est un modèle ARMA auquel on applique un caractère de non stationnarité (le rajout d'un terme de tendance ou de saisonnalité par exemple). Le processus ARIMA s'écrit avec la symbolique suivante :

$\varphi_p(B)Y_t \; \nabla^d Y_t = \theta_q(B)\varepsilon_t$

Le terme $\nabla^d Y_t$ est un filtre différence d'ordre d : $\nabla^d = (1 - B)^d$

Ceci mérite des explications :

La différence à l'ordre 1 de Y_t est la série $\nabla^1 Y_t = Y_t - Y_{t-1}$;

La différence à l'ordre 2 est la série $\nabla^2 Y_t = \nabla^1\left(\nabla^1 Y_t\right) = \nabla^1 Y_t - \nabla^1 Y_{t-1} = Y_t - 2Y_{t-1} + Y_{t-2}$;

Etc.

Donc $\nabla^d Y_t$ est une série transformée par différences successives à partir de Y_t. Or, nous avons vu plus haut que cette manipulation supprimait la tendance. Les différences successives permettent ainsi de rendre la série stationnaire.

On suppose souvent que Y_t est un processus aléatoire normal stationnaire du deuxième ordre : les moments du 1er ordre ($E(Y_t)$) et du 2ème ordre ($E(Y_t^2)$) sont invariants par translation dans le temps [21].

Pour s et t, deux instants quelconques :

$$E(Y_t) = m,$$

$$Cov(Y_t, Y_s) = \gamma_{s-t}$$

avec γ_{s-t} ne dépendant que de (s-t).

Les modèles utilisés le plus fréquemment sont les modèles MA (stationnaire), AR (stationnaire sous certaines conditions), ARMA (stationnaire sous certaines conditions) et ARIMA (non stationnaire),

[21] Pour plus de précision, voir le paragraphe 2.2. (Processus aléatoires).

SARIMA (non stationnaire). Citons un dernier type de modèles : le modèle ARCH, utilisé fréquemment en économie, est défini de la façon suivante [16] :

$$Y_t = \varepsilon_t h_t^{\frac{1}{2}} \quad \text{avec :}$$

$$\varepsilon_t \sim N(0,\sigma^2) \text{ et}$$

$$h_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i Y_{t-1}^2$$

Ce modèle est intéressant de par sa forme non linéaire et parce que sa variance conditionnelle dépend du temps (d'où le H de ARCH qui évoque l'hétéroscédasticité).

2.5.3. Processus de Markov

Les processus ou chaînes de Markov [17] [(22)] sont des outils particulièrement intéressants pour modéliser les phénomènes liés au temps lorsque ceux-ci sont caractérisés par une autocorrélation. Ainsi, les modèles à compartiments ou multi-étapes, fort utiles pour décrire l'histoire de la maladie en cancérologie (pas de cancer → cancer dépistable → cancer symptomatique → décès par cancer), en infectiologie (susceptible → infecté → résistant) et dans de nombreux autres domaines, tirent un bénéfice certain de l'utilisation de ces processus. Les processus de Markov servent aussi, comme base des méthodes de Monte Carlo par chaînes de Markov (MCMC), outil d'échantillonnage puissant, permettant de résoudre de nombreux problèmes de calcul d'intégrales inaccessibles à la résolution formelle.

Une chaîne de Markov est un processus temporel discret (c'est-à-dire discontinu) Y_t, dont la distribution au temps t, conditionnellement aux valeurs précédentes (Y_{t-1}, Y_{t-2}, …, Y_{t-k}, …, Y_0) ne dépend que de Y_{t-1}. Dit autrement, le processus de Markov a une mémoire mais une mémoire du temps t-1 uniquement. Ce qui peut s'écrire :

$$P (Y_t \in A \mid Y_0, Y_1, \ldots, Y_{t-1}) = P (Y_t \in A \mid Y_{t-1})$$

Avec A, sous-ensemble quelconque de valeurs possibles du processus. A est appelé aussi « un des sous-ensembles des états du processus markovien ».

Cette définition signifie que la probablité de se retrouver à l'intérieur d'un ensemble d'états (donc de se retrouver dans un des états d'un sous-ensemble donné), conditionnellement à tous les états précédents ne dépend que de l'état « juste avant ».

Il est possible d'exprimer cette probabilité d'une autre façon : grâce aux probabilités de transition :

$$P_{ij} (t) = P(Y_t = j \mid Y_0 = i)$$

Les probabilités de transition P_{ij} (t) sont attribuées à tous les couples d'états i et j et dépendent du temps. Pij représente la probabilité que le processus se trouve dans l'état j au temps t, sachant qu'il se trouvait dans l'état i au temps 0.

Un exemple simple de processus de Markov est le processus AR(1) c'est-à-dire :

$$Y_t = \varphi Y_{t-1} + \varepsilon_t$$

On voit ici que : $P (Y_t \mid Y_0, Y_1, \ldots, Y_{t-1}) = P (Y_t \mid Y_{t-1})$

L'écriture du processus exprime Y_t avec une fonction linéaire de Y_{t-1}. On l'appelle processus markovien linéaire.

Lorsque φ est égal à 1, Y_t est la marche aléatoire.

---

[22] La chaîne de Markov est un processus de Markov particulier puisque les t sont discrets tout comme les valeurs prises par les variables (ce n'est pas le cas pour le processus qui peut prendre ses valeurs dans IR). Voir plus bas.

2.5.4. Systèmes dynamiques déterministes et aléatoires, chaos

Un système dynamique est un système qui évolue avec le temps. Il peut être d'ordre physique, chimique, écologique, épidémiologique, physiologique, économique, géographique, géologique, informatique, mécanique, zoologique, statistique, acoustique, etc. Son évolution temporelle peut être traduite par un ensemble fini d'équations : équations différentielles lorsque le temps est considéré comme continu, équations aux différences quand le temps est considéré comme étant discret [18].

Il est impossible, dans le cadre de ce manuel, de traiter du vaste chapitre des systèmes dynamiques et du non moins vaste chapitre du chaos déterministe. Cependant, comme la matière est passionnante, nous ne résisterons pas à la tentation d'en évoquer au moins le principe. Et, en fait de principe, il s'agira d'un exemple. Un des plus célèbres, sinon le plus célèbre des exemples de chaos déterministe. Il y a une quarantaine d'années encore, les biologistes pensaient que les systèmes dynamiques (populations, influx nerveux, etc...) évoluaient systématiquement vers un état d'équilibre (grâce à des facteurs régulateurs) ou, tout au plus, se comportaient de façon périodique. Là encore, comme ailleurs, quand on observait un comportement erratique, on pensait avoir affaire à un artefact, à l'influence des conditions extérieures ou à une erreur de mesure. Au début des années 70, des écologistes ont eu l'intuition que le désordre observé dans les oscillations de la taille des populations animales et végétales était inhérent à ce système : les équations étaient déterministes. Robert May a repris les équations utilisées par les entomologistes pour caractériser l'évolution de la taille d'une population et, en particulier ce qu'on appelle l'équation logistique [19]. Attardons nous quelques instants sur cette notion.

Soit N_t, l'effectif de la population au temps t, t étant considéré comme discret (l'année, le mois, etc.). Soit N_{max}, l'effectif maximum atteint par N_t et soit $x_t = \frac{N_t}{N_{max}}$.

x_t est donc la fraction de la population maximale au temps t et on a : $0 \leq x_t \leq 1$.

Le modèle utilisé pour définir x_t est le suivant :

$$x_{t+1} = r\, x_t\, (1 - x_t)$$

Où r représente un taux de croissance de la population.

Cette formulation signifie que l'effectif de la population à l'instant t+1 est « proportionnel » au nombre de descendants issus de la population telle qu'elle était au temps t. C'est la partie de l'équation :

$$x_{t+1} = r\, x_t$$

Mais $1-x_t$ est un facteur de régulation *rétroactif* lié à la disponibilité des ressources (nourriture) : plus la population est grande à l'instant t, plus elle consomme de nourriture et moins elle en laisse à la génération suivante qui voit du coup son effectif diminuer. Ainsi x au temps t+1 est une fonction croissante de x et de $1-x$, au temps t. Il s'agit d'une fonction non linéaire (quadratique, en fait).

Cette équation permet de calculer, à partir de la valeur de x_t celle de x au temps t+1 puis, par itération, la valeur de x à un instant quelconque. Robert May s'est aperçu que, suivant la valeur de r, le comportement de la population est totalement différent :

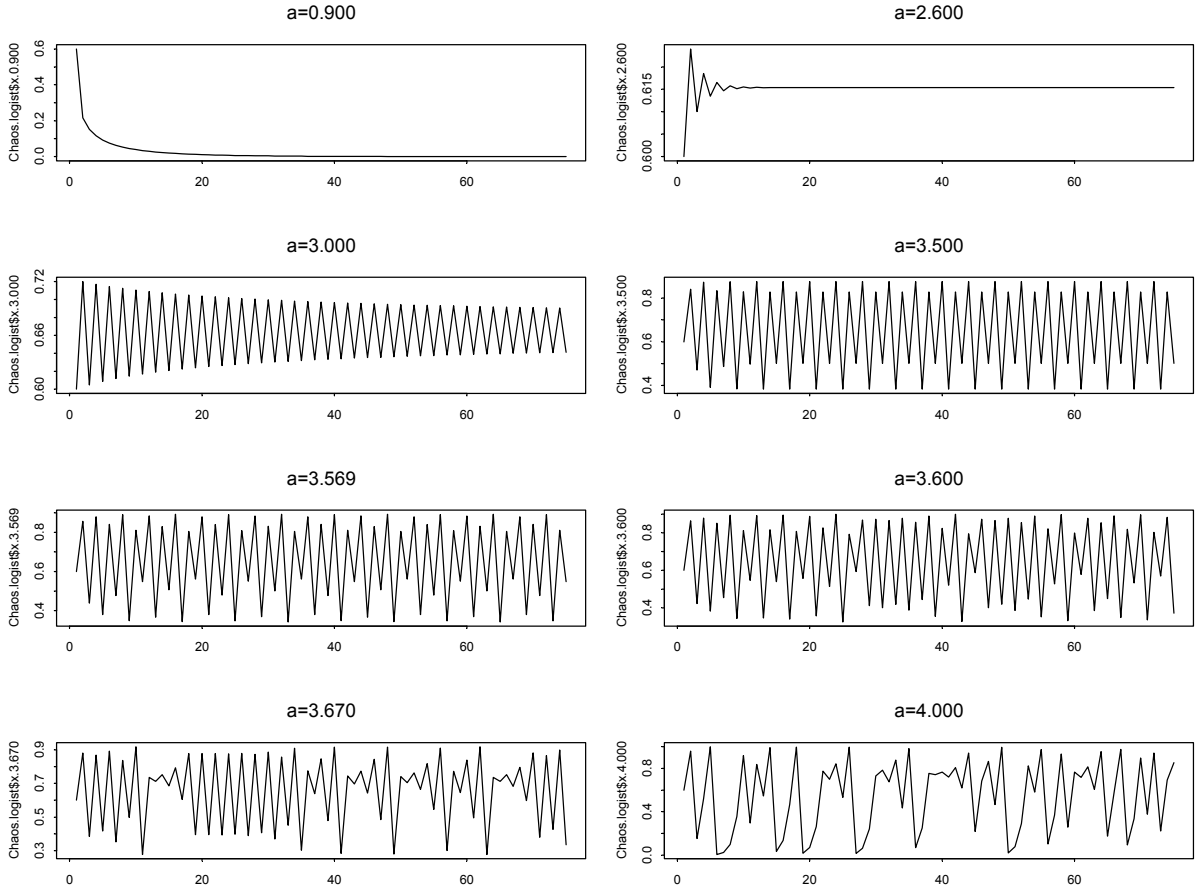
- Si $r \leq 1$: $x_t \rightarrow 0$ quand $t \rightarrow \infty$;
- Si $1 < r < 3$: $x_t \rightarrow x_{lim}$ (en oscillant), x_{lim} étant une valeur dépendant de r ;
- Si $3 \leq r < 3{,}57$: apparition de cycles ; en effet, lorsqu'on fait augmenter la valeur du paramètre, au bout d'un certain nombre d'itérations, x oscille entre 2 valeurs puis, alors que r augmente encore, x oscille entre 4 valeurs puis 8, 16, 32,... ;
- Si $3{,}57 \leq r < 3{,}6786$: x se comporte de façon totalement chaotique ;
- Si $3{,}6786 \leq r < 4$: le chaos disparaît et des cycles de périodes 3, 6, 12, 24,... puis 7, 14, 28,... s'installent puis le chaos réapparaît etc...

La figure 38 représente les variations du processus logistique pour différentes valeurs du paramètre : 0,9 (x_t tend vers 0), 2,6 (x_t tend vers 0,615384615 après quelques oscillations), 3 (x_t oscille entre 2 valeurs : 0,690509805 et 0,641118043), 3,5 (x_t oscille entre 4 valeurs : 0,874997264, 0,382819683, 0,826940707 et 0,50088421), 3,569 (x_t oscille entre 160 valeurs), 3,6 (comportement chaotique), 3,67 (comportement chaotique) et 4 (alternance de chaos et de cycles de périodes impaires).

En épidémiologie les systèmes chaotiques ont été utilisés pour modéliser les séries temporelles représentatives des maladies infectieuses, comme la rougeole [20-21].

Remarque. Il est possible d'adjoindre au modèle déterministe une composante aléatoire.

Figure 38. Comportement de l'équation logistique en fonction de la valeur du paramètre.



2.6. Intérêt des séries temporelles

2.6.1. De façon générale

Il est d'usage de considérer l'intérêt des séries temporelles selon trois perspectives : descriptive, explicative et prévisionnelle.

Description

- L'analyse temporelle permet de connaître la structure de la série de données étudiée ;

- Elle peut être utilisée pour comparer une série à d'autres séries (varicelle et oreillons, par exemple) ;

Explication

- Les variations d'une série peuvent être expliquées par une autre série (exposition météorologique, pollution atmosphérique, etc.) ;
- Il est possible de modéliser une intervention externe grâce à l'analyse de séries temporelles ;
- Ces analyses permettent de réaliser des scénarios pour la période contemporaine : en agissant sur une variable explicative, il est possible d'observer le comportement de la variable expliquée ;

Prévision

La prévision *a priori* permet la planification ;

La prévision *a posteriori* permet d'estimer l'impact d'une perturbation (dépistage, par exemple) sur la variable expliquée ;

Des scénarios pour le futur, enfin, peuvent être réalisés.

2.6.2. Dans le domaine environnemental

L'analyse de séries temporelles n'est pas la seule façon de traiter des relations entre un indicateur d'exposition et un indicateur de santé. Les études exposés-non exposés, les études cas-témoins, de panel, les études écologiques géographiques, les études en *case cross over* sont utilisables aussi. Cependant certaines sont plus adaptées à l'étude des effets à long terme (études exposés-non exposés), d'autres sont parfois contestables méthodologiquement parlant (études géographiques), certaines sont plus appropriées au suivi d'individus fragilisés (enfant, asthmatiques, etc.), d'autres font encore l'objet de discussions d'indication (*case cross over*). Le grand avantage des études de séries temporelles est d'analyser des données facilement accessibles en général car mesurées en routine (données de mortalité en population, données d'hospitalisation, données d'exposition, etc.). D'autre part, les analyses de séries temporelles, bénéficiant souvent de longues périodes de données, voient leur puissance statistique être tout à fait honorable.

Dans le champ de l'épidémiologie environnementale, cependant, les séries nécessitent une modélisation incluant des facteurs exogènes. Ceci fera l'objet du chapitre suivant…

3. Modèle

3.1. Principe de la modélisation

Le principe de la modélisation peut se résumer à ces trois questions :

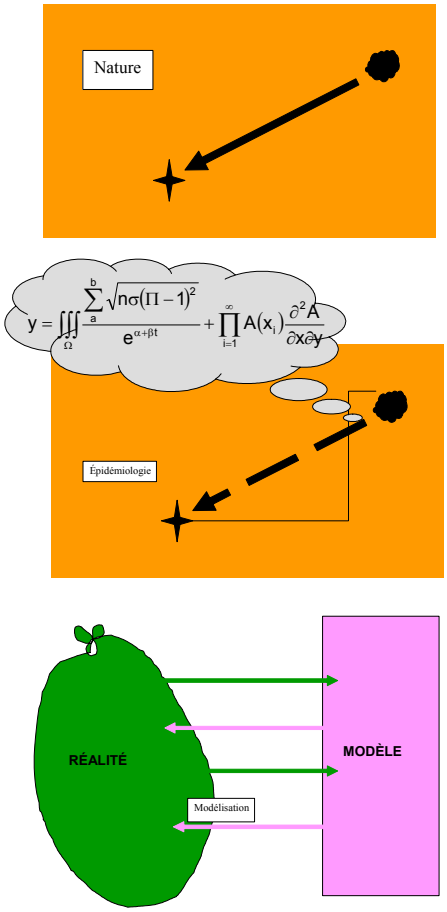
- Pourquoi modéliser ?
- Que modélise-t-on ?
- Comment modéliser ?

La modélisation trouve sa justification dans l'explicitation et la prédiction du phénomène temporel. Dans un modèle, on explique les variations de la variable d'intérêt, dite variable dépendante ou variable expliquée (variable sanitaire, par exemple) par les variations d'un ensemble d'autres variables dites *explicatives* ou *facteurs* ou *variables indépendantes*. Ce peut être le temps lui-même (tendance, variations saisonnières) ou des grandeurs pouvant varier avec le temps (variables météorologiques, charge pollinique, etc.) *ce qui ne veut pas dire qu'elles ne peuvent pas être constantes* (sexe, statut, domicile, etc.). L'explicitation permet de prédire les valeurs que peut prendre la variable expliquée selon celles prises par les facteurs.

La modélisation représente le comportement d'une grandeur naturelle par une expression comportant une partie déterministe (une fonction) et une partie aléatoire (figure 39). La partie déterministe est ce qui permet de décrire le comportement de la moyenne du phénomène (le comportement moyen). La partie aléatoire est le différentiel entre la vraie valeur de la variable étudiée et la partie déterministe.

La modélisation se « fabrique » donc sur deux plans : déterministe, en essayant d'ajuster une forme mathématique à la variation « en moyenne » du phénomène, par des méthodes que nous verrons plus loin ; aléatoire, en donnant une forme à la variabilité du phénomène autour de sa moyenne, en d'autres termes, en donnant une forme au hasard.

Figure 39. Nature, modélisation et épidémiologie



3.2. Modèles

Les deux modèles de régression dont il sera question sont le modèle linéaire généralisé et le modèle additif généralisé.

3.2.1. Modèle linéaire généralisé (GLM)

Le modèle linéaire généralisé (GLM) [22-25] est sans doute l'outil le plus général, le plus utile et, par conséquent, le plus utilisé de la panoplie des instruments dévolus à la modélisation.

L'ancêtre : la régression linéaire simple

Elle est connue et se présente de la façon suivante :

$$Y = \alpha + \beta X + \varepsilon$$

Elle exprime une variable Y en fonction d'une variable X. L'erreur ε (encore appelée « résidu aléatoire ») est non corrélée linéairement avec X. De plus son espérance $E(\varepsilon) = 0$.

La problématique

Avant de définir le GLM, il convient de préciser quelques notations et définitions :

Soit y une grandeur à expliquer, y_i, i de 1 à n, les valeurs prises par la grandeur y lors de n mesures.

Soient x_1, x_2, …, x_j, …, x_p, p grandeurs explicatives. La variable générique x_j prend n valeurs x_{ij}, i de 1 à n.

Les n mesures sont présentées sous forme d'un tableau, comme suit (tableau 2).

Tableau 2. Variable expliquée et variables explicatives. Notations.

n° observation	**y**	$\mathbf{x_1}$	$\mathbf{x_2}$	…	$\mathbf{x_j}$	…	$\mathbf{x_p}$
1	y_1	x_{11}	x_{12}	…	x_{1j}	…	x_{1p}
2	y_2	x_{21}	x_{22}	…	x_{2j}	…	x_{2p}
…	…	…	…	…	…	…	…
i	y_i	x_{i1}	x_{i2}	…	x_{ij}	…	x_{ip}
…	…	…	…	…	…	…	…
n	y_n	x_{n1}	x_{n2}	…	x_{nj}	…	x_{np}

$\mathbf{y} = (y_1, y_2, \ldots, y_i, \ldots, y_n)'$ [23] est le vecteur des observations à expliquer ;

x_{ij} sont les valeurs prises par le facteur $\mathbf{x_j}$ lequel peut s'écrire : $\mathbf{x_j} = (x_{1j}, x_{2j}, \ldots, x_{ij}, \ldots, x_{nj})'$, avec j = 1, 2, …p

[23] **x'** désigne le transposé du vecteur **x**.

Un intermédiaire : le modèle linéaire général

Il met en relation l'espérance des Y_i avec les valeurs prises par les p cofacteurs de la façon suivante :

$$E[Y_i] = \mu_i = \sum_{j=1}^{p} x_{ij}\beta_j \quad i = 1,2,...,n$$

Où les x_{ij} sont les valeurs prises par les covariables (voir tableau ci-dessus) et les β_j, p coefficients à déterminer.

Ceci peut s'écrire aussi de façon vectorielle :

$$E[Y] = \mu = \sum_{j=1}^{p} x_j\beta_j$$

avec $\mu = (\mu_1, \mu_2, \ldots, \mu_i, \ldots, \mu_n)'$

Et de façon matricielle :

$$E[Y] = \mu = X\beta$$

avec $\beta = (\beta_1,\beta_2,\ldots,\beta_j,\ldots,\beta_p)'$

Les variables Y_i sont supposées être normales indépendantes et de variance constante σ^2.

Ceci peut s'exprimer de façon équivalente : les erreurs ε_i (telles que $\varepsilon_i = Y_i - \mu_i$, i = 1, 2, …, n) sont des variables aléatoires indépendantes des X_j et présentant les propriétés suivantes :

$$E(\varepsilon_i) = 0 \text{ et } var(\varepsilon_i) = \sigma_i^2, \forall\, i \in [1,n]$$

$$Cov(\varepsilon_i,\varepsilon_k) = 0, \forall\, (i,k) \in [1,n]^2$$

On suppose aussi en général que ε_i suit une loi normale donc que :

$$\varepsilon_i \sim N(0, \sigma^2) , \forall\, i \in [1,n]$$

Définition du GLM

Le GLM s'écrit [23] :

$$Y_i \sim L_{exp} \qquad \text{et} \qquad \mu_i = E[Y_i]$$

$$\eta_i = g(\mu_i)$$

$$\eta_i = \sum_{j=1}^{p} \beta_j x_{ij}$$

Y_i est une variable aléatoire correspondant à la valeur mesurée y_i,

L_{exp} est une loi quelconque de la famille exponentielle [(24)],

μ_i est l'espérance de Y_i,

η_i est appelé prédicteur,

g() désigne une fonction, dite *fonction de lien*, strictement monotone et différentiable

β_j est un des paramètres à déterminer.

---

[24] La notion de famille exponentielle sera définie plus loin.

Remarque. La fonction de lien donne la forme de la relation entre (l'espérance de) la variable expliquée et les variables explicatives. Cette relation, condensée se présente comme ci-dessous :

$$g(E[Y_i]) = \sum_{j=1}^{p} \beta_j x_{ij}$$

Ceci donne plus de souplesse au modèle mais permet aussi de faciliter le calcul de la vraisemblance en choisissant correctement cette fonction, compte tenu de la loi de probabilité qui a été attribuée à la variable expliquée (voir, plus bas, la notion de fonction de lien canonique). Cette fonction est strictement monotone, c'est-à-dire ou bien croissante strictement ou bien strictement décroissante sur l'ensemble de son intervalle de définition. Elle est différentiable, ce qui signifie qu'il est possible de calculer sa dérivée première.

Le GLM peut encore s'écrire, en utilisant la notation vectorielle et matricielle :

$$\mathbf{Y} \sim L_{exp} \qquad \text{et} \qquad \boldsymbol{\mu} = E[\mathbf{Y}]$$
$$\boldsymbol{\eta} = g(\boldsymbol{\mu})$$
$$\boldsymbol{\eta} = \mathbf{X}\beta$$

$\mathbf{Y} = (Y_1, Y_2, \ldots, Y_i, \ldots, Y_n)'$ est le vecteur des variables aléatoires,

$\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_i, \ldots, \mu_n)'$ est le vecteur des espérances,

$\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_i, \ldots, \eta_n)'$ est le vecteur des prédicteurs,

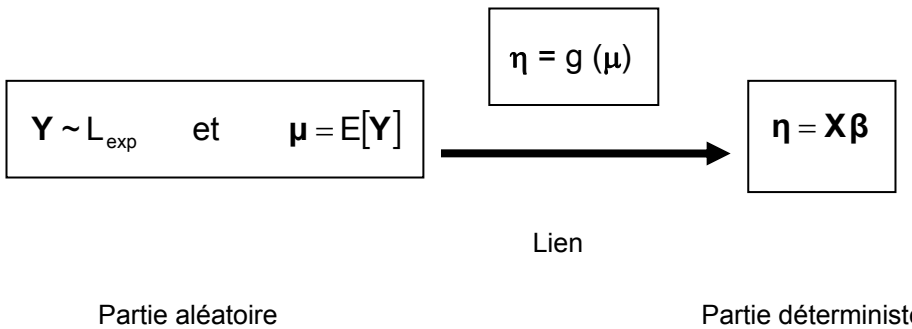
$\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_j, \ldots, \beta_p)'$ est le vecteur des paramètres à déterminer,

$\mathbf{X}$ est la matrice des cofacteurs :

$$X = \begin{bmatrix} x_{11} & \ldots & x_{1j} & \ldots & x_{1p} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{i1} & \ldots & x_{ij} & \ldots & x_{ip} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{n1} & \ldots & x_{nj} & \ldots & x_{np} \end{bmatrix}$$

On reconnaît ainsi deux parties à l'expression du GLM : une partie déterministe (mathématique, si l'on préfère), c'est la combinaison linéaire des covariables, et une partie aléatoire, c'est la distribution de la variable expliquée. Entre les deux : la fonction de lien (figure 40).

Figure 40. Écriture du GLM



Ainsi, de façon générale, le GLM met en relation l'espérance d'une variable dépendante **Y** avec un ensemble de covariables $\mathbf{X_j}$ selon certaines conditions [23] : 1°) la fonction de distribution de la variable **Y** appartient à la famille exponentielle, 2°) l'espérance de **Y** est reliée à une grandeur η (prédicteur linéaire) par l'intermédiaire d'une fonction g (fonction de lien), monotone et différentiable, 3°) le prédicteur linéaire s'exprime comme combinaison linéaire des covariables.

Famille exponentielle

Nous avons vu, plus haut, la forme générale du GLM. Il était question de la famille exponentielle. Explorons plus avant cette notion. La famille exponentielle est un ensemble de lois dont l'écriture est résumée par une formule unique et possédant des propriétés communes. L'intérêt de cette écriture est qu'un ensemble de résultats peut être obtenu de façon globale puis décliné selon les particularités propres à chaque loi.

Une loi de probabilité est définie par sa densité [(25)] : l'écriture générale de la densité de probabilité d'une loi exponentielle se présente de la façon suivante :

$$f_Y(y \mid \theta,\phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right)$$

Cette écriture, un peu compliquée résume la forme de toutes les densités des lois appartenant à cette famille. L'expression $f_Y(y \mid \theta,\phi)$ représente la fonction densité de Y (ou des observations y) *étant donnés* θ et ϕ, autrement dit conditionnellement à θ et ϕ.

Quelques définitions :

θ est le « paramètre naturel » de la famille exponentielle appelé paramètre « canonique ».

ϕ est le paramètre de dispersion.

θ est une fonction de μ : $\theta(\mu)$, définie par la relation :

$$E(Y) = \mu = b'(\theta) = \frac{\partial b(\theta)}{\partial \theta}$$

La variance de Y est :

$$\mathrm{var}(Y) = b''(\theta)a(\phi) = \frac{\partial^2 b(\theta)}{\partial \theta^2}a(\phi)$$

La variance de Y est donc composée de deux termes distincts. L'un est une fonction de θ, l'autre est une fonction de ϕ.

---

[25] La notion de densité de probabilté en un point (ici, une valeur de y) n'est pas simple à expliquer mais elle se comprend intuitivement. En effet, la probabilité en un point ne peut être écrite. Ce qu'on peut exprimer formellement c'est la probabilité sur un intervalle très (infiniment) petit dy. Cette probabilité (infiniment petite, en général, elle aussi) s'écrit :

$$dP = f(y)\,dy$$

Avec f(y), la densité de probabilité en y. L'expression précédente peut s'écrire :

$$f(y) = \frac{dP}{dy}$$

On en déduit que la densité de probabilité est la dérivée de la fonction probabilité.

Puisque θ dépend de μ, $b''(\theta)$ dépend de μ.

$b''(\theta)$ est appelé fonction de variance. La fonction de variance est définie par la relation :

$$v(\mu) = b''(\theta) = \frac{\partial^2 b(\theta)}{\partial \theta^2}$$

Quant à la fonction $a(\phi)$, elle se présente sous la forme :

$$a(\phi) = \frac{\phi}{\omega}$$

Le paramètre de dispersion ϕ qui s'écrit aussi σ^2, est constant alors que ω est un poids propre à chaque observation.

Le choix de la fonction de lien dépend de la loi exponentielle : pour chaque élément de la famille exponentielle, il existe une fonction de lien « naturelle » dite « fonction de lien canonique » telle que :

$$\theta = \theta(\mu) = \eta = X\beta$$

Donc telle que

$$g(\mu) = \theta(\mu)$$

Les modalités relatives aux différentes familles exponentielles (paramètre de dispersion, fonction de lien, fonction de variance, etc.) sont présentées dans de nombreux ouvrages [23,25].

En voici un ou deux exemples.

Commençons par la loi normale :

$a(\phi) = \phi = \sigma^2$, $b(\theta) = \frac{\theta^2}{2}$, $c(y,\phi) = -\frac{1}{2}\left(\frac{y^2}{\phi} + \ln(2\Pi\phi)\right)$, $\theta(\mu) = \mu$, g est la fonction identité, c'est-à-dire $g(\mu) = \theta(\mu) = \mu$, $v(\mu) = 1$

Pour la loi de Poisson, on a :

$a(\phi) = \phi = 1$, $b(\theta) = e^{\theta}$, $c(y,\phi) = -\ln(y!)$, $\mu(\theta) = e^{\theta}$, g est la fonction ln, $v(\mu) = \mu$.

La loi binomiale et la loi gamma, entre autres, sont aussi des éléments de la famille exponentielle. Pour l'expression de leur densité, on se reportera aux ouvrages précités [23,25].

Application à un échantillon

Revenons à notre échantillon $\mathbf{y} = (y_1, y_2, \ldots, y_i, \ldots, y_n)'$

Nous pouvons écrire la densité pour chacune des observations :

$$f_Y(y_i \mid \theta_i, \phi) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right)$$

θ_i est propre à chaque variable Y_i alors que ϕ est supposé constant.

Vraisemblance

Il sera question plus loin de l'ajustement du modèle aux données, c'est-à-dire du calcul, ou plutôt, de l'estimation des paramètres β_j. Pour ce faire, il est nécessaire de définir une notion qui sera utilisée à

cette occasion. C'est la notion de log-vraisemblance. Celle-ci est, comme son nom l'indique, le logarithme népérien de la vraisemblance.

La vraisemblance est une mesure (de probabilité) associée à un échantillon donné. Plus précisément, la vraisemblance d'un échantillon est la probabilité d'observer cet échantillon. Un échantillon est un groupe ordonné de réalisations y_i de n variables Y_i, i de 1 à n. Si on suppose que les variables sont indépendantes (n tirages indépendants), la probabilité d'observer cet échantillon est le produit des probabilités d'observer chacune des réalisations. Aussi, la vraisemblance de l'échantillon, notée L, est :

$L(y_1, y_2, \ldots, y_n) = P[(Y_1, Y_2, \ldots, Y_n) = (y_1, y_2, \ldots, y_n)] = P[Y_1 = y_1] \,.\, P[Y_2 = y_2] \,.\, \ldots \,.\, P[Y_n = y_n]$

Cette expression est adaptée à des lois de probabilité discrète. Quand la loi est continue, il faut exprimer cette propriété à l'aide des densités de probabilité.

Prenons un exemple. Si l'échantillon est tiré d'une loi de Poisson P :

$$P(Y = y) = \frac{e^{-\lambda}.\lambda^{y}}{y!}$$, avec y, nombre entier positif ou nul.

Ainsi pour la réalisation $(y_1, y_2, \ldots, y_n)$ de l'échantillon, on a

$$L(y_1, y_2, ..., y_n) = \prod_{i=1}^{n} P(Y_i = y_i) = \prod_{i=1}^{n} \frac{e^{-\lambda}.\lambda^{y_i}}{y_i!} = \frac{e^{-n\lambda}.\lambda^{\sum_{i=1}^{n} y_i}}{\prod_{i=1}^{n} y_i!}$$

La probabilité de chacune des réalisations individuelles $P(Y_i = y_i)$ est appelée contribution de l'observation à la vraisemblance de l'échantillon.

Remarque. Si les variables sont liées, l'expression de la vraisemblance est légèrement différente (voir plus bas).

Dans ce que nous venons de voir, la distribution de la variable (ou des variables) est supposée connue, c'est-à-dire que l'on suppose connaître les paramètres caractérisant la densité de probabilité, θ et ϕ. En fait, la vraisemblance trouve tout son intérêt quand les paramètres θ et ϕ ne sont pas connus. Il convient alors d'exprimer la vraisemblance comme une fonction du (des) paramètre(s).

On écrira que la contribution de l'observation y_i à la vraisemblance est :

$$L_i(\theta_i , \phi \mid y_i) = f_i (y_i \mid \theta_i , \phi)$$

Avec $f_i (y_i \mid \theta_i , \phi)$, la densité de probabilité de Y_i [(26)].

Nous supposons, comme précédemment que θ_i est propre à chaque variable Y_i et que ϕ est constant.

Pour l'ensemble de l'échantillon, la vraisemblance est, comme on l'a vu plus haut, le produit des contributions individuelles à la vraisemblance :

$$L(\theta, \phi | y) = \prod_{i=1}^{n} f_i(y_i | \theta_i, \phi)$$

La log-vraisemblance s'écrit alors :

$$l(\theta, \phi | y) = \sum_{i=1}^{n} \ln[f_i(y_i | \theta_i, \phi)]$$

---

[26] Remarquons que la densité $f_i (y_i \mid \theta_i , \phi)$ est une fonction de y_i (conditionnellement aux paramètres) alors que $L_i(\theta_i , \phi \mid y_i)$ est une fonction de θ_i et ϕ (conditionnellement à y_i).

Dans le cas où la distribution de probabilité de la variable dépendante appartient à la famille exponentielle, la densité $f_i\ (y_i \mid \theta_i\ ,\ \phi\ )$ a pour forme celle de la densité des lois exponentielles comme il a été vu plus haut.

$L_i(\theta_i\ ,\ \phi \mid y_i)$, contribution de l'observation y_i à l'espérance, se présente sous la forme :

$$L_i\left(\theta_i \mid \phi, y_i\right) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right)$$

Aussi la contribution de y_i à la log-vraisemblance s'écrit :

$$l_i\left(\theta_i \mid \phi, y_i\right) = \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)$$

Il est possible de rencontrer, cependant, des variables expliquées dont la densité de probabilité ne fait pas partie de la famille exponentielle ou, tout au moins, dont on ne connaît pas la forme avec certitude. Dans ces cas, la formulation de la vraisemblance telle qu'elle a été vue précédemment n'est pas applicable et l'on a recours à la quasi-vraisemblance. Cette dernière présente une forme semblable mais non identique. Elle permet cependant, comme la vraisemblance, de réaliser l'estimation des paramètres. La fonction de quasi-vraisemblance est largement décrite dans le chapitre 8 de l'ouvrage de McCullagh et Nelder [23]. L'hypothèse fondamentale dans la définition des GLM est que la distribution de la variable dépendante appartient à la famille exponentielle. Les développements ultérieurs des GLM se fondent sur la remarque fondamentale que l'équation du maximum de vraisemblance (cf. ci-dessus) ne fait intervenir que les deux premiers moments (l'espérance et la variance) [26]. Par extension, Wedderburn [27] propose de l'utiliser comme méthode d'estimation y compris dans le cas où la distribution de la variable dépendante n'est pas entièrement connue. L'hypothèse contraignante des distributions de probabilité étant relâchée, on parle alors de quasi-vraisemblance. Wedderburn [27] et McCullagh [28] ont montré que les paramètres estimés par maximum de quasi-vraisemblance ont beaucoup de propriétés communes à ceux estimés par maximum de vraisemblance. La plupart des fonctions de vraisemblance utilisées dans le cadre de la famille exponentielle (loi de Poisson, par exemple) peuvent être vues comme des fonctions de quasi-vraisemblance si le choix de la fonction de variance correspond à la fonction de variance naturelle associée à une distribution de la famille exponentielle [29]. Dans ce cas, la fonction quasi-score (la dérivée première de la fonction de log-quasi-vraisemblance) correspond à la fonction score (la dérivée première de la fonction de log-vraisemblance) et les estimateurs du maximum de quasi-vraisemblance correspondent aux estimateurs du maximum de vraisemblance. Cependant, et c'est à ce moment que la fonction de quasi-vraisemblance peut jouer un rôle intéressant, il est aussi possible pour les distributions où le paramètre de dispersion est égal à 1 (distribution binomiale et de Poisson) de considérer un paramètre de dispersion variable ; c'est le cas, par exemple, des modèles log-linéaires avec erreurs de Poisson, pour lesquels il est proposé d'introduire un paramètre de surdispersion différent de 1 dans l'expression de la variance. Il est également possible d'envisager d'autres fonctions de variance que celles précisément associées aux distributions classiques de la famille exponentielle. On peut ainsi poursuivre une inférence où les seules suppositions concernant la distribution des données portent sur les deux premiers moments [26].

Le GLM et le temps

Les hypothèses, vues ci-dessus, se traduisent, dans le cas d'une variable indexée par le temps (processus), de la façon suivante :

$$Y_t \sim L_{exp} \qquad \text{et} \qquad \mu_t = E[Y_t]$$

$$\eta_t = g(\mu_t)$$

$$\eta_t = \sum_{j=1}^{p} \beta_j x_{tj}$$

Y_t est la variable expliquée au temps t (t = 1, 2,…, T), L_{exp} est une loi de probabilité de la famille exponentielle comme plus haut, μ_t est l'espérance de Y_t, au temps t, η_t est le prédicteur linéaire au temps t, g est la fonction de lien, (β_1, β_2,… β_j,… β_p)' est le vecteur des paramètres, x_{tj} est la valeur du facteur j au temps t. La variance de Y_t et son espérance (μ_t) sont reliées par la relation $Var(Y_t) = \phi\, v(\mu_t)$ ou ϕ est le paramètre de dispersion ; $v(\mu_t)$ est la fonction de variance.

Dans le cas particulier d'un processus, les variables Y_t successives ne sont pas indépendantes, en général et la vraisemblance se présente sur la forme suivante :

$L(y_1, y_2, \ldots, y_n) = P[(Y_1, Y_2, \ldots, Y_n) = (y_1, y_2, \ldots, y_n) \mid \mathbf{X}] = P[Y_1 = y_1 \mid \mathbf{x_1}] \,.\, P[Y_2 = y_2 \mid y_1 , \mathbf{x_1, x_2}] \,.\, . P[Y_3 = y_3 \mid y_1 , y_2 , \mathbf{x_1, x_2, x_3}] \,.\, \ldots \,.\, . P[Y_n = y_n \mid y_1 , y_2 , \ldots y_{n-1} , \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \ldots, \mathbf{x_n}]$ [25].

Les applications du GLM

Dans tout ce qui a précédé, il n'a pas été question de la nature de la variable Y. Ce qu'on sait d'elle, c'est que sa distribution de probabilité appartient à la famille exponentielle. Comme il a été mentionné plus haut, le GLM est un outil de modélisation « quasi-universel ». Ceci veut dire que Y peut revêtir de nombreuses formes. Un compte, bien sûr – c'est l'objet du manuel – mais aussi une variable réelle représentant une mesure, ou une fonction de risque propre à l'analyse de survie, etc.

Dans ce dernier cas, si l'on considère les modèles à risques proportionnels la fonction de risque instantanée s'écrit, par exemple :

$$\lambda(t,Z) = \lambda_0(t)e^{\beta' Z}$$

Où, λ est la fonction de risque instantanée, t le temps, Z l'individu, z les caractéristiques de l'individu Z, β le vecteur des paramètres [27].

Ceci peut s'écrire, pour l'individu Z_i présentant les caractéristiques $z_{i1}, z_{i2}, \ldots, z_{ip}$:

$$\lambda(t,Z_i) = \lambda_0(t)e^{\sum_{j=1}^{p} \beta_1 z_{i1} + \beta_2 z_{i2} + \ldots + \beta_p z_{ip}}$$

La régression logistique utilise aussi le GLM :

$$\ln\left(\frac{\pi_i}{1-\pi_i}\right) = \sum_{j=1}^{p} x_{ij}\beta_j \qquad i = 1,2,\ldots,n$$

Avec π_i représentant la probabilité qu'une variable dichotomique (prenant la valeur 0 ou 1) soit égale à 1.

[27] Rappelons que la notation β' désigne le transposé du vecteur β.

3.2.2. Modèle additif généralisé (GAM)

Le modèle additif généralisé est une extension du GLM [30]. En fait, une grande partie de la définition du GLM vaut aussi pour lui. Ce qui les différencie c'est le prédicteur. Il est linéaire dans le GLM, il est dit additif dans le GAM, c'est-à-dire qu'il n'est pas obligé d'être linéaire… De plus il est composé d'une somme de fonctions qui ne sont pas forcément paramétriques. Le GAM suppose que la fonction non paramétrique de n variables X_1, X_2, …, X_n dont l'estimation et la représentation sont complexes puisse être approchée par une combinaison linéaire de fonctions non paramétriques : $S(x_1, \ldots, x_n) \sim f_1(x_1) + \ldots + f_n(x_n)$.

Définition du GAM

Comme pour le GLM, le GAM dispose d'une version simplifiée, le modèle additif (qui est donc au GAM, ce que le modèle linéaire général est au GLM). Il se présente ainsi :

$$E[Y] = \mu = \alpha + \sum_{j=1}^{p} f_j(X_j)$$

Comme pour le modèle linéaire général, les erreurs ε_i (telles que $\varepsilon_i = Y_i - \mu_i$, i = 1, 2, …, n) sont des variables aléatoires indépendantes des X_j et présentant les propriétés suivantes :

$$E(\varepsilon_i) = 0 \text{ et } var(\varepsilon_i) = \sigma_i^2, \forall\, i \in [1,n].$$

Les fonctions f_j sont des fonctions quelconques d'une ou plusieurs variables. Elles peuvent être paramétriques (polynomiales, trigonométriques, *splines* de régression, etc.) ou non paramétriques comme les fonctions *splines* non paramétriques (ou *splines* de lissage) ou les fonctions de régression locale pondérée (*locally-weighted running-line smoother* [31] ou fonctions *loess* dans le logiciel S-PLUS [32]).

Le GAM, quant à lui, se présente sous la forme :

$$Y_i \sim L_{exp} \qquad \text{et} \qquad \mu_i = E[Y_i]$$
$$\eta_i = g(\mu_i)$$
$$\eta_i = \alpha + \sum_{j=1}^{p} f_j(x_{ij})$$

Les autres notations sont les mêmes que pour le GLM.

Comme précédemment, il est possible d'écrire le modèle sous forme vectorielle et matricielle :

$$\mathbf{Y} \sim L_{exp} \qquad \text{et} \qquad \boldsymbol{\mu} = E[\mathbf{Y}]$$
$$\boldsymbol{\eta} = g(\boldsymbol{\mu})$$
$$\boldsymbol{\eta} = F(\mathbf{X})$$

Là aussi, pour la signification des notations, on se reportera à ce qui a été dit précédemment.

En ce qui concerne les hypothèses, comme pour le GLM, le GAM repose sur les deux conditions vues plus haut : 1°) la fonction de distribution de **Y** appartient à la famille exponentielle, 2°) la fonction g (fonction de lien) est monotone et différentiable.

Nous avons vu que la différence entre les deux modèles réside dans les fonctions composant le prédicteur η. Ce sont ces fonctions qu'il faut voir à présent.

Fonctions

Nous avons vu que le prédicteur pouvait contenir des fonctions trigonométriques, des fonctions polynomiales mais aussi des fonctions qui n'entrent pas dans ces catégories. Il en existe de plusieurs sortes mais celles qui sont les plus utilisées dans les modélisations de séries temporelles sont les fonctions *splines* et les fonctions *loess*. Il n'est pas question, là encore, d'entrer dans les détails. On pourra, pour cela, se référer à des ouvrages spécialisés [30] qui d'ailleurs traitent d'autres fonctions de ce type. Ces fonctions sont des outils très utiles car elles permettent de réaliser des ajustements « souples » aux données et d'exercer un lissage de celles-ci.

Avant de voir plus précisément les fonctions intervenant dans les modèles additifs, donnons en une expression formelle et (donc) générale. Pour cela, il faut supposer qu'il existe une relation réelle mais non connue entre la variable expliquée et les variables explicatives. On peut écrire :

$$Y = f(X) + \varepsilon$$

Avec ε indépendantes, $E(\varepsilon) = 0$, $var(\varepsilon) = \sigma^2$ (ce qui veut dire que la variance est constante).

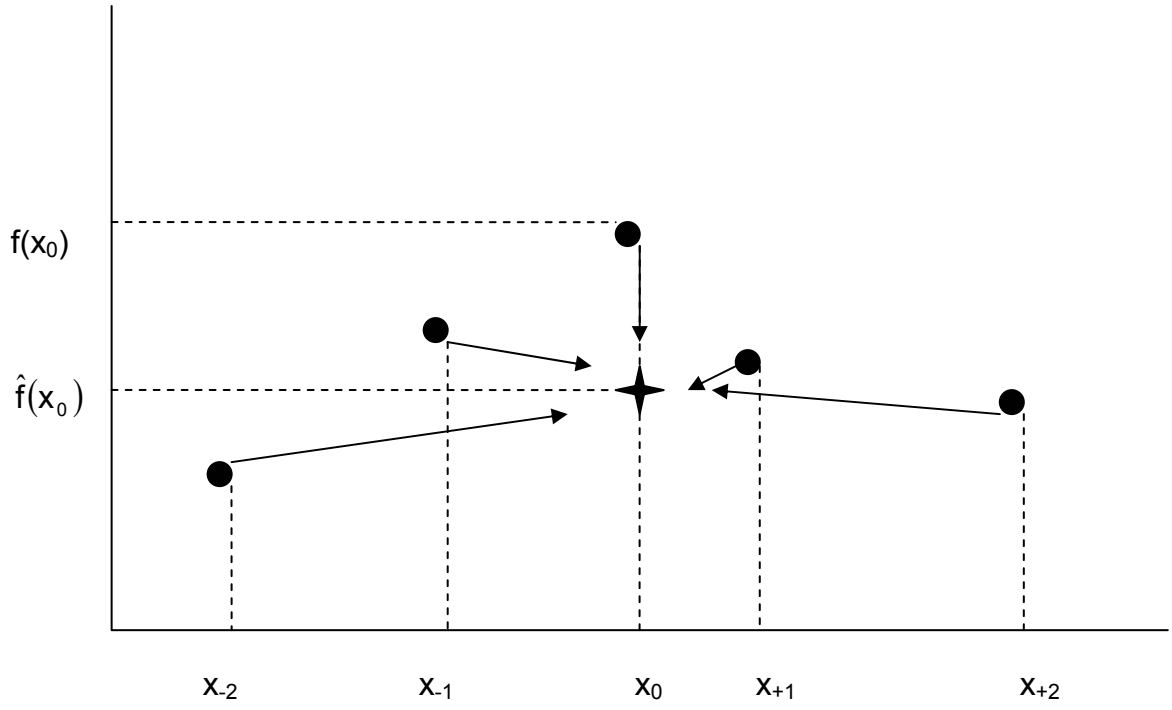
Nous savons que la partie mathématique (explicative) de la modélisation représente l'espérance de Y, conditionnellement aux variables explicatives. Donc :

$$E(Y \mid X = x) = f(x)$$

Ce qui veut dire que quand X prend la (les) valeurs x, alors l'espérance conditionnelle de Y vaut f(x).

Le lissage « fabrique », pour une valeur de x donnée, x_0 par exemple, des valeurs $\hat{f}(x_0)$, proches des vraies valeurs $f(x_0)$, en général, en moyennant les valeurs de y proches de $f(x_0)$. La figure 41 montre l'approximation de la vraie valeur $f(x_0)$ par un lissage (représenté par une étoile à quatre branches) obtenu par un procédé déduit des valeurs des « points » du voisinage, $f(x_{-2})$, $f(x_{-1})$, $f(x_0)$, $f(x_{+1})$ et $f(x_{+2})$.

Figure 41. Lissage



Les fonctions splines

Ces fonctions réalisent, en gros, une régression polynomiale (cubique ou de degré 4, le plus souvent mais rien ne s'oppose *a priori* à ce que le degré soit 2 ou 11….) par morceaux (dans des intervalles) en imposant des conditions de continuité, de pente et de courbure aux frontières des intervalles.

Ces fonctions sont diverses et se répartissent en deux groupes : les *splines* paramétriques et les *splines* non paramétriques, dites encore *splines* « de lissage » [(28)]. « Paramétrique » veut dire que la fonction a une expression algébrique (dépendant d'un ensemble de paramètres). « Non paramétrique » veut dire qu'une partie de la fonction est définie sans paramètres accessibles au calcul analytique ; la fonction est construite, alors, sur la base d'un ou plusieurs algorithmes.

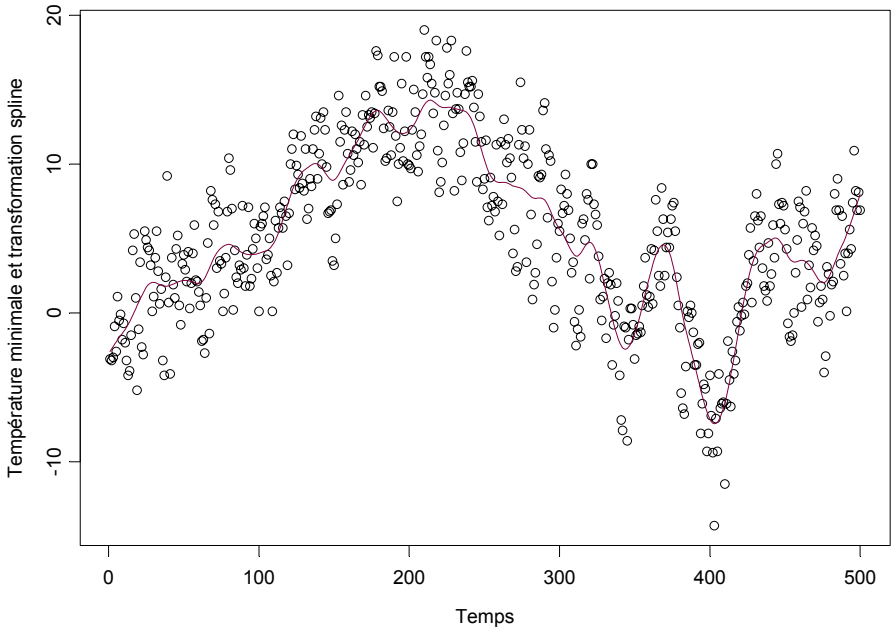
Nous donnons ci-dessous quelques exemples de *splines* paramétriques (*splines* de régression, *natural splines*, splines pénalisées) et non paramétriques (*splines* de lissage) [30].

Splines de régression

Supposons que nous ayons un ensemble de points (x_i , y_i). Le domaine sur lequel cet ensemble de points est étudié est [x_{min} , x_{max}].

Pour appliquer une transformation par *spline* de regression (*regression spline*, en anglais), il faut d'abord découper [x_{min} , x_{max}] en un certain nombre d'intervalles séparés par des points dit nœuds (*knots*, *breakpoints*) ξ_1, ξ_2, …, ξ_K. La fonction *spline* réalise une régression polynomiale (généralement cubique) dans chaque intervalle [ξ_j, ξ_{j+1}], c'est-à-dire ajuste un polynôme (de degré 3 en général, c'est-à-dire du type $a_3x^3 + a_2x^2 + a_1x + a_0$) aux points contenus dans l'intervalle (raison pour laquelle ces fonctions sont aussi appelées *splines* polynomiales) (figure 42).Les régressions ont une forme différente selon l'intervalle, c'est-à-dire que les paramètres sont différents d'un intervalle à l'autre.

Figure 42. Fonction *spline* de la température minimale à Strasbourg



[28] En anglais, *smoothing splines*.

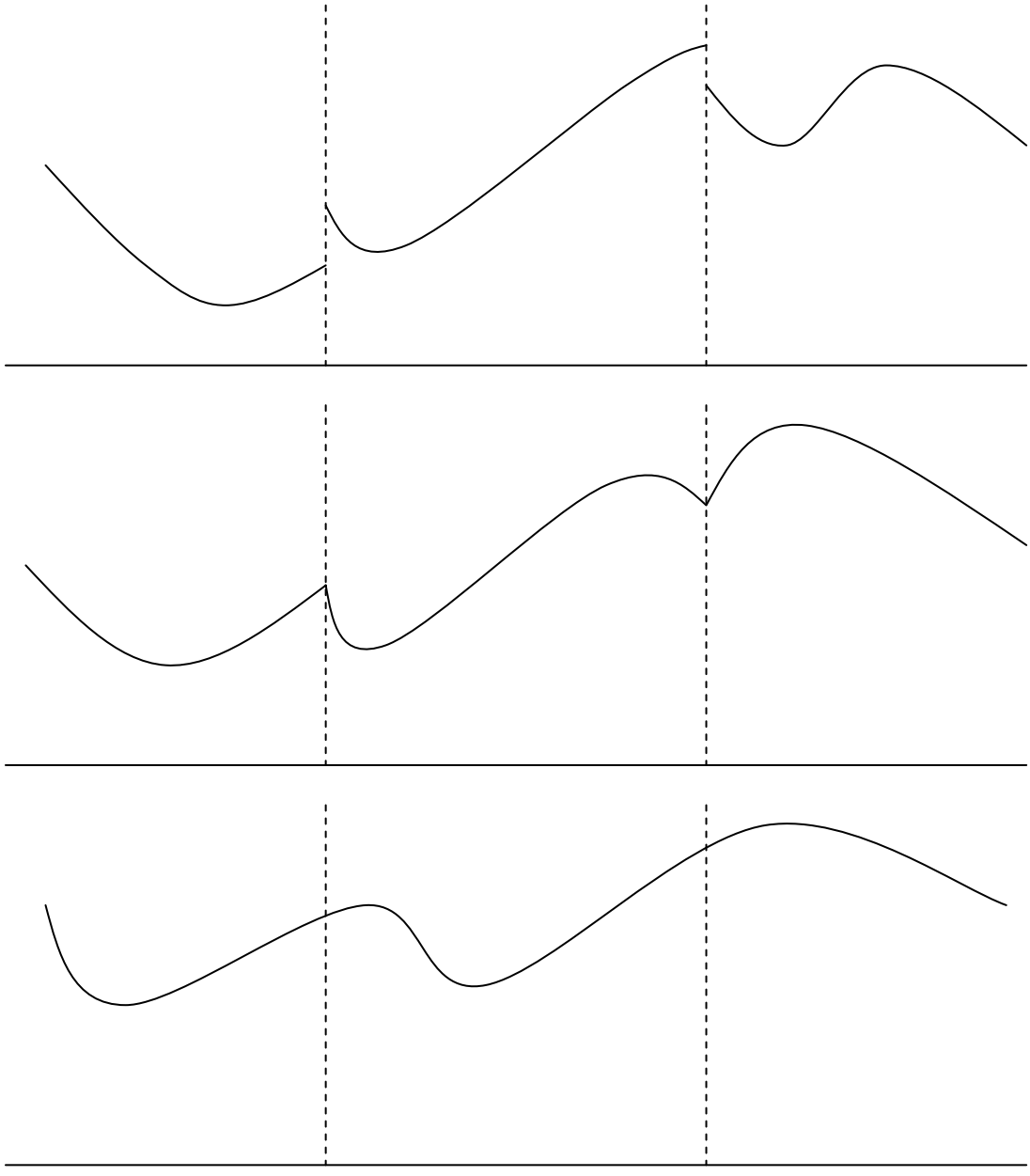
Si l'on s'en tient là, les courbes représentatives ne seront pas en continuité, en général, au niveau des nœuds. Il faut donc imposer à la régression *spline* de « jointer » les extrémités des courbes de régression des différents intervalles, c'est-à-dire imposer à la fonction *spline* d'être continue au niveau des nœuds. Mais cela ne suffit pas : la courbe a beau être continue – si on l'impose, bien sûr – la jonction n'a aucune raison d'être harmonieuse (la jonction se fait avec un angle). Pour assurer cette harmonie, il faut imposer que la pente et la courbure de la courbe varient régulièrement, sans accroc au niveau des nœuds. Ceci est obtenu en forçant les dérivées première et seconde à être continues au niveau de nœuds (figure 43).

La fonction *spline* se présente sous la forme suivante [30] :

$$s(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{j=1}^{K} \theta_j \left(x - \xi_j\right)_+^3$$

Le dernier terme (le + qui se trouve en indice représente la partie positive de l'expression) permet de satisfaire aux diverses conditions de continuité.

Figure 43. Différentes fonctions *splines* : fonction discontinue au niveau des nœuds, fonction continue, fonction de dérivées première et seconde continues



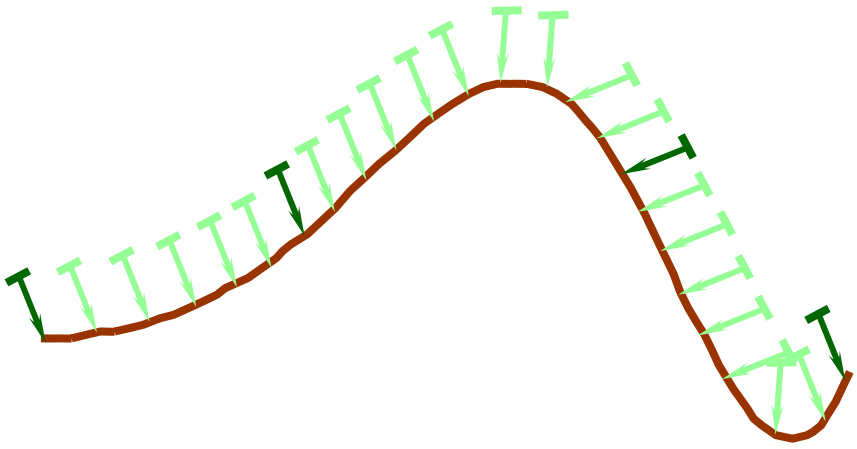
Natural splines

Les « *splines* naturelles » sont une variante proche des *splines* polynomiales vues ci-dessus. Elles supportent une contrainte supplémentaire : il faut qu'au delà des nœuds extrêmes, la fonction soit linéaire, ce qui veut dire qu'au voisinage de ces nœuds, la dérivée seconde et la dérivée d'ordre 3 doivent être nulles (la dérivée seconde, rappelons le, donne la valeur de la courbure ; si elle est égale à zéro, la courbe est… « droite »). Ces *splines* bénéficient de l'ajout de deux nœuds supplémentaires (aux extrémités), ce qui laisse moins de liberté à la fonction mais stabilise la variance des estimations aux extrémités, variance qui est élevée dans le cas des *splines* polynomiales classiques.

Splines pénalisées

Les *splines* pénalisées ou *penalized splines* sont une réponse à un défaut du logiciel S-PLUS qui, pour la construction des *natural splines* et des *B-splines*, localise les nœuds au niveau des quantiles de la distribution des observations. Or, ce type de *splines* est très sensible à la position des nœuds en question. La solution est de disposer de très nombreux nœuds puis de supprimer ceux qui sont inutiles. Autrement dit, il s'agit de positionner beaucoup **plus de nœuds que nécessaire** et de **ne conserver que les nœuds apportant suffisamment d'information** en regard de la pénalité (figure 44). Cette dernière opération est réalisée grâce à un paramètre de pénalisation lors de l'ajustement et la commande est disponible dans le logiciel jumeau, R.

Figure 44. *Spline* pénalisée : positionnement des nœuds et conservation des nœuds « utiles » (en vert foncé)



Splines de lissage

Les *splines* de lissage (*cubic smoothing splines*) sont des fonctions qui minimisent la somme des carrés des résidus (comme il est classique) mais cette somme est « pénalisée » par un terme représentant la courbure de la fonction :

$$\sum_{i=1}^{n}\left[y_i - f(x_i)\right]^2 + \lambda\int_a^b \left[f''(t)\right]^2 dt$$

f est la fonction *spline* recherchée, λ est un paramètre, a et b sont tels que :

$$a \leq x_1 \leq x_2 \leq \ldots \leq x_n \leq b$$

Le paramètre λ joue le rôle d'un facteur de lissage. Plus sa valeur est élevée, plus le lissage est important.

Les autres fonctions de lissage

Les fonctions de régression locale pondérée

En anglais *locally-weighted running-line smoother* [31], ces fonctions représentent actuellement, avec les *splines*, les plus utilisés des outils dédiés à la modélisation des séries temporelles, dans le domaine de l'environnement, en tout cas.

Les fonctions de régression locale pondérée (appelées encore fonctions *loess* dans le logiciel S-PLUS [32]) remplacent un point (x_0,y_0) par une régression linéaire sur les points (x_i,y_i) du voisinage de (x_0,y_0), affectés d'une pondération dépendant de l'éloignement $|x_i - x_0|$ par rapport au point (x_0,y_0). Les fonctions *loess* réalisent ainsi un lissage non paramétrique déterminé par l'étendue du voisinage de points participant aux régressions locales. Cette étendue est appelée *fenêtre de lissage* [31,33]. Le lissage est d'autant plus important que la fenêtre est large.

Fonctions diverses

D'autres fonctions de lissage (non paramétriques) répondent aux noms de *running-mean* (moyenne mobile), *running-line*, *kernel*, *bin smoother*, etc. Pour plus de précisions, il est possible de se référer au livre de Hastie et Tibshirani [30].

Le temps

Les variables du GAM peuvent être aussi, comme celles du GLM, indexées par le temps :

$$Y_t \sim L_{exp} \qquad \text{et} \qquad \mu_t = E[Y_t]$$

$$\eta_t = g(\mu_t)$$

$$\eta_t = \alpha + \sum_{j=1}^{p} f_j(x_{tj})$$

Nombre de degrés de liberté

Dans le cas du GLM, les choses sont claires : le nombre de degrés de liberté (ddl) est égal, en gros, à la différence entre le nombre d'observations et le nombre de paramètres à estimer.

Pour le modèle additif, on a affaire à des fonctions de lissage et le calcul du nombre de ddl est plus compliqué. Pour ce faire, il faut définir, d'abord, ce qu'est une fonction de lissage linéaire :

Une fonction de lissage S est linéaire si (formule classique car cette définition est valable quelle que soit la fonction) :

$$S(a\mathbf{y_1} + b\mathbf{y_2} \mid \mathbf{x}) = a\,S(\mathbf{y_1} \mid \mathbf{x}) + b\,S(\mathbf{y_2} \mid \mathbf{x})$$

La transformation de **y**, c'est-à-dire de (y_1, y_2, …, y_n) en ($\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$) par la fonction de lissage peut s'écrire :

$$\hat{\mathbf{f}} = \mathbf{S}\,\mathbf{y}$$

Avec $\hat{\mathbf{f}}$, la transformation par la fonction de lissage, **S**, une matrice carrée de type M_n, c'est-à-dire à n lignes et n colonnes (appelée matrice de lissage) et **y** = (y_1, y_2, …, y_n)'.

La moyenne mobile, les fonctions *loess*, les *splines* de lissage sont des fonctions linéaires.

Nous allons définir à quoi correspond le nombre de ddl. Mais, avant cela, nous devons rappeler quelques notions concernant les matrices.

Une matrice est un tableau rectangulaire fait de lignes et de colonnes identifiées, contenant des nombres. Elle sert d'outil à un ensemble d'opérations. L'une des fonctions d'une matrice, est de représenter les transformations. Un vecteur, par exemple, est transformé par une matrice le plus souvent en un autre vecteur différent du premier par sa norme (longueur) et sa direction. Ceci s'écrit comme le produit d'une matrice par un vecteur :

$$\mathbf{v_2} = \mathbf{M}\,\mathbf{v_1}$$

$\mathbf{v_1}$ est le vecteur initial, $\mathbf{v_2}$ est son transformé par la matrice **M**.

Considérons à présent un cas particulier : la matrice carrée. Il s'agit d'une matrice contenant autant de lignes que de colonnes. Il peut arriver, dans ce cas, qu'un vecteur soit transformé par celle-ci en un vecteur colinéaire (de même direction) et ceci s'écrit :

$$\mathbf{v_2} = \mathbf{M}\,\mathbf{v_1} = \alpha\,\mathbf{v_1}$$

α étant un scalaire [29].

Le vecteur $\mathbf{v_1}$ et le scalaire α, définis comme tels, s'appellent, respectivement, vecteur propre et valeur propre de la matrice. En général ils ne sont pas uniques et il existe un certain nombre de vecteurs propres et de valeurs propres associées, pour chaque matrice. Ces vecteurs et valeurs propres ont un ensemble de propriétés mais l'une d'entre elles permet de diagonaliser la matrice, c'est-à-dire de la transformer en une matrice diagonale. Cette dernière est une matrice dont tous les éléments sont nuls sauf la première diagonale (celle qui part du haut à gauche pour arriver en bas à droite). Cette première diagonale est composée des valeurs propres (figure 45).

Figure 45. Diagonalisation d'une matrice et valeurs propres.

$$\begin{pmatrix} a_{11} & & & a_{1n} \\ & a_{22} & & \\ & & \ldots & \\ a_{n1} & & & a_{nn} \end{pmatrix} \xrightarrow{\text{diagonalisation}} \begin{pmatrix} \alpha_1 & 0 & & 0 \\ 0 & \alpha_2 & \ldots & \\ & \ldots & \ldots & \ldots \\ 0 & & 0 & \alpha_n \end{pmatrix}$$

Dernière notion avant que nous abordions la définition du ddl : la trace d'une matrice, notée tr(M). La trace est la somme des valeurs propres c'est-à-dire des éléments de la première diagonale de la matrice diagonalisée. Cette trace est d'ailleurs aussi la somme des éléments de la première diagonale de la matrice de départ.

[29] Nombre.

On a, pour l'exemple de la matrice M, ci-dessus :

$$\text{tr(M)} = \sum_{i=1}^{n} a_{ii} = \sum_{j=1}^{n} \alpha_j$$

Pour une fonction de lissage linéaire, le nombre de ddl est défini de différentes façons et sa valeur diffère légèrement, selon le cas. Dans toutes les définitions, cependant, intervient la trace de la matrice de lissage $\mathbf{S}_\lambda$ [30]. Ainsi, selon les auteurs [30], le nombre de ddl est égal à :

- $\text{tr}(\mathbf{S}_\lambda)$, elle-même ;
- $n - \text{tr}(2\mathbf{S}_\lambda - \mathbf{S}_\lambda \mathbf{S'}_\lambda)$ [31] ;
- $n - \text{tr}(\mathbf{S}_\lambda \mathbf{S'}_\lambda)$.

Ces expressions ont été élaborées pour se rapprocher le plus possible de la définition du ddl dans le cas du modèle de régression linéaire (dont le GLM).

3.2.3. Ajustement du modèle

3.2.3.1. Méthodes d'ajustement

3.2.3.1.1. Principes

De façon générale, deux approches peuvent être utilisées pour estimer des paramètres. La méthode du maximum de vraisemblance et la méthode des moindres carrés. Ces deux approches ne donnent pas exactement les mêmes résultats à une exception près : quand la loi de probabilité est normale.

Méthode du maximum de vraisemblance

Nous avons vu, précédemment que la vraisemblance, comme fonction du (ou) des paramètre(s), est la probabilité d'observer l'échantillon… observé (*i.e.* celui qui constitue les données). Lorsqu'on fait varier le (ou les) paramètre(s), la loi de probabilité change et, du coup, la probabilité d'observer un échantillon identique à celui qui a été observé. Par conséquent, pour l'échantillon considéré et pour le type de loi de probabilité que l'on pense être adapté, il doit exister une valeur de paramètre pour laquelle la probabilité d'observer l'échantillon est maximale. Cette valeur est donc celle qui convient le mieux au jeu de données.

Par conséquent, le paramètre qui maximise la vraisemblance est un estimateur du paramètre de la loi de probabilité. Qui dit maximiser la vraisemblance dit maximiser la log-vraisemblance (puisque la fonction logarithme népérien est continue, monotone croissante). Le paramètre recherché est donc celui qui maximise la log-vraisemblance. Or nous savons que les extremums (maximum ou minimum) d'une fonction sont obtenus pour les valeurs de la variable qui rendent la dérivée première nulle. Dans le cas présent, il y a plusieurs paramètres donc il faudra considérer la dérivée première partielle.

Ainsi, si $\hat{\theta}$ est la valeur de θ telle que : $\left.\frac{\partial l(\theta,\varphi|y)}{\partial\theta}\right|_{\hat{\theta}} = 0$ [32], alors $\hat{\theta}$ est un estimateur du maximum de vraisemblance. Cette dérivée partielle est encore appelée fonction *score*.

---

[30] On écrit $\mathbf{S}_\lambda$ pour rappeler que le lissage dépend d'un paramètre qu'on nomme λ.

[31] Rappelons que la notation « ' » veut dire transposé.

[32] Cette expression représente la valeur prise par la dérivée première partielle de $l(\theta, \phi \mid y)$ pour la valeur $\hat{\theta}$ du paramètre θ.

À ce stade, on verra émerger deux problèmes :

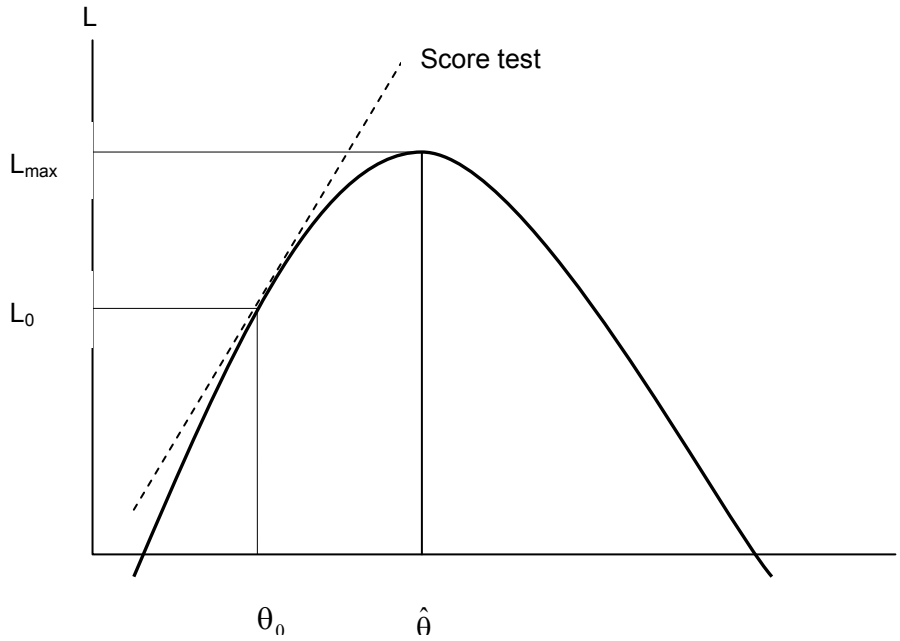
1) Rendre la dérivée nulle veut dire trouver un extremum. Mais pourquoi sagit-il forcément d'un maximum et pas d'un minimum ?

La réponse est que la courbe représentant la log-vraisemblance est strictement concave (vers le bas) [34] ce qui veut dire présentant une concavité orientée vers le « bas du plan » et donc l'extrémum est un maximum (figure 46).

2) Y-a-t-il plusieurs extrémums et si oui que faire ?

Là aussi, la réponse est contenue dans cette notion de concavité stricte : avant l'extremum la courbe croît strictement, après elle décroît strictement aussi. Il n'y a donc qu'un extremum.

Figure 46. Maximum de vraisemblance



Méthode des moindres carrés (pondérés)

Cette méthode est bien connue. Et en particulier, la méthode des moindres carrés sans pondération, dite des moindres carrés ordinaires (MCO).

La méthode des MCO consiste à minimiser, pour la variable expliquée, la somme des carrés des écarts entre la valeur prédite par le modèle et la valeur observée, ce qui revient à minimiser les résidus du modèle, soit :

$$S_2 = \sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2$$

Puisque la variable expliquée est une combinaison de fonctions d'autres variables, la somme S_2 est une fonction, elle aussi, des variables explicatives, bien sûr mais aussi, et surtout, des paramètres. Il faut là aussi minimiser S_2 et donc calculer les valeurs des paramètres qui rendent les dérivées partielles premières nulles. Ce qui, d'un point de vue géométrique et, lorsqu'on a affaire à une variable expliquée par une et une seule variable explicative, correspond à une droite, dite droite des moindres carrés.

Remarque. Quand la variable expliquée est munie d'une loi de distribution de probabilité normale, les méthodes du maximum de (log-)vraisemblance et la MMC donnent les mêmes estimateurs [33] mais pas exactement la même variance [34].

Disons encore un mot de la méthode des moindres carrés pondérés qui est utilisée dans certains cas.

Cette notion ne présente en fait – au niveau de sa définition en tout cas – aucune difficulté.

Cette méthode consiste à multiplier (pondérer) chaque carré des écarts par un poids, w_i.

La formule précédente devient :

$$S_2 = \sum_{i=1}^{n} w_i \left(y_i - \hat{y}_i\right)^2$$

L'ensemble des w_i peut prendre différentes formes selon le cas. Lorsque l'ensemble des observations est partitionné en un ensemble de classes numérotées 1, 2, …j, …, par exemple, w_i peut être pris égal au nombre d'observations dans chaque classe. Ce qui revient à donner plus d'importance là où il y a plus d'observations. On peut aussi pondérer le carré indicé par i par l'inverse de la variance de y_i, ce qui revient à donner plus d'importance aux observations qui ont une variance faible (donc peu dispersées).

3.2.3.1.2. Mise en œuvre

La recherche de l'estimateur par maximisation de la log-vraisemblance, on l'a vu, consiste à trouver la valeur du paramètre qui annule la dérivée de cette log-vraisemblance. Ceci a l'air d'être simple (!) au vu de l'écriture formelle mais, justement, le calcul formel est difficile en raison de la forme complexe de la fonction densité de probabilité. On a, alors, recours à des algorithmes (des méthodes itératives).

Méthodes propres aux GLM

Les techniques d'ajustement du GLM se résument ici à une méthode : la **méthode des scores de Fisher**.

La méthode des scores de Fisher est une variante de l'algorithme de Newton-Raphson lequel repose sur un développement de Taylor.

La formule de Taylor exprime la valeur d'une fonction au voisinage d'un « point ». Elle peut se présenter sous plusieurs formes mais nous allons utiliser la formule de Taylor avec reste de Young.

Soit une fonction f qui à tout x (d'un intervalle de IR donné) associe la valeur f(x). On suppose que f est définie, continuement dérivable à l'ordre p (*i.e.* les dérivées première, seconde, …, d'ordre p existent et sont continues) [34].

Alors on a :

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \ldots \frac{f^{(p)}(x_0)}{p!}(x - x_0)^p + (x - x_0)^p \varepsilon(x - x_0)$$

Avec $\varepsilon(x - x_0)$ très petit ou, mieux dit, avec $\lim_{x \to x_0} \varepsilon(x - x_0) = 0$ [35].

---

[33] C'est pour des raisons de courbure de la surface de projection : si les variables sont normales, la surface est plane.

[34] On dit que la fonction est de classe C^p.

Cette formule montre qu'on peut linéariser une fonction au voisinage d'un point x_0, pour peu que la fonction ait des propiétés de dérivabilité suffisantes.

Si on applique cette formule à l'ordre 1, on a :

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + (x - x_0)\varepsilon(x - x_0)$$

Comme $(x - x_0)\varepsilon(x - x_0)$ est très petit [(36)], on le néglige. Et :

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

Donc si $\hat{x}$ est la racine de f, on a : $f(\hat{x}) = 0$ et si l'on remplace x par $\hat{x}$, l'équation ci-dessus devient :

$0 \approx f(x_0) + f'(x_0)(\hat{x} - x_0)$, d'où :

$$\hat{x} \approx x_0 - \frac{f(x_0)}{f'(x_0)}$$

Le **processus de Newton** utilise cette propriété pour trouver la racine de f, c'est-à-dire la (ou les) valeurs(s) de x qui sont telles que f(x) = 0. Cette méthode consiste à créer une suite $\{x_n\}$ définie par :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

On peut montrer que si le premier terme de l'itération (x_1) est choisi suffisamment proche de la racine, alors $f(x_n)$ tend vers 0 quand n tend vers l'infini donc que les x_n approchent la racine de la fonction quand n tend vers l'infini. On remplace ainsi un calcul formel difficile (trouver analytiquement, le 0 de la fonction [(37)]) par un calcul algorithmique.

Il s'agit d'une estimation puisque l'on sait qu'on n'atteindra la valeur recherchée qu'après une infinité d'itérations ! On considère, en fait, qu'on a obtenu la valeur $\hat{x}$ quand les x_n ne varient plus beaucoup, ce qui suppose qu'on choisisse à l'avance un différentiel $| x_{n+1} - x_n |$ « seuil », en dessous duquel on interrompt l'itération.

Nous n'allons dire que quelques mots de l'**algorithme de Newton-Raphson** puisque c'est une variante (la méthode des scores de Fischer) qui est utilisée.

Le but de la méthode de Newton-Raphson est la recherche d'un extremum (maximum ou minimum) d'une fonction donnée. Cette fonction peut être multidimensionnelle (fonction à plusieurs variables). Or nous savons que rechercher un extremum, revient à trouver la (les) valeur(s) d'une (des) variable(s) qui rend(ent) la dérivée première nulle, en d'autres termes les 0 de la dérivée. Si F est la fonction, il faut donc trouver $\hat{X}$, tel que :

$$F'(\hat{X}) = 0$$

On utilise des majuscules, ici, car X est un vecteur, F est une fonction multidimensionnelle.

On peut écrire la formule de Taylor, comme plus haut mais pour la fonction dérivée :

$$F'(X) \approx F'(X_0) + F''(X_0)(X - X_0)$$

---

[35] Notons qu'ε, ici, n'a rien à voir avec la variable aléatoire « erreur » de la modélisation mais représente une grandeur très petite ou, en tous cas, qui devient très petite.

[36] Voir note [(35)]

[37] La racine de f.

En fait, comme on a dit précédemment, F(X), F'(X) sont des vecteurs.

$F'(X_0)$, par exemple, est composé des dérivées partielles de F par rapport aux différentes variables (cette fonction est aussi appelé « gradient ») :

$$F'(X_0) = \begin{pmatrix} \frac{\partial F}{\partial x_1}(X_0) \\ \frac{\partial F}{\partial x_2}(X_0) \\ \dots \\ \frac{\partial F}{\partial x_p}(X_0) \end{pmatrix}$$

Quant à F''(X), c'est une matrice que nous ne représenterons pas ici.

Pour estimer $\hat{X}$, on fait comme précédemment : le processus de Newton généralisé est appliqué et le calcul de X_{n+1} en fonction de X_n est réalisé par un processus itératif basé sur la résolution d'un système d'équations linéaires à chaque itération. Les valeurs X_n tendent alors vers $\hat{X}$ quand n tend vers l'infini.

Nous abordons, à présent, la **méthode des scores de Fisher**. Celle-ci est appliquée aux paramètres β et la fonction étudiée est la log-vraisemblance. Dans la formule ci-dessus, il faut remplacer X par β et F par l. Rappelons que la méthode des scores est une variante de l'algorithme de Newton-Raphson : en effet, dans la formule ci-dessus, la dérivée seconde au dénominateur est remplacée par son espérance.

Donc la formule itérative s'écrit :

$$\hat{\beta}_{n+1} = \hat{\beta}_n - \frac{l'(\hat{\beta}_n)}{E[l''(\hat{\beta}_n)]}$$

Nous avons vu plus haut que la dérivée première est la fonction score notée $s(\beta)$. D'autre part, nous l'avons vu aussi, la dérivée seconde est une matrice, son espérance aussi. Tout ceci fait que la formule s'écrit, de façon plus rigoureuse :

$$\hat{\beta}_{n+1} = \hat{\beta}_n - \frac{s(\hat{\beta}_n)}{E\left[-\frac{\partial^2 l(\hat{\beta}_n | y)}{\partial\hat{\beta}\partial\hat{\beta}'}\right]}$$

Avec (rappel) l, la vraisemblance qui, comme on l'a vu précédemment, est de la forme:

$$l_i(\theta_i \mid \phi, y_i) = \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)$$

Méthodes propres aux GAM

Elles sont nombreuses et dépendent des contraintes de lissage imposées aux fonctions (de lissage) du modèle, fonctions habituelles du modèle additif et du modèle additif généralisé. On se reportera avec intérêt à l'ouvrage de Hastie et Tibshirani [30]. Citons-en deux ou trois : régression linéaire

multiple après remplacement des fonctions non-paramétriques par des fonctions paramétriques (logarithme, racine carrée, etc.), après transformation des variables en polynomes orthogonaux [38], après décomposition des fonctions en un ensemble de fonctions de base (les B-splines, par exemple) et, enfin, la plus usitée des méthodes : celle qui estime chaque fonction par une fonction de lissage propre.

L'algorithme qui permet de faire face à toutes les situations de régression dans le cas d'un modèle additif (simple ; on ne parle pas encore de GAM, ici) est le ***backfitting algorithm***.

Reprenons notre modèle additif en l'écrivant de façon simplifiée :

$$\mathbf{Y} = \boldsymbol{\alpha} + \sum_{j=1}^{p} f_j(\mathbf{X}_j) + \boldsymbol{\varepsilon}$$

Cette notation est un raccourci de celle de la définition (prédicteur, etc.) [39]. On suppose que $\boldsymbol{\varepsilon}$ est indépendant des $\mathbf{X_j}$, que $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ et que $var(\boldsymbol{\varepsilon}) = \boldsymbol{\sigma}^2$.

On déduit de la formule précédente, en isolant $f_k(\mathbf{X_k})$, que :

$$\mathbf{Y} - \boldsymbol{\alpha} - \sum_{j=1}^{p} f_{j\neq k}(\mathbf{X}_j) = f_k(\mathbf{X}_k) + \boldsymbol{\varepsilon}$$

D'où

$$E\left(\mathbf{Y} - \boldsymbol{\alpha} - \sum_{j=1}^{p} f_{j\neq k}(\mathbf{X}_j) \middle| \mathbf{X}_k \right) = f_k(\mathbf{X}_k)$$

D'où en intervertissant les indices j et k, pour plus de commodité et de cohérence pour la suite (ceci ne change rien aux résultats) :

$$E\left(\mathbf{Y} - \boldsymbol{\alpha} - \sum_{k=1}^{p} f_{k\neq j}(\mathbf{X}_k) \middle| \mathbf{X}_j \right) = f_j(\mathbf{X}_j)$$

Ceci signifie que la valeur de $f_j(X_j)$ ou, ce qui revient au même, les valeurs $\{ f_j(X_{1j}), f_j(X_{2j}), \ldots, f_j(X_{nj}) \}$ est (sont) exprimable(s) à partir des fonctions des autres variables.

Le principe de l'algorithme est qu'à chaque étape, la valeur prise par la fonction d'une variable donnée est calculée à partir des valeurs prises par les fonctions des autres variables à l'étape précédente.

L'algorithme « d'ajustement arrière » (ou « rétrograde ») repose sur les étapes suivantes :

1° On choisit des valeurs initiales pour les $f_j(\mathbf{x}_j)$: $\boldsymbol{\alpha} = \dfrac{\sum_{i=1}^{n} y_i}{n}$, en d'autres termes, on prend la moyenne des y_i pour valeur initiale de α ; pour les autres, on choisit p valeurs de départ, $f_1^0, f_2^0, \ldots, f_p^0$. Comme on n'a pas d'*a priori*, on peut choisir des valeurs déduites d'une régression de y sur les $\mathbf{x}_j$.

---

[38] Orthogonaux : non colinéaires.

[39] Les notations en gras représentent des vecteurs.

2° On « fabrique » f_1^1 tel que $f_1^1 = S_j\left(\mathbf{y} - \alpha - \sum_{k=1}^{p} f_{k\neq j}^0 \middle| \mathbf{x}_j \right)$; ceci veut dire que l'on calcule f_1 au stade 1 à partir des f_k au stade 0 à l'aide d'une fonction de lissage prédéterminée.

3° On fait la même chose avec $f_2^1, f_3^1, ..., f_p^1$.

4° On recommence pour $f_2^2, f_3^2, ..., f_p^2$, etc.

5°) On s'arrête quand les fonctions ne diffèrent plus trop d'une étape à l'autre. Il est donc nécessaire de définir un seuil.

Normalement si tout se passe bien, l'algorithme converge dans le cas des *splines* de lissage mais dans le cas des fonctions *loess*, ceci est moins sûr [30].

Le ***local scoring algorithm*** répond à la problématique de l'ajustement du GAM.

Cet algorithme reprend, en le généralisant, celui de la méthode des scores de Fischer. On retrouve l'expression de Taylor mais avec la notion de lissage (toutes les fonctions de lissage peuvent convenir).

Rappelons la définition du GAM :

$$Y_i \sim L_{exp} \qquad \text{et} \qquad \mu_i = E[Y_i]$$

$$\eta_i = g(\mu_i)$$

$$\eta_i = \alpha + \sum_{j=1}^{p} f_j(x_{ij})$$

Il faut estimer α et les f_j.

1) On démarre l'algorithme en donnant une valeur arbitraire à α et aux f_j :

$$\alpha^0 = g\left(\sum_{j=1}^{n} \frac{y_i}{n} \right) \quad \text{et} \quad f_1^0 = f_2^0 = ... f_p^0 = 0$$

Ce qui revient à prendre, pour α, la moyenne des y_i (valeurs de la variable expliquée, mesurées) et donner aux fonctions initiales f_j^0, la valeur 0.

D'où, l'on sait calculer $\eta_i^0 = \alpha^0 + \sum_{j=1}^{p} f_j^0(x_{ij})$ et $\mu_i^0 = g^{-1}(\eta_i^0)$

Les $f_j^0 = 0$ et on connaît g (fonction logarithme népérien, par exemple).

2) On construit une nouvelle variable dépendante que l'on pondère

Soit $z_i = \eta_i^0 + (y_i - \mu_i^0)\left(\frac{\partial \eta_i}{\partial \mu_i} \right)_0$, la nouvelle variable déduite de y_i ;

Soit w_i une pondération de z_i ; l'expression de cette pondération dépend, en gros, de la dérivée de g^{-1} et de l'inverse de la variance [30].

3) On ajuste un modèle additif pondéré sur les valeurs de la nouvelle variable dépendante z_i

L'ajustement se fait par le « *backfitting algorithm* ». On obtient ainsi une estimation des fonctions f_j^1, la valeur des η_i^1 et des μ_i^1 ;

On calcule le critère de convergence basé sur la somme relative des valeurs absolues des différences entre les f_j^1 et f_j^0

4) On répète 2) en remplaçant les valeurs du stade 0 par les valeurs du stade 1

5) On répète 3) etc.

L'algorithme opère jusqu'à ce que le critère de convergence soit suffisament proche de 0. Ceci suppose, là encore, le choix d'un seuil de convergence.

3.2.3.2. Mesure de la qualité de l'ajustement

Déviance

La définition de la déviance repose sur la log-vraisemblance.

Rappelons que la log-vraisemblance est

$$l_i(\theta_i \mid \phi, y_i) = \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)$$

On peut simplifier son écriture, en prenant pour paramètre μ (à la place de θ qui est la forme générique du paramètre avec $\mu(\theta) = \theta$, comme il a été dit plus haut), en négligeant le paramètre de dispersion ϕ et en remplaçant le symbole « | » (de probabilité conditionnelle) par « ; » pour être conforme à la notation :

$$l(\mu; y)$$

Alors, si $\hat{\mu}$ est le paramètre estimé par le modèle et μ_{max} est la valeur du paramètre correspondant au modèle saturé (ce modèle qui donne aux composantes y_i de la variable dépendante y, les valeurs mesurées des y_i, maximise $l(\mu ; y)$), la déviance s'écrit :

$$D(\mathbf{y}; \hat{\boldsymbol{\mu}}) = 2l(\boldsymbol{\mu}_{max}; \mathbf{y}) - 2l(\hat{\boldsymbol{\mu}}; \mathbf{y})$$

La sélection d'un modèle se fait sur la base d'une minimisation de la déviance.

Rien n'empêche de remplacer μ par η dans la formule de la déviance :

$$D(\mathbf{y}; \hat{\boldsymbol{\eta}}) = 2l(\boldsymbol{\eta}_{\mathbf{max}}; \mathbf{y}) - 2l(\hat{\boldsymbol{\eta}}; \mathbf{y})$$

Celle-ci permet de comparer deux GLM, le premier étant emboîté dans le second.

Alors, on montre que sous certaines conditions, la quantité :

$$D(\hat{\boldsymbol{\eta}}_2 ; \hat{\boldsymbol{\eta}}_1) = D(\mathbf{y}; \hat{\boldsymbol{\eta}}_1) - 2l(\mathbf{y}; \hat{\boldsymbol{\eta}}_2)$$

Suit une loi du χ^2 avec un nombre de ddl égal à la différence des dimensions des deux modèles [40].

Pour les GAM, la déviance est encore valable pour la comparaison mais l'approche théorique de la distribution est encore mal développée à ce jour. La distribution du χ^2 peut être utilisée en première approximation, cependant.

Critère d'Akaike

Le critère d'Akaike ou AIC (*Akaike-information criterion*) est égal à la déviance pénalisée par un terme dépendant du nombre de paramètres du modèle [35-37]. L'expression de l'AIC est :

$$\mathrm{AIC} = \frac{\mathrm{D}\left(\mathbf{y}; \hat{\boldsymbol{\mu}}\right)}{n} + \frac{2\mathrm{df}\,\phi}{n}$$

Nous reconnaissons un premier terme comportant la déviance divisée par le nombre d'observations. Le deuxième terme comporte au numérateur le produit du paramètre de dispersion par le nombre de degré de liberté df. Ce dernier est la trace de la matrice $\mathbf{R}$, laquelle est l'opérateur [41] qui permet d'ajuster un modèle additif avec pondération sur les données, autrement dit qui traite la variable pondérée $\mathbf{z}$ vue plus haut. On a :

$$\hat{\boldsymbol{\eta}} = \mathbf{R}\,\mathbf{z}$$

Cette formule est à rapprocher de celle vue plus haut, elle aussi :

$$\hat{\mathbf{f}} = \mathbf{S}\,\mathbf{y}$$

Le critère d'Akaike est utilisé pour choisir le meilleur modèle. Ce choix s'effectue sur la base de la minimisation du critère. **En toute rigueur, il est utilisé pour des modèles linéaires et emboîtés**.

[40] La déviance est une autre façon de nommer le test du rapport de vraisemblance puisque là où il y a rapport de deux grandeurs, il y a différence de leurs logarithmes.

[41] Nous avons vu plusieurs fois le terme « opérateur » avec des applications différentes. Ici, « opérateur » veut dire matrice.

4. Principe de la modélisation des séries temporelles

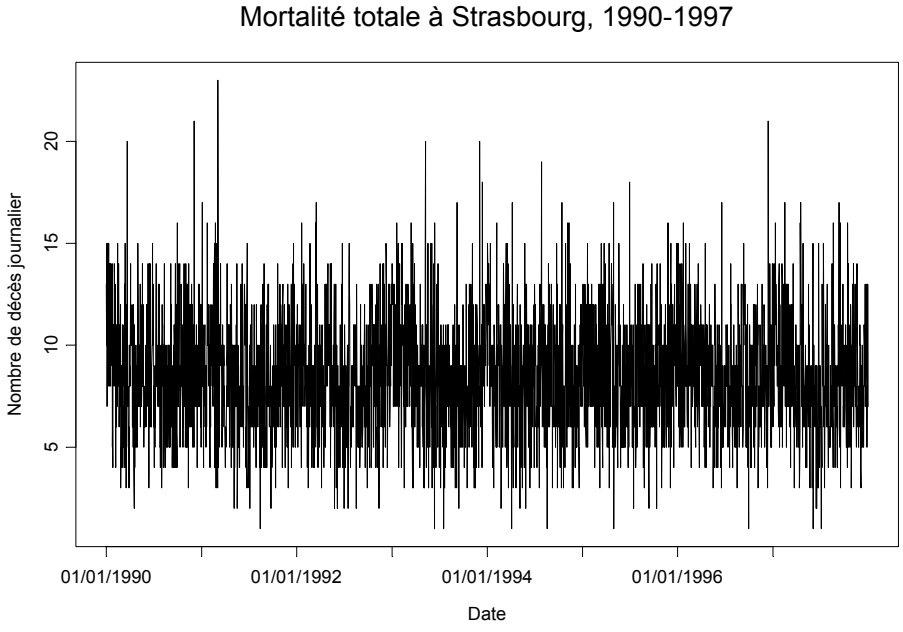
Nous traiterons ici, à titre d'exemple, la modélisation des liens entre la pollution atmosphérique et les indicateurs de santé. Mais les méthodes indiquées, comme il a été dit plus haut, s'appliquent tout autant à d'autres problématiques.

4.1. Problématique de la modélisation

Les indicateurs sanitaires sont généralement caractérisés par un faible nombre d'événements journaliers, des variations temporelles à long terme (tendance), à moyen et court termes (variations saisonnières, hebdomadaires, …) (figure 47), par une autocorrélation des comptes journaliers et une surdispersion.

L'autocorrélation signifie qu'il existe des liaisons temporelles (*ie.* des covariations) dans la série de comptes journaliers (un bruit blanc, *a contrario*, comme on l'a vu, est un exemple de phénomène non autocorrélé). L'autocorrélation, pour un décalage donné entre deux éléments de la série, est mesurée par le coefficient d'autocorrélation ou par le coefficient d'autocorrélation partielle (c.f. § 2.3.2. et § 2.3.3.). Ce dernier, rappelons le, mesure la liaison entre deux valeurs de la série, comme le premier, mais en éliminant l'effet des liaisons existant au sein des valeurs correspondant aux temps intermédiaires.

Figure 47. Exemple d'une série temporelle sanitaire

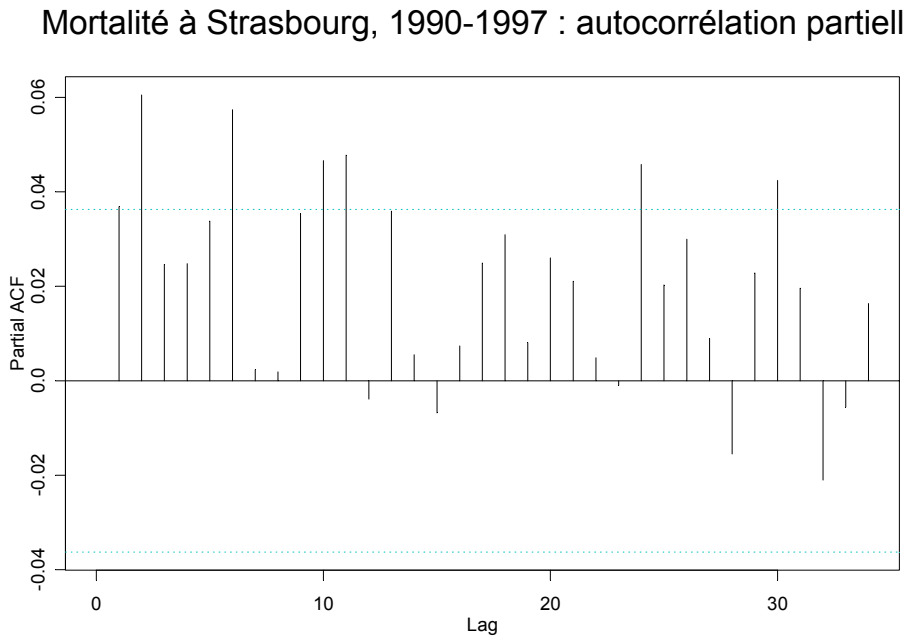


Le corrélogramme partiel représente le coefficient d'autocorrélation partielle en fonction du décalage (figure 48) (voir aussi, plus haut, § 2.3.3.). L'*autocorrélation* est repérable grâce aux pics du corrélogramme partiel, c'est-à-dire les valeurs du coefficient significativement différentes de 0 (*i.e.* valeurs en dehors de l'intervalle plus ou moins deux écart-types) (figure 48).

La *surdispersion* existe quand la variance des données est supérieure à la variance théorique, laquelle est égale à l'espérance si la variable *compte journalier* suit une loi de Poisson.

Ces deux dernières caractéristiques (autocorrélation et surdispersion) sont liées directement à des facteurs externes (météorologie, épidémies de grippe, pollens, par exemple) et à des variations saisonnières indirectement représentatives de variables non mesurées.

Figure 48. Corrélogramme partiel



Les *indicateurs de pollution* sont également soumis à des variations à court, moyen et long termes, dues essentiellement aux émissions et à des facteurs météorologiques. Par ailleurs, la température, l'humidité relative, les épidémies de grippe ou les pollens interviennent comme *facteurs de confusion* : leurs variations temporelles, saisonnières et à court terme elles aussi, sont liées à la fois à celles des polluants et à celles des données sanitaires [1,38-40].

L'étude de la relation entre les indicateurs sanitaires et les indicateurs de pollution doit donc tenir compte : 1°) de la tendance et des variations saisonnières des différentes variables, 2°) de ces facteurs de confusion, 3°) de l'autocorrélation des indicateurs sanitaires et environnementaux, 4°) de la surdispersion de la variable expliquée.

4.2. Les différents types de modélisation

Nous verrons en détail la modélisation (GAM) plus loin. Voyons, ici, de façon générale, comment se présente la modélisation d'une série temporelle.

Résumons ce que nous avons vu plus haut (§ 3.1.).

Nous disposons d'une série temporelle y_t, t de 1 à n. Cette série temporelle est une réalisation d'un processus aléatoire temporel (suite de variables aléatoires indicées par le temps) Y_t. Modéliser signifie :

1) Donner une loi de probabilité à Y_t (partie statistique du modèle) ;

2) Expliquer l'espérance de Y_t à l'aide d'autres variables (partie mathématique du modèle).

4.2.1. Les modèles courants

Modèles ARIMA

Rappelons que les modèles ARMA sont applicables à des séries stationnaires (§ 2.3.4., § 2.5.2.). Si le processus n'est pas stationnaire (*i.e.* possède une tendance), il est possible de réécrire l'expression du modèle ARMA mais en l'appliquant à une différence à l'ordre 1, ou à un ordre plus élevé.

Le modèle ARMA s'écrit :

$$Y_t + \varphi_1 Y_{t-1} + \ldots + \varphi_p Y_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

Avec ε_t est un bruit blanc $N(0,\sigma^2)$.

Le modèle ARIMA s'écrira :

$$\Delta^d Y_t + \varphi_1 \Delta^d Y_{t-1} + \ldots + \varphi_p \Delta^d Y_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

$\Delta^d Y_t$ représente la différence de Y_t à l'ordre d.

Modèles linéaire et additif généralisés

Le GLM s'écrit :

$$Y_t \sim L_{exp} \qquad \text{et} \qquad \mu_t = E[Y_t]$$
$$\eta_t = g(\mu_t)$$
$$\eta_t = \sum_{j=1}^{p} \beta_j x_{tj}$$

Le GAM s'écrit :

$$Y_t \sim L_{exp} \qquad \text{et} \qquad \mu_t = E[Y_t]$$
$$\eta_t = g(\mu_t)$$
$$\eta_t = \alpha + \sum_{j=1}^{p} f_j(x_{tj})$$

Pour ces deux modèles, respectivement, les variables x_{tj} et les fonctions $f_j(x_{tj})$ peuvent prendre différentes formes.

Pour modéliser la tendance, on se servira d'un terme du type « $a.t^b$ » pour le GLM (a et b sont des scalaires), *loess* (t , *largeur.fenêtre*) pour le GAM. Les fonctions *splines* sont utilisables dans les deux cas (paramétriques pour le GLM, (paramétriques et/ou non paramétriques pour le GAM).

Pour la saisonnalité, dans le cas du GLM, des termes trigonométriques comme $\sin(\omega t+\varphi)$ ou $\cos(\omega t+\varphi)$ pourront rendre service. Il est possible d'en faire figurer plusieurs. Pour le GAM, les fonctions *loess* et *splines* de la tendance s'adaptent aux variations saisonnières aussi.

Pour les variables explicatives (les facteurs), dans le cas du GLM, on peut utiliser des transformations diverses de la variables (linéaire, fonction logarithme, quadratique, racine carrée, etc.) ; le GAM permet d'introduire les variables avec des fonctions de lissage (*splines*, *loess*).

4.2.2. Justification du choix du type de modélisation

Le choix de l'un ou l'autre des modèles, repose sur la nature des données dont on dispose et sur ce que l'on veut faire. Si les données se réduisent à la série à étudier, on aura tendance à utiliser les modèles ARIMA. Si on dispose de données relatives à des variables explicatives, on se servira des modèles GLM ou GAM. Le choix entre ces deux derniers dépend de la nécessité éprouvée (ou non) d'un lissage de certaines séries.

On peut mettre à contribution les deux types de modèles, ARMA et explicatif. Ainsi, si l'on applique un modèle GLM ou GAM et qu'il persiste une autocorrélation dans les résidus, il peut être utile d'introduire un terme autorégressif, donc de type AR.

4.3. La démarche

Citons rapidement les grandes étapes car ceci sera vu en détail plus loin. La démarche adoptée actuellement, sachant qu'on dispose de variables explicatives est la suivante :

- Analyse descriptive de la série
- Modélisation de la tendance, des variations saisonnières (saison, mois, jour)
- Modélisation des autres facteurs (dans le cadre de la pollution atmosphérique)
- Recherche d'une relation paramétrique entre le polluant et l'indicateur de santé ou exploration des effets retardés
- Analyse de sensibilité
- Prise en compte de l'autocorrélation résiduelle

Notons que la représentation graphique de la série temporelle et son observation est une étape importante du processus de l'analyse. Elle devrait précéder tout calcul car elle permet de visualiser le « comportement » de la série et, par là, d'orienter la démarche exploratoire.

4.4. Qualités et défauts des différents modèles

- GLM : rigidité des fonctions mais l'apport des splines pénalisés permet de réduire fortement ce problème. Il existe plusieurs algorithmes de résolution robustes et mathématiquement bien définis.
- GAM : problèmes d'estimation liés à la concurvité (lissage non paramétrique voir § 5.3.3) et recours au lissage paramétrique ; sous-estimation de la variance des paramètres, en voie d'être résolue.
- ARIMA : le versant explicatif n'est pas ou peu exploré en épidémiologie. Les modèles ARIMAX (X pour les variables explicatives) sont utilisés depuis longtemps mais dans d'autres domaines.

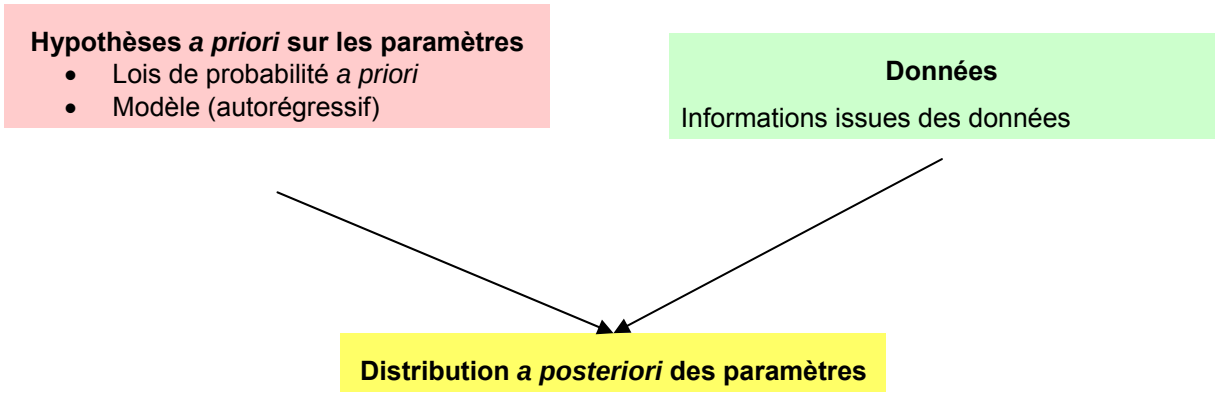
4.5. Approche bayésienne

Ici encore, nous ne ferons que frôler le sujet. Nous verrons les grands principes de l'approche bayésienne. Elle fait l'objet d'une littérature abondante mais on peut conseiller, deux-trois textes sur le principe [41-45]

4.5.1. Principes généraux de l'approche bayésienne

En statistique fréquentiste, on estime des paramètres sur la population à partir des données tirées des échantillons analysés. Partant des fameuses formules de Bayes, il est possible de montrer qu'on peut associer à l'information tirée de l'échantillon, une information provenant d'une autre source (littérature, avis d'experts, expérience antérieure, etc.) pour inférer l'estimation sur la population. Cette information additionnelle (valeur d'un paramètre, variance, etc.) est appelée information a *priori*. L'estimation finale des paramètres et de leur distribution s'appelle information *a posteriori* (figure 49).

Figure 49. Principe de l'inférence bayésienne



La vraisemblance est calculée comme d'habitude à partir de la densité de probabilité. Il faut alors choisir une loi *a priori* sur les paramètres. Le choix de ces lois ne se fait pas au hasard mais sur la base d'une simplification de la vraisemblance [42]. On les appelle des lois conjuguées de la loi de probabilité des observations. Il est possible (d'usage…), de plus, de choisir des lois dites peu informatives (à variance élevée) pour ne pas donner forcément trop d'importance aux *a priori* qu'on a sur la valeur des paramètres. En effet, l'avis d'expert peut être inexistant ou peu fiable, l'expérience peut manquer, etc.

Mais tout n'est pas aussi simple ! Le plus souvent, les lois *a priori* sur les paramètres ne sont pas conjuguées (pour des raisons diverses, dont le souci d'adéquation à la réalité : en effet, il n'est pas toujours pertinent de choisir une loi qui nous facilite les calculs au détriment de la vraisemblance du modèle [43]). On remarque alors vite que, dans ce cas, les calculs liés à l'estimation de la vraisemblance [44] deviennent très compliqués (les intégrales intervenant dans les formules sont difficiles à résoudre par le calcul formel) et on a recours à des algorithmes de calcul issus des méthodes de Monte Carlo par chaînes de Markov (MCMC), tels l'algorithme de Metropolis-Hastings et l'échantillonneur de Gibbs [44,46]. Depuis peu (milieu des années 90) ces calculs bénéficient de la puissance de nos ordinateurs actuels.

4.5.2. Application aux études de séries temporelles et exemples

Les modèles AR, MA et AR(I)MA, par exemple, ont été analysés avec cette approche.

Ainsi pour un modèle AR, par exemple [45], si les observations sont notées y_t, alors la variable Y_t est définie conditionnellement à y_{t-1}, y_{t-2}, …, y_{t-p} de la façon suivante (autorégression) :

$$Y_t \sim N\left(\mu - \sum_{i=1}^{p} \rho_i \left(y_{t-i} - \mu\right), \sigma^2\right)$$

Ici, les lois *a priori* sont choisies, normale pour μ, inverse gamma pour σ^2, normale pour les paramètres autorégressifs ρ_i.

Lorsque les données contredisent les hypothèses *a priori*, le modèle bayésien peut s'adapter en tenant compte de celles-là plus que de celles-ci. Il en est de même lorsque les données montrent un changement relatif aux hypothèses *a priori*. En cas de modification de la variance au cours du développement de la série temporelle, les modèles bayésiens s'adaptent plus souplement que

[42] C'est un peu ce qu'on fait aussi dans le cas du choix des fonctions de lien dans les modèles linéaires généralisés.

[43] Vraisemblance, au sens de conformité à la réalité.

[44] Vraisemblance, au sens statistique …

certains modèles fréquentistes tels les modèles ARCH qui, comme nous l'avons vu, permettent de tenir compte de cette variation mais tout en fixant celle-ci structurellement (§ 2.5.2.).

Il existe un exemple d'analyse de séries temporelles basées sur des données utilisées par les courtiers en assurance dont le taux de chômage [43]. Les auteurs utilisent un modèle permettant la variation de niveau de base ainsi que l'augmentation de la variance des erreurs :

$y_t = \mu_t + x_t$

y_t est la variable expliquée (nombre de cas incidents), μ_t est la composante niveau de base, x_t est la composante autorégressive.

Un ensemble de paramètres permet de faire varier les deux composantes :

$\mu_t = \mu_{t-1} + \delta_t * \beta_t$

δ_t est une indicatrice, elle prend les valeurs 0 ou 1 avec une probabilité de type loi de Bernouilli $B(\varepsilon_1)$, β_t mesure l'importance de la variation du niveau si elle a lieu.

$$x_t = \sum_{j=1}^{p} \phi_j x_{t-j} + e_t$$

Les ϕ_j sont les paramètes autorégressifs, e_t est l'erreur et suit une loi normale $N(0 , \sigma_t^2)$.

Enfin :

σ_t peut être constant ou être modifié ; le choix se fait sur la base d'une loi de Bernouilli.

Ainsi $\sigma_t = \sigma_{t-1}$ ou $\sigma_t = (\beta\upsilon)_t * \sigma_{t-1}$

$(\beta\upsilon)_t$ mesure l'importance de la variation de la variance.

À présent, l'approche bayésienne attribue à chacun des paramètres une loi de distribution *a priori*. Puis on calcule la distribution de chacun des paramètres conditionnellement aux observations et aux autres paramètres. Ces distributions permettent de générer des échantillons à partir des distributions *a posteriori*. Enfin, à partir de cet ensemble de distributions *a posteriori*, on calcule les valeurs prédites de la variable représentant les observations. L'adéquation du modèle et le choix du meilleur modèle sont testés grâce à un critère d'Akaike (AIC) ou un *Bayesian Information criterion* (BIC) modifiés. Le choix du modèle le meilleur étant fait, les auteurs peuvent considérer l'écart entre les données observées et les données prédites (avec l'intervalle de prédiction *a posteriori*) pour voir si certaines variations du niveau ou de la variance observées sont accidentelles ou dues simplement au hasard[45] (45).

[45] Ce qui est accidentel n'est pas dû forcément au hasard, c'est ce qui est inattendu, imprévu et indépendant de la volonté humaine.

5. Logiciel S-PLUS (et… logiciel R)

Ce chapitre est dédié principalement à la connaissance et à la manipulation de S-PLUS. Le logiciel « R » est la « version gratuite » de S-PLUS. Les commandes sont très proches. Donc ce qui vaut pour l'un vaut pour l'autre ou à peu près. Comme on le verra, « R » dispose, cependant, de fonctions supplémentaires telles les *splines* pénalisées. S-PLUS propose une interface graphique « plus conviviale » que son *jumeau*.

5.1. Introduction

S-PLUS en est à sa version 6.1. *Insightful Corp.* (figure 50) a pris la suite de *MathSoft Inc.* (figure 51). Le logiciel n'a pas beaucoup changé, en tout cas sur le fond. Quelques modifications de forme que remarqueront les anciens utilisateurs, comme l'Object Explorer (version revue de l' *Object Browser*) qui range tous les objets (données, modèles, etc.) dans un seul répertoire, etc.

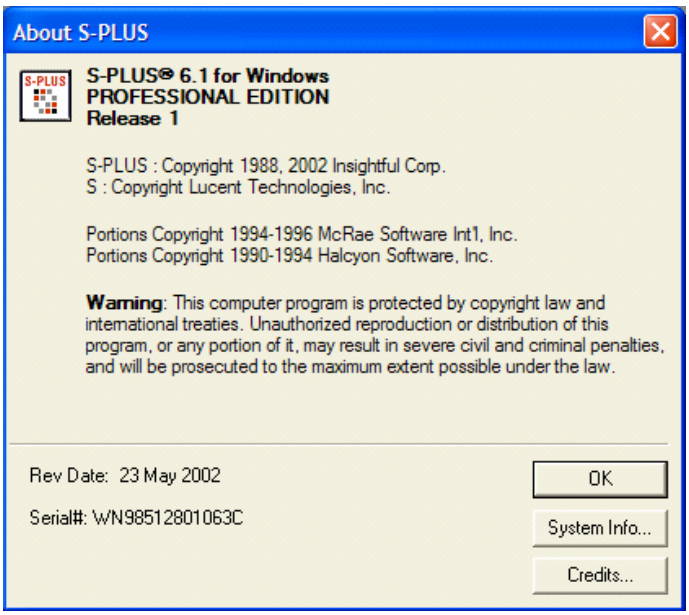
L'intérêt de S-PLUS réside en ce qu'il est adapté à la modélisation des GAM et qu'il est bien équipé sur le plan graphique. Ce qui n'est pas peu, vu l'utilité du recours à la représentation graphique, de façon générale et dans le cas de l'étude des séries temporelles, en particulier. Il a été fait état de problèmes de convergence et de problèmes de conception relatifs aux algorithmes d'ajustement. La correction de ces problèmes devrait être réalisée à plus ou moins court terme. « R » est muni de ce qu'il faut pour traiter la difficulté, donc le recours à ce logiciel, dont les commandes et la structure, rappelons le, sont proches de celles de S-PLUS, est toujours possible.

Dans ce qui va suivre, on le verra, sera réalisé un survol rapide des potentialités de S-PLUS (et, par conséquent, de R). Fonctions, commandes, manipulations de données, fonctions graphiques seront présentées dans l'optique d'une utilisation ciblée sur l'analyse des séries temporelles. Le lecteur qui désire connaître tel ou tel point précis échappant à l'objet de cette présentation pourra se référer à un ensemble d'ouvrages conseillés en fin de document.

De façon générale, il est possible de se référer systématiquement à l'**aide en ligne** dès le moindre doute. Cette aide est relativement bien faite et permet de s'extraire, le plus souvent, d'une situation difficile.

***Avertissement* : les exemples et les calculs ont été testés sur S-PLUS Version 6 (figure 50)** :

Figure 50. S-PLUS version 6.1



Il se peut qu'entre les diverses versions existent des différences (figure 51). Il en est bien sûr de même avec R (figure 52).

Figure 51. S-PLUS version 2000

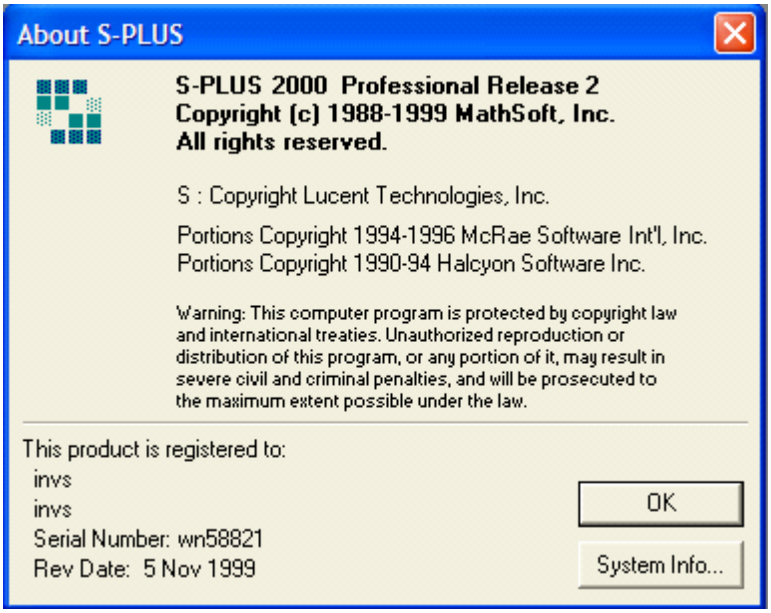
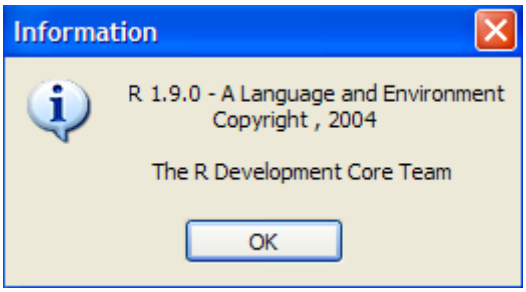


Figure 52. R version 1.9.0



5.2. Langage

5.2.1. Commandes de lancement

Pour lancer S-PLUS, il faut cliquer sur son icône placé, si tout se passe normalement, sur le bureau après installation du logiciel. Un certain nombre de questions sont posées : emplacement du répertoire de travail, etc.

Il est utile, cependant, de créer sur le bureau plusieurs raccourcis, à raison d'un icône par problématique (par sujet, en somme) : un pour la première étude mortalité - pollution atmosphérique, par exemple, un pour les hospitalisations, un pour l'analyse de tendance du cancer du poumon, un pour les calculs divers qu'on peut être amené à faire, etc.

Aussi avant de créer le raccourci, on crée le répertoire de travail ; par exemple : « `D:\reptrav\splus\9v2morta` ».

Le raccourci se présente de la façon suivante (figure 53) :

Figure 53. Raccourci de lancement S-PLUS

9villes2 morta.lnk

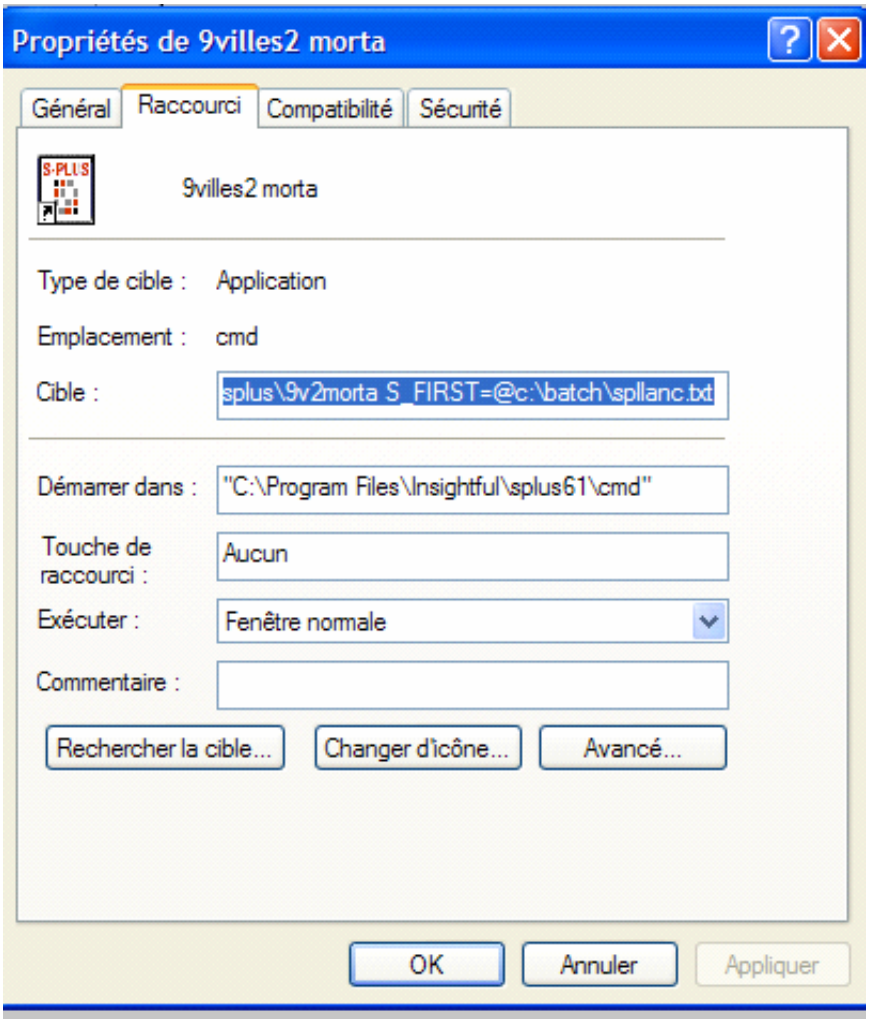
Si l'on ouvre le raccourci (en cliquant sur le bouton droit de la souris on fait apparaître le menu contextuel et on choisit « Propriétés »), on trouve ce qui est représenté sur la figure 54 :

Dans la ligne de commande « Cible », il est intéressant de remplacer la ligne par une commande du type :

« `"C:\Program Files\Insightful\splus61\cmd\SPLUS.exe" S_PROJ=D:\reptrav\splus\9v2morta S_FIRST=@c:\batch\spllanc.txt` »

La commande « "`C:\Program Files\Insightful\splus61\cmd\SPLUS.exe`" » lance le fichier de commande de S-PLUS.

Figure 54. Ouverture du raccourci de lancement S-PLUS



La commande « `S_PROJ=D:\reptrav\splus\9v2morta` » indique le répertoire de travail.

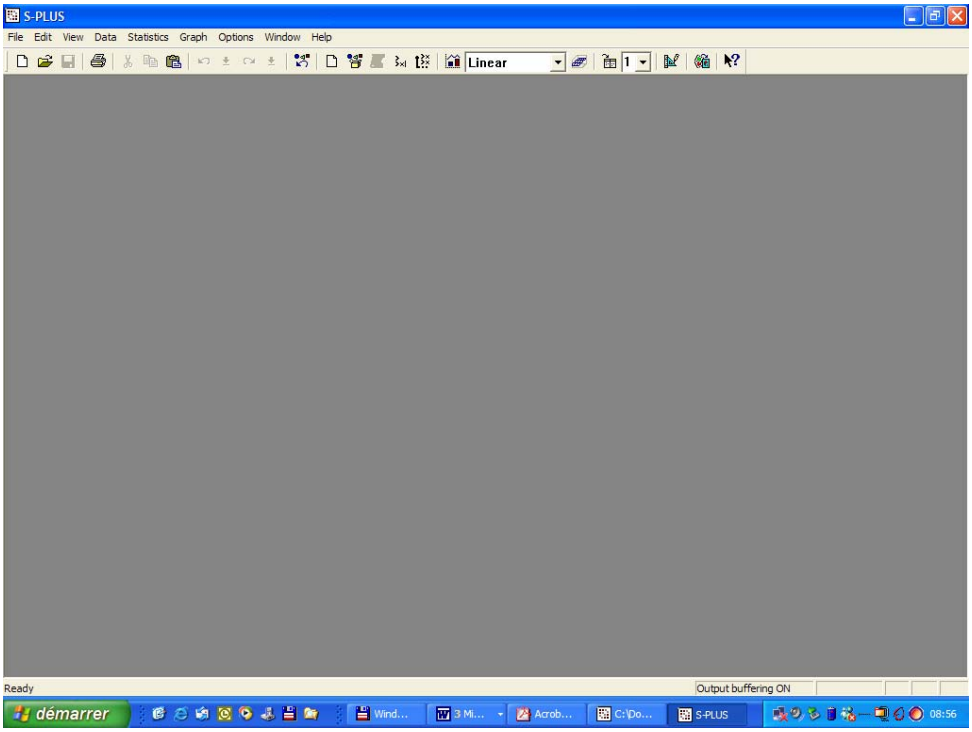
La commande « `S_FIRST=@c:\batch\spllanc.txt` » indique au programme de charger des instructions utiles avant de démarrer. Ces instructions sont écrites par l'utilisateur dans un fichier de commandes texte (ici « spllanc.txt » qu'il aura créé auparavant et qui permet de s'affranchir de l'écriture des conditions en question à chaque fois. Il conviendra d'écrire dans ce fichier les commandes suivantes :

```
options(object.size=1000000000,memory=3*2147483647,digits=17,contrasts=c("c
ontr.treatment","contr.poly"));library(mass,first=T);library(hmisc,T);libra
ry(design,T);masked()
```

Les instructions réunies sous la rubrique « `options` » autorisent une taille suffisante aux objets créés (« `object.size` »), une place en mémoire suffisante (« `memory` »), donnent la precision des calculs (« `digits` »), attribuent à chaque niveau d'une variable une variable accessoire type « *dummy variable* » (« `contr.treatment` ») : par exemple, si une variable a quatre valeurs possibles, 1, 2, 3 et 4, ces niveaux sont remplacés dans les calculs, respectivement, par (0,0,0), (1,0,0), (0,1,0) et (0,0,1). Les instructions « `library` » chargent des librairies de fonctions supplémentaires.

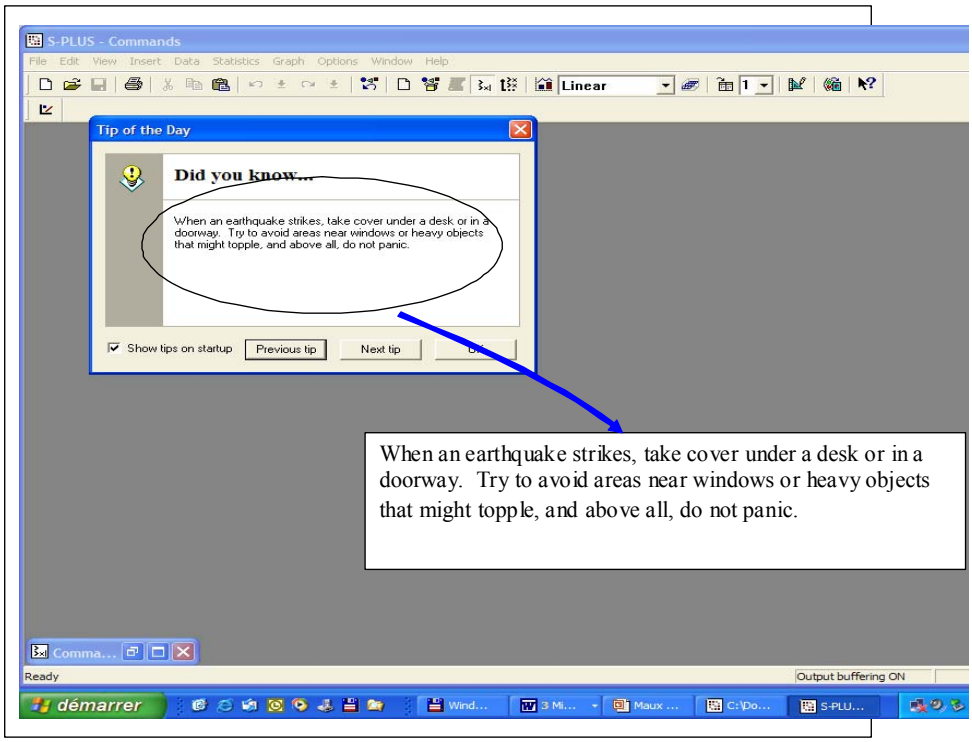
Quand le raccourci est prêt, on clique deux fois dessus et la fenêtre principale de S-PLUS s'ouvre (figure 55) :

Figure 55. Fenêtre principale de S-PLUS



Lorsqu'on coche la case adéquate, on peut avoir un ensemble de conseils relatifs au logiciel ou plus généraux (figure 56) !

Figure 56. Conseil du jour



Et, rappelons le, avant tout, avoir un réflexe ! La barre des menus, menu « *Help* » avec ses mots-clefs (« *Langage Reference* », la recherche des termes du langage S-PLUS (« *Search S-PLUS Help* »), l'aide raisonnée (S-PLUS Help) (figure 57) et les manuels (figure 58).

Figure 57. Aide en ligne

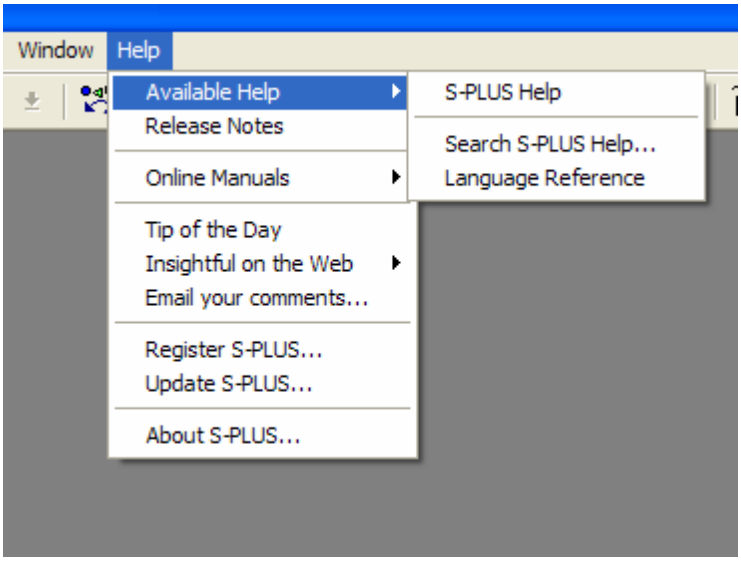
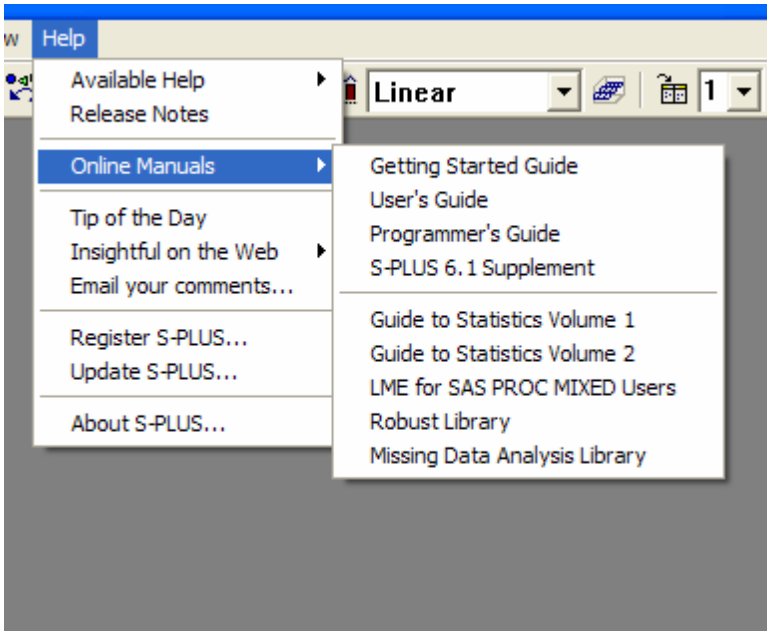


Figure 58. Les manuels



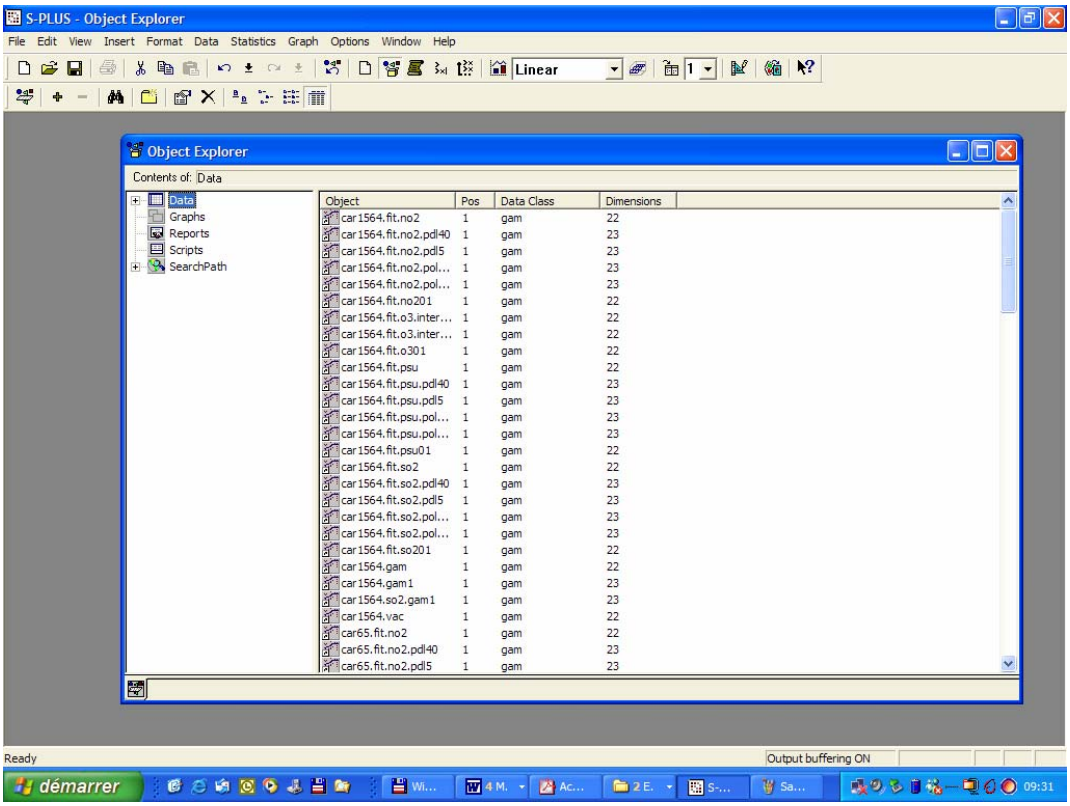
Deux icônes de la barre de commande sont intéressants (figure 59)

Figure 59. Icônes *Object Explorer* et *Commands Window*

Cliquer sur l'icône *Commands Window* (icône de droite sur la figure 59) fait apparaître la fenêtre de commande, lieu de saisie de toutes les commandes.

L'icône *Object Exporer* (à gauche sur la figure 59) ouvre la fenêtre de l'équivalent S-PLUS de l'Explorateur de Windows. On y voit tous les objets du répertoire de travail, ceux qu'on a créés et ceux qui y sont d'office (librairies, etc.). Sans être essentiel (pour voir les objets sur lequel on travaille, il suffit d'écrire « `objects()` » ou « `ls()` » sur la ligne de commande de la fenêtre de commandes pour les afficher dans cette même fenêtre), il n'en est pas moins « convivial » (figure 60).

Figure 60. Fenêtre de l'*Object Explorer*



Notion d'objet

Objet veut dire… objet. Comme beaucoup de logiciels, S-PLUS fonctionne avec des objets. Ceux-ci peuvent être des données (nombres, vecteurs, tableaux), des graphes, des modèles, des listes, etc. Ces objets sont composés et restent dans l'état après création jusqu'à destruction.

Exemples :

Le modèle « `morcard.fit.no2` », par exemple qui a été créé avec une ligne de commande spécifique (que nous verrons plus loin) est un objet composé. En effet :

L'instruction « `names(morcard.fit.no2)` » demande au logiciel de donner les noms des composants de l'objet `morcard.fit.no2`. Cette commande donne, ici :

```
 [1] "coefficients"             "residuals"                "fitted.values"
"effects"              "R"                        "rank"
 [7] "assign"                   "df.residual"              "contrasts"
"weights"              "smooth"                   "nl.df"
[13] "var"                      "terms"                    "call"
"formula"              "family"                   "nl.chisq"
[19] "y"                        "iter"                     "additive.predictors"
"deviance"             "null.deviance"
```

Nous voyons tous les composants d'un modèle : les coefficients, le vecteurs des résidus, les variables, les valeurs prédites, etc.

Nous verrons plus tard que « $ » est le symbole d'appartenance, c'est-à-dire que A$a désigne l'objet *a* appartenant à l'objet *A*.

Si nous écrivons « `morcard.fit.no2$coefficients` », nous devons obtenir l'expression de l'objet `coefficient` de l'objet `morcard.fit.no2`, soit :

```
(Intercept) lo(trend, 400/2922) s(dowf.num, 4)       j.feries          vac
lo(grip7, 0.9) lo(tempmin, 0.9) lo(hummin, 0.9) lo(tempmax2, 0.9)
    1.087884             -0.3503382    -0.002693843 -0.002949975 0.03055843
1.477998      -0.01299716      -0.7700738         -2.354468
     no224h01
 0.0007115992
```

Ce sont les valeurs estimées des coefficients relatifs aux différentes variables du modèle.

5.2.2. Données : manipulations et opérations

Importation / exportation

Fichiers de données étrangers à S-PLUS

Concernant les données étudiées, la situation la plus classique consiste à disposer d'un fichier Excel. Sur la première feuille du fichier (c'est celle-là qui sera importée *de facto*) figure le tableau des données, une colonne par variable avec, sur la première ligne, le nom des variables.

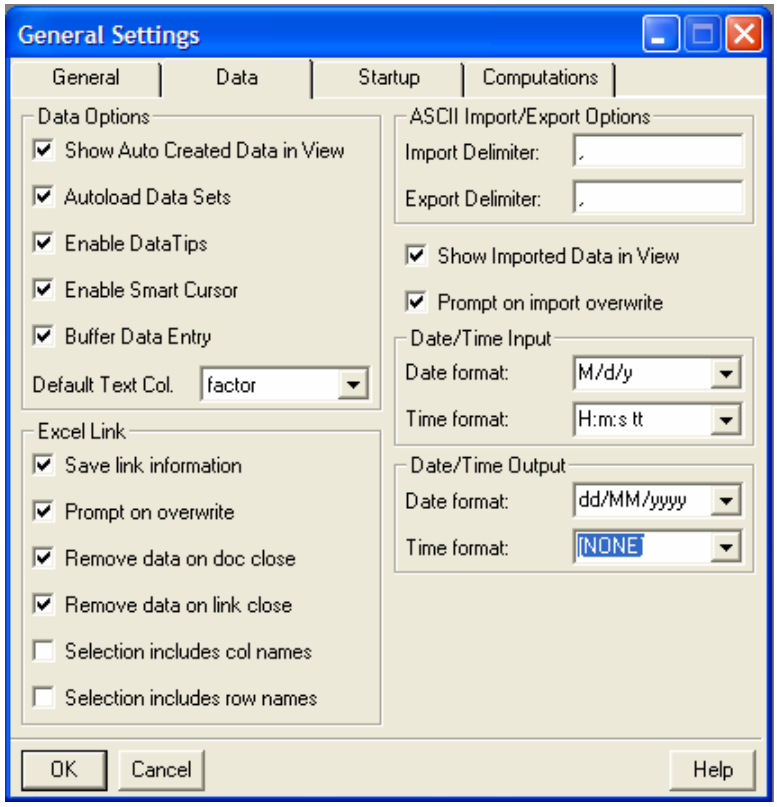
Dans la fenêtre principale de S-PLUS, il faut cliquer sur « *File* » puis sélectionner « *Import Data* » puis « *From File* », à la suite de quoi une fenêtre s'ouvre demandant le nom du fichier et son format (ici Excel). Bien sûr, d'autres formats peuvent être utilisés, comme il est indiqué dans la fenêtre.

On peut bien sûr exporter vers Excel ou un autre format.

Remarque. Pour importer correctement un tableau Excel avec des dates, il faut avant de réaliser cette manœuvre, choisir l'option de format *date* correcte :

Cliquer sur le menu « *Options* », puis choisir le sous-menu « *General Settings...* », puis l'onglet « *Data* » puis la rubrique « *Date/Time Output* ». Choisir pour « *Date Format* » le format « dd/MM/yyyy » et pour « *Time Format* » la réponse « [NONE] » (figure 61).

Figure 61. Formatage des dates avant importation d'un tableau Excel



Fichiers propres à S-PLUS

Un autre moyen d'importer et d'exporter des données, est d'utiliser les commandes « *data.dump* » et « *data.restore* ». Ces deux fonctions permettent, respectivement, d'exporter et d'importer des listes d'objets contenant des données. Nous verrons plus tard en quoi consistent ces objets (chapitre suivant) mais nous donnons, d'ores et déjà, l'instruction à saisir sur la ligne de commande (fenêtre de commandes).

Si un seul objet est concerné par l'exportation (un tableau de données, une matrice, un vecteur etc.) :

« `data.dump("objet","d:\\dir\\mon.fichier.dumpé")` »

« `objet` » est le nom qu'a l'objet dans S-PLUS (*i.e.* celui qui figure dans l'*Object Explorer* ou qui apparaît dans la fenêtre de commande quand on saisit « `objects()` »). « `d:\\dir\\mon.fichier.dumpé` » est le chemin choisi et le nom que l'on donne au fichier d'exportation.

Remarque. Ne pas oublier les guillemets « à l'anglaise » [46] dans la ligne de commande.

S'il faut exporter plusieurs objets de données en une fois, il faut écrire :

« `data.dump(c("objet.1",...,"objet.k"),"d:\\dir\\mon.fichier.dumpé")` »

La différence par rapport à la commande précédente est introduite par l'expression « `c("objet.1",...,"objet.k")` ».

La notation « `c(…)` » désigne une liste d'objets. Ces objets peuvent être de natures différentes, bien sûr.

Pour importer, c'est plus simple ! La commande est :

« `data.restore("d:\\chemin\\mon.fichier.dumpé")` »

Nature des variables

Les variables (*i.e.* les données) se présentent, comme dans les autres logiciels de statistiques ou de programmation, sous différentes formes.

Ce sont, des plus simples aux plus « composées » pour ne pas dire complexes : les constantes, les nombres (dates, heures comprises), les caractères, les vecteurs, les matrices, les tableaux (*data.frame*), les listes.

Les constantes s'écrivent avec des chiffres… Ne pas oublier que la virgule est un point ! Par exemple « `34.5432` ».

Les **variables numériques** (scalaires, dates, heures, etc.) sont appelées par un nom qui s'écrit de façon alphanumérique et commence par une lettre. Par exemple « psas.9 ». Comme on le voit, on peut insérer un point dans le nom (et même plusieurs).

Quand on veut attribuer une valeur (numérique) à une variable numérique, on écrit :

`nom.variable` ← `nombre`

« `nom.variable` » est le nom choisi pour la variable, `nombre` est la valeur entrée dans cette variable.

---

[46] The *quotes*.

La flèche est le symbole d'assignation de base mais l'écrire de nombreuses fois est fastidieux, aussi on la remplace par le signe « _ » (*underline*), donc l'instruction précédente devient

```
nom.variable_nombre
```

Exemple :

```
psas.9_12.3
```

Remarque. La version S-PLUS 6. autorise l'utilisation du signe « = ».

Les **variables alphanumériques** se nomment comme les variables numériques et on leur attribue une valeur (alphanumérique) de la façon suivante :

```
psas.9_"Derrière"
```

Si on tape « `psas.9` » dans la fenêtre de commande, on obtient ceci :

```
[1] "Derri\350re"
```

Donc il vaut mieux éviter les accents :

```
psas.9_"Derriere"
```

Si l'on ouvre l'*Object Explorer*, notre variable apparaît dans le répertoire *Data*.

Pour connaître la valeur d'une variable, il suffit d'écrire son nom dans la ligne de commande et d'appuyer sur la touche « ENTRÉE ».

Si l'on veut supprimer l'objet, on écrit sur la ligne de commande

```
rm(nom.variable)
```

Avec « `rm` » pour *remove*.

Les **vecteurs** sont constitués par une colonne de constantes ou de variables, numériques ou alphanumériques. On retrouve la lettre « c ».

Exemple :

```
psas.9_c(17,4,7.8)
```

Attention ne pas oublier le « `c(…)` » !

Que se passe-t-il si on inclue un vecteur dans un autre ?

Exemple :

```
psas.9_c(17,4,7.8)
aphea_c(5,psas.9,6.7)
```

Si l'on écrit « `aphea` » et si on appuie sur « ENTRÉE », la sortie affichée dans la fenêtre de commande est :

```
[1]  5.0 17.0  4.0  7.8  6.7
```

Donc…

On peut également aller dans l'*Object Explorer* et cliquer sur le nom du vecteur (ou déplacer le curseur sur le nom du vecteur et appuyer sur « ENTRÉE ») ; le vecteur apparaît alors comme colonne d'un tableau comme dans Excel (figure 62).

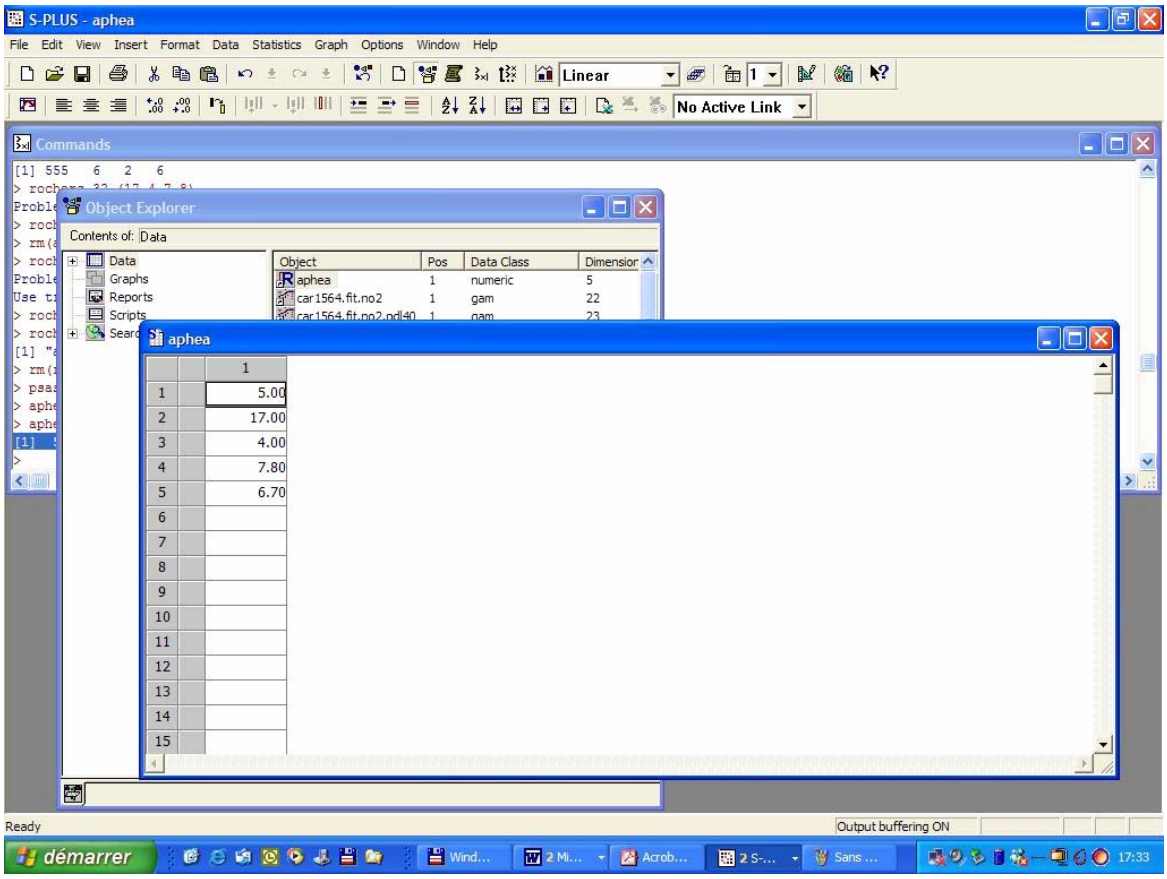
Les vecteurs peuvent bien sur être alphanumériques :

```
psas.9_c("lille","marseille","7.8","bordeaux")
```

Donne si l'on saisit son nom sur la ligne de commandes et on tape « ENTRÉE », on obtient ce qui suit :

```
[1] "lille"     "Marseille" "7.8"       "bordeaux"
```

Figure 62. Vecteur dans S-PLUS.



Les **matrices** sont des tableaux de données numériques ou alphanumériques de dimension n*m. Les données sont toutes du même type (nombres ou mots, etc.)

Exemple :

```
psas.9_matrix(c(1,5,7.8,98,2,100),3,2)
```

Donne une matrice contenant les nombres 1, 5, 7.8, 98, 2 et 100 disposés en 3 lignes et 2 colonnes. Si l'on tape

```
psas.9
```

On obtient :

```
     [,1] [,2]
[1,]  1.0   98
[2,]  5.0    2
[3,]  7.8  100
```

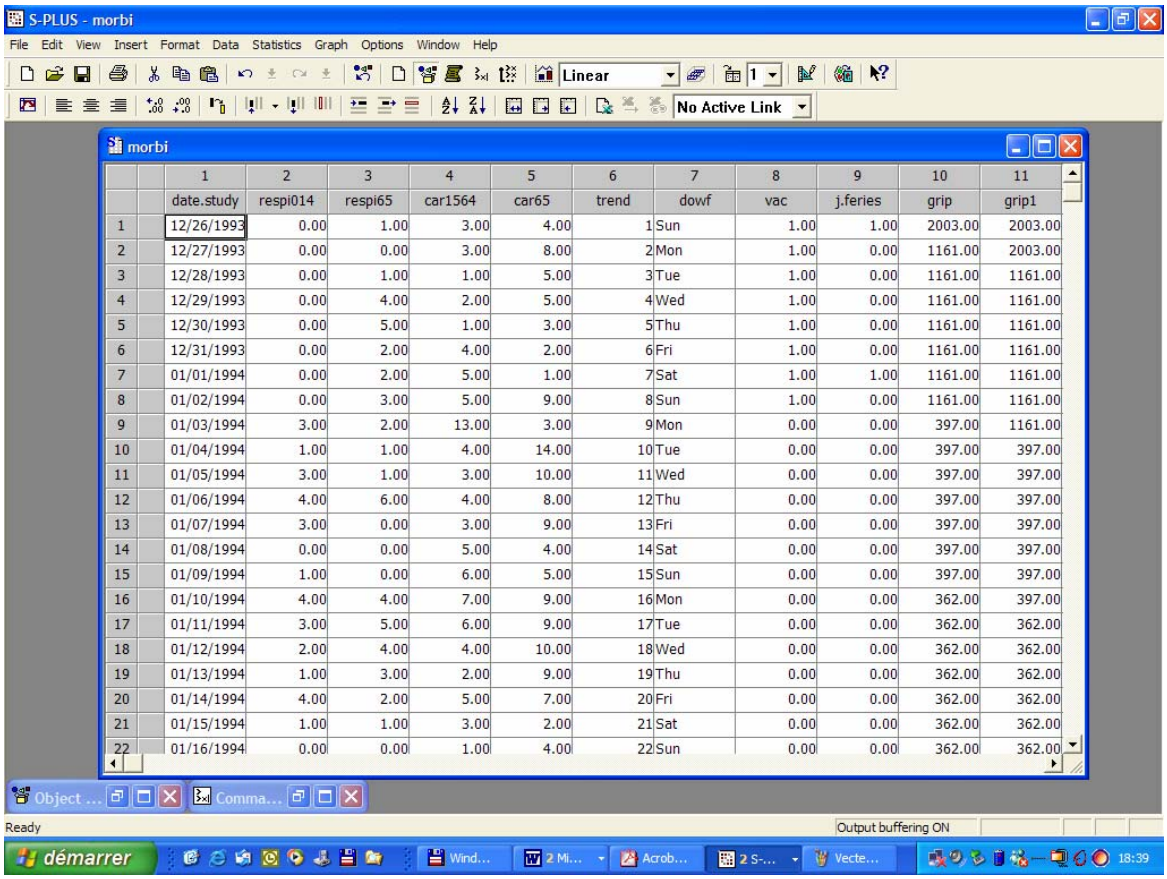
Si on clique sur l'objet « `psas.9` » dans l'*Object Explorer*, on obtient un tableau type « Excel » avec 2 colonnes.

Il y a de nombreuses autres façons d'écrire la commande permettant de fabriquer une matrice : à partir d'un fichier, par « apposition » de vecteurs, etc.

Les **tableaux proprement dits** (*data.frame*), sont des tableaux de données de tous types. Dans un *data.frame*, il peut y avoir à la fois des colonnes contenant des nombres, des valeurs logiques (0 / 1), des caractères alphanumériques, des dates, etc. Mais dans une colonne du *data.frame* ne figure qu'un type de données. De plus, c'est un tableau classique rectangulaire, donc toutes les colonnes ont la même longueur, c'est-à-dire le même nombre d'éléments.

Lorsqu'on importe une feuille de calcul Excel, c'est sous la forme d'un *data.frame* qu'elle apparaît dans S-PLUS (figure 63).

Figure 63. Tableau de données (*data.frame*)



	1	2	3	4	5	6	7	8	9	10	11
	date.study	respi014	respi65	car1564	car65	trend	dowf	vac	j.feries	grip	grip1
1	12/26/1993	0.00	1.00	3.00	4.00	1	Sun	1.00	1.00	2003.00	2003.00
2	12/27/1993	0.00	0.00	3.00	8.00	2	Mon	1.00	0.00	1161.00	2003.00
3	12/28/1993	0.00	1.00	1.00	5.00	3	Tue	1.00	0.00	1161.00	1161.00
4	12/29/1993	0.00	4.00	2.00	5.00	4	Wed	1.00	0.00	1161.00	1161.00
5	12/30/1993	0.00	5.00	1.00	3.00	5	Thu	1.00	0.00	1161.00	1161.00
6	12/31/1993	0.00	2.00	4.00	2.00	6	Fri	1.00	0.00	1161.00	1161.00
7	01/01/1994	0.00	2.00	5.00	1.00	7	Sat	1.00	1.00	1161.00	1161.00
8	01/02/1994	0.00	3.00	5.00	9.00	8	Sun	1.00	0.00	1161.00	1161.00
9	01/03/1994	3.00	2.00	13.00	3.00	9	Mon	0.00	0.00	397.00	1161.00
10	01/04/1994	1.00	1.00	4.00	14.00	10	Tue	0.00	0.00	397.00	397.00
11	01/05/1994	3.00	1.00	3.00	10.00	11	Wed	0.00	0.00	397.00	397.00
12	01/06/1994	4.00	6.00	4.00	8.00	12	Thu	0.00	0.00	397.00	397.00
13	01/07/1994	3.00	0.00	3.00	9.00	13	Fri	0.00	0.00	397.00	397.00
14	01/08/1994	0.00	0.00	5.00	4.00	14	Sat	0.00	0.00	397.00	397.00
15	01/09/1994	1.00	0.00	6.00	5.00	15	Sun	0.00	0.00	397.00	397.00
16	01/10/1994	4.00	4.00	7.00	9.00	16	Mon	0.00	0.00	362.00	397.00
17	01/11/1994	3.00	5.00	6.00	9.00	17	Tue	0.00	0.00	362.00	362.00
18	01/12/1994	2.00	4.00	4.00	10.00	18	Wed	0.00	0.00	362.00	362.00
19	01/13/1994	1.00	3.00	2.00	9.00	19	Thu	0.00	0.00	362.00	362.00
20	01/14/1994	4.00	2.00	5.00	7.00	20	Fri	0.00	0.00	362.00	362.00
21	01/15/1994	1.00	1.00	3.00	2.00	21	Sat	0.00	0.00	362.00	362.00
22	01/16/1994	0.00	0.00	1.00	4.00	22	Sun	0.00	0.00	362.00	362.00

Chaque colonne de l'objet *data.frame* peut être individualisée comme objet à son tour. Le *data.frame* pris comme exemple, ci-dessus, s'appelle `morbi`. Les colonnes sont identifiées par un numéro (1ère ligne du tableau) et par un nom (2ème ligne du tableau). Ce dernier n'est pas obligatoire ; il figure ici parce que `morbi` a été importé d'un fichier Excel contenant le nom des colonnes. La première colonne a comme nom *date.study*. Si l'on veut traiter cette colonne (la transformer, l'intégrer dans des calculs, etc.) il faut la nommer, en rappelant d'où elle vient : « `morbi$date.study` ». Le symbole « **$** » indique l'appartenance, nous le savons.

Lorsqu'on tape ce nom dans la fenêtre de commandes, on obtient :

```
[1] 12/26/1993 12/27/1993 12/28/1993 12/29/1993 12/30/1993 12/31/1993 01/01/1994 …
[13] 01/07/1994 01/08/1994 01/09/1994 01/10/1994 01/11/1994 01/12/1994 01/13/1994 …
[25] 01/19/1994 01/20/1994 01/21/1994 01/22/1994 01/23/1994 01/ …
...
[2173] 12/07/1999 12/08/1999 12/09/1999 12/10/1999 12/11/1999 12/12/1999 …
[2185] 12/19/1999 12/20/1999 12/21/1999 12/22/1999 12/23/1999 12/24/1999 …
[2197] 12/31/1999
```

La colonne peut être appelée autrement. Si on tape « `morbi[1]` » dans la fenêtre de commandes , on obtient :

```
   date.study
 1 12/26/1993
 2 12/27/1993
 3 12/28/1993
 4 12/29/1993
 …
2192 12/26/1999
2193 12/27/1999
     date.study
2194 12/28/1999
2195 12/29/1999
2196 12/30/1999
2197 12/31/1999
```

Il s'agit de la même colonne mais écrite sous forme d'un vecteur. [n] indique donc le n^{ième} élément du *data frame*.

Si on tape « `morbi[,1]` », on obtient le même résultat qu'avec *morbi$date.study*. [,n] représente donc la n^{ième} colonne, le blanc laissé devant la virgule voulant dire que le numéro de ligne est indifférent.

Si on tape « `morbi[1,]` », on obtient :

```
  date.study respi014 respi65 car1564 car65 trend dowf vac j.feries grip grip1 …
1 12/26/1993        0       1       3     4     1  Sun   1        1 2003  2003 …
   tempmin2 tempmin3 tempmax tempmax1 tempmax2 tempmax3 hummin hummin1 hummin2 …
1      2.2      3.9     2.6      3.2      4.9      6.6     62      77      62
…
```

Il s'agit de la première ligne, le blanc après la virgule voulant dire que le numéro de colonne est indifférent.

Enfin, si l'on tape « `morbi[3,5]` », on obtient :

```
[1] 5
```

Il s'agit de l'élément indicé, comme dans une matrice, par le numéro de la ligne et le numéro de la colonne dans lesquelles il se trouve.

Remarque. Il y a encore de nombreuses commandes possibles pour désigner ces éléments.

Une **liste**, enfin, est une collection d'objets de natures quelconques, de longueurs quelconques : vecteurs, modèles, *data.frame*, etc.

Manipulation des données

Nous avons déjà découvert les opérations de base destinées à créer les variables avec le symbole « _ » (ou « = », finalement). À présent, nous allons passer en revue un ensemble d'instructions permettant de construire des variables utiles. Nous en donnerons à chaque fois la formulation générale et un ou plusieurs exemples.

Créer une variable tendance

```
nom.data.frame$trend_c(1:T)
```

Cette commande crée un vecteur qu'on dénomme « *trend* » (de longueur T) contenant les nombres de 1 à T.

Ex : `infarct$trend_c(1:2134)`

Créer une variable « jour de la semaine », « mois de l'année » ou « saison »

Jour de la semaine

```
nom.var_weekdays(nom.vecteur.date)
```

Ex 1 : `ramses$jsem_weekdays(ramses$date.study)`

Ex 2 : `weekdays (dates("13/1/04", format="d/m/y"))`

Donne:

```
[1] Tue
Levels (first 5 out of 7):
[1] "Sun" "Mon" "Tue" "Wed" "Thu"
 Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

Mois de l'année

```
nom.var_months(nom.vecteur.date)
```

Ex : `months((dates("13/1/04", format="d/m/y"))`

On obtient :

```
[1] Jan
Levels (first 5 out of 12):
[1] "Jan" "Feb" "Mar" "Apr" "May"
 Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < Oct < Nov < Dec
```

Saison

```
mois=as.numeric(months(nom.vecteur.date))
nom.vecteur.saison=mois>numéro.mois1&mois<numéro.mois2
```

Exemple : création des variables *été* et *hiver*

```
mois=as.numeric(months(morta$date.study))
morta$summer=mois>3&mois<10
morta$winter=mois<=3|mois>=10
```

La première instruction fabrique un vecteur `mois` numérique, composé de nombres compris entre 1 et 12, la valeur 1 correspondant à janvier, la valeur 2 à février, etc. La deuxième instruction fabrique un

vecteur `morta$summer` avec des T (*true*) et des F (*false*) selon que le numéro du mois est compris strictement entre 3 et 10 ou non. La troisième instruction fabrique un vecteur `morta$winter`.

Remarque. Les symboles « & » et « | » correspondent, respectivement, aux opérateurs logiques ET et OU (voir plus bas).

Transformer une variable numérique en variable catégorielle

Rappelons qu'une variable catégorielle est une variable à niveaux [47].

```
variable.categor_as.factor(variable.numerique)
```

Ex : `ramses$vacances_as.factor(ramses$vacances)`

Transformer une variable catégorielle en variable numérique

```
variable.numerique_as.numeric(variable.categor)
```

Ex: `mois_as.numeric(months(morta$date.study))`

L'instruction `months` fabrique un vecteur contenant les mois correspondant aux dates.

Créer une variable pouvant prendre 2 valeurs selon 1 condition

1) Première formulation

```
variable2_ifelse(variable1==valeurdecondition,valeur1,valeur2)
```

Cette instruction impose à `variable2` la `valeur1` si la `variable1` est égale à `valeurdecondition` et la `valeur2` sinon.

Ex : `med_ifelse(ramses$nbmed==0,.1,ramses$nbmed)`

Ici, `med` prend la valeur 0.1 si `ramses$nbmed` égale 0 et la valeur `ramses$nbmed` sinon (et donc ne change pas de valeur)

2) Deuxième formulation

```
variable2_variable1>valinf & variable1<valsup
```

Ou

```
variable2_variable1<=valinf | variable1>valsup
```

Dans le premier cas, la `variable2` prend la valeur `T` (vrai) si la valeur de `variable1` est comprise entre `valinf` et `valsup` et prend la valeur `F`(faux) sinon.

Dans le second cas, la `variable2` prend la valeur `T` (vrai) si la valeur de `variable1` est inférieure ou égale à `valinf` ou supérieure strictement à `valsup` et prend la valeur `F` (faux) sinon

Remarque. Le symbole « & » correspond à l'opérateur logique ET (voir plus bas).

Ex : fabrication d'une variable « premier trimestre »

[47] Et un « passage catégoriel », c'est quoi ?

```
mois_as.numeric(months(morta$date.study))
```

PUIS

```
morta$t1_mois>=1&mois<=3
```

La première instruction fabrique un vecteur `mois` numérique, composé de nombres compris entre 1 et 12, la valeur 1 correspondant à janvier, la valeur 2 à février, etc. La deuxième instruction fabrique un vecteur `morta$t1` avec des T (*true*) et des F (*false*) selon que le numéro du mois est compris entre 3 et 10 ou non.

Fabriquer une variable, moyenne de plusieurs autres

Il faut écrire :

```
nouvvar_(var1+var2+...+varN)/N
```

Ex : `so224h2to4_ifelse(is.na=T,NA,(so224h2+so224h3+so224h4)/3)`

Remarque. La mention « `is.na` » signifie « la valeur est manquante »

Remplacement de certaines valeurs d'un vecteur par d'autres

L'instruction est :

```
nom.var[nom.var==valeur.à.remplacer]_ valeur.de.remplacement
```

Ex : `x[x==6]_10` remplace tous les 6 par des 10.

Répéter un vecteur

Il existe plusieurs cas.

Répéter un vecteur plusieurs fois

```
rep(nom.vecteur, nb.fois)
```

Ex : `rep(1:5, 3)` donne le vecteur `(1,2,3,4,5,1,2,3,4,5,1,2,3,4,5)`

Répéter chaque élément d'un vecteur un certain nombre de fois

```
rep(nom.vecteur, each=nb.fois)
```

Ex 1 : `rep(1:5, each=2)` donne le vecteur `(1,1,2,2,3,3,4,4,5,5)`

Ex 2 : `rep(grip,each=7)` donne le vecteur
`(grip[1],grip[1],..,grip[1],grip[2],grip[2],....`

Décaler une variable d'un rang ou de plusieurs rangs

Il convient d'écrire :

```
nom.var.decal1_c(rep(NA,1),nom.var[1:(length(nom.var)-1)])
nom.var.decal2_c(rep(NA,2),nom.var[1:(length(nom.var)-2)])
```

`nom.var.decal3_c(rep(NA,3),nom.var[1:(length(nom.var)-3)])`

etc.

Dans ces lignes, les instructions décalent le vecteur d'une valeur vers le bas et remplacent la première valeur par NA (valeur manquante).

Remarque. Le mot « `length` » donne la longueur du vecteur dont le nom est entre parenthèses.

Ex :

```
morta$gripa1_c(rep(NA,1),morta$gripa[1:(length(morta$gripa)-1)])
morta$gripa2_c(rep(NA,2),morta$gripa[1:(length(morta$gripa)-2)])
morta$gripa3_c(rep(NA,3),morta$gripa[1:(length(morta$gripa)-3)])
```

Destruction de variable

Nous l'avons vu précédemment :

`remove(nom.objet1,nom.objet2,...)` ou `rm(nom.objet1,nom.objet2,...)`

Rajouter une ligne ou une colonne à un data.frame

Si `Z` est un *data.frame*, `ligne` un vecteur ligne et `colonne` un vecteur colonne, alors

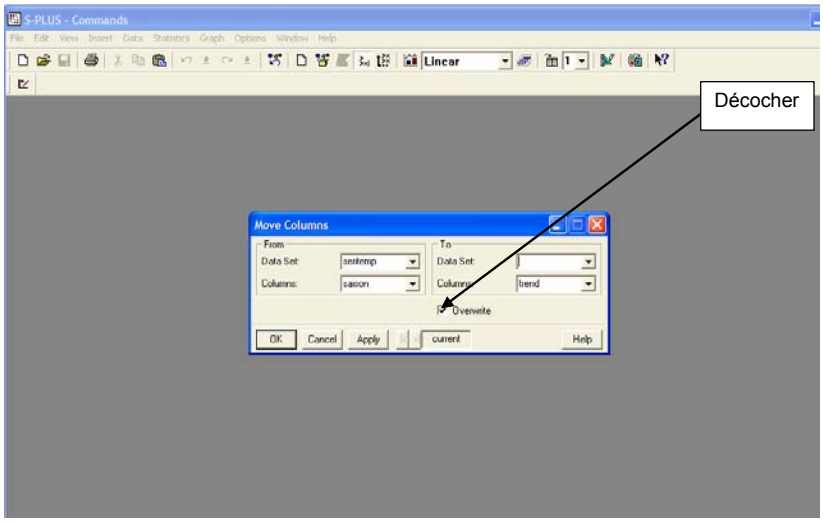
`Z'_rbind(ligne,Z)` fabrique un *data.frame* (`Z'`) avec la ligne `ligne` en première ligne !

`Z″_cbind(colonne,Z)` fabrique un *data.frame* (`Z″`) avec la colonne `colonne` comme première colonne.

On peut permuter les éléments `colonne` ou `ligne` avec `Z` dans l'instruction pour les placer, respectivement à la droite et en bas du *data.frame*.

On peut aussi déplacer une colonne avec le menu « Data ». Les actions sont « move » puis « colonne ». On obtient la fenêtre suivante où l'on choisit le data.frame, la colonne à déplacer, la colonne à côté de laquelle on déplace la colonne. Attention à ne pas écraser la colonne cible : il faut décocher « overwrite » (figure 64).

Figure 64. Déplacer une colonne dans un *data.frame*



Construire un « sous-data.frame » formé par certaines colonnes d'un autre data.frame

Z est Z' sont des *data.frames*, var1 et var2 sont des colonnes de Z. L'instruction suivante accole les colonnes désignées, extraites du premier tableau.

```
Z'_as.data.frame(cbind(Z$nom.var1,Z$nom.var2, ..))
```

Ex :
```
ficicor_as.data.frame(cbind(fifi$tempmin,fifi$hummin,fifi$co24h,fifi$no24h,
fifi$no224h,fifi$o38h,fifi$so224h,fifi$pm1324h))
```

Extraire une portion d'un data.frame

Selon la valeur d'une variable

a) 1ère méthode

```
portion.fichier_fichier[fichier$variable[limiteinf:limitesup],]
```
ou
```
portion.fichier_fichier[fichier$variable==valeur,]
```

Dans ces deux écritures, le vecteur portion.fichier est constitué d'une partie du fichier fichier. Dans le premier cas, les lignes qui sont conservées sont celles dont la variable impliquée (variable) voit ses valeurs comprises entre les valeurs limiteinf et limitesup. Dans le second cas, les lignes conservées sont celles dans lesquelles la variable impliquée prend la valeur valeur.

Ex 1 : `mortab_morta[morta$trend[785:2005],]`

Ex 2 : `mora.summer_mortaa[mortaa$summer[T],]`

Ex 3 : `seinv[seinv$age==1,]`

b) 2ème méthode

```
..., subset=(var!=val1&var>val2)
```

Ex :
```
..., subset=(trend!=418&trend>100)
```

Cette instruction n'est pas utilisée comme telle mais comme partie de l'écriture d'un modèle (voir plus loin).

Selon des numéros de lignes ou de colonne

a) Selon des numéros de lignes

```
portion.fichier_fichier[ligne.min:ligne.max,]
```

Ex : `grip.matrice.portion_grip.matrice[14:2010,]`

b) Selon des numéros de colonnes

```
portion.fichier_fichier[,colonne.min:colonne.max]
```

Extraire une case, une colonne ou une ligne d'un data.frame

```
case_{i,j}_tableau[i,j]
ligne_i_tableau[i,]
colonne_j_tableau[,j]
```

Pour extraire une colonne d'un data.frame afin d'en faire un vecteur : `vecteur_c(data.frame$Vx)` avec x, numéro de la colonne

Donner un nom aux colonnes d'un data.frame

Si Z est un tableau, l'instruction suivante attribue un nom à chaque colonne :

```
names(Z)_c("nom.var1","nom.var2",...)
```

Ex :

```
names(fificor)_c("tempmin","hummin","co224h","no24h","no224h","o38h","so224
h","pm1324h")
```

Il est possible de modifier la casse des noms des colonnes du data.frame. Si l'on veut transformer les majuscules en minuscules, l'instruction est:

```
names(nom.data.frame)=casefold(names(nom.data.frame))
```

Ex : `names(morta)=casefold(names(morta))`

Matrices et arrays

Nous l'avons vu précédemment, une matrice, tableau à 2 dimensions, contenant des nombres, se dit `matrix` en langage S-PLUS.

L'instruction générale la plus simple permettant d'en créer une est :

```
nom.matrice_matrix(data,nrow=nombre.lignes,ncol=nombre.colonnes)
```

`data` contient les données

Si l'on veut vraiment simplifier encore on se contente de :

```
nom.matrice_matrix(data,nombre.lignes,nombre.colonnes)
```

Fabriquons à titre d'exemple, une matrice 5 x 6 (5 lignes, 6 colonnes) avec des valeurs manquantes (NA) :

```
xy_matrix(data=NA, 5, 6)
```

On obtient :

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]   NA   NA   NA   NA   NA   NA
[2,]   NA   NA   NA   NA   NA   NA
[3,]   NA   NA   NA   NA   NA   NA
[4,]   NA   NA   NA   NA   NA   NA
[5,]   NA   NA   NA   NA   NA   NA
```

Construisons une matrice 3 x 4, contenant les 12 premiers nombres entiers avec l'écriture simple précédente :

```
xy_matrix(data=c(1:12),3,4)
```

On obtient :

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

Nous voyons que la matrice a été remplie par colonne, c'est-à-dire que les 3 premières valeurs ont été placées dans la 1ère colonne, les 3 suivantes dans la 2ème, etc.

Si l'on veut à présent remplir la matrice par ligne, c'est-à-dire placer les 4 premières valeurs dans la première ligne, les 4 suivantes dans la 2ème, etc. il faut écrire :

```
xy_matrix(data=c(1:12),3,4,byrow=T)
```

On obtient alors :

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

Lorsqu'on ne spécifie pas « byrow », ceci revient à l'instruction « byrow=F »

Il est possible de donner des noms aux colonnes et aux lignes de la matrice :

```
dimnames(xy)_list(c("a","b","c"),c("c","e","f","g"))
```

« xy » est le nom de la matrice, crée précédemment, « a », « b », … sont les noms attribués aux lignes et aux colonnes.

On obtient en tapant le nom de la matrice (« xy ») dans la fenêtre de commande :

```
  c  e  f  g
a 1  2  3  4
b 5  6  7  8
c 9 10 11 12
```

Si l'on veut en une ligne créer la matrice x et les noms des lignes et colonnes, on écrit :

```
xy_matrix(data=c(1:12),3,4,byrow=T,dimnames_list(c("a","b","c"),c("c","e","
f","g")))
```

Les « *arrays* » sont aux matrices ce que les cubes et hyper-cubes sont au carré.

Ce sont des tableaux à n dimensions, avec n quelconque.

L'instruction permettant de créer un tel tableau est :

```
xyz_array(data,dim=c(n1,n2,…np))
```

`data` représente les valeurs à placer dans le tableau, n_1, n_2, …n_p sont les « longueurs » respectives des p dimensions du tableau.

Ex :

```
xyz_array(c(1:24),dim=c(4,3,2))
```

Cette instruction fabrique un tableau à 3 dimensions (un cube) dont les longueurs sont, respectivement, 4, 3 et 2.

Visualisons ce tableau :

```
, , 1
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12

, , 2
     [,1] [,2] [,3]
[1,]   13   17   21
[2,]   14   18   22
[3,]   15   19   23
[4,]   16   20   24
```

Ainsi le tableau est présenté par couches successives.

Les éléments (« les cases ») du tableau (matrices comprises) sont accessibles par une expression du type :

`nom.tableau[`x_1`,`x_2`,…, `x_p`]`

x_1,x_2,…, x_p représentent la position de l'élément dans le tableau.

Par exemple, pour le tableau précédent, on peut écrire :

```
xyz[3,2,1]_111
```

Le tableau xyz devient :

```
, , 1
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3  111   11
[4,]    4    8   12

, , 2
     [,1] [,2] [,3]
[1,]   13   17   21
[2,]   14   18   22
[3,]   15   19   23
[4,]   16   20   24
```

Détruire un objet

L'instruction est la même que pour une variable simple. Il faut écrire :

```
rm(objet.à.détruire)
```

Il est possible de détruire plusieurs objets en une ligne. Par exemple, pour détruire les matrices `xy` et `xyz`, créées précédemment :

```
rm(xy,xyz)
```

Opérations sur les variables

Opérations algébriques

Les opérations arithmétiques et algébriques, classiques, sur les nombres sont exprimées avec une symbolique tout aussi classique :

« `+` », « `-` », « `*` », « `/` », pour les quatre opérations principales, « `**` » ou « `^` » pour l'exponentiation.

« `log` », « `exp` », « `sqrt` », pour le logarithme népérien (et non « `ln` »), pour l'exponentielle et la racine carrée. Pour le logarithme en base 10 (celui qui est noté log en mathématique, il faut écrire « `log10(nombre)` ».

Il est possible de concaténer des expressions composées de caractères alphanumériques. Ex.

Définissons trois variables alphanumériques

```
psas.9_"Derriere"
```

```
apheis_"la"
```

```
aphea_"porte"
```

Pour concaténer les variables il faut utiliser la commande « `paste` » :

```
paste(psas.9,apheis,aphea)
```

```
[1] "Derriere la porte"
```

Opérations logiques

Notations « F » et « V »

Nous avons vu plus haut que : `F` = 0 ; `V` = 1.

Les signes « == », « < », « > », « <= », « >= », « != »

La signification de ces signes apparaît dans les expressions suivantes (instructions et sorties S-PLUS) et se passe d'explication :

```
3==3
```

```
[1] T
```

```
3==4
```

```
[1] F
```

```
3<4
```

```
[1] T
```

```
3>4
```

```
[1] F
```

```
3<=6
```

```
[1] T
3<=1
[1] F
3!=1
[1] T
```

Remarque. Comme on le verra plus bas, le signe « ! » signifie « non ». Corollaire : la dernière instruction logique (« != ») signifie « différent de ».

Les signes « & » et « | »

Le signe « & » signifie « ET », le signe « | » signifie « OU » (inclusif) :

```
(3<=6)&(5>0)
[1] T
(3<=6)&(5<0)
[1] F
(3<=6)|(5>0)
[1] T
(3<=6)|(5<0)
[1] T
(3>=6)|(5<0)
[1] F
```

Le signe « ! »

Ce signe indique la négation de la valeur qui suit.

Si « p » est une proposition, « !p » est sa négation (« non p »). Par exemple l'expression « !is.na » veut dire « il n'y a pas de valeur manquante », « !is.na(morta$gripa7) » veut dire, selon le contexte, « la variable gripa7 du *data.frame* morta n'a pas de valeur manquante » ou « ne pas considérer les valeurs manquantes de la variable gripa7 ».

Voici encore quelques expressions avec le signe « ! » :

```
3!=7
[1] T
3!=3
[1] F
```

Statistiques descriptives

Statistiques descriptives de base

Les minimum, maximum, moyenne, quartiles s'obtiennent avec la commande :

```
summary(nom.variable)
```

Cette commande produit, comme son nom l'indique, un résumé des propriétés de l'objet auquel elle s'applique (un *synopsis* comme il est dit dans l'aide de S-PLUS). Si cet objet est une variable,

`summary` donne les caractéristiques principales de cette variable. Si c'est un modèle, elle donne un ensemble d'informations relatives aux paramètres du modèle, comme nous le verrons plus loin. La sortie de cette commande dépend donc de l'objet qui figure entre les parenthèses.

Ex :

```
summary(morta$mortot)
```

Cette instruction donne la sortie suivante:

```
   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
  1.000  6.000   8.000  8.512 10.000  23.000
```

La variance est calculée avec :

```
var(nom.variable)
```

La covariance et le coefficient de corrélation avec :

```
var(nom.variable1,nom.variable2)
```

et

```
cor(nom.variable1,nom.variable2)
```

Descriptif d'un vecteur par histogramme

Le petit programme suivant donne la distribution des valeurs d'un vecteur selon un critère de découpage (nombre de valeurs par tranche de valeurs) :

```
for(i        in        1:length(tapply(morta$no224h,cut(morta$no224h,breaks=
c(0,10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,180,190,200)
),table)))
print(sum(tapply(morta$no224h,cut(morta$no224h,breaks=c(0,10,20,30,40,50,60
,70,80,90,100,110,120,130,140,150,160,170,180,190,200)),table)[[i]]))
```

Les intervalles définis sont [0 ; 10 [, [10 ; 20 [, …, [90 ; 200 [. Cette expression donne un résultat du type suivant :

```
[1] 1
[1] 43
[1] 256
[1] 635
[1] 796
[1] 587
[1] 310
[1] 143
[1] 66
[1] 32
[1] 12
[1] 6
[1] 2
[1] 0
```

```
[1] 2
[1] 3
[1] 0
[1] 0
[1] 0
[1] 0
```

Nombre de valeurs d'un vecteur

L'expression suivante donne le nombre de valeurs du vecteur égales à une valeur donnée :

```
length(nom.vecteur[nom.vecteur==n])
```

Ex :

```
length(morta$mortot[morta$mortot==3])
```

Cette expression donne le nombre de valeurs du vecteur égales à des valeurs données :

```
for(i in n1:n2) print(c(i,length(nom.vecteur[nom.vecteur==i])))
```

Ex :

```
 for(i in 0:23) print(c(i,length(morta$mortot[morta$mortot==i])))
```

Calcul des quantiles

Voici différents cas.

Quantiles pour intervalles identiques

L'écriture générale est du type : `0,1,k/100`

L'expression est :

```
quantile(nom.variable,probs=seq(0,1,.05),na.rm=T) : donne les quantiles 0, 5, 10, ... 100
```

Quantiles pour intervalles variables

```
quantile(nom.variable,
c(0,.05,.1,.15,.2,.25,.3,.35,.4,.45,.50,.55,.6,.65,.7,.75,.8,.85,.9,.95,.98,.99,1),na.rm=T)
```

Colonne composée des quantiles avec mélange d'intervalles identiques et variables et avec moyenne

```
cbind(c(quantile(variable, seq(0, .95, .05), na.rm = T), quantile(variable, seq(.98, 1, .01),na.rm = T), mean(variable, na.rm = T)))
```

Cette expression donne les quantiles 0, 5, 10, ...95, 98, 99, 100 et la moyenne des valeurs.

Ex :

```
cbind(c(quantile(mortaa$o38h, seq(0, 0.95, 0.05), na.rm = T), quantile(mortaa$o38h, seq(0.98, 1, 0.01),na.rm = T), mean(mortaa$o38h, na.rm = T)))
```

Calcul corrélations des variables d'un data.frame ou d'une matrice

La commande est la suivante :

```
cor(nom.tableau,na.method="available")
```

L'expression `na.method=` … indique la façon de traiter les valeurs manquantes.

Calcul des paramètres statistiques d'un ensemble de variables d'un data.frame selon les niveaux d'une variable qualitative

```
mois_as.numeric(months(morta$date.study))
morta$summer_mois>3&mois<10
```

La première ligne fabrique un vecteur contenant les numéros correspondant aux mois (1 pour janvier, 2 pour février, etc.), la seconde fabrique un vecteur contenant des V et des F, selon que la date correspond à un numéro de mois compris entre 3 et 10 (été) ou en dehors de cet intervalle (hiver).

```
summary(morta[morta$summer[T],])
```

Cette expression donne les statistiques des éléments du tableau pour l'été :

```
    rosee1             rosee2             rosee3             premin
    Min.:-99.9000      Min.:-99.9000      Min.:-99.9000      Min.:-9999.9000
 1st Qu.:  7.4000   1st Qu.:  7.3000   1st Qu.:  7.2000   1st Qu.: 1010.4000
  Median: 10.8000    Median: 10.8000    Median: 10.8000    Median: 1014.5000
    Mean:  9.9641      Mean:  9.9167      Mean:  9.8051      Mean:  999.1031
 3rd Qu.: 13.9000   3rd Qu.: 13.9000   3rd Qu.: 13.9000   3rd Qu.: 1018.4000
    Max.: 20.5000      Max.: 20.5000      Max.: 20.5000      Max.: 1030.6000
```

```
length(Dimdp[Dimdp$summer[T],]$date)
```

Cette instruction donne le nombre de lignes du tableau correspondant à l'été

Attention.

```
summary(Dimdp[Dimdp$summer[F],])
```

Ne fonctionne pas (ne donne que des « `NA` ») aussi pour sélectionner l'hiver il faut créer

une variable `winter` :

```
Dimdp$winter_mois<4|mois>9
```

Puis calculer les paramètres

```
summary(Dimdp[Dimdp$winter[T],])
```

5.2.3. Chemins, répertoires, attachement, Explorateur (*Object Explorer*)

Nous avons déjà vu l'*Object Explorer*, anciennement *Object Browser* (§ 5.2.1). Il s'agit, nous le savons d'un explorateur comme dans Windows. En bref il se présente avec deux fenêtres.

Celle de gauche contient les noms des répertoires des objets (figure 65). Elle permet de naviguer dans l'arborescence des objets en question. Il y a au départ (et à l'arrivée si l'on ne change rien) cinq répertoires (figure 66) : « `Data` » qui contient les objets de données (variables, matrices, vecteurs, *data.frames*, etc.), « `Graphs` » qui contient les graphes créés et conservés, « `Reports` » qui contient les fichiers textes créés par S-PLUS (appelés justement « *reports* » ; nous verrons ces fichiers en détail un peu plus loin : § 5.2.5.), « `Scripts` » qui contient les fichiers de commande permettant de

rassembler un ensemble d'intructions à faire exécuter au logiciel (ces fichiers serons vus plus bas : § 5.2.5.) et le répertoire « `SearchPath` » qui contient un ensemble d'autres répertoires. Le premier d'entre eux est notre répertoire de travail (celui qu'on a créé au début : voir § 5.2.1.), les autres contiennent des bibliothèques de données, d'exemples, de fonctions permettant de tester et d'apprendre à manier S-PLUS. Ces répertoires s'attachent (ils sont mis à disposition de l'utilisateur) et se détachent (ils sont momentanément mis hors du champ de travail) facilement : il suffit de sélectionner le répertoire et de cliquer avec le bouton droit de la souris puis de choisir « `Attach` » dans le menu contextuel.

La fenêtre de droite (nommée « `Right Pane` ») contient les objets (figures 65 et 66).

Figure 65. Object Explorer, fenêtres et répertoires

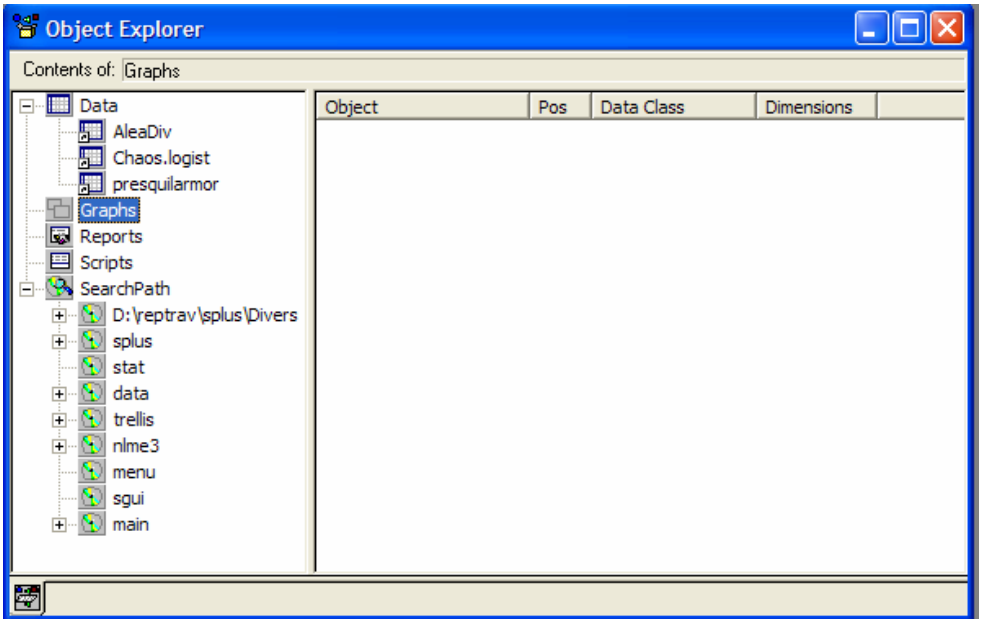
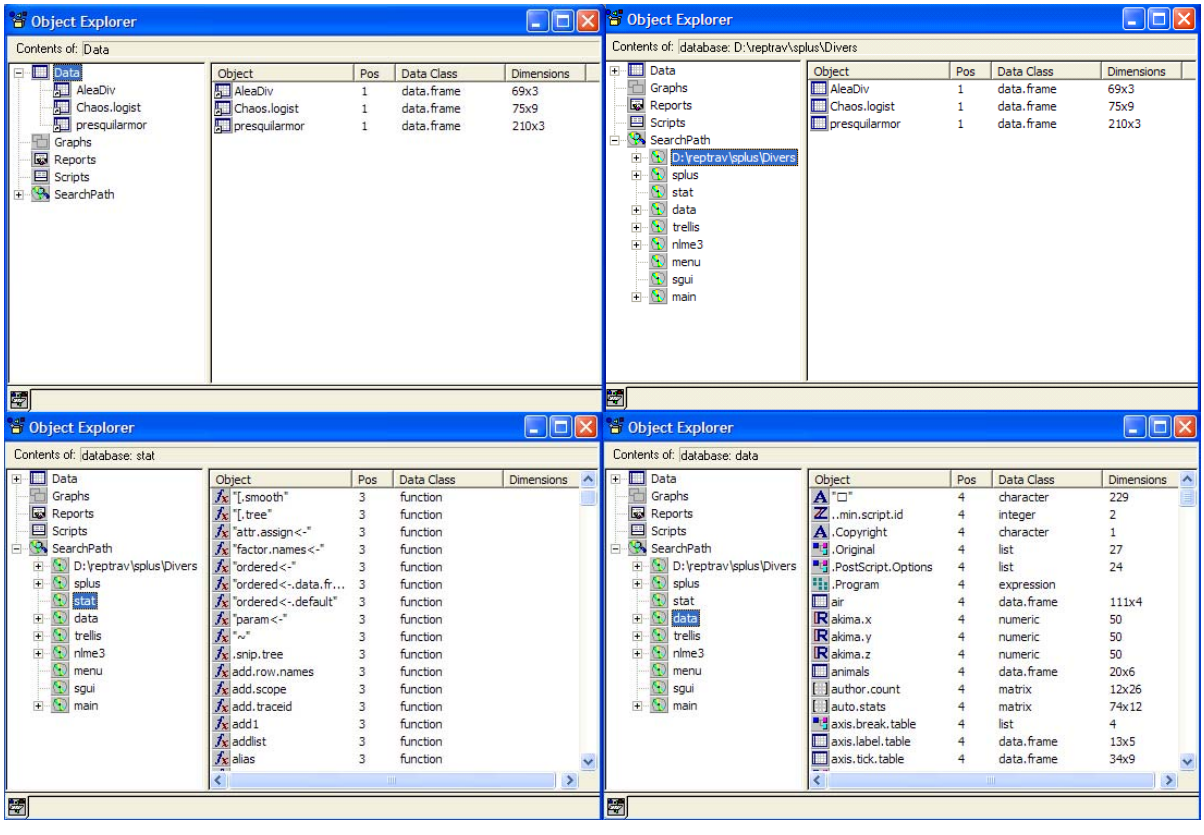


Figure 66. Object Explorer. Différents répertoires



5.2.4. Graphes

Instruction de base

L'instruction de base pour créer un graphe est :

```
plot(varA,varB,type="lettre",pch=nombre1,lty=nombre2, col=nombre3)
```

`lettre` est : `p`, `l` ou `b` (il en existe d'autres)

`p` impose un graphe formé de points, `l` un graphe formé de ligne(s), `b` un graphe formé de points et de lignes ;

« `nombre1` » indique la forme de la marque (cercle, carré, triangle, etc.) ;

« `nombre2` » indique la forme de la ligne (1 pour continue, 2 pour discontinue) ;

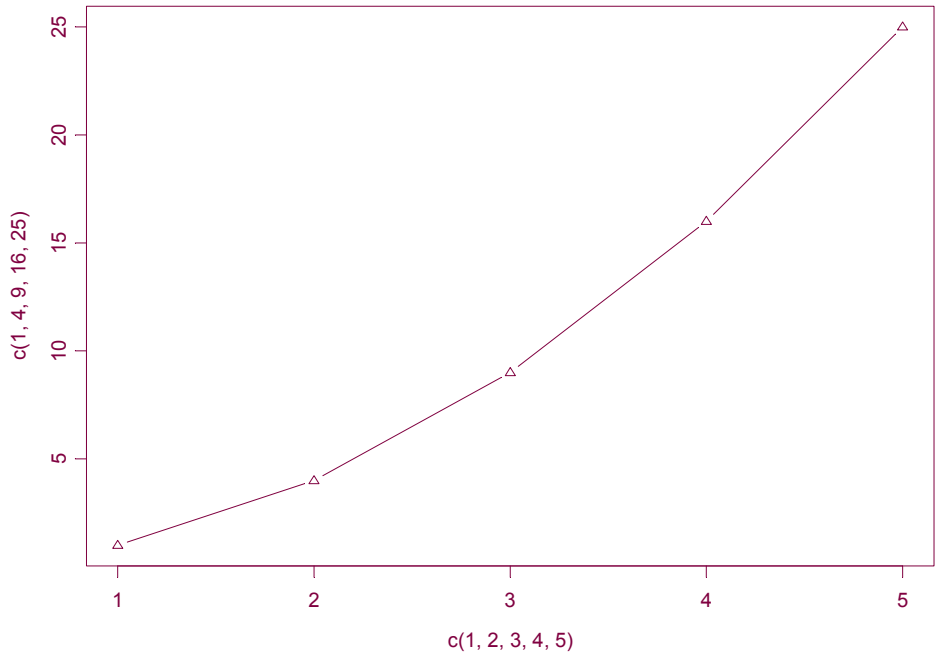
« `nombre3` » indique la couleur.

Ex :

```
plot(c(1,2,3,4,5),c(1,4,9,16,25),type="b",pch=2,col=3)
```

Le graphe obtenu est (figure 67) :

Figure 67. Exemple de graphe obtenu avec la commande « `plot` »



Pour rajouter une courbe (ligne ou points) sur un graphe déjà réalisé

L'instruction à utiliser est « `lines` » ou « `points` ».

```
lines(x,y,…)
```

```
points(x,y,…)
```

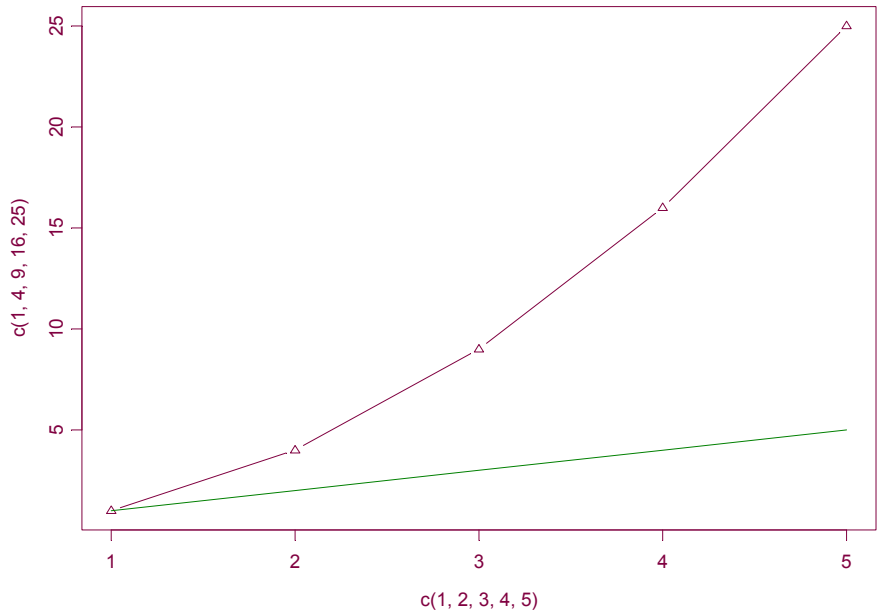
Les trois petits points derrière la virgule signifient que l'ont peut utiliser les mêmes instructions que pour la commande « `plot` ».

Ex :

```
plot(c(1,2,3,4,5),c(1,4,9,16,25),type="b",pch=2,col=3)
lines(c(1,2,3,4,5),c(1,2,3,4,5),type="l",col=4)
```

Ces instructions donnent (figure 68) :

Figure 68. Graphe obtenu avec « `plot` » et « `lines` »



Graphes avec lissage

On peut lisser La courbe :

```
plot(fct.lissage(varA,varB))
```

Avec `fct.lissage` prenant la forme `lowess`, `supsmu` ou `spline`.

On peut aussi superposer la courbe et son lissage

```
plot(varA,varB)
lines(fct.lissage(varA,varB),col=couleur)
```

Avec `fct.lissage` prenant la forme `loess.smooth`, `supsmu` ou `smooth.spline`.

Remarque. La fonction « `supsmu` » est un lissage semblable à « `lowess` » mais il « choisit » plus ou moins librement la taille des fenêtres selon la variabilité locale.

Ex 1 :

```
plot(lowess(fitted(drpha.gam),resid(drpha.gam)^2))
plot(lowess(fitted(drpha.gam),resid(drpha.gam,type="deviance")^2))
plot(lowess(fitted(drpha.gam),resid(drpha.gam,type="working")^2))
```

```
plot(supsmu(fitted(drpha.gam),resid(drpha.gam)^2))
```

Ex 2 :

Il est possible de faire figurer le graphe initial et les différents lissages possibles. Les instructions se présentent sous la forme générale :

`plot(x,y)`

`lines(loess.smooth(x,y))` ou `lines(smooth.spline(x,y))` ou `lines(supsmu(x,y))`

Par exemple, les commandent suivantes donnent les graphes de la figure 69.

```
plot(morta$trend,morta$mortot,type="p",pch='.',ylim=c(7,11),xlab="Temps",yl
ab="Mortalité journalière, Strasbourg")
lines(smooth.spline(morta$trend, morta$mortot),lty=2,lwd=4)
lines(loess.smooth(morta$trend, morta$mortot),lty=3,lwd=2)
lines(supsmu(morta$trend,morta$mortot))
```

Remarque.

« `type="p"` » commande le mode « point »,

« `ylim=c(7,11)` » limite le graphe en hauteur

« `lty=2` » commande le type de ligne (pointillé, continu, tiret, etc.)

« `lwd=4` » commande l'épaisseur du trait

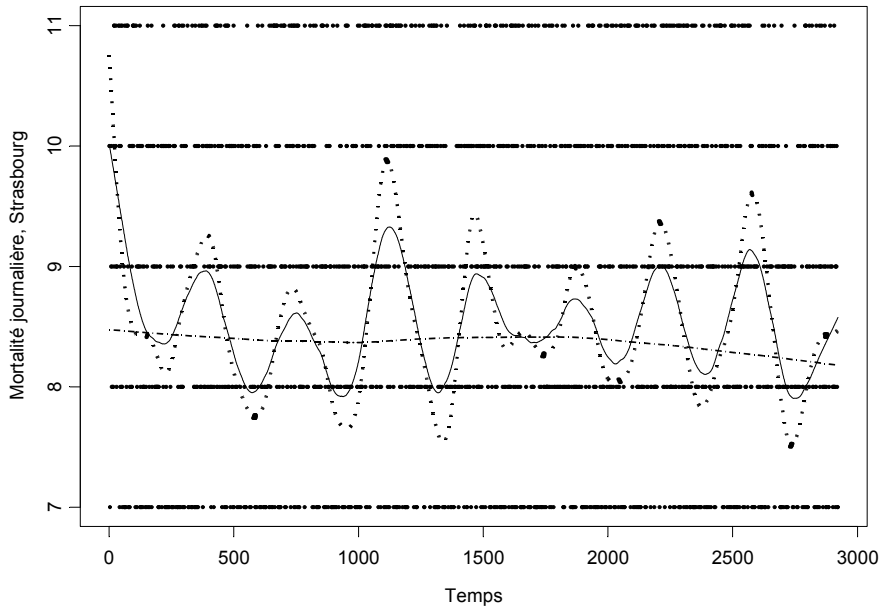
Remarque.

Avant de créer le graphe, S-PLUS fait défiler une liste d'avertissements du type :

« `WARNING: Point out of bounds: x = 2921.000000, y = 13.000000` »

Ceci est en rapport avec les limites qui ont été imposées au graphe.

Figure 69. Exemple de série temporelle discrète (pointillés horizontaux) avec lissage spline (· · ·), loess (– – –) et supsmu (——)



Graphes relatifs à la modélisation

Pour tracer un ensemble de graphes relatifs à un modèle, il suffit d'écrire :

```
plot(nom.modèle)
```

Ou, ce qui revient au même :

```
plot.gam(nom.modèle)
```

On peut obtenir un autre ensemble de graphes avec :

```
plot.glm(nom.modèle)
```
[48]

L'instruction peut être un peu plus précise :

```
plot(fifi$date.study,fifi$no24h,type="l",xlim=c(x1,x2),ylim=c(y1,y2),main="
Moyenne journalière de NO",xlab="",ylab="Concentration de NO en microg/m3")
```

Cette écriture permet de spécifier un ensemble de caractéristiques, telles que l'étendue des valeurs sur l'axe des abscisses (« `xlim` »), des ordonnées (« `ylim` »), le titre principal (« `main` »), les titres des axes (« `xlab` » et « `ylab` »).

Créer une fenêtre graphique

Avec le menu

D'abord deux solutions :

- Icône **New**

OU

- Menu **File** - **New** et Touche **Enter**

Puis :
Menu **Graph Sheet** et Touche **Enter**

Ces manipulations créent une fenêtre graphique appelée « **GS1** ».

Avec les commandes

Pour faire apparaître une feuille « Graphsheet 2 » (appelée « **GSD2** ».), la commande est « `graphsheet()` ». Cette fenêtre est disposée en **paysage**.

Pour faire apparaître une feuille Graphsheet 2 en **portrait**, il faut écrire : `graphsheet(orientation="portrait")`

La commande « `par()` » crée aussi une fenêtre graphique « **GSD2** ». Cette commande a la particularité de faire figurer la fenêtre graphique en dessous de la fenêtre de commande, ce qui n'est pas très gênant car en cliquant dessus on la met au premier plan. Elle a une autre particularité c'est d'écrire une ensemble de lignes sur la fenêtre de commande dont voici un extrait :

---

[48] Cette commande produit l'écart type de la variance en fonction des valeurs prédites (ceci permet de voir si il y a une tendance et si c'est le cas le choix de la famille de lois est mauvais), un graphe « `total / fitted` » qui, si tous les points sont sur la première bissectrice montre une bonne adéquation du modèle aux données, un graphe « *qq-plot* » qui compare les résidus de Pearson à une distribution théorique (la relation doit être linéaire sinon la dsitribution n'est pas normale).

```
$"1em":
[1] 0.02391402 0.03627594
$adj:
[1] 0.5
$ask:
[1] F
$bty:
[1] "o"
…
$yaxp:
[1] 0 1 5
$yaxs:
[1] "r"
$yaxt:
[1] "s"
```

Remarque. La fenêtre graphique, se crée toute seule comme une grande quand on introduit la commande « `plot…` ».

*Remarque***.** Voici quelques commandes à utiliser dans « R » pour manipuler les fenêtres graphiques :

Pour afficher la liste des fenêtres graphiques ouvertes

```
dev.list()
```

Pour rajouter une fenêtre graphique d'un certain type (Windows, pdf, etc.)

```
x11()
```

ou

```
pdf
```

ou...

Pour fermer une fenêtre graphique

```
set.off(numéro.de.fenêtre)
```

Pour afficher le numéro de la fenêtre graphique active

```
dev.cur()
```

Pour changer de fenêtre active

```
dev.set(numéro.de.fenêtre)
```

Choisir l'orientation d'un graphe

On peut suivre la suite d'activations de boutons et/ou de menus (la manipulation commence avec la création de la fenêtre comme précédemment) :

1) Icône **New** OU Menu **File** - **New** et Touche **Enter**

2) Menu **Graph Sheet** et Touche **Enter**

La fenêtre graphique est créée

3) Menu **Format** - **Sheet** et Touche **Enter**

La fenêtre "Graph Sheet Properties" apparaît

3) Ligne **Orientation** choisir **Landscape** (paysage) OU **Portrait** et Touche **Enter**

4) Menu **Options** et Touche **Enter**

5) Menu **Save Windows Size/..**. et Touche **Enter**

Placer plusieurs graphes sur la même page

La commande est :

```
par(mfrow=c(m,n))
```

Elle crée la place pour des graphes placés selon m lignes et n colonnes

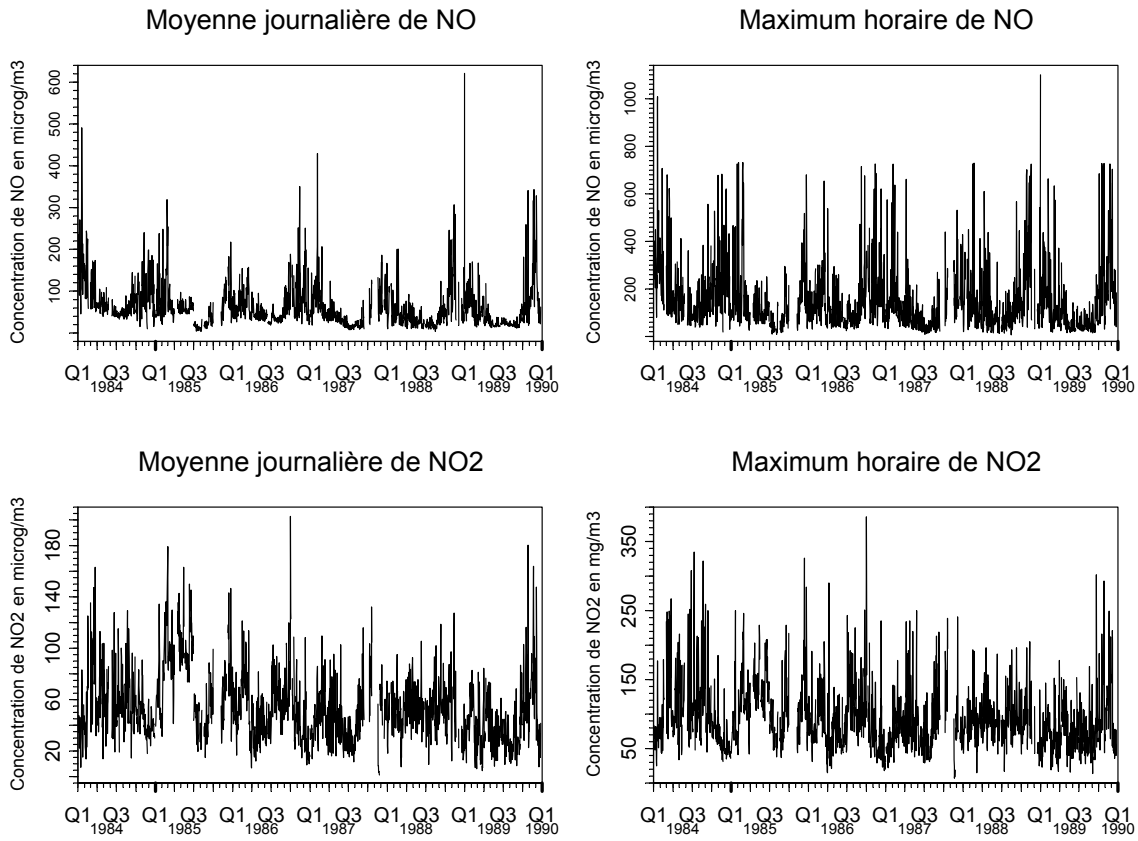
Remarque. La commande est identique dans R.

Ex :

```
par(mfrow=c(2,2))
plot(fifi$date.study,fifi$no24h,type="l",main="Moyenne journalière de NO",xlab="",ylab="Concentration de NO en microg/m3")
plot(fifi$date.study,fifi$no1h,type="l",main="Maximum horaire de NO",xlab="",ylab="Concentration de NO en microg/m3")
plot(fifi$date.study,fifi$no224h,type="l",main="Moyenne journalière de NO2",xlab="",ylab="Concentration de NO2 en microg/m3")
plot(fifi$date.study,fifi$no21h,type="l",main="Maximum horaire de NO2",xlab="",ylab="Concentration de NO2 en mg/m3")
par(mfrow=c(1,1))
```

La première ligne crée une structure de page à 4 figures. Les lignes suivantes créent les schémas (figure 70). La dernière revient à la présentation standard.

Figure 70. Page multi-figures



Créer un tableau n x n de figures représentant les variations d'un ensemble de variables *versus* les mêmes variables.

L'instruction est :

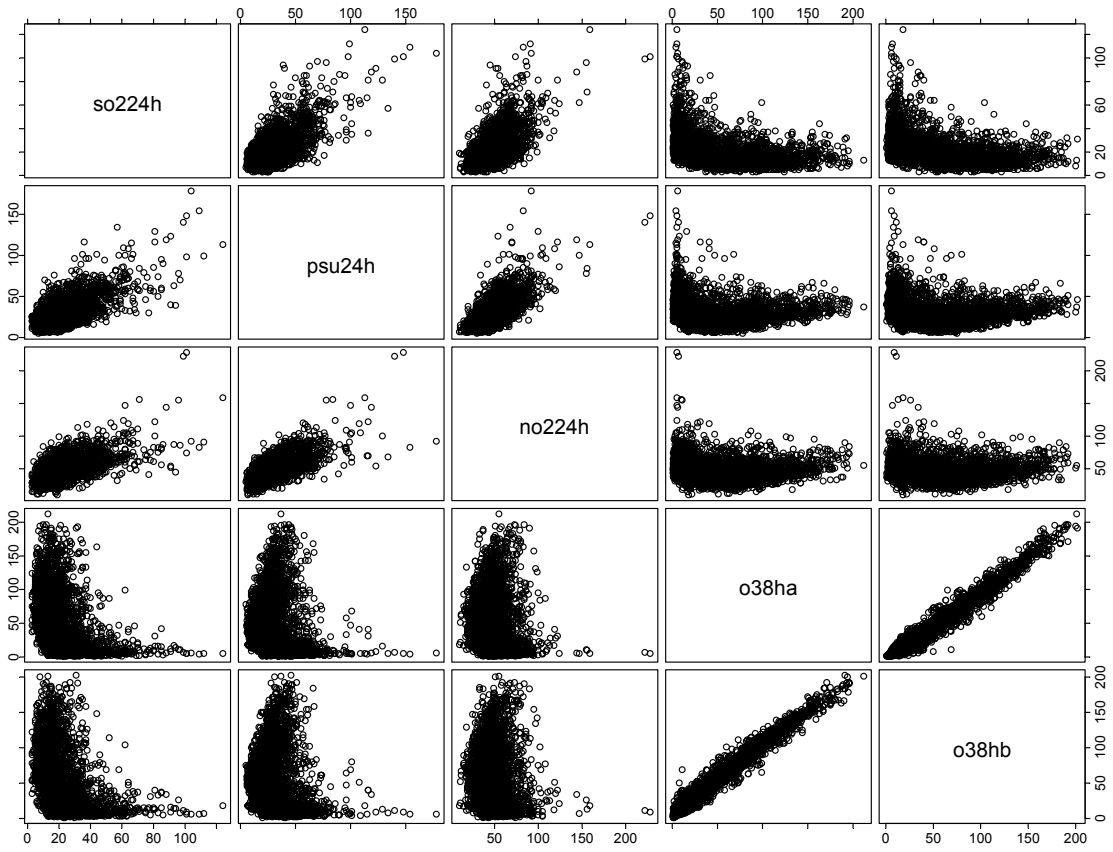
`pairs(data)`

Ex : `pairs(morta[c(43,49,55,61,67)])`

c(43,49,…) représente les colonnes 43, 49, etc. du *data.frame*.

Le résultat est visible sur la figure 71.

Figure 71. Matrice de graphes



Il est aussi possible de créer la matrice de figures précédente avec des courbes de lissage.

Ex :

```
pairs(ozone.data, panel=function(x,y) { points(x,y);lines (lowess (x,y))} )
```

On peut rajouter des courbes de lissage colorées (ici le lissage *supsmu*, par exemple car il tolère des valeurs manquantes).

Ex :

```
pairs(morta[c(43,49,55,61,67)],panel=function(x,y) { points(x,y);
lines(supsmu(x,y),col=2)} )
```

Graphe d'une variable *versus* une autre selon les niveaux d'une troisième

Ce type de graphe peut aider à détecter une interaction ;

La fonction correspondante est « `coplot` »

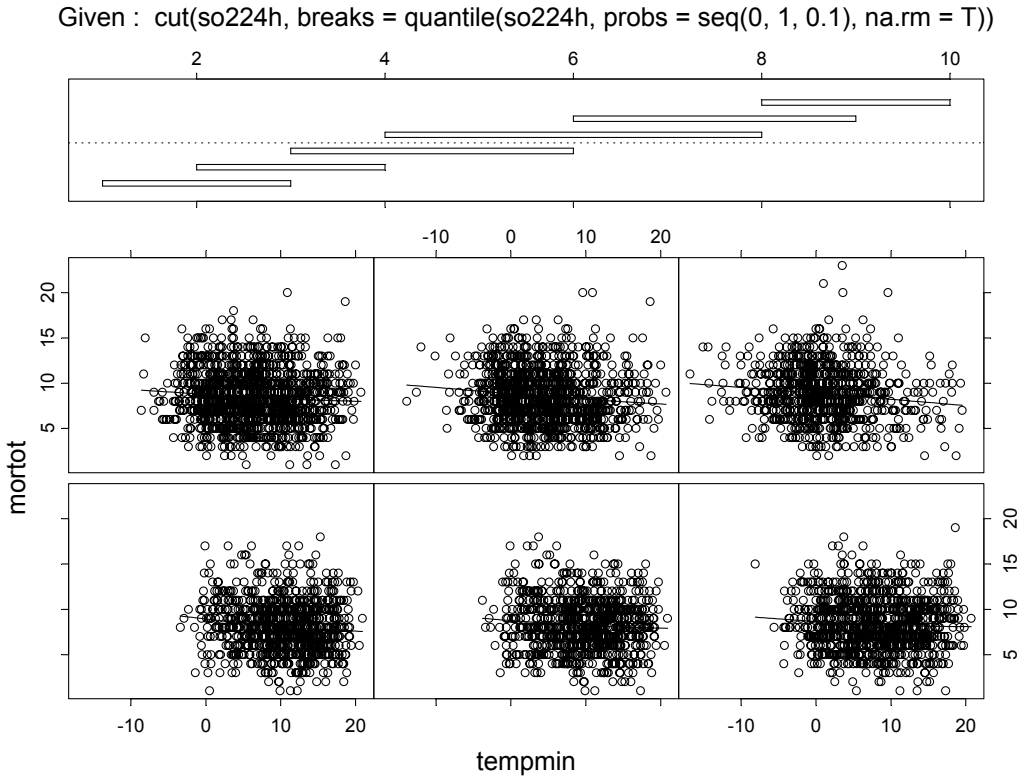
Ex :

```
coplot(mortot~tempmin|cut(so224h,breaks=quantile(so224h,probs=seq(0,1,.1),na.rm=T)),data=morta,panel=function(x,y)panel.smooth(x,y,span=1))
```

Cette fonction représente la variable `mortot` *versus* la variable `tempmin` selon les niveaux de la variable `so224h`, déterminés par les découpages 0, 0,1, 0,2, …1.

La figure obtenue est aussi une matrice (figure 72).

Figure 72. Relation entre deux variables selon les niveaux d'une troisième



Autres types de graphiques

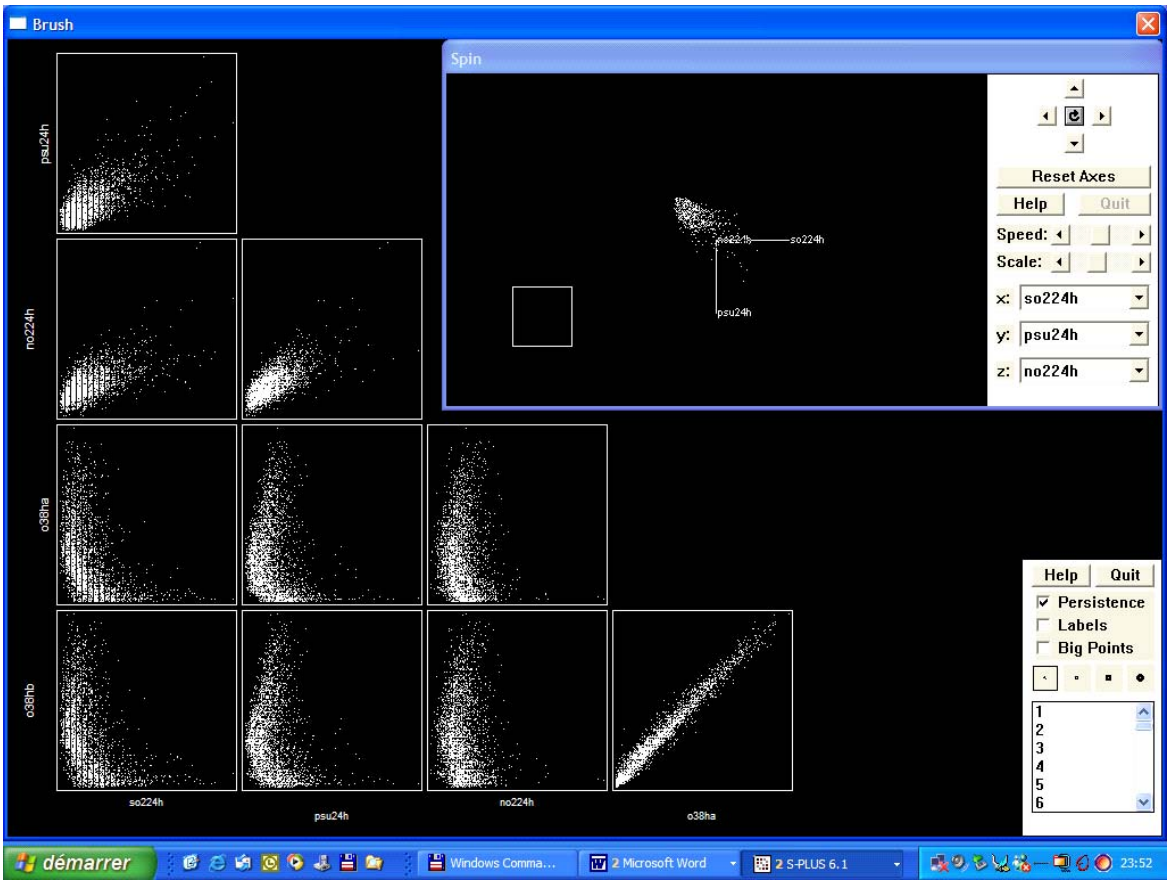
De nombreuses commandes permettent de réaliser et de travailler les graphiques dans S-PLUS. Nous ne les détaillerons pas mais les citerons. On pourra se reporter avec profit à l'aide de S-PLUS pour les explorer plus avant.

La commande « `panel.smooth` » rajoute des courbes de lissage (loess) à une "matrice" de graphe (pairs ou coplot), la commande « `brush` » place plusieurs graphes sur la même page (un peu comme « `pairs` »") avec des boutons permettant de modifier l'orientation des repères. Voici un exemple de commande :

```
brush(morta[c(43,49,55,61,67)])
```

Le résultat (figure 73) :

Figure 73. Matrice de graphes obtenue avec la commande "*brush*".



Écrire des marques sur un axe

L'écriture générale est

```
axis(numéro.axe,at=c(...),labels=T,ticks=T,distn=NULL,
line=0,pos=...,outer=F)
```

« `numéro.axe` » est 1 pour l'axe des abscisses, 2 pour l'axe des ordonnées, « `at=c(…)` » correspond aux valeurs des abscisses, « `labels=T` » veut dire que l'écriture des valeurs est autorisée, « `ticks=T` » autorise l'insertion des marques (les traits) sur l'axe. Les autres instructions sont possibles mais pas obligatoires (on se référera à l'aide pour avoir de plus amples renseignements)

Ex :

```
axis(1,at=c(1975:1996),labels=c("1975","1976","1977","1978","1979","1980","
1981","1982","1983","1984","1985","1986","1987","1988","1989","1990","1991"
,"1992","1993","1994","1995","1996"),ticks=T)
```

Donner des limites (supérieure et inférieure) aux axes

L'instruction est :

```
plot(...,xlim=c(x1,x2),ylim=c(y1,y2))
```

Formats des dates

Différentes instructions sont utilisables. Voici une commande qui se révèle utile dans certains calculs, en ce qu'elle modifie la présentation de la date.

Ex 1 :

L'instruction

```
date.lettres_dates("27/12/1997",format="d/m/y",out="day month year")
```

Donne

```
 [1] 27 December 1997
```

Ex 2 :

L'instruction

```
date.chiffres_dates("27/12/1997",format="d/m/y",out="m/y/d")
```

Donne

```
 [1] 12/97/27
```

Exporter un graphe

Il est souvent utile d'exporter un graphe et de l'importer dans Word (pour une publication, par exemple).

Pour exporter le graphe

Dans S-PLUS :

- Il faut sélectionner la fenêtre graphique en cours (le fichier) puis la page correspondant au graphe que l'on veut exporter.
- Puis sélectionner Menu **Files**, **Menu Export Graph…**
- La fenêtre **Export Graph** s'ouvre (figure 74)
- Choisir le nom du fichier et le répertoire et choisir le format Windows Metafile (extension `.WMF`).

Pour importer le graphe dans Word

Menu **Insertion**

Menu **Image**

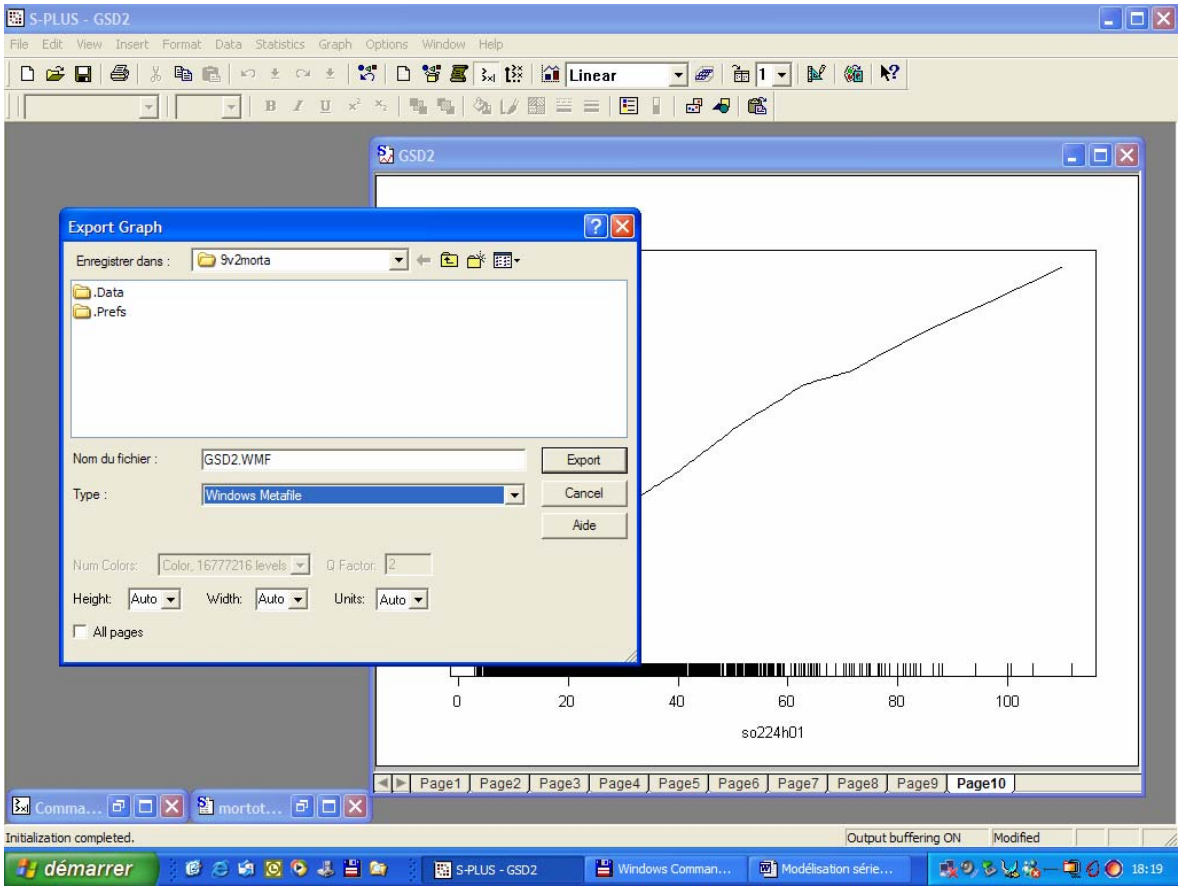
Menu **A partir d'un fichier**

La fenêtre **Insérer une image s'ouvre**

Sélectionner le fichier image exporté sous S-PLUS (fichier `.WMF`).

Remarque. Dans « R », la manipulation est quasiment identique.

Figure 74. Exportation d'un graphique sous S-PLUS



5.2.5. Travailler avec les fichiers textes S-PLUS

Écrire des commandes les unes à la suite des autres est roboratif. Heureusement, nous disposons de deux types de fichiers qui rendent de grands services. Un vrai fichier de commande, appelé « *Script File* » et un fichier texte (`.rtf`, en fait), appelé « *Report File* ».

Script File

Il est accessible par

Menu **File** - **New** ou Icône **New**

Puis Touche **Enter**

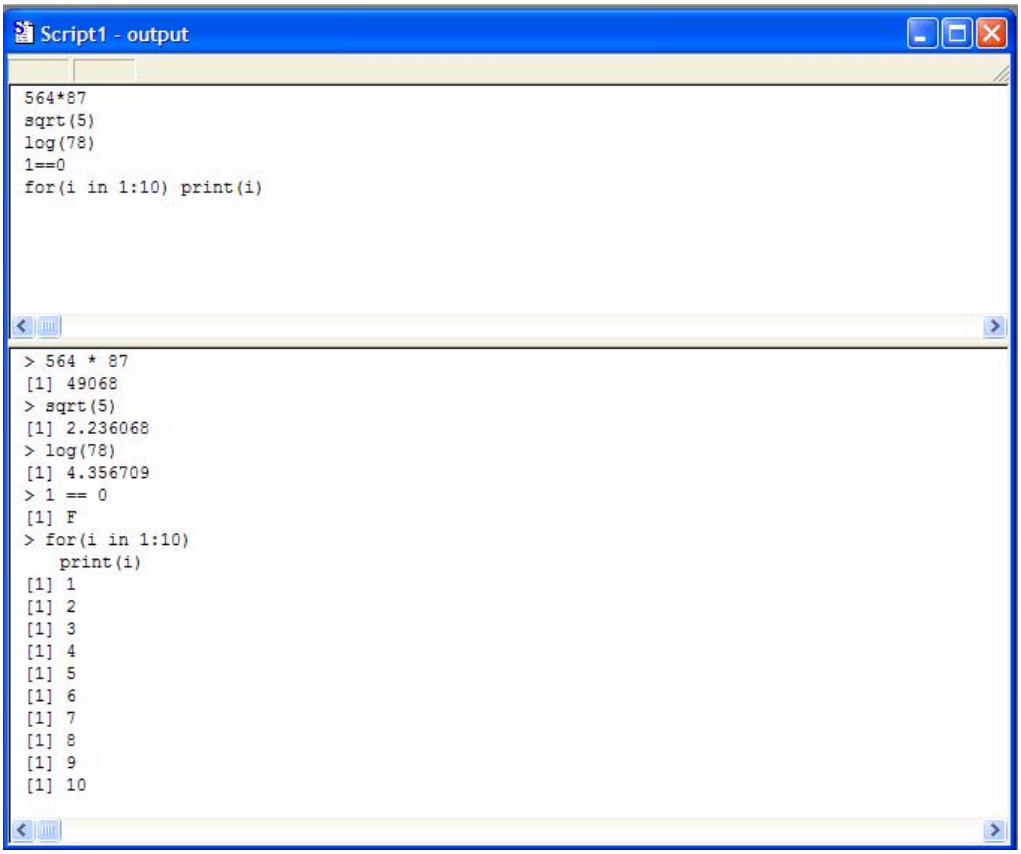
La fenêtre **New** s'ouvre

On choisit **Script File**

La fenêtre *Script File* s'ouvre (figure 75).

Figure 75. Fenêtre "Script File"

g



Cette fenêtre est composée de deux parties. Une fenêtre supérieure où l'on écrit l'instruction ou la série d'instructions devant être réalisées. Une fenêtre inférieure où apparaissent les résultats des commandes et opérations. Les opérations sont effectuées quand on clique sur l'icône de lancement

de commandes qui se présente sous forme d'un triangle du côté gauche de la barre d'outils « *Script* » (figure 76).

Figure 76. Bouton permettant d'effectuer les commandes "*Script*"

Report File

Il est accessible, comme *Scrip File* par

Menu **File** - **New** ou Icône **New**

Puis Touche **Enter**

La fenêtre **New** s'ouvre

On choisit **Report File**

La fenêtre *Report File* s'ouvre ()

Ce fichier texte « riche » (`.rtf`) peut servir à écrire des séries de commandes et les effectuer en réalisant un « copier-coller » dans la fenêtre de commandes. Il sert également à stocker une bibliothèque de commandes (« *Script File* » aussi). Il sert enfin à réaliser des sorties de résultats.

Pour l'exécution des commandes, il fait peut-être un peu double-emploi avec le *Script File* mais il permet d'améliorer la présentation (« mise en forme des caractères ») et peut contenir des graphiques S-PLUS sans passer par l'exportation/importation de graphes.

5.3. Modélisation : écriture d'un modèle

Nous présentons, ici, quelques notions concernant l'écriture des modèles sans entrer dans le détail puisque ceci sera vu plus loin. En particulier, nous ne verrons pas dans cette partie les commandes et instructions nécessaires à l'analyse et à la construction du modèle.

5.3.1. Principe de l'écriture du modèle

Écriture générale

L'écriture du modèle est, en général du type :

`nom.modèle_type.modèle(`**`écriture.du.modèle, loi.de.probabilité, données,conditions`**`)`

Plus précisément :

```
nom.modèle_type.modèle(variable.expliquée ~
fonction1(variable.explicative1,paramètres.fonction1)+
fonction2(variable.explicative2, paramètres.fonction2)+ … +
fonctionN(variable.explicativeN,paramètres.fonctionN),family=famille.lois,d
ata=nom.fichier.données,na=na.omit,subset=conditions.restrictives)
```

Ex :

```
mortotso201.inter.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,hummin,.9)+lo(tempmax2,.9)+lo(hu
mmin12,.9)+lo(so224h01,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)
```

Variables

Les variables expliquées et explicatives sont de tous types : variables réelles, catégorielles, etc. Elles se présentent sous forme non transformée ou sous forme d'une fonction (logarithme, polynomiale, exponentielle et autres fonctions paramétriques ou non paramétriques).

Nature du modèle

On précise la nature du modèle (`lm`, `glm`, `gam`, etc.), avant d'écrire les variables.

Familles

La loi de distribution ou la famille de lois figure également. Ce peut-être : la loi binomiale (`binomial`), la loi normale (`gaussian`) , la loi gamma (`Gamma`) , la loi normale inverse (`inverse.gaussian`) , la loi de Poisson (`poisson`) , la famille de lois dépendant d'une quasi-vraisemblance (`quasi`).

5.3.2. Détails de l'écriture

Fonctions de lissage

Nous avons vu que les variables peuvent apparaître sous une forme non transformée ou sous forme d'une fonction. Dans ce dernier cas, un ensemble de fonctions de lissage peut être utilisé. Ces fonctions vues plus haut sont les fonctions *splines* et les fonctions *loess*, essentiellement.

Les premières s'écrivent :

`s(variable,degré)`

Les secondes :

`lo(variable,span=largeur.fenêtre)`

Restriction de la modélisation à une partie des données

Lorsqu'on ne veut prendre qu'une partie des données, on peut utiliser :

`subset=série.de.conditions`

Ex :

```
gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(tempmin,0.9)+lo(hummin,0
.9)+lo(tempmax123,0.9)+lo(o38hb,0.9),family=quasi(log,mu),
data=morta,subset=summer&mortot<16,na.action=na.omit)
```

Il est possible aussi d'utiliser :

`data[condition.sur.data]`

Il est possible de combiner les méthodes

Ex :

Ici, on utilise les deux méthodes

```
toux.gam_gam(rhino~lo(trend,.3)+ ...
lo(so224h1),family=quasi(log,mu),data=ramses[ramses$nbmed>0,],subset=(trend
>100&trend!=418),na=na.omit)
```

Ex :

```
toux.gam_gam(rhino~lo(trend,.3)+                                          ...
lo(o38h),family=quasi(log,mu),data=ramses[ramses$trend[4:1754],],
subset=(summer),na=na.omit)
```

Methode « *stepwise* »
L'ajustement d'un modèle selon la méthode du « stepwise » (pas-à-pas) se fait par l'écriture suivante (cette écriture peut être complexifiée ; pour ce faire voir l'aide en ligne) :

`step(nom.modèle,direction,trace=T,)`

`nom.modèle` est le nom d'un modèle déjà construit ;

`direction` est le sens de l'ajustement pas-à-pas : ce peut-être « `forward` », « `backward` » ou « `both`», selon que l'on décide d'entrer les variables l'une après l'autre ou de les inclure toutes d'emblée et les retirer au fur et à mesure ou laisser au modèle le choix. L'option par défaut est « `backward` ».

« `trace=T` » fait que les résultats intermédiaires de l'ajustement apparaîssent à l'écran. Ceci est intéressant pour contrôler la progression de la modélisation pas-à-pas.

Ex : `ramses.step_step(ramses.nbmed)`

Modélisation générale avec *offset*

La commande « `offset` » sert à spécifier un terme qui intervient dans le modèle mais qu'on ne veut pas tester. Par exemple, lorsqu'on veut exprimer l'incidence d'une pathologie et que le modèle (loi de probabilité, fonction de lien et prédicteur) attribue la loi de probabilité au nombre de cas incidents mais écrit l'incidence (et non le nombre de cas incidents) comme fonction des variables explicatives, il faut introduire le dénominateur de l'incidence (nombre de personnes-années) sous la forme d'un « offset » :

Ainsi, si i est l'incidence, k, le nombre de cas incidents, m, le nombre de personnes-années :

$$i = \frac{k}{m}$$

Si , de plus k suit une loi de Poisson :

$$k \sim P(\lambda)$$

Et si l'espérance de k est telle que :

$$\ln\left(\frac{E(k)}{m}\right) = g(x_1, x_2, ..., x_n)$$

Alors, il faut introduire offset(m) et non m dans le modèle :

`nom.modèle_nature.modèle(k~offset(ln(m))+g(`x_1`, `x_2`, …, `x_n`))`

Ex : `rhino.gam_gam(rhino~offset(log(med))+lo(trend,.3)+dowf+vacances+ j.feries+lo(tempmin,.7)+lo(hummin,.7)+lo(grip,.7)+lo(urti,.7)+lo(so224h1),f amily=quasi(log,mu),data=ramses,na=na.omit)`

Utilisation de la commande « `update` »

L'instruction « `update` » permet de modifier un modèle déjà construit sans le réécrire totalement :

`modèle2_update(modèle1, .~. -ancienne.expression + nouvelle.expression)`

Ex :

Premier modèle

```
rh.gam1_gam(rhino~offset(log(nbmedof))+lo(trend,.3)+dowf+j.feries+
vacances+lo(grip,.7)
+lo(gram1,.7)+lo(urti1,.7),family=quasi(log,mu),data=ramses, na=na.omit)
```

Deuxième modèle

`rh.gam2_update(rh.gam1, .~. – lo(urti1,.7) + lo(gram3,.7))`

Dans ce modèle, on a remplacé la variable urti1 par la variable gram3

On peut, bien sur, ne faire qu'ôter une ou plusieurs variables (en se limitant à l'expression qui suit le « - ») ou ne faire qu'ajouter une ou plusieurs variables (en se limitant à l'expression qui suit le « + »).

Les instructions « `fitted` » et « `predict` »

Ces deux instructions permettent d'estimer des valeurs prédites par le modèle.

L'instruction « `fitted` » fournit les valeurs prédites de la **variable expliquée** (pour être complet, rajoutons que les coefficients estimés sont obtenus grâce à l'instruction « `coeff` » et les résidus grâce à l'instruction « `resid` ».

L'instruction « `predict` » fournit les valeurs du **prédicteur** prévues par le modèle.

Par conséquent, si le prédicteur est linéaire (modèle linéaire), les deux instructions donnent les valeurs prédites de la variable expliquée. Si, par contre, comme le GLM ou le GAM le permettent, la fonction de lien n'est pas linéaire (logarithmique, etc.), alors les deux instructions ne donnent pas le même résultat : « `fitted` » donne les valeurs prédites (variable expliquée) et « `predict` » fournit les valeurs prédites sous forme de fonction inverse de la fonction de lien.

Ex :

```
is7584.glm_glm(is~offset(log(po)/100000)+agef+p7584+p7584.2,
data=isiv,family=poisson,subset=per<=10)
```

La fonction de lien est la fonction logarithme népérien (car la loi est une loi de Poisson).

`fitted(is7584.glm)` donne les valeurs prédites de `is` ;

`predict(is7584.glm)` donne les valeurs prédites de exp(`is`).

De plus si on écrit :

`predict(nom.modèle,nom.data.frame)`

Avec `nom.data.frame` contenant les valeurs futures des variables expliquées, alors les valeurs sont prédites pour le passé et le futur.

Ex :

```
predict.glm(object, newdata, type = c("link", "response", "terms"), se.fit
= F, terms = labels(object), ...)
```

5.3.3. Problèmes

Un problème est apparu, lié à l'utilisation du logiciel S-Plus [47] : une combinaison de deux algorithmes est utilisée pour les estimations : *local scoring algorithm* et *backfitting algorithm* (§ 3.2.3.1.2.). Or, selon le nombre d'itérations programmé, le logiciel ne converge pas vers les mêmes estimations des valeurs centrales et des intervalles de confiance des paramètres (en raison des difficultés à la convergence de l'algorithme vers son optimum). La responsabilité de ce problème a été révélé lors de la ré-analyse des données de l'étude NMMAPS [48,49]. Ici il est donc prudent d'être plus exigeant quant aux paramètres de convergence des deux algorithmes (augmentation du nombre d'itérations et de la précision de la convergence).

De plus, en raison de la non prise en compte de phénomènes de *concurvité* (équivalent non paramétrique de la colinéarité) entre les variables explicatives, les modèles GAM (fonction *loess*, *splines* de lissage) sous-estiment l'intervalle de confiance du coefficient du polluant et, par là, celui du RR.

6. Démarche de la modélisation

Le déroulement de la modélisation sera décrit et expliqué à chaque étape en indiquant les instructions adéquates dans le logiciel S-PLUS (encadrés) et en montrant les « sorties » du logiciel sous forme de résultats numériques (estimation des paramètres, tests, etc.) et sous forme de graphes.

Les instructions S-PLUS sont, selon le cas, génériques (le nom donné aux variables est le plus général possible) ou plus spécifiques (vraies commandes utilisées afin de donner des résultats concrets).

Rappelons que l'observation graphe de la série temporelle doit être la toute première étape de l'analyse car elle permet de visualiser le « comportement » de la série et d'orienter la démarche exploratoire.

Rappelons aussi que l'exemple qui figure dans ce chapitre (l'analyse des relations entre un polluant et un indicateur sanitaire) peut être extrapolé à d'autres thématiques, sans difficulté.

6.1. Nature des variables introduites dans le modèle

Le **modèle initial** contient les variables *nombre de cas incidents* (en fait, l'espérance du nombre de cas incident sous forme d'un logarithme népérien), *tendance*, *jour de la semaine*, *jours fériés*, *périodes de vacances scolaires*, *épidémies de grippe*, (*comptes polliniques*,) *température minimale journalière*, *humidité minimale journalière*, *température maximale journalière* et *un indicateur de pollution.*

Les différentes variables dont la nature est présentée dans le tableau 3 apparaissent pour la plupart sans **décalage temporel** dans le modèle initial [(49)]. La *température maximale* journalière est cependant affectée d'un retard de 1. En effet, les températures maximales reflètent plutôt l'exposition hivernale diurne car, en cette saison, les personnes sont protégées des températures minimales nocturnes (durant la nuit, les fenêtres sont généralement fermées) et, par ailleurs, l'effet des faibles températures est généralement retardé. En revanche, les températures minimales sont plutôt le reflet de l'exposition nocturne estivale, saison pendant laquelle les personnes se protègent le jour des plus fortes températures mais ouvrent leurs fenêtres la nuit. Or, l'effet des températures élevées est généralement immédiat et la variable *température minimale* n'est pas décalée. Le décalage imposé trouve aussi sa justification dans le fait que la température maximale est corrélée à la température minimale du même jour.

Les variables *tendance*, *grippe*, *température minimale journalière*, *humidité minimale journalière*, *pollen* et les *vacances d'été*, sont introduites sous la **forme de fonctions** *loess* ou *splines* (tableau 3). Le choix de la largeur de la fenêtre de lissage permet une prise en compte plus ou moins fine des variations temporelles de la variable : une fenêtre étroite prend en compte les variations à court terme, une fenêtre plus large filtre celles-ci pour ne garder dans le modèle que les variations à long terme. Dans le modèle initial, selon les variables, le choix de la largeur de la fenêtre est fondé sur la base d'une durée *considérée comme pertinente* (six mois pour la tendance, par exemple) ou d'une fraction de la durée totale d'observation (70 % de la durée de l'étude, par exemple, pour les autres variables lissées). La fenêtre de la tendance correspond ainsi à 6 mois, durée permettant en général une bonne prise en compte de la saisonnalité : il faut, en effet, capter l'effet global de ces variables sans « récupérer trop de saisonnalité ». L'AIC (§ 3.2.3.2) pourrait être utilisé pour choisir la taille de la fenêtre mais celui-ci a tendance à privilégier les fenêtres larges.

L'*indicateur de pollution* apparaît sous forme de la moyenne des niveaux du jour même et du jour précédent (moyenne 0-1 jours) ou sans décalage. Alors que les autres polluants figurent sous la forme d'une variable simple, l'ozone dont les concentrations sont très contrastées selon la saison et selon l'heure de la journée, apparaît sous forme d'une interaction avec une variable *été* ; cette dernière vaut

[49] Les transformations et les décalages appliqués aux variables apparaissant dans ce tableau ne sont qu'indicatifs. Ces propositions sont fondées sur l'expérience de la modélisation des relations entre la pollution atmosphérique et la santé et sur la consultation de la littérature.

1 en été (avril à septembre) et 0 en hiver. Au cours de l'hiver, le coefficient de l'ozone et son écart-type sont respectivement égaux au coefficient et à l'écart-type de la variable *ozone* seul. En été, le coefficient de l'ozone est la somme du coefficient de la variable *ozone* et de celui de l'interaction *été-ozone*. Son écart type est égal à :

$$\sqrt{(\mathrm{Var}(\mathrm{coef}_{o_3}) + \mathrm{Var}(\mathrm{coef}_{o_3 * \text{été}}) + 2\mathrm{Cov}(\mathrm{coef}_{o_3}, \mathrm{coef}_{o_3 * \text{été}}))}$$

La variable *polluant* est introduite

- Soit sans transformation dans le modèle ;
- Soit sous forme de fonction *loess* ou *spline*.

Pour tenir compte de la surdispersion de la variable sanitaire, le modèle doit permettre au paramètre de dispersion de prendre des valeurs différentes de 1.

Tableau 3. Variables du modèle initial, nature, transformation et décalage.

Variable	Nature	Transformation	Décalage (jours)
variable sanitaire[(a)]	discrète	logarithme népérien	-
tendance	discrète	*loess* / *spline*	-
jour de la semaine	qualitative	aucune / *spline*	0
jours fériés	binaire	aucune	0
vacances scolaires	binaire / discrète	aucune / *loess* / *spline*	0
grippe	discrète	*loess* / *spline*	0
pollens	discrète	*loess* / *spline*	0
température minimale	continue	*loess* / *spline*	0
température maximale	continue	*loess* / *spline*	1
humidité	continue	*loess* / *spline*	0
polluant	continue	aucune / *loess* / *spline*	0-1

[(a)] En fait c'est l'espérance de la variable nombre de cas incident qui apparaît dans le modèle.

Le modèle initial se présente de la façon suivante (respectivement, pour la mortalité et les hospitalisations) :

```
morta.gam _ gam (indic.sanit ~ lo(tendance,183/(nb.jours.total)) +
j.semaine + j.féries + vacances + lo (grippe.décalage.0,.7) +
lo(tempé.min.décalage.0,.7) + lo(tempé.min.décalage.0,.7) +
lo(tempé.max.décalage.1,.7) + lo(polluant.décalage0,.7) , family =
quasi(log,mu) , data=morta , na=na.omit)

morbi.gam _ gam (indic.sanit ~ lo(tendance,183/(nb.jours.total)) +
j.semaine + j.féries + vacances + lo (grippe.décalage.0,.7) +
lo(tempé.min.décalage.0,.7) + lo(tempé.min.décalage.0,.7) +
lo(tempé.max.décalage.1,.7) + lo(pollen1_décalage0,.7) +
lo(pollen2_décalage01,.7) + . . . + lo(pollenN_décalage0,.7)+
lo(polluant.décalage0,.7) + lo(polluant.décalage0,.7) , family =
quasi(log,mu) , data=morbi , na=na.omit)
```

Ou, si le polluant est introduit sous la forme d'une relation linéaire 0-1 jours (ce qui est préférable) :

```
morta.gam _ gam (indic.sanit ~ lo(tendance,183/(nb.jours.total)) +
j.semaine + j.féries + vacances + lo (grippe.décalage.0,.7) +
lo(tempé.min.décalage.0,.7) + lo(tempé.min.décalage.0,.7) +
lo(tempé.max.décalage.1,.7) + polluant.décalage01 , family = quasi(log,mu)
, data=morta , na=na.omit)


morbi.gam _ gam (indic.sanit ~ lo(tendance,183/(nb.jours.total)) +
j.semaine + j.féries + vacances + lo (grippe.décalage.0,.7) +
lo(tempé.min.décalage.0,.7) + lo(tempé.min.décalage.0,.7) +
lo(tempé.max.décalage.1,.7) + lo(pollen1_décalage0,.7) +
lo(pollen2_décalage0,.7) + . . . + lo(pollenN_décalage0,.7)+
lo(polluant.décalage0,.7) + polluant.décalage01 , family = quasi(log,mu) ,
data=morbi , na=na.omit)
```

Remarque. La variable *vacances* (congés scolaires) sera utilisée pour les hospitalisations. Pour la mortalité, après contrôle de la saison, on ne devrait pas en avoir besoin. Si l'on tient à l'inclure dans le modèle de la mortalité, il faut calculer le RR associé à cette variable et la garder si ce RR est significatif (voir plus loin pour le calcul du RR et son test).

6.2. Outils de l'analyse

Un ensemble d'outils statistiques et graphiques aide à la détermination du modèle le mieux ajusté aux données et aux relations entre les données. Ces outils sont les suivants :

- Autocorrélation partielle des résidus (PACF)
- Observation du graphe des résidus
- Comparaison du graphe de la série prédite par le modèle et du graphe de la série initiale
- Effet partiel de chaque facteur sur la variable sanitaire
- Critère d'Akaike
- Paramètre de dispersion
- Les « résumés de modèles » (*summaries*)

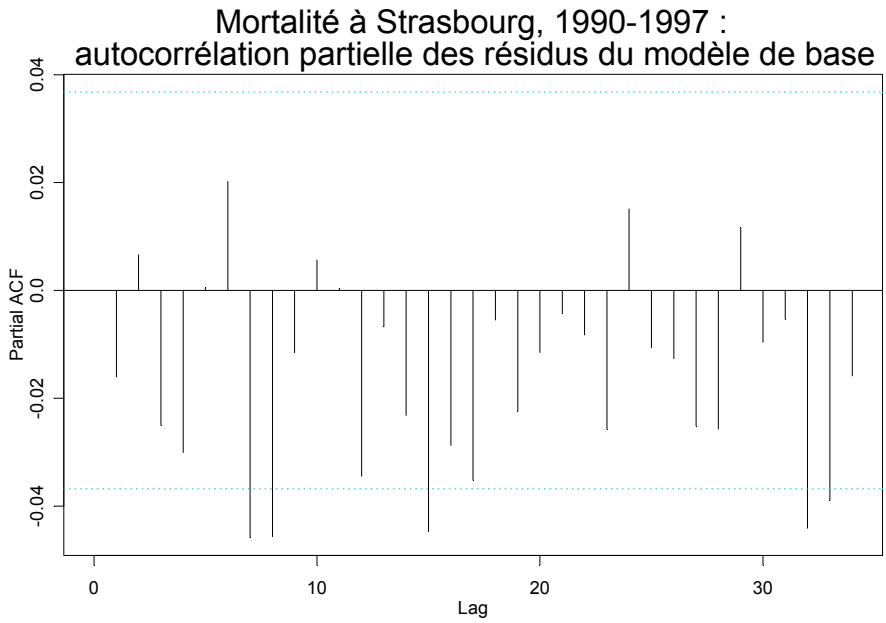
Ces outils sont détaillés dans les paragraphes ci-dessous.

6.2.1. Autocorrélation partielle des résidus (PACF)

Au cours de la progression de l'analyse, la prise en compte des variations temporelles et des facteurs de confusion permet de réduire l'autocorrélation dans les résidus du modèle [39] (figure 77). Pour la **mortalité**, la fonction d'autocorrélation doit être, en fin de modélisation, celle d'un bruit blanc quel que soit le retard (les résidus du modèle – dont l'ensemble forme un processus également – sont idéalement des variables d'espérance nulle, non corrélées et de variances égales) car l'autocorrélation n'est due qu'à des facteurs extrinsèques (météo, saison, niveau de pollution). Pour la **morbidité**, l'autocorrélation, due en plus à des facteurs intrinsèques (fonctionnement de l'hôpital dans le cas des hospitalisations, par exemple), est plus forte et plus difficile à supprimer, aussi il persiste parfois une autocorrélation résiduelle importante pour les premiers retards ; il est cependant nécessaire d'obtenir un bruit blanc au delà des dix premiers retards et, au moins, une réduction de l'autocorrélation sur les premiers retards.

```
sum(acf(resid(mortot.gam),type="p")$acf)
```

Figure 77. Corrélogramme des résidus



Sur ce graphe, il existe des pics d'autocorrélation (négatifs) sur les retards (*lags*) 7, 8, etc.

6.2.2. Observation du graphe des résidus

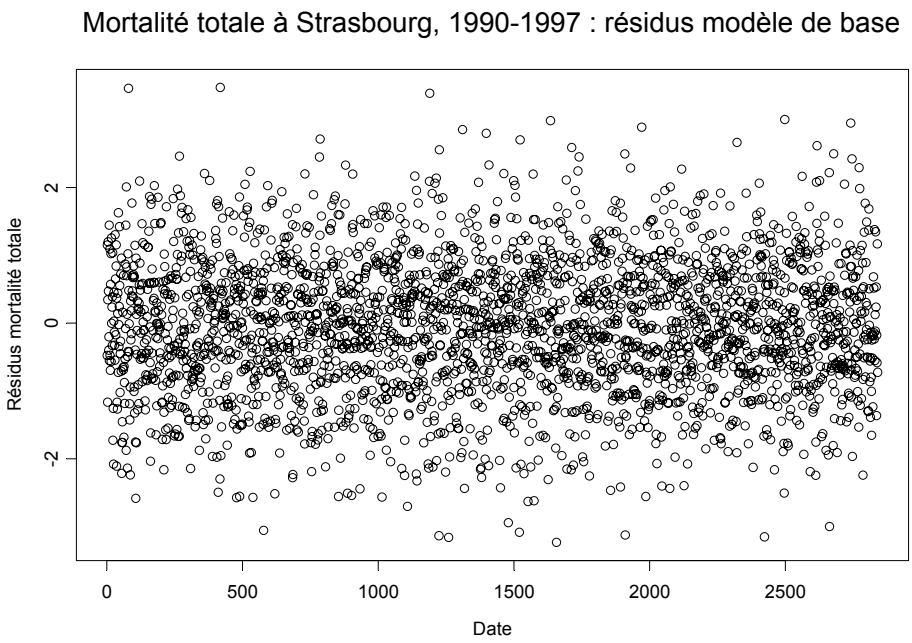
L'observation du graphe des résidus (figure 78) permet de vérifier si la tendance et la saisonnalité disparaissent au fur et à mesure de la construction du modèle.

```
plot(resid(mortot.gam))
```

```
Ex 1
plot(resid(mortot.gam))
lines(supsmu(morta$trend,resid(mortot.gam),span=.4),col=2)
abline(c(0,0),col=2)

Ex 2
plot(resid(respi014.gam))
lines(supsmu(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$
respi014<6],resid(respi014.gam),span=.1),col=3)
```

Le graphe des résidus *versus* les valeurs prédites peut aussi montrer la structure des résidus.

Figure 78. Résidus du modèle



Ce graphe ne montre pas (plus) de structure au sein des résidus.

6.2.3. Comparaison du graphe de la série prédite par le modèle et du graphe de la série observée

La comparaison du graphe de la série prédite par le modèle et du graphe de la série observée permet de juger de la qualité de la modélisation (*i.e.* la contemporanéité des deux séries) c'est-à-dire de la façon dont sont contrôlées les variations de l'indicateur sanitaire, lors d'un épisode de grippe, d'un événement de courte durée, par exemple ou durant une saison particulière. La comparaison ne se fait pas de façon absolue car les données observées sont beaucoup plus dispersées (ce qui est normal puisque les valeurs prédites en représentent l'espérance). Il faut comparer les tendances des deux courbes.

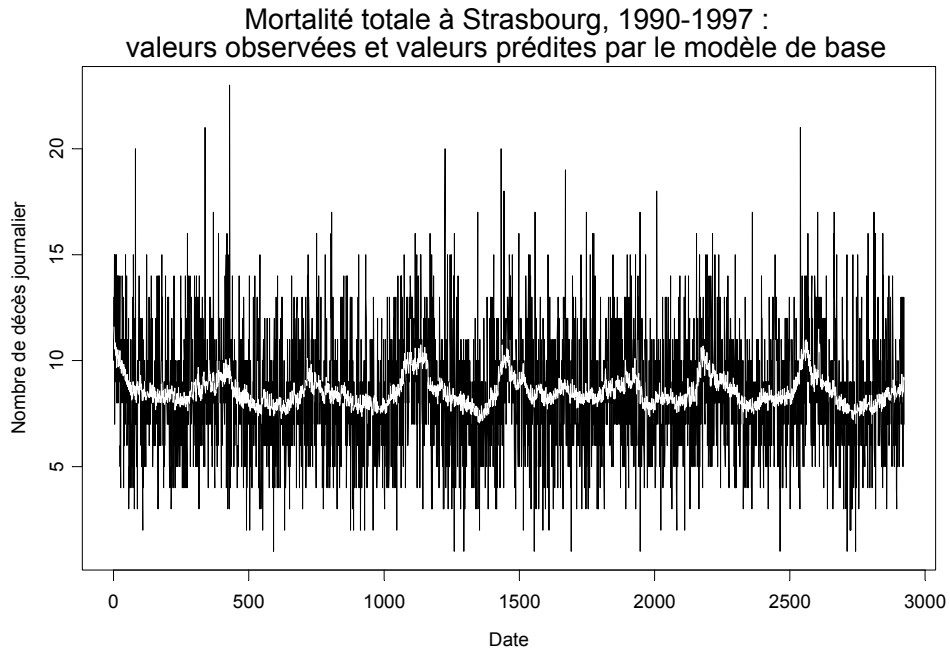
Pratiquement, il faut faire figurer l'indicateur sanitaire (mortalité ou morbidité) et le modèle sur un même graphique. Il existe différentes façons de représenter les deux séries.

6.2.3.1. Courbes superposées

Les deux courbes sont tracées dans le même repère. Elles sont correctement superposées si les deux séries sont proches l'une de l'autre (figure 79).

```
Prédites et observées sur le même graphe

Ex 1 : mortalité
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$gripa7)&!is.na(morta$gripb3)&!is.na(morta$te
   mpmax1)&!is.na(morta$so224h)&morta$mortot<16],fitted(mortot.gam),col=2)

Ex 2 : hospitalisations
plot(morbi$trend,morbi$respi014,type="l")
lines(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$respi01
4<6],fitted(respi014.gam),col=2)

Ex 3 : hospitalisations ; même figure que précédemment mais avec les mêmes
restrictions pour les 2 séries
plot(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$respi014
<6],morbi$respi014[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$respi0
14<6],type="l")
lines(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$respi01
4<6],fitted(respi014.gam),col=2)

Ex 4 : pour s'affranchir de l'effet jour de la semaine et jours fériés, on
peut mettre en jeu les instructions suivantes :
x1_predict(respi014.gam,type="terms")
x1.const_attr(x1,"constant")
plot(morbi$date.study[!is.na(morbi$so224h)&!is.na(morbi$grip)&morbi$respi01
4<12],morbi$car1264[!is.na(morbi$so224h)&!is.na(morbi$grip)&morbi$respi014<
12],col=2)
lines(morbi$date.study[!is.na(morbi$so224h)&!is.na(morbi$grip)&morbi$respi0
14<12],fitted(respi014.gam)/exp(x1[,3]+x1[,4]))
```

Figure 79. Comparaison des valeurs prédites par le modèle et des valeurs observées



Lorsque les series sont lissées, on obtient (figures 80 et 81) :

Ex: Représentation séries observée et prédites, lissées par des fonctions *spline*.

```
observ.liss_smooth.spline(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$car65<16],morbi$car65[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$car65<16])[["y"]]

predit.liss_smooth.spline(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$car65<16],fitted(car65.gam))[["y"]]

plot(observ.liss)

lines(predit.liss,col=2)
```

Figure 80. Comparaison des valeurs prédites par le modèle et des valeurs observées avec lissage

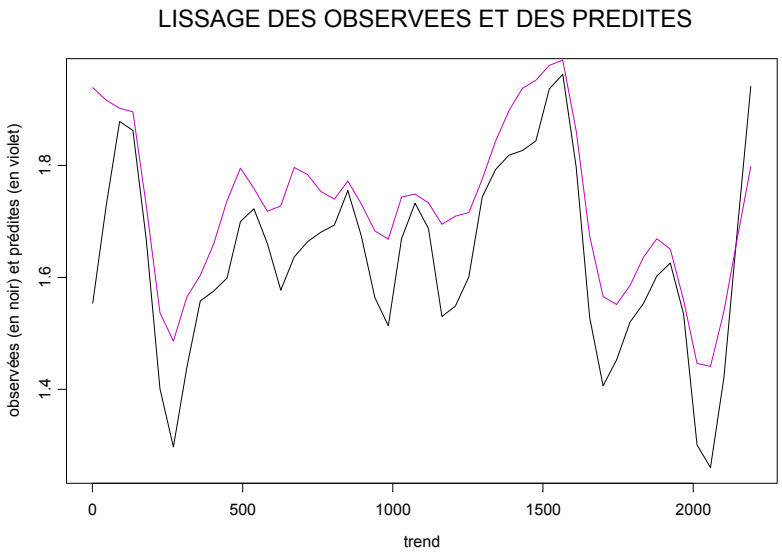
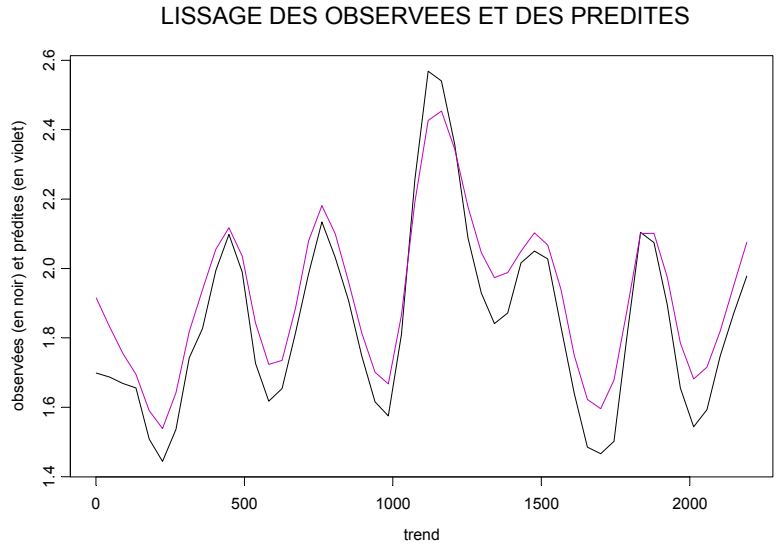


Figure 81. Bonne adéquation des valeurs prédites par le modèle aux valeurs observées



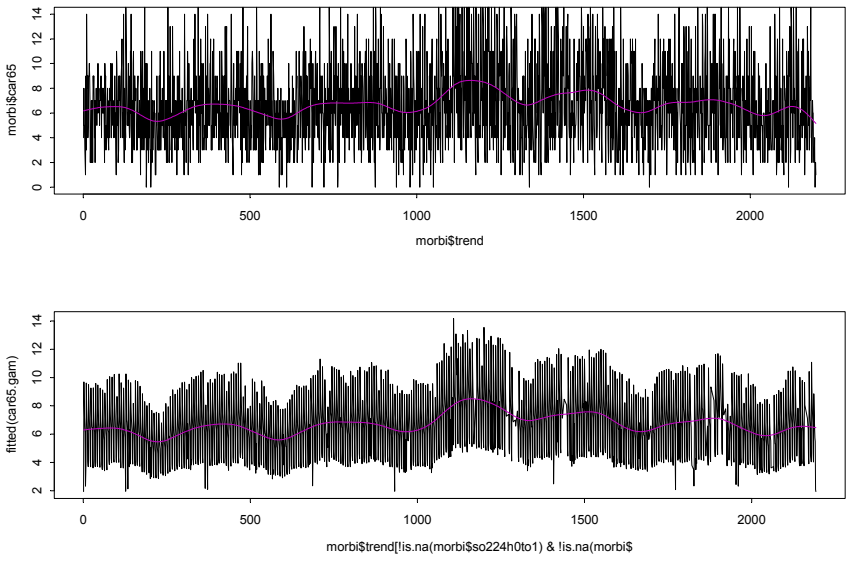
6.2.3.2. Courbes séparées

Les courbes correspondant aux données et aux valeurs prédites sont tracées dans des repères séparés. Il est plus facile, aux fins de comparaison, de rajouter une courbe lissée (*spline* ou *loess*) à chacun des graphes (observées et prédites).

```
Observées et prédites avec lissage sur des graphes séparés 1


Ex 1 : observées avec smooth.spline sur graphe du haut et prédites avec
smooth.pline sur graphe du bas (figure 82)
par(mfrow=c(2,1))
plot(morbi$trend,morbi$car65,type="l",ylim=c(0,14))
lines(smooth.spline(morbi$trend,morbi$car65),col=3)
plot(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)],fitted(car65.
gam),type="l")
lines(smooth.spline(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)
],fitted(car65.gam)),col=3)
par(mfrow=c(1,1))
```

La figure ci-dessous (figure 82) montre les graphes de la série observée (graphe supérieur) avec lissage *spline* (fonction *smooth.spline* de S-PLUS) et de la série des valeurs prédites par le modèle (graphe du bas) avec lissage *spline*.

Figure 82. Observées et prédites avec lissage sur des graphes séparés



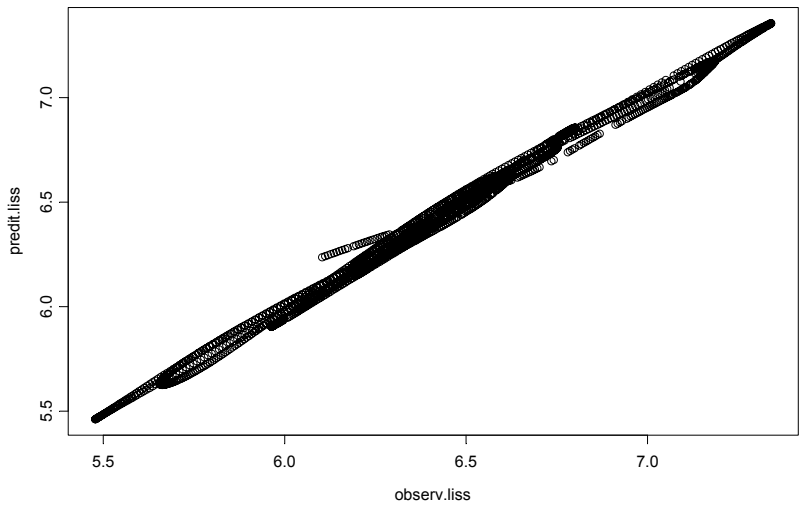
```
Observées et prédites avec lissage sur des graphes séparés 2

Ex 2 : observées avec loess.smooth sur graphe du haut et prédites avec
loess.smooth sur graphe du bas
par(mfrow=c(2,1))
plot(morbi$trend,morbi$respi014,type="l",ylim=c(min(fitted(respi014.gam)),m
ax(fitted(respi014.gam))))
lines(loess.smooth(morbi$trend,morbi$respi014,span=.1),col=3)
plot(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$respi014
<6],fitted(respi014.gam),type="l")
lines(loess.smooth(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&
morbi$respi014<6],fitted(respi014.gam),span=.1),col=3)
par(mfrow=c(1,1))

Ex 3 : même figure que précédemment mais avec mêmes restrictions pour les 2
séries
par(mfrow=c(2,1))
plot(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$respi014
<6],morbi$respi014[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$respi0
14<6],type="l",ylim=c(min(fitted(respi014.gam)),max(fitted(respi014.gam))))
lines(loess.smooth(morbi$trend,morbi$respi014,span=.1),col=3)
plot(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&morbi$respi014
<6],fitted(respi014.gam),type="l")
lines(loess.smooth(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$grip)&
morbi$respi014<6],fitted(respi014.gam),span=.1),col=3)
par(mfrow=c(1,1))
```

6.2.3.3. Représentation de la série prédite versus la série observée

On peut représenter les deux séries lissées (par la fonction *smooth.spline*) l'une par rapport à l'autre (figure 83).

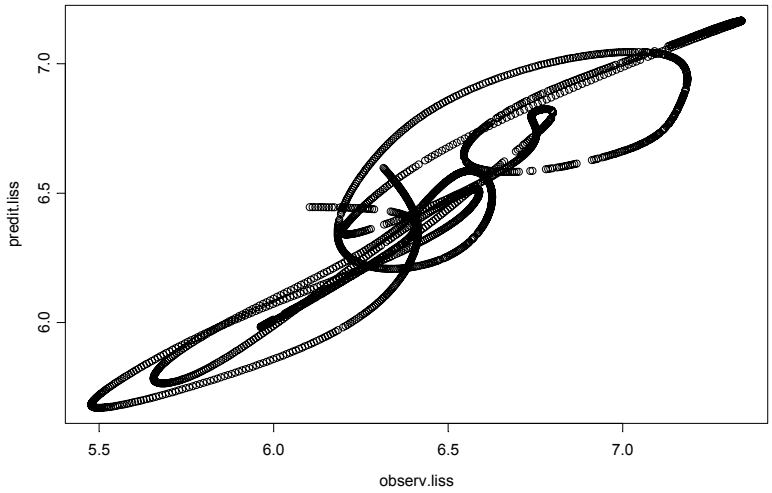
```
plot(observ.liss,predit.liss)
```

Figure 83. Représentation de la série prédite versus la série observée



Das les cas où les deux series (observée et predite par le modèle) ne sont pas en adéquation, on obtient (figure 84).

Figure 84. Mauvaise adéquation de la série prédite à la série observée

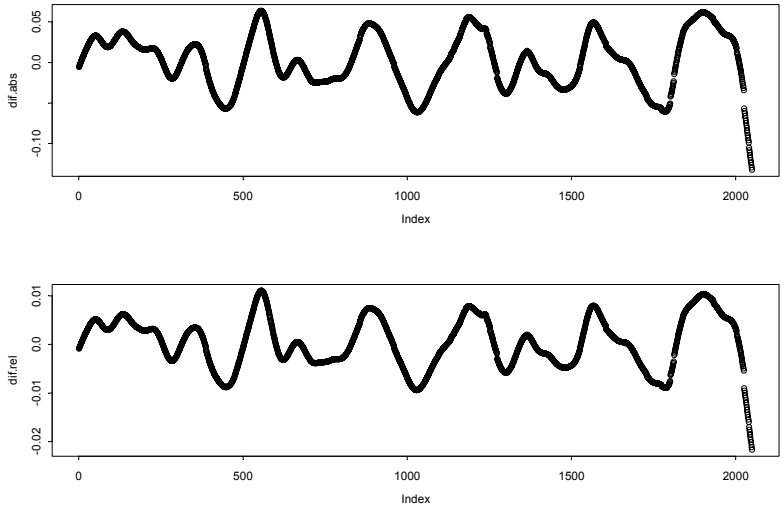


6.2.3.4. Écart entre les courbes lissées des séries observée et prédite

Il est souvent difficile de décider si l'adéquation des deux courbes (séries elles-mêmes ou courbes de lissage) est correcte. On peut alors s'intéresser à la courbe représentative de la différence absolue ou relative des séries observée et prédite lissées.

```
Ex 1 : Représentation des différences absolues et relatives des deux séries
lissées par des fonctions spline.

observ.liss_smooth.spline(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi
$grip)&morbi$car65<16],morbi$car65[!is.na(morbi$so224h0to1)&!is.na(morbi$gr
ip)&morbi$car65<16])[["y"]]

predit.liss_smooth.spline(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi
$grip)&morbi$car65<16],fitted(car65.gam))[["y"]]

dif.abs_ observ.liss-predit.liss

dif.rel_dif.abs/observ.liss

par(mfrow=c(2,1))

plot(dif.abs)

plot(dif.rel)

par(mfrow=c(1,1))
```

Figure 85. Représentation des différences absolue et relative des deux séries lissées par des fonctions *spline*



En haut : différence absolue ; en bas : différence relative.

```
Ex 2 : Représentation des différences absolues et relatives des deux séries
lissées par des fonctions loess


observ.liss_loess.smooth(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$
grip)&morbi$respi014<6],morbi$respi014[!is.na(morbi$so224h0to1)&!is.na(morb
i$grip)&morbi$respi014<6],span=.1)[["y"]]

predit.liss_loess.smooth(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morbi$
grip)&morbi$respi014<6],fitted(respi014.gam),span=.1)[["y"]]

dif.abs_ observ.liss-predit.liss

dif.rel_dif.abs/observ.liss

par(mfrow=c(2,1))

plot(dif.abs,type="l")

plot(dif.rel,type="l")

par(mfrow=c(1,1))
```

```
Ex 3 : Comparaison des lissages des prédites et des mesurées :
représentation des deux courbes sur le même graphe, des différences
absolues et relatives ainsi que des deux séries l'une par rapport à
l'autre.


#Fabrication des objets (lists) représentatifs des observées et des
prédites

observ.smooth_loess.smooth(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morb
i$grip)&morbi$respi014<6],morbi$respi014[!is.na(morbi$so224h0to1)&!is.na(mo
rbi$grip)&morbi$respi014<6],span=.1)

predit.smooth_loess.smooth(morbi$trend[!is.na(morbi$so224h0to1)&!is.na(morb
i$grip)&morbi$respi014<6],fitted(respi014.gam),span=.1)

#Fabrication du tableau (data.frame « obspre » ) contenant les observées,
les prédites et leurs différences  ainsi que la concordance des variations
des obs et des préd (0 si variation dans le même sens, 1 si variations en
sens contraires)

obspre$x_observ.smooth[["x"]]

obspre$dif_observ.smooth[["y"]]-predit.smooth[["y"]]

obspre$obs_observ.smooth[["y"]]

obspre$pre_predit.smooth[["y"]]

obspre$concord_as.integer(ifelse((obspre$obs-
c(NA,obspre$obs[1:length(obspre$obs)-1]))*(obspre$pre-
c(NA,obspre$pre[1:length(obspre$pre)-1]))>0,"0","1"))

#Représentation graphique des observées et des prédites lissées sur le même
schéma

plot(observ.smooth,type="l",main="LISSAGE DES OBSERVEES ET DES
PREDITES",xlab="trend",ylab="observées (en noir) et prédites (en
violet)",lab=c(20,,))

lines(predit.smooth,col=3)

#Fabrication des vecteurs des différences absolue et relative des observées
et des prédites lissées

dif.abs_observ.smooth[["y"]]-predit.smooth[["y"]]

dif.rel_dif.abs/observ.smooth[["y"]]

#Représentation graphique des différences absolue et relative des observées
et des prédites lissées sur deux schémas séparés empilés

par(mfrow=c(2,1))

plot(observ.smooth[["x"]],dif.abs,type="l",main="DIFFERENCES LISSAGES
OBSERVEES-PREDITES",xlab="trend",ylab="Différence absolue")

plot(dif.rel,type="l",xlab="Rang de l'abscisse de lissage",ylab="Différence
relative")

par(mfrow=c(1,1))

#Et graphe représentant les séries lissées l'une par rapport à l'autre

plot(observ.smooth[["y"]],predit.smooth[["y"]],xlim=c(min(observ.smooth[["y
"]],predit.smooth[["y"]]),max(observ.smooth[["y"]],predit.smooth[["y"]])),y
lim=c(min(observ.smooth[["y"]],predit.smooth[["y"]]),max(observ.smooth[["y"
]],predit.smooth[["y"]])),main="PREDITES VERSUS OBSERVEES",sub="(En violet
la droite y=x)",xlab="Observées",ylab="Prédites")

lines(observ.smooth[["y"]],observ.smooth[["y"]],col=3)
```

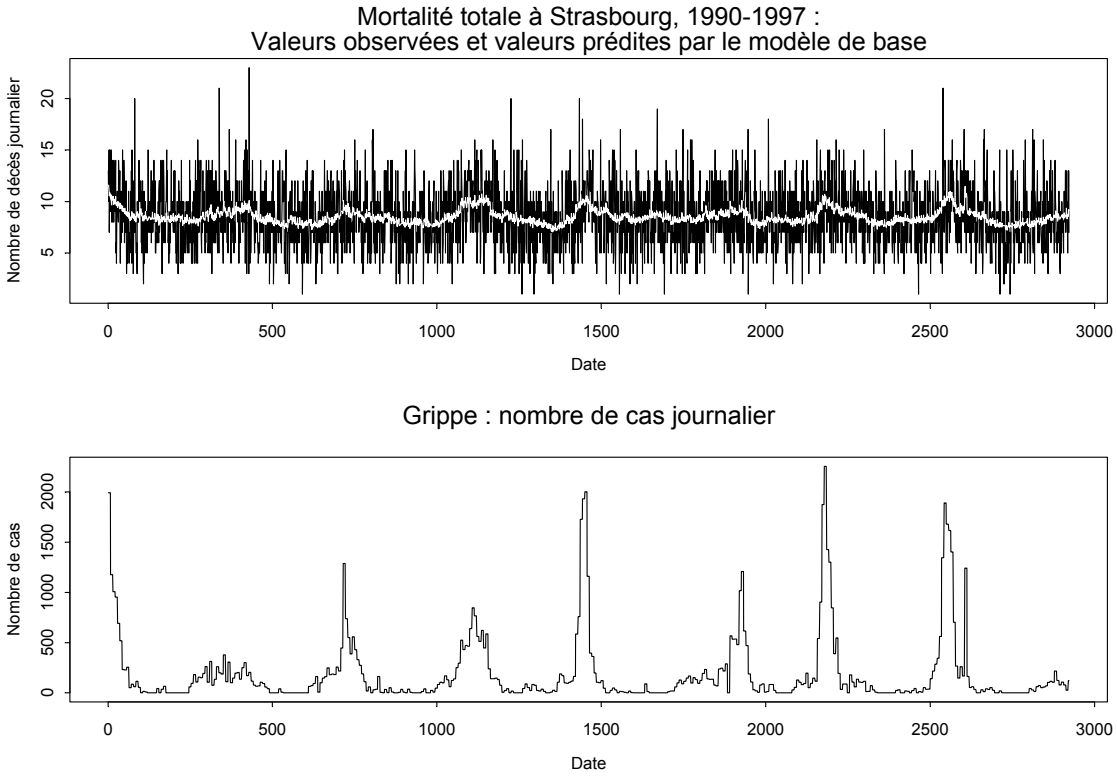
6.2.3.5. Confrontation des séries observée et prédite avec un facteur explicatif

S'il existe une mauvaise adéquation des graphes des séries sanitaires observée et prédite (pics ne correspondant pas) ceci peut être dû à un facteur externe comme la grippe, par exemple. Il est alors utile de tracer les courbes des deux séries sanitaires ainsi que celle du facteur externe.

```
Prédites et observées sur le graphe du haut, autre variable (grippe par
exemple) selon la date sur le graphe du bas. 1


Ex 1
par(mfrow=c(2,1))
plot(morbi$trend,morbi$car1564,type="l")
lines(morbi$trend[!is.na(morbi$so224h)&!is.na(morbi$grip)&morbi$car1564<12]
,fitted(car1564.gam),col=2)
plot(morbi$date.study,morbi$grip,type="l")
par(mfrow=c(1,1))
```

On obtient la figure ci-dessous (figure 86).

Figure 86. Comparaison des séries obsevée, prédite et du facteur explicatif



Ce graphe permet de vérifier que les variations et accidents que l'on trouve dans la courbe lissée (donc modélisée) se retrouvent dans la variable d'intérêt (ici la grippe). Dans le cas présent ceci est à peu près vrai.

```
Prédites et observées sur le graphe du haut, autre variable (grippe par
exemple) selon la date sur le graphe du bas. 2


Ex 2 : prédites et observées lissées sur le même schéma et grippe sur
schéma en dessous
par(mfrow=c(2,1))
plot(observ.smooth,type="l",main="LISSAGE DES OBSERVEES ET DES
PREDITES",xlab="trend",ylab="observes (en noir) et prédites (en
violet)",lab=c(20,,))
lines(predit.smooth,col=3)
plot(morbi$date.study,morbi$grip,type="l")
par(mfrow=c(1,1))
```

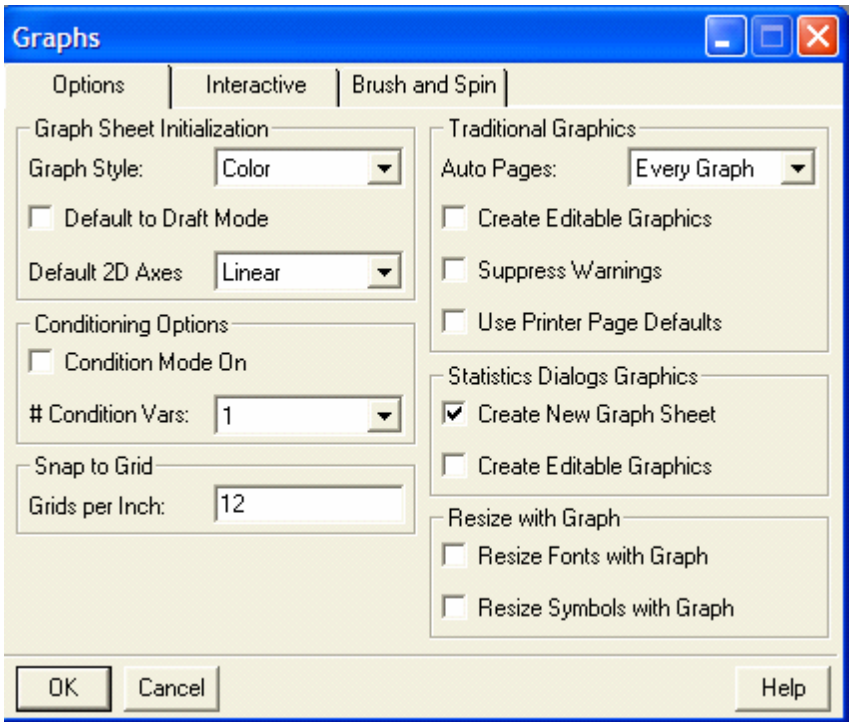
6.2.4. Effet partiel de chaque facteur sur la variable sanitaire

L'effet partiel de chacun des facteurs sur la variable sanitaire peut être mis en évidence de façon graphique. Il est possible de représenter la forme de la relation partielle de chacun des facteurs avec la variable sanitaire.

```
plot.gam(mortot.gam)
```

Le résultat de cette commande est un fichier avec autant de feuilles que de graphes (une par facteur) si l'on a pris la précaution de paramétrer les options correctement. Il faut aller dans le menu **Options** puis Graph Options. On obtient la fenêtre suivante (figure 87) :

Figure 87. Options graphiques



Il faut choisir l'option **Every Graph** dans la ligne **Auto Pages**.

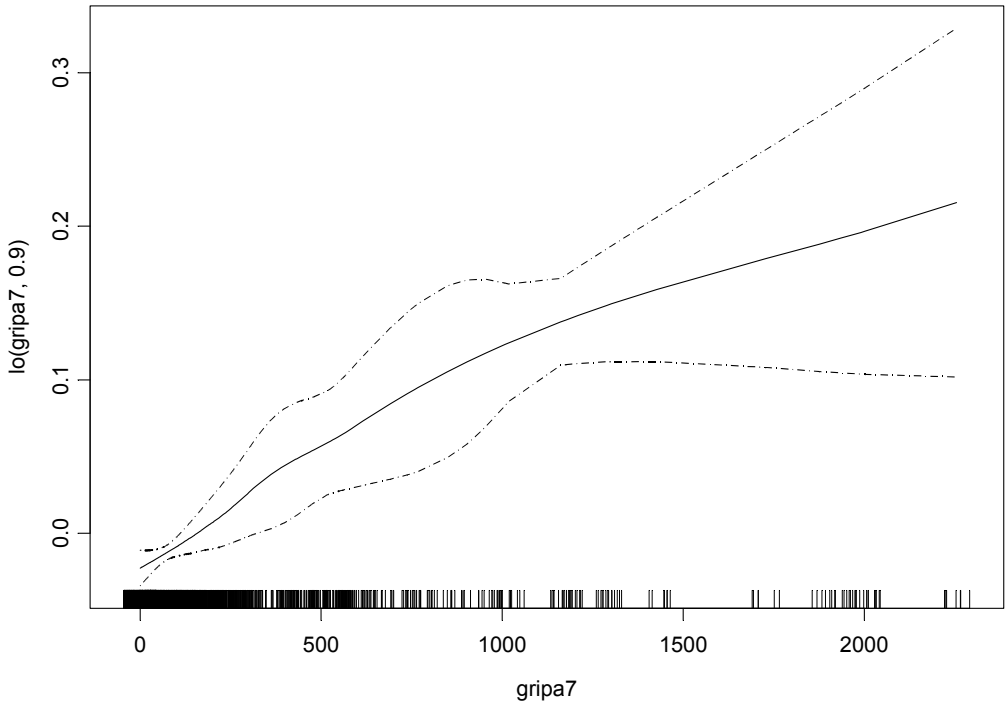
Remarque. Dans « R », pour obtenir les graphes des effets partiels, on écrit « `plot` » et non « `plot.gam` ». En fait, dans S-PLUS les deux commandes sont possibles avec un modèle gam et sont équivalentes.

Si l'on veut faire figurer les intervalles de confiance, la commande est :

```
plot.gam(mortot.gam,se=T)
```

On obtient des graphes comme ci-dessous (ici, effet partiel de la grippe affectée d'un retard de 7 jours) (figure 88) :

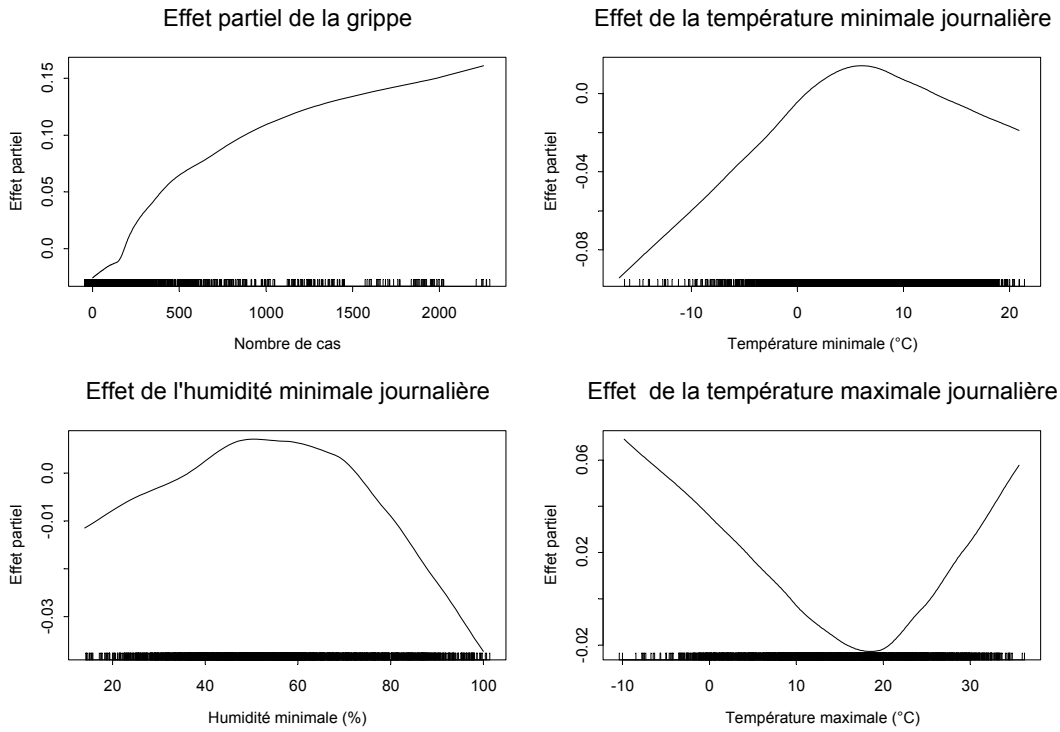
Figure 88. Effet partiel avec intervalle de confiance



Si l'on veut placer **plusieurs graphes sur une même feuille**, on utilise la commande « `par(mfrow=c(l,c))` » avec « `l` » le nombre de lignes et « c » le nombre de colonnes souhaités (§ 5.2.4). Dans « R », la commande est la même.

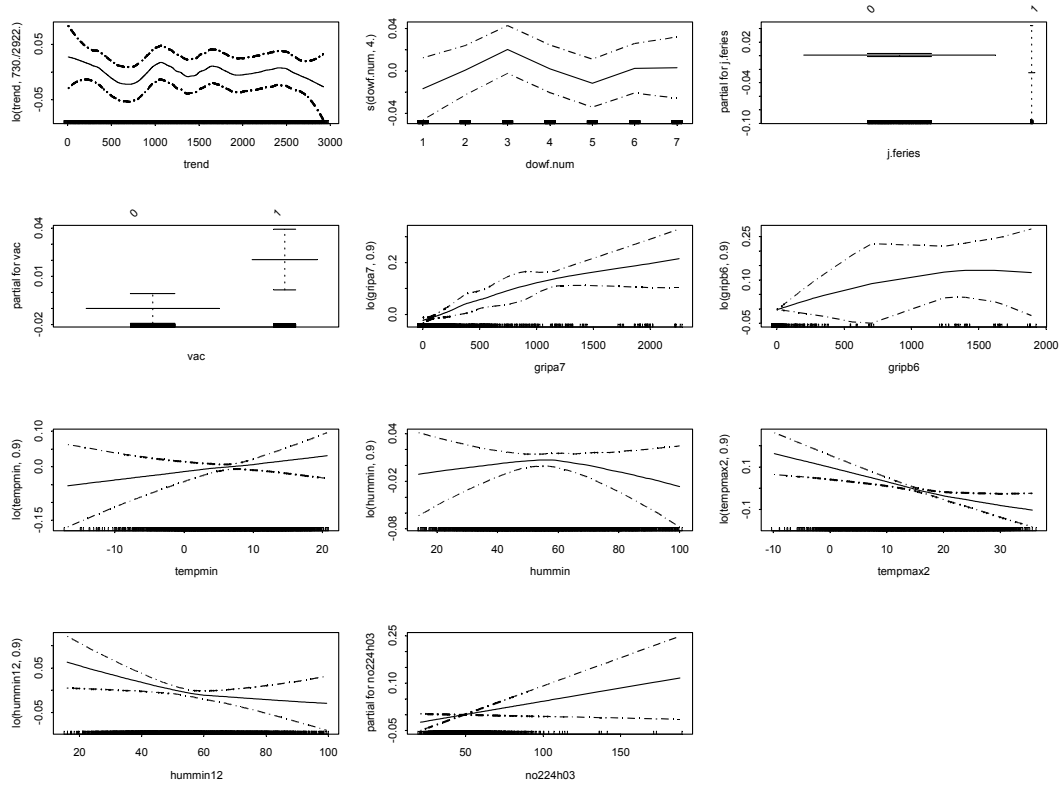
Ici, par exemple, on a créé une matrice de figures 2 x 2 avec l = c = 2 (figure 89).

Figure 89. Relation partielle de chaque facteur avec la variable sanitaire



Avec les intervalles de confiance (figure 90) :

Figure 90. Relation partielle de chaque facteur avec la variable sanitaire et intervalle de confiance



Il peut être intéressant aussi de représenter toutes les variables sur un graphe unique :

```
par(mfrow=c(1,1))
plot(morta$trend,morta$mortot,type="l",ylim=c(-3,40))
lines(morta$trend,fitted(mortot.gam),col=2)
lines(morta$trend,morta$grip/300,col=8)
lines(morta$trend,morta$tempmin/3,col=5)
lines(morta$trend,morta$tempmax/3,col=4)
lines(morta$trend,morta$bouleau/50,col=8)
lines(morta$trend,30+morta$so2424h/10,col=3)
```

6.2.5. Critère d'Akaike

Le critère d'Akaike (AIC pour *Akaike Information Criteria*) [35-37] permet de faire un choix – ou tout au moins, d'aider au choix – entre différents modèles (§ 3.2.3.2), le modèle retenu étant celui qui présente l'AIC le plus faible. Il « opère » de la même façon que la déviance mais en pénalisant celle-ci par un terme dépendant du nombre de paramètres du modèle :

AIC = -2 * (maximum de la log-vraisemblance) + 2 * (nombre de paramètres)

```
AIC(mortot.gam)
```

L'AIC permet ainsi d'orienter le choix des décalages attribués aux températures et à l'humidité, aide à décider de l'intérêt de conserver ou non la température maximale et les comptes polliniques dans le modèle, à tester la pertinence de l'introduction d'une variable humidité supplémentaire décalée et à tester l'interaction entre la température et l'humidité. Le critère d'Akaike, ici, ne détermine pas mais *oriente* les choix précédents car il n'est pas défini pour les modèles à quasi-vraisemblance (voir § 3.2.3.2). Cette dernière s'impose ici, en effet, en raison de la surdispersion de la variable sanitaire. D'autre part, pour appliquer ce critère, il est nécessaire que les modèles à comparer soient emboîtés, ce qui n'est pas toujours le cas.

`AIC(mortot.gam)` donne un résultat du type :

```
gam(formula = mortot ~ lo(trend, 183/2922) + dowf + j.feries + vac +
lo(gripa, 0.9) + lo(gripb, 0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
lo(tempmax1, 0.9) + lo(so224h, 0.9), family = quasi(log, mu), data = morta,
subset = mortot < 16, na.action = na.omit)

Degrees of Freedom Total =  2833

Degrees of Freedom Residual =  2784.27

Residual Deviance =  2576.629

AIC= 2662.942
```

Remarque. Dans « R », l'Akaike est obtenu avec la commande « `Bic(mortot.gam,k=2)` ». La commande « `Bic(mortot.gam)` » donne la valeur du *Bayesian information criteria* (BIC).

Ainsi :

```
Bic(mortot.gam) donne la sortie suivante :

gam(formula = mortot ~ s(trend, k = 150) + dowf + feries + typvac +

s(grip, k = 226) + s(tempmin, k = 39) + s(tempmax1, k = 49) + oz, family =
quasipoisson, data = mortac, sp = c(1e+07, 1e+07, 1000, 5000))

Degrees of Freedom Total =  2920

Degrees of Freedom Residual =  2878.257

Residual Deviance =  2821.958

BIC= 5833.913

Penalty= 7.9793

Et

Bic(mortot.gam,k=2) donne ce type de résultat :

gam(formula = mortot ~ s(trend, k = 150) + dowf + feries + typvac +
s(grip, k = 226) + s(tempmin, k = 39) + s(tempmax1, k = 49) + oz, family =
quasipoisson, data = mortac, sp = c(1e+07, 1e+07, 1000, 5000))

Degrees of Freedom Total =  2920

Degrees of Freedom Residual =  2878.257

Residual Deviance =  2821.958

BIC= 3576.896

Penalty= 2
```

Remarque. BIC = -2 * log-vraisemblance + (nombre de paramètres) * logarithme (nombre d'observations).

6.2.6. Paramètre de dispersion

Ce paramètre devrait être égal à 1 puisque l'analyse se base sur une régression de Poisson. Si le paramètre est supérieur à 1, cela signifie qu'il existe une surdispersion, témoin d'une variation extrapoissonienne à expliquer et/ou modéliser. À l'inverse, un paramètre de dispersion inférieur à 1 peut témoigner d'une sur-spécification du modèle (trop de paramètres introduits dans le modèle).

Pour calculer le coefficient de dispersion, il faut écrire la ligne suivante :

```
summary(mortot.gam)$dispersion
```

Le résultat du calcul pourrait être :

```
Ex

Dispersion Parameter for Quasi-likelihood family taken to be 0.9710607
```

Dans ce cas, il y a sous-dispersion.

De façon générale, les deux tests les plus utiles pour la construction du modèle sont la PACF et le graphe des résidus. Puis vient la comparaison entre les courbes observées et les courbes prédites. L'AIC sert à choisir les décalages relatifs aux températures et à l'humidité, à décider de l'intérêt ou non de conserver la température maximale et les pollens dans le modèle, à tester la pertinence de

l'introduction d'une variable *humidité* décalée (ou d'une variable *vacances*) et à tester l'interaction de la température et de l'humidité.

6.2.7. Résumés des modèles

La commande `summary` que nous connaissons déjà, lorsqu'elle est appliquée à un modèle, donne un résumé des caractéristiques de ce modèle. Elle peut être utilisée à tout moment mais, en fin de la modélisation, elle permet de fournir l'estimation des coefficients de la régression ainsi que leur intervalle de confiance.

Les deux types de `summary` que l'on utilise le plus souvent en modélisation sont la commande `summary.glm` et la commmande `summary.gam`.

Commande « `summary.glm` »

Pour voir ce que cette commande permet d'obtenir, appliquons là à un exemple (`mortot901.gam` est le nom du modèle).

```
summary.glm(mortot901.gam)
```

La sortie est classique (valeurs et dispersion des estimateurs des coefficients de la régression, significativité, coefficient de dispersion, déviance, déviance résiduelle, corrélations etc. :

```
Call: gam(formula = mortot ~ lo(trend, 730./2922.) + s(dowf.num, 3.) +
j.feries + vac + lo(gripa7, 0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
 lo(  hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12, 0.9) + lo(so224h01,
0.9), family = quasi(log, mu), data = morta, subset = mortot <16.,
na.action = na.omit)
Deviance Residuals:
        Min         1Q      Median        3Q       Max
 -0.8779967 -0.2352778 -0.01019811 0.2290521 0.9325031

Coefficients:
                          Value  Std. Error     t value
        (Intercept)  2.108540476 0.021527054  97.9484012
lo(trend, 730/2922)  0.184460238 0.391546082   0.4711074
     s(dowf.num, 3)  0.001760766 0.003099084   0.5681570
           j.feries -0.015743256 0.017847766  -0.8820855
                vac  0.015250270 0.006971424   2.1875400
    lo(gripa7, 0.9)  2.123733872 0.351396149   6.0437027
    lo(gripb6, 0.9)  0.961699192 0.343989115   2.7957257
   lo(tempmin, 0.9)  1.137091983 0.739968391   1.5366764
    lo(hummin, 0.9) -0.257667875 0.492564600  -0.5231149
  lo(tempmax2, 0.9) -2.843198518 0.852903043  -3.3335542
  lo(hummin12, 0.9) -1.004207216 0.561644698  -1.7879760
  lo(so224h01, 0.9)  0.982975746 0.462950159   2.1232863

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8887326 )

    Null Deviance: 2698.185 on 2798 degrees of freedom

Residual Deviance: 2570.518 on 2773.0284484 degrees of freedom

Number of Fisher Scoring Iterations: 3
```

```
Correlation of Coefficients:
                   (Intercept) lo(trend, 730/2922) s(dowf.num, 3)
j.feries        vac lo(gripa7, 0.9) lo(gripb6, 0.9) lo(tempmin, 0.9)
lo(trend, 730/2922)  0.0505620
     s(dowf.num, 3) -0.5670078  -0.0468292
           j.feries  0.7585259   0.0300408           0.0112109
                vac -0.0003247  -0.0224722           0.0262237     -
0.1197353
    lo(gripa7, 0.9) -0.0509614   0.1160658          -0.0195694     -
0.0467046 -0.1222055
    lo(gripb6, 0.9) -0.0598071  -0.2375846           0.0148128     -
0.0411643 -0.1216654  0.2099343
   lo(tempmin, 0.9) -0.0065453   0.1084568           0.0105778
0.0103023 -0.0542207  0.1049667       0.0919613
    lo(hummin, 0.9) -0.0032059  -0.0878063           0.0067405     -
0.0091276  0.0701343 -0.1177697      -0.0548393      -0.1137254
  lo(tempmax2, 0.9)  0.0334349   0.0384223          -0.0366664
0.0203241 -0.0793500  0.1002862       0.0012642      -0.7698933
  lo(hummin12, 0.9)  0.0021293  -0.0274698           0.0128320
0.0015778  0.0310565 -0.0791808      -0.0625068      -0.2650310
  lo(so224h01, 0.9)  0.1038461   0.4578421          -0.0896570
0.0710070 -0.0088959  0.0974079      -0.1427750       0.2752635

                   lo(hummin, 0.9) lo(tempmax2, 0.9) lo(hummin12, 0.9)
lo(trend, 730/2922)
     s(dowf.num, 3)
           j.feries
                vac
    lo(gripa7, 0.9)
    lo(gripb6, 0.9)
   lo(tempmin, 0.9)
    lo(hummin, 0.9)
  lo(tempmax2, 0.9)  0.0859058
  lo(hummin12, 0.9) -0.5270893       0.4451520
  lo(so224h01, 0.9) -0.1304722       0.0965972         0.0221096
```

Remarque. Si l'on n'a pas besoin des corrélations, il suffit d'écrire « `summary.glm(mortot901.gam,cor=F)` » ou même, plus simplement « `summary.glm(mortot901.gam,c=F)` ».

Remarque. Dans le cas d'un modèle GAM comme ci-dessus, les coefficients des termes non paramétriques (lissés) ne peuvent pas être interprétés comme dans un GLM. Cette commande est utile en fin de modélisation quand, comme nous le verrons, le modèle est rendu « linéaire » pour le polluant afin d'estimer le coefficient de celui-ci ainsi que son intervalle de confiance.

Commande « `summary.gam` »

Dans le cas d'un modèle GAM, cette commande peut s'écrire tout court `summary(nom.modèle)`.

Par exemple :

```
summary.gam(mortot901.gam)
```

Donne :

```
Call: gam(formula = mortot ~ lo(trend, 730./2922.) + s(dowf.num, 3.) +
j.feries + vac + lo(gripa7, 0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9)
 +lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12, 0.9) + lo(so224h01,
0.9), family = quasi(log, mu), data = morta, subset = mortot
<16., na.action = na.omit)
Deviance Residuals:
        Min         1Q      Median        3Q       Max
 -0.8779967 -0.2352778 -0.01019811 0.2290521 0.9325031

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8887326 )

    Null Deviance: 2698.185 on 2798 degrees of freedom

Residual Deviance: 2570.518 on 2773.028 degrees of freedom

Number of Local Scoring Iterations: 3

DF for Terms and F-values for Nonparametric Effects

                    Df Npar Df   Npar F     Pr(F)
        (Intercept)  1
lo(trend, 730/2922)  1     5.9 1.096575 0.3616601
     s(dowf.num, 3)  1     2.0 2.263621 0.1040581
           j.feries  1
                vac  1
    lo(gripa7, 0.9)  1     1.7 0.832274 0.4180459
    lo(gripb6, 0.9)  1     1.1 2.394487 0.1182205
   lo(tempmin, 0.9)  1     0.6 0.339518 0.4533888
    lo(hummin, 0.9)  1     0.6 4.133959 0.0615453
  lo(tempmax2, 0.9)  1     0.6 4.218146 0.0571956
  lo(hummin12, 0.9)  1     0.6 4.147662 0.0617217
  lo(so224h01, 0.9)  1     0.9 0.624227 0.4183870
```

Ici, outre les déviances et le paramètre de dispersion, la sortie comporte un ensemble de résultats concernant les variables explicatives sous forme non paramétrique. Ainsi, la probabilité `Pr(F)` représente la significativité de la dimension non linéaire du coefficient : si le résultat est significatif, cela veut dire que la part non linéaire (la courbure ?) n'est pas négligeable et que ne tenir compte que de la composante linéaire n'est pas suffisant.

6.3. Analyse descriptive

Après détermination de l'ensemble des variables (variable sanitaire et facteurs) nécessaires à l'analyse il est indispensable d'en calculer les paramètres principaux, de tracer les graphes et les *boxplots*, de réaliser un lissage des données pour voir les variations saisonnières, une PACF pour détecter l'autocorrélation dans la série. Cette analyse préliminaire devrait être suivie d'une recherche des valeurs aberrantes des différentes variables qu'il serait judicieux d'enlever avant de débuter la modélisation.

6.3.1. Paramètres

6.3.1.1. Minimum, premier quartile, médiane, moyenne, troisième quartile, maximum

La ligne à écrire dans S-Plus est :

```
summary(morta$mortot)
```

Exemple de sortie S-PLUS :

```
   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
  1.000  6.000   8.000  8.512 10.000  23.000
```

6.3.1.2. Variance

```
var(morta$mortot)
```

6.3.1.3. Coefficient de variation

Le coefficient de variation est égal au rapport de l'écart-type à la moyenne.

```
sqrt(var(morta$mortot)) / mean(morta$mortot)
```

6.3.1.4. Dispersion (variance / moyenne)

```
var(morta$mortot) / mean(morta$mortot)
```

Ceci permet de voir rapidement s'il y a sur-dispersion (dans l'éventualité où le coefficient de dispersion est supérieur à 1) ou sous-dispersion (cas où le coefficient est inférieur à 1).

6.3.2. Graphes

6.3.2.1. Indicateur sanitaire en fonction de la date

```
plot(morta$date.study,morta$mortot,type="l")
```

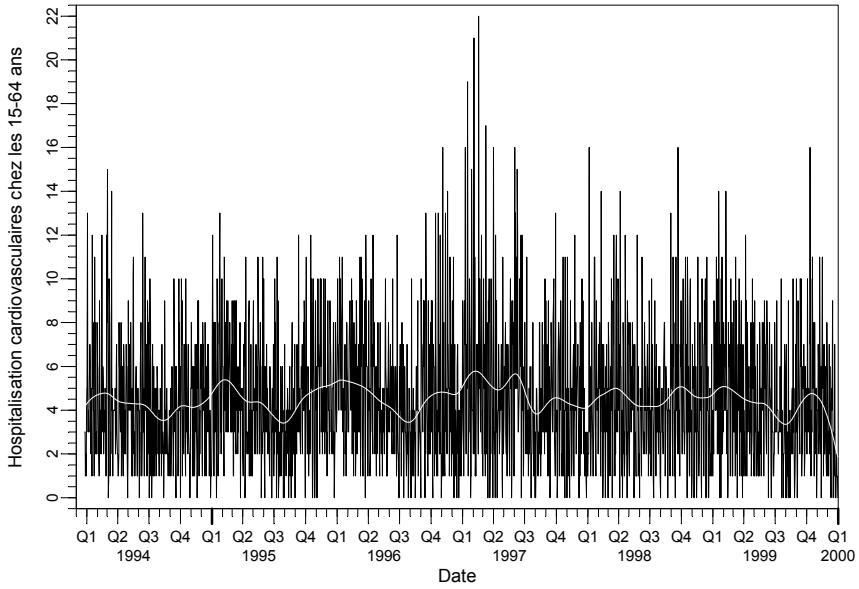
Il est possible avec plus ou moins de bonheur de tracer des graphes comprenant des lissages.

Un lissage « *spline* » :

```
plot(morbi$date.study,morbi$car1564,type="l",xlab="Date",ylab="Hospitalisat
ion cardiovasculaires chez les 15-64 ans")
lines(smooth.spline(morbi$date.study,morbi$car1564),col=0)
```

Ces lignes donnent le graphe suivant (figure 91) :

Figure 91. Lissage *spline*. Hospitalisations pour pathologie cardio-vasculaires à Strasbourg



Un lissage « *loess* » :

```
plot(morbi$date.study,morbi$car1564,type="l",col=4)
lines(loess.smooth(morbi$date.study,morbi$car1564,span=.05))
```

Le lissage « *supersmoother* »

```
plot(morbi$date.study,morbi$car1564,type="l",col=2)
lines(supsmu(morbi$date.study,morbi$car1564))
```

Donne le message suivant :

```
Error in Ops.dates(is.inf(x)): unary ! not defined for dates objects
Dumped
OU
Problem in is.number(x) & !is.inf(x): length of longer operand (2197)
should be a multiple of length of shorter (5)
```

Et ne donne pas de graphe !

6.3.2.2. Indicateur sanitaire et différents lissages en fonction de la tendance (ou numéro du jour)

Les commandes sont :

```
plot(morta$trend,morta$mortot,type="l")
lines(smooth.spline(morta$trend,morta$mortot),col=2)
lines(loess.smooth(morta$trend,morta$mortot),col=3)
lines(supsmu(morta$trend,morta$mortot),col=4)
```

Remarque. Pour affiner l'analyse des variations, il est possible de modifier la fenêtre de la fonction :

```
lines(supsmu(morta$trend,morta$mortot,span=.4),col=4)
```

Il est possible de mêler les deux types de graphes (selon la tendance et la date) :

```
par(mfrow=c(2,1))
plot(morbi$date.study,morbi$car65,type="l")
plot(morbi$trend,morbi$car65,col=2)
lines(smooth.spline(morbi$trend,morbi$car65))
par(mfrow=c(1,1))
```

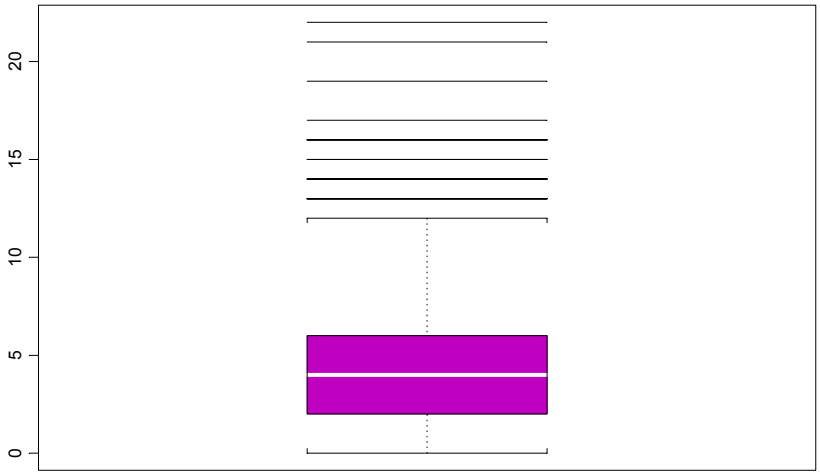
6.3.3. Boxplot

```
boxplot(morta$mortot)
```

Le *boxplot* permet de voir, pour chaque variable s'il existe des valeurs aberrantes (*outliers*) (figure 92).

Si c'est le cas, il faut déterminer la limite supérieure des valeurs *non exotiques*. La valeur de cette limite est fixée à la médiane plus deux écarts interquartiles. Les valeurs supérieures à cette limite sont supprimées de l'analyse.

Figure 92. *Box plot* de la variable sanitaire



6.3.4. PACF (autocorrélation partielle)

```
acf(morta$mortot,,type="p")
```

S'il n'est pas nécessaire de faire figurer la liste des autocorrélations :

```
sum(acf(morta$mortot,type="p")$acf)
```

Dans « R », on écrirait :

```
sum(pacf(resid(mortot01.gam)[,30])$acf)
```

`[,30]` veux dire que l'on désire afficher les 30 premiers retards (les crochets signifient que cette instruction n'est pas obligatoire).

Une condition nécessaire (et non suffisante) est que la somme des PACF soit proche de 0 (~ PACF d'un bruit blanc).

6.4. Processus de l'analyse

L'écriture du modèle initial est la suivante (cf. § 6.1)

```
morta.gam _ gam (indic.sanit ~ lo(tendance,183/(nb.jours.total)) +
j.semaine + j.féries + vacances + lo (grippe.décalage.0,.7) +
lo(tempé.min.décalage.0,.7) + lo(hummid.min.décalage.0,.7) +
lo(tempé.max.décalage.1,.7) + lo(polluant.décalage0,.7) , family =
quasi(log,mu) , data=morta , na=na.omit)


morbi.gam _ gam (indic.sanit ~ lo(tendance,183/(nb.jours.total)) +
j.semaine + j.féries + vacances + lo (grippe.décalage.0,.7) +
lo(tempé.min.décalage.0,.7) + lo(hummid.min.décalage.0,.7) +
lo(tempé.max.décalage.1,.7) + lo(pollen1_décalage0,.7) +
lo(pollen2_décalage0,.7) + . . . + lo(pollenN_décalage0,.7)+
lo(polluant.décalage0,.7) , family = quasi(log,mu) , data=morbi ,
na=na.omit)
```

Les outils décrits plus haut (§ 6.2) sont appliqués au modèle initial et servent de référence pour l'étape suivante. Ce sont, à titre de rappel :

- L'autocorrélation partielle des résidus (PACF)
- L'observation du graphe des résidus
- La comparaison du graphe de la série prédite par le modèle et du graphe de la série initiale
- L'effet partiel de chaque facteur sur la variable sanitaire
- Le critère d'Akaike
- Le paramètre de dispersion
- Les « *summaries* »

À partir du modèle initial et aidée des différents outils décrits précédemment, la poursuite de l'analyse consiste à modifier progressivement certains paramètres du modèle afin d'en améliorer l'ajustement. Les différentes étapes de l'analyse sont les suivantes :

6.4.1. Analyse de sensibilité aux valeurs extrêmes de la variable sanitaire

La présence de valeurs extrêmes peut être en relation avec des phénomènes accidentels non environnementaux (problème d'enregistrement, etc.) et peut biaiser la relation entre la variable sanitaire et les variables explicatives. L'élimination des valeurs extrêmes des données sanitaires c'est à dire des valeurs supérieures à la médiane plus deux écarts interquartiles (cf. § 6.3.3) limite la surdispersion : si le paramètre de dispersion est trop éloigné de 1, il est préférable de retenir le modèle sans valeurs extrêmes.

Ex

Dans S-Plus, la ligne de commande :

```
summary(morta$mortot)
```

Donne la sortie suivante :

```
 Min. 1st Qu. Median  Mean 3rd Qu. Max.
    1       6      8 8.512      10   23
```

La limite supérieure des valeurs non exotiques est :

```
8+2*(10-6) = 16
```

Remarque. La ligne de commande S-Plus permettant de trouver cette valeur immédiatement est :

```
summary(morta$mortot)[3]+2*(summary(morta$mortot)[5]-
summary(morta$mortot)[2])
```

Le modèle est réécrit en supprimant les valeurs aberrantes :

```
mortot2.gam_gam(mortot~lo(trend,183/2922)+j.sem+j.feries+vac+lo(grip,.7)+lo
(tempmin,.7)+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<16)
```

Si l'on veut exclure les valeurs limites

Ou

```
mortot2.gam_gam(mortot~lo(trend,183/2922)+j.sem+j.feries+vac+lo(grip,.7)+lo
(tempmin,.7)+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<=16)
```

Si l'on veut conserver les valeurs limites

Les différents tests sont appliqués au modèle sans valeurs aberrantes

- PACF des résidus

```
sum(acf(resid(mortot2.gam),type="p")$acf)
```

A priori, peu de valeurs sont exclues (les valeurs aberrantes représentent 2 à 3% de l'ensemble des données). Par conséquent, cette opération ne doit pas modifier la PACF de façon notable. Si tel n'était pas le cas (variation défavorable de la PACF), il serait nécessaire de modifier la fenêtre de la tendance.

- Graphe des résidus

```
plot(resid(mortot2.gam))
```

- Graphe des séries prédite et observée (courbes superposées)

```
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax1)&morta$mortot<
16],fitted(mortot2.gam),col=2)
plot(morta$grip,type="l")
```

- Effet partiel des différents facteurs

```
plot.gam(mortot2.gam)
```

À ce stade, la connaissance de l'AIC et du paramètre de dispersion n'est pas utile.

Remarque. Lorsqu'on introduit une variable *jours fériés* et une variable *vacances*, cette dernière peut apparaître comme un facteur protecteur (effet plus important quant elle est égale à 1). De façon générale ceci peut se passer si l'effet est faible et l'intervalle de confiance est important. La variable *vacances* peut ainsi avoir un effet positif dû au hasard. Si les vacances sont fortement reliées à la mortalité, il faut regarder ce qui se passe à ce moment. D'un point de vue pratique, on ajuste la saison et, peut-être de manière particulière, ce qui se passe en été. A ce moment, la variable *vacances* ne devrait plus être nécessaire. Si elle est encore significative, il faut vérifier si son impact est fort. Si c'est le cas, il faut essayer de voir quelles vacances scolaires induisent cet effet. Ce dernier est peut-être lié uniquement à la semaine entre noël et le 1er de l'an. Dans ce cas, il faut ajuster spécifiquement pour cette période.

6.4.2. Traitement préliminaire de la variable grippe

Si, au cours d'une période d'épidémie grippale, l'**adéquation des graphes** respectifs des valeurs observées et des valeurs prédites est mauvaise, ceci peut vouloir dire que l'épisode grippal n'est pas homogène. Il peut être pertinent de créer deux (voire plusieurs) variables *grippe* (permettant d'individualiser les périodes épidémiques correspondant à des virus différents) et de tester l'effet de cette modification sur la congruence des courbes.

Il est possible, par exemple, de créer une variable *grippe* propre à la période (ou plusieurs variables *grippe* propres aux périodes) où se pose le problème de congruence et une autre variable pour l'ensemble des autres périodes.

```
Ex
gripa est égale à grip en dehors de la grippe de l'hiver 1996-1887
gripb est égale à grip lors de la grippe de l'hiver 1996-1887

morta$gripa_ifelse(morta$trend<2466|morta$trend>2677,morta$grip,0)
morta$gripa1_c(rep(NA,1),morta$gripa[1:(length(morta$gripa)-1)])
morta$gripa2_c(rep(NA,2),morta$gripa[1:(length(morta$gripa)-2)])
morta$gripa3_c(rep(NA,3),morta$gripa[1:(length(morta$gripa)-3)])
morta$gripa4_c(rep(NA,4),morta$gripa[1:(length(morta$gripa)-4)])
morta$gripa5_c(rep(NA,5),morta$gripa[1:(length(morta$gripa)-5)])
morta$gripa6_c(rep(NA,6),morta$gripa[1:(length(morta$gripa)-6)])
morta$gripa7_c(rep(NA,7),morta$gripa[1:(length(morta$gripa)-7)])

morta$gripb_ifelse(morta$trend<2466|morta$trend>2677,0,morta$grip)
morta$gripb1_c(rep(NA,1),morta$gripb[1:(length(morta$gripb)-1)])
morta$gripb2_c(rep(NA,2),morta$gripb[1:(length(morta$gripb)-2)])
morta$gripb3_c(rep(NA,3),morta$gripb[1:(length(morta$gripb)-3)])
morta$gripb4_c(rep(NA,4),morta$gripb[1:(length(morta$gripb)-4)])
morta$gripb5_c(rep(NA,5),morta$gripb[1:(length(morta$gripb)-5)])
morta$gripb6_c(rep(NA,6),morta$gripb[1:(length(morta$gripb)-6)])
morta$gripb7_c(rep(NA,7),morta$gripb[1:(length(morta$gripb)-7)])
```

Le modèle est construit avec les nouvelles variables *grippe*.

```
Ex 1
mortot2ab.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+
lo(gripa,.7)+lo(gripb,.7)+lo(tempmin,.7)+lo(hummin,.7)+
lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)


Ex 2 respi642ab.gam_gam(respi64~lo(trend,183/2922)+dowf+j.feries+vac+
lo(gripa,.7)+lo(gripb,.7)+lo(tempmin,.7)+lo(hummin,.7)+
lo(tempmax1,.7)+lo(bouleau,.7)+lo(graminee,.7)+
lo(urticacee,.7)+lo(so224h,.7),family=quasi(log,mu),data=morbi,
na=na.omit,subset=respi64<16)
```

L'appréciation de la qualité du contrôle est basée sur la comparaison du graphe de la variable sanitaire prédite et du graphe de la variable sanitaire enregistrée. Ainsi, par exemple, si un pic hivernal de la mortalité (ou de l'hospitalisation) réelle correspond à une période d'épidémie de grippe et si la mortalité (ou l'hospitalisation) prédite ne présente pas un tel pic, ceci est dû au fait que cette épidémie n'est pas bien contrôlée par le modèle.

6.4.3. Modification éventuelle des variables *tendance* et/ou *vacances*

Comme précédemment, s'il y a mauvaise adéquation des courbes représentant les séries sanitaires mesurée et prédite lors de certaines périodes de l'année ou au cours de vacances, il est utile de créer des variables *tendance* et/ou *vacances* (trend.a, trend.b, etc., vac.a, vac.b, etc.).

6.4.3.1. Tendance

La tendance générale (trend) peut être conservée mais des tendances partielles lui sont adjointes :

```
Ex
morbi$trend96.1_ifelse(morbi$trend<1025|morbi$trend>1075,0, morbi$trend)
morbi$trend96.2_ifelse(morbi$trend<1100|morbi$trend>1175,0, morbi$trend)
morbi$trend96.3_ifelse(morbi$trend<1225|morbi$trend>1275,0, morbi$trend)
```

6.4.3.2. Vacances

Les différentes possibilités sont :

- une variable *vacances* de type tendance (*vac.trend*) c'est à dire équivalente d'une variable tendance (*trend*) ; cette variable peut prendre des valeurs entières (1 à n) ou la valeur contemporaine de la tendance elle-même ; ceci peut être utilisé pour des durées suffisamment longue, les vacances scolaires d'été, par exemple.
- une variable *vac.niveau* avec une valeur entière spécifique pour chaque type de vacances ; par exemple noël=1, février=2, pâques=3, été=4, toussaint=5.
- une variable *vac.année* pour un type de vacance particulier, noël par exemple, ayant une valeur différente selon l'année. Exemple : pour noël 92, la variable vaut 1, pour noël 93, elle vaut 2, en 94, 3 etc. En dehors, elle vaut 0.

Il est possible, bien sur, de combiner les trois types d'approches.

6.4.4. Traitement de la taille des fenêtres de lissage

La diminution de la taille des fenêtres de lissage (de l'ensemble des variables, celle de la tendance étant traitée en dernier) diminue l'autocorrélation dans la série des résidus. La taille est choisie sur la base d'une minimisation de la somme des autocorrélations et d'une **PACF** la plus conforme possible (*i.e.* comportant le moins possible de pics d'autocorrélation, *i.e.* la plus proche possible de celle d'un bruit blanc). La **comparaison des graphes des valeurs observées et prédites** est utile à cette étape pour confirmer l'option choisie. Il en est de même pour les **effets partiels** : lorsque les courbes représentatives des effets partiels des différentes variables, autres que la tendance ne sont pas réalistes sur le plan biologique (croissance, forme, etc.), il est licite, en effet, de modifier la fenêtre.

6.4.4.1. Augmentation des fenêtres des variables autres que la tendance

L'écriture du modèle est, par exemple :

```
mortot3.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo
(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+
lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit, subset=mortot<16)
```

Les différents tests sont réalisés.

- PACF des résidus

```
sum(acf(resid(mortot3.gam),type="p")$acf)
```

Différentes largeurs de fenêtre sont testées. Le processus s'arrête quand le minimum de la somme des autocorrélations et la PACF la plus conforme ont été obtenus.

- Prédites et observées (courbes superposées)

```
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax1)&morta$mortot<
16],fitted(mortot3.gam),col=2)
```

- Effets partiels

```
plot.gam(mortot3.gam)
```

- AIC

Ici l'AIC peut être utilisé et est un critère d'aide à la décision mais non pas un critère déterminant.

Remarque. L'AIC a tendance à privilégier les fenêtres larges ce qui a pour conséquence un moins bon ajustement pour les valeurs extrêmes.

6.4.4.2. Variation de la fenêtre de la (des) tendance(s)

Remarque. Ici, les fenêtres des tendances et des variables vacances sont traitées simultanément et de la même façon lorsque ces dernières sont considérées comme variables quantitatives discrètes (équivalentes de variables *tendance*).

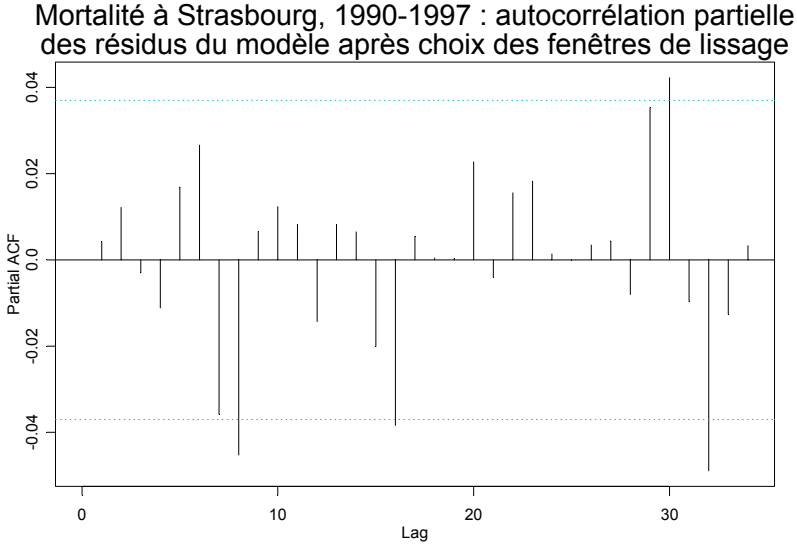
```
mortot3a.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(gripa,.9)+l
o(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),fami
ly=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot3b.gam_gam(mortot~lo(trend,300/2922)+dowf+j.feries+vac+
lo(gripa,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)
mortot3c.gam_gam(mortot~lo(trend,500/2922)+dowf+j.feries+vac+
lo(gripa,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)
mortot3d.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)
mortot3e.gam_gam(mortot~lo(trend,800/2922)+dowf+j.feries+vac+
lo(gripa,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)
.....
```

- PACF des résidus

```
sum(acf(resid(mortot3a.gam),type="p")$acf)
sum(acf(resid(mortot3b.gam),type="p")$acf)
sum(acf(resid(mortot3c.gam),type="p")$acf)
sum(acf(resid(mortot3d.gam),type="p")$acf)
sum(acf(resid(mortot3e.gam),type="p")$acf)
.....
```

La recherche de la taille de la fenêtre s'arrête quand le minimum de la somme des autocorrélations et la PACF la plus conforme ont été obtenus (figure 93).

Figure 93. Corrélogramme partiel des résidus après traitement des fenêtres de lissage



- AIC

Il sert à confirmer le choix de la fenêtre (en cas de conflit avec le choix issu du calcul de la PACF, ce dernier conserve la préférence).

- Graphes des effets partiels

Ils permettent de vérifier la cohérence des effets : par exemple, l'effet croissant de la grippe, l'effet protecteur des jours fériés, etc.

- Prédites et observées (courbes superposées)

Le graphe des résidus et la dispersion sont moins utiles à cette étape.

6.4.5. Ajustement de la variable grippe à différents retards ainsi que pour différentes fenêtres

Il faut expliquer au mieux les valeurs extrêmes de la variable sanitaire contemporaines de la période d'épidémie grippale. La recherche de la minimisation de l'**AIC** aide à choisir la largeur de la fenêtre de lissage puis à tester les différents retards, puis à changer de fenêtre, etc. Si la modélisation est correcte (*i.e.* si le meilleur retard est sélectionné), le **graphe représentant l'effet partiel** de la grippe est, en général, relativement lissé.

Ainsi, si par exemple, il a été nécessaire d'introduire deux variables *grippe* (*gripa* et *gripb*) dans le modèle, il faut tester *gripa* à différents retards *puis* gripb à différents retards

Le modèle choisi au stade précédent est :

```
mortot3.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)
```

1) Test de *gripa*

a) Variation du décalage

```
mortot4a0.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot4a1.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa1,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot4a2.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa2,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot4a3.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa3,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)
```

```
mortot4a4.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa4,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot4a5.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa5,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot4a6.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa6,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot4a7.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)
```

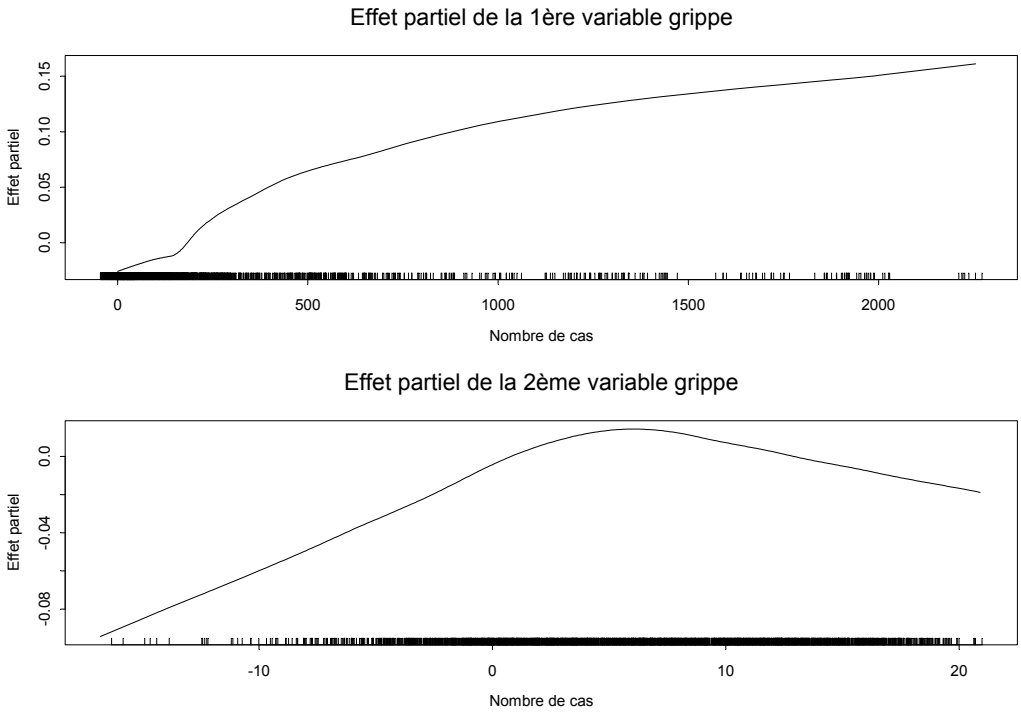
- AIC

```
AIC(mortot4a)

AIC(mortot4a1)

.....
```

- Effets partiels

Ils permettent de vérifier la cohérence du modèle (*gripa* doit entraîner une augmentation de la mortalité, pas exemple) (figure 94).

Figure 94. Effets partiels des deux types de variables *grippe*



b) Puis variation de la fenêtre

- AIC

- Effets partiels

Si la fenêtre diminue trop, le risque de trouver une relation non monotone entre la grippe et la mortalité (croissante, décroissante puis décroissante, par exemple) n'est pas négligeable.

2) Test de *gripb*

a) Variation du décalage

- AIC

- Effets partiels

b) Puis variation de la fenêtre

Les mêmes remarques peuvent être formulées que pour la variable *gripa*.

6.4.6. Choix et ajustement des variables pollens

L'**AIC** est utilisé pour sélectionner, dans le cas de l'analyse des admissions hospitalières, les indicateurs polliniques pertinents et pour choisir les retards à leur affecter. L'analyse se fait, pollen par pollen.

6.4.7. Traitement des variables *température*

6.4.7.1. Ajuster la variable température maximale

Celle-ci est testée avec différents retards bruts (niveaux de température mesurés 1, 2 et 3 jours avant) ainsi qu'avec des retards moyennés (moyenne des mesures réalisées 1 et 2 jours avant, 2 et 3 jours avant et 1, 2 et 3 jours avant). Les retards moyennés sont destinés à détecter un éventuel effet cumulatif de la température. La température minimale n'est pas modifiée et reste dans le modèle avec un décalage de 0 jour. Le choix du meilleur modèle se base sur l'**AIC** (minimisation) tout en contrôlant systématiquement l'impact des modifications sur la **PACF**.

La variable *tempmax* est testée d'emblée à différents retards (1-3 j) et moyennée (1-2, 2-3 et 1-2-3) soit `tempmax1, tempmax2, tempmax3, tempmax12, tempmax23, tempmax123`. Rappelons que la variable *tempmin* n'est pas modifiée pour l'instant et doit rester dans le modèle avec un décalage 0.

```
morta$tempmax12_(morta$tempmax1+morta$tempmax2)/2
morta$tempmax23_(morta$tempmax2+morta$tempmin3)/2
morta$tempmax123_(morta$tempmax1+morta$tempmax2+morta$tempmin3)/3
```

Les modèles testés sont :

```
mortot51.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot52.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot53.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax3,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot512.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax12,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot523.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax23,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)

mortot5123.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax123,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,
na=na.omit,subset=mortot<16)
```

Le choix du meilleur modèle est basé sur la minimisation de l'AIC tout en contrôlant systématiquement l'impact sur la PACF (on ne tient pas compte, à ce stade de la forme de l'effet du polluant). Si l'AIC ne présente pas beaucoup de variation, il faut alors choisir le décalage de *tempmax* sur la base du graphique de l'effet partiel (forme de la relation mortalité – tempmax) et de la PACF.

- AIC

- PACF des résidus

- Graphe des résidus

- Prédites et observées (courbes superposées)

- Graphe des effets partiels

Concernant les graphes des effets partiels des deux types de températures (minimale et maximale), il faut que l'une des deux courbes (les effets partiels de la température minimale et de la température maximale) soit croissante (en général, `tempmin`), l'autre décroissante (en général, `tempmax`).

6.4.7.2. Ajuster la variable température minimale

Si l'AIC ne permet pas de choisir le décalage de *tempmax*, il peut paraître opportun de tester le modèle avec et sans *tempmax*. Si l'AIC ne peut pas faire la différence, il est licite de supprimer *tempmax*.

Dans ce cas, il est possible d'ajuster *tempmin*. La procédure est superposable à la précédente (choix du décalage).

Remarque. Dans le cas où les deux variables subsistent dans le modèle, il faut éviter d'attribuer le même décalage à la température maximale qu'à la température minimale et, par là, éviter le risque de colinéarité entre ces deux variables.

6.4.8. Étude de l'opportunité d'une variable humidité supplémentaire

Tout en gardant la variable *humidité* avec un décalage 0, il peut être nécessaire d'introduire une nouvelle variable *humidité* avec un décalage supérieur à 0 dans le modèle. Cette variable est testée, comme la température maximale, à différents retards bruts (1 à 3 jours) et moyennés (1-2 jours, 2-3 jours et 1-2-3 jours). Là aussi, le choix du décalage à conserver est basé sur la minimisation de l'**AIC** : ainsi, la présence d'une variable *humidité* décalée est pertinente si l'AIC diminue lorsqu'elle est introduite dans le modèle.

Remarque. L'AIC peut diminuer même si l'on rajoute une nouvelle variable ; en effet, l'AIC dans son principe sélectionne les variables dans le but de prédire au mieux ; si une variable apporte un gain assez important par rapport au coût en terme de degré de liberté qu'elle engendre, l'AIC diminuera et la nouvelle variable sera conservée dans le modèle.

Création des variables humidité décalées :

```
morta$hummin12_(morta$hummin1+morta$hummin2)/2
morta$hummin23_(morta$hummin2+morta$hummin3)/2
morta$hummin123_(morta$hummin1+morta$hummin2+morta$hummin3)/3
```

Les modèles :

```
mortot61.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin1,.9)+lo(so224h,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)
mortot62.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin2,.9)+lo(so224h,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)
mortot63.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin3,.9)+lo(so224h,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)
mortot612.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)
mortot623.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin23,.9)+lo(so224h,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)
mortot6123.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+
lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin123,.9)+lo(so224h,.9),family=quasi(log,mu),data=mo
rta,na=na.omit,subset=mortot<16)
```

Si l'AIC ne diminue pas ou pas beaucoup, il n'est pas nécessaire de conserver la variable humidité décalée. En cas d'hésitation, il est possible de considérer la PACF.

- AIC

- PACF des résidus

- Prédites et observées (courbes superposées)

- Effets partiels

Remarque. L'introduction de deux décalages d'une même variable (ici, un décalage 0 et un décalage supérieur ou égal à 1 pour les variables météorologiques) pourrait faire craindre d'inclure des variables colinéaires. En fait, ici ce sont les effets à court terme des variables météorologiques qui sont considérés. Donc, après avoir tenu compte de l'effet « saisons », la corrélation entre les variables décalées 0 et 1 jour (et *a fortiori*, pour un décalage plus grand) diminue fortement. De plus, en ce qui concerne la température, deux variables différentes sont utilisées (*tempmin* et *tempmax*), ce qui réduit d'autant la corrélation.

6.4.9. Traitement de la variable jour de la semaine

Pour l'analyse de la mortalité, il peut être intéressant de tester une transformation *spline* de la variable *jour de la semaine*. Dans ce cas, en effet, l'influence des différents jours de la semaine est relativement faible. Cette transformation permet de réduire le nombre de degrés de liberté utilisés. Si l'**AIC** diminue significativement, la transformation est justifiée.

Pour la morbidité, il vaut mieux traiter la variable *jour de la semaine* sous forme qualitative car celle-ci intervient de façon importante et contrastée sur l'activité des systèmes de soins.

Pour transformer la variable jour de la semaine par une fonction *spline* il faut tout d'abord créer une variable quantitative

```
morta$dowf.num_as.numeric(weekdays(morta$date.study)
```

Puis il faut tester `s(dowf,3)` ou `s(dowf,4)` et calculer l'AIC.

Les modèles sont les suivants :

```
mortot73.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,3)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)

mortot74.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)
```

- AIC

```
AIC(mortot73.gam)

AIC(mortot74.gam)
```

- PACF des résidus

```
sum(acf(resid(mortot73.gam),type="p")$acf)
sum(acf(resid(mortot74.gam),type="p")$acf)
```

- Graphe des résidus

```
plot(resid(mortot73.gam))
plot(resid(mortot74.gam))
```

- Prédites et observées (courbes superposées)

```
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax2)&
!is.na(morta$hummin2)&!is.na(morta$gripa7)&!is.na(morta$gripb6)&
morta$mortot<16],fitted(mortot73.gam),col=2)


plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax2)&
!is.na(morta$hummin2)&!is.na(morta$gripa7)&!is.na(morta$gripb6)&
morta$mortot<16],fitted(mortot74.gam),col=3)
```

- Graphes des effets partiels

```
plot.gam(mortot73.gam)
plot.gam(mortot74.gam)
```

6.4.10. Traitement de la variable indicateur de pollution

Lorsqu'on écrit le modèle initial, comme on l'a vu, plusieurs approches sont possibles dont deux principales.

6.4.10.1. Si le polluant a été introduit sans transformation avec un décalage 0-1 jour dans le modèle initial

L'hypothèse retenue est celle d'une relation linéaire sans seuil entre la moyenne des niveaux du polluant du jour même et du jour j-1. La variable *polluant* est donc introduite dans le modèle sans transformation, sous forme d'une moyenne des concentrations du jour même et de la veille.

L'ensemble du modèle est testé avec un polluant quelconque (à part l'ozone qui est particulier en raison de la faiblesse de ses taux hivernaux).

Pour les autres polluants excepté l'ozone, le modèle reste le même. En effet, un polluant est choisi au départ pour prendre en compte son effet sur la PACF ; il faut changer de modèle si les périodes d'étude changent pour chaque polluant.

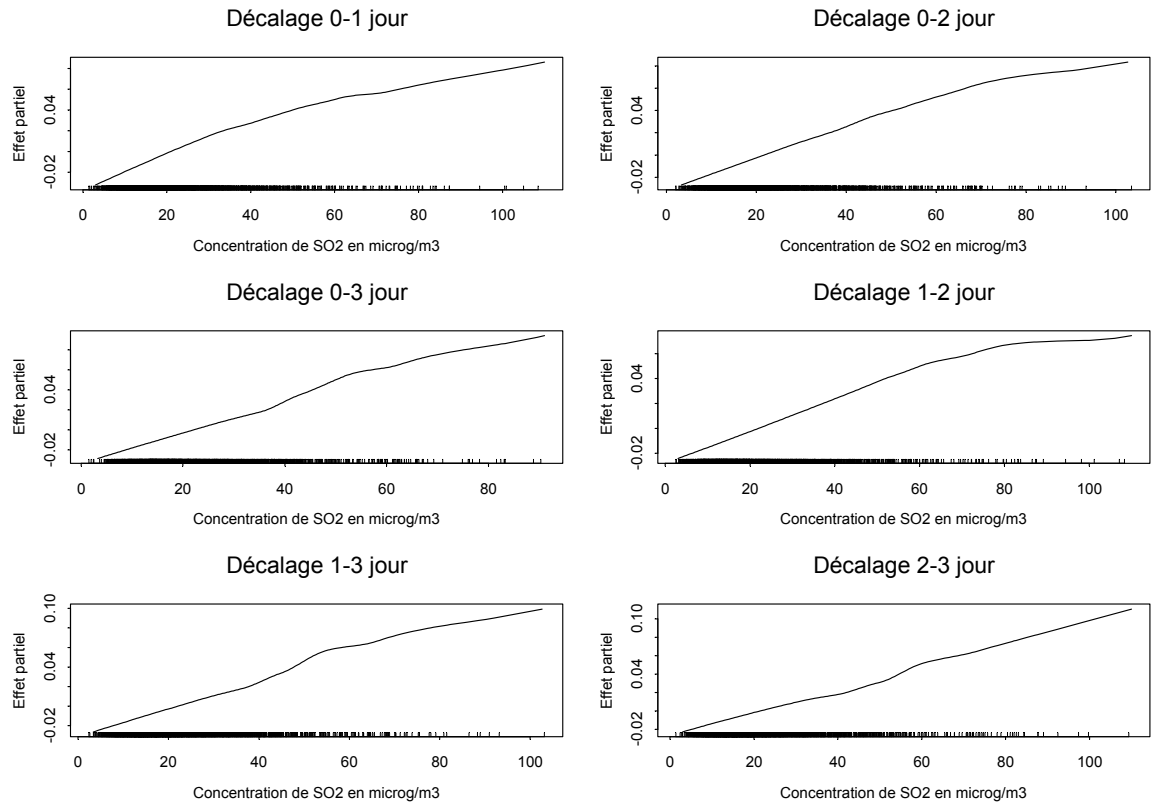
6.4.10.2. Introduction non paramétrique du polluant avec un décalage 0-1 jour dans le modèle initial puis test du décalage et de la forme paramétrique de la relation polluant - variable sanitaire

6.4.10.2.1. Choix du décalage

Il n'y a pas de test à proprement parler pour le choix du décalage. Deux critères sont utilisés :

- La façon dont varie le coefficient du polluant dans la mesure de l'adéquation du modèle et le calcul des effets des différents facteurs ;
- Le graphe de la relation polluant-variable sanitaire (surtout s'il y a une cohérence quant à la croissance de la pente qui doit être monotone croissante, d'un décalage à l'autre) (figure 95) ;
- L'AIC peut aider.

Figure 95. Graphes représentatifs des effets des différents décalages du polluant sur l'indicateur de santé.




```
summary.glm(mortot.gam)
plot.gam(mortot.gam)
```

Il ne faut tenir compte que du fait qu'un ou (mieux) plusieurs décalages présente(nt) une relation croissante. Si un décalage a une relation croissante avec la mortalité, il est retenu. Quand plusieurs décalages consécutifs ont une pente croissante, il faut calculer la relation pour la moyenne des valeurs du polluant à ces différents décalages (dans le graphe ci-dessus, on choisirait le décalage 0-3 jours).

En cas de doute, il convient de donner la préférence aux retards cumulés. Il peut arriver que les effets des retards successifs ne soient pas cohérents (effets positifs puis négatifs pour des décalages consécutifs, par exemple) ; dans ce cas, il faut être prudent et ne pas conclure à un effet significatif du polluant.

Remarque. Ceci peut permettre de diminuer le bruit dans la relation (diminution de l'intervalle de confiance du coefficient) si les relations sont homogènes sur les différents jours : en effet, si l'analyse porte sur la moyenne, la variance est *a priori* plus faible. De plus, comme la pollution a un effet sur plusieurs jours, le fait d'estimer son impact sur ces jours donne une estimation plus juste (i.e. « plus proche de la réalité ») car moins liée à des conditions d'expositions particulières à une journée.

Sur la base de ce qui vient d'être dit, Il faut tester le modèle avec des décalages 0, 1, 2, 3, 4 et 5 jours puis avec les moyennes 0-1, 0-2, 0-3 jours puis 1-2, 1-3, 2-3 jours.

Les décalages sont calculés de la façon suivante :

```
morta$so224h01_(morta$so224h+morta$so224h1)/2

……….

morta$so224h12_(morta$so224h1+morta$so224h2)/2

……….

morta$so224h13_(morta$so224h1+morta$so224h2+morta$so224h3)/3

……….
```

Construction des modèles :

```
mortot80.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),family=quasi(log,mu),
data=morta,na=na.omit,subset=mortot<16)

mortot81.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h1,.9),family=quasi(log,mu),data=mo
rta,na=na.omit,subset=mortot<16)

mortot82.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h2,.9),family=quasi(log,mu),data=mo
rta,na=na.omit,subset=mortot<16)

mortot83.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h3,.9),family=quasi(log,mu),data=mo
rta,na=na.omit,subset=mortot<16)

mortot84.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h4,.9),family=quasi(log,mu),data=mo
rta,na=na.omit,subset=mortot<16)

mortot85.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h5,.9),family=quasi(log,mu),data=mo
rta,na=na.omit,subset=mortot<16)

mortot801.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h01,.9),
family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

```
mortot802.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h02,.9),
family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot803.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h03,.9),
family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot812.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h12,.9),
family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot813.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h13,.9),
family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot823.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h23,.9),
family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

Contrôle des paramètres :

```
summary.glm(mortot80.gam,cor=F)

summary.glm(mortot81.gam,cor=F)

summary.glm(mortot82.gam,cor=F)

……
```

6.4.10.2.2. Détermination de la forme de la relation paramétrique

Les relations paramétriques entre le polluant et l'indicateur sanitaire les plus fréquemment testées sont les relations linéaire, logarithmique, racine carrée et quadratique.

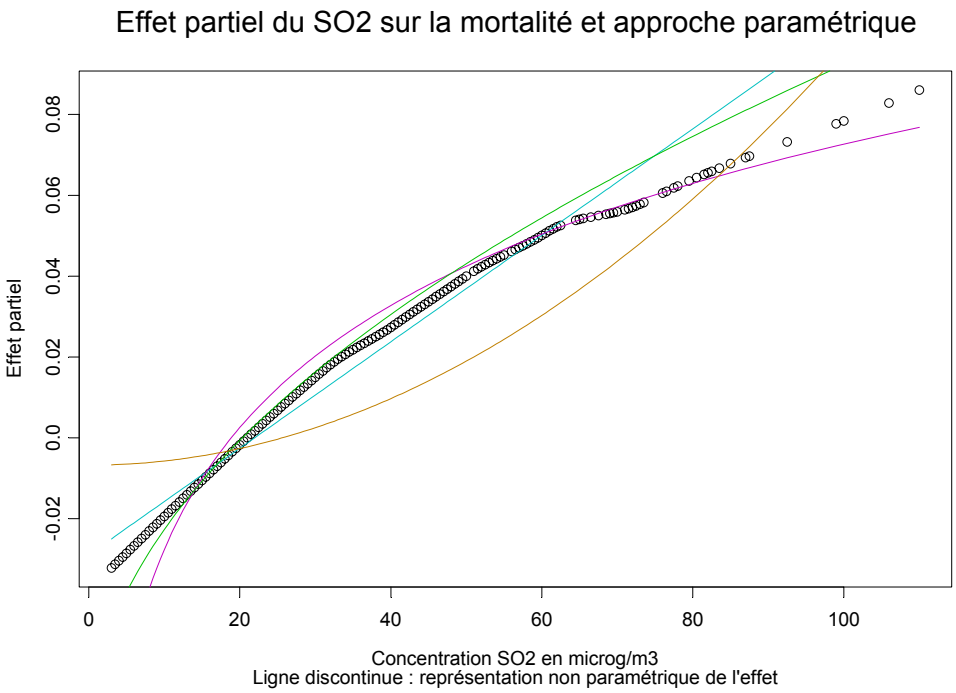
L'observation de la forme du graphe de l'effet partiel du polluant (`plot.gam` dans S-PLUS) permet d'orienter la recherche de la meilleure fonction paramétrique (figure 96).

Les modèles sont les suivants :

```
Modèle de base (loess)

mortot.fit.so224h0to1_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+
j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+
lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h01,.9),
family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)


Modèle linéaire

mortot.fit.so224h0to1.lin_gam(mortot~lo(trend,730/2922)+
s(dowf.num,4)+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+
lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+
so224h01,family=quasi(log,mu),data=morta,na=na.omit, subset=mortot<16)
```

```
Modèle logarithme népérien

mortot.fit.so224h0to1.log_gam(mortot~lo(trend,730/2922)+
s(dowf.num,4)+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+
lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)
+log(so224h01),family=quasi(log,mu),data=morta,na=na.omit,
subset=mortot<16)


Modèle racine quarrée

mortot.fit.so224h0to1.sqrt_gam(mortot~lo(trend,730/2922)+
s(dowf.num,4)+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+
lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+
sqrt(so224h01),family=quasi(log,mu),data=morta,na=na.omit,
subset=mortot<16)


Modèle quadratique

mortot.fit.so224h0to1.quad_gam(mortot~lo(trend,730/2922)+
s(dowf.num,4)+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+
lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+
so224h01^2,family=quasi(log,mu),data=morta,na=na.omit, subset=mortot<16)
```

Figure 96. Forme de la relation exposition-indicateur sanitaire et différentes approches paramétriques.



Le modèle choisi, il faut réaliser un *summary.glm* pour connaître les estimations des coefficients ainsi que les intervalles de confiance correspondants.

```
summary.glm(mortot.fit.so224h0to1.FONCTION.PARAMETRIQUE,cor=F)
```

Exemple :

```
summary.glm(morresp.fit.no224h0.lin,cor=F)
```

```
Call: gam(formula = morresp ~ lo(trend, 310./2922.) + s(dowf.num, 3.) +
j.feries + vac + lo(gripa7, 0.9) + lo(gripb5, 0.9) + lo(tempmin, 0.9)+
lo(hummin, 0.9) + lo(tempmax3, 0.9) + no224h, family = quasi(log, mu), data
= morta, subset = morresp <= 2., na.action = na.omit)

Deviance Residuals:
       Min         1Q     Median         3Q      Max
 -1.002568 -0.9975722 -0.9958124 0.7940866 3.969267

Coefficients:
                          Value  Std. Error    t value
        (Intercept) -0.577797862 0.089534431 -6.4533594
lo(trend, 310/2922)  4.096796819 1.226931177  3.3390600
     s(dowf.num, 3) -0.009204480 0.011245858 -0.8184773
           j.feries  0.084310171 0.060761533  1.3875583
                vac  0.001135781 0.025482055  0.0445718
    lo(gripa7, 0.9)  3.873578867 1.279449056  3.0275366
    lo(gripb5, 0.9)  2.487897223 1.093535562  2.2750949
   lo(tempmin, 0.9) -1.155873118 2.277872871 -0.5074353
    lo(hummin, 0.9) -3.528187608 1.468235970 -2.4030113
  lo(tempmax3, 0.9) -1.901053723 2.430011487 -0.7823229
             no224h  0.003271580 0.001329852  2.4601084

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8190512 )

    Null Deviance: 2697.662 on 2790 degrees of freedom

Residual Deviance: 2618.365 on 2757.7694306 degrees of freedom

Number of Fisher Scoring Iterations: 4
```

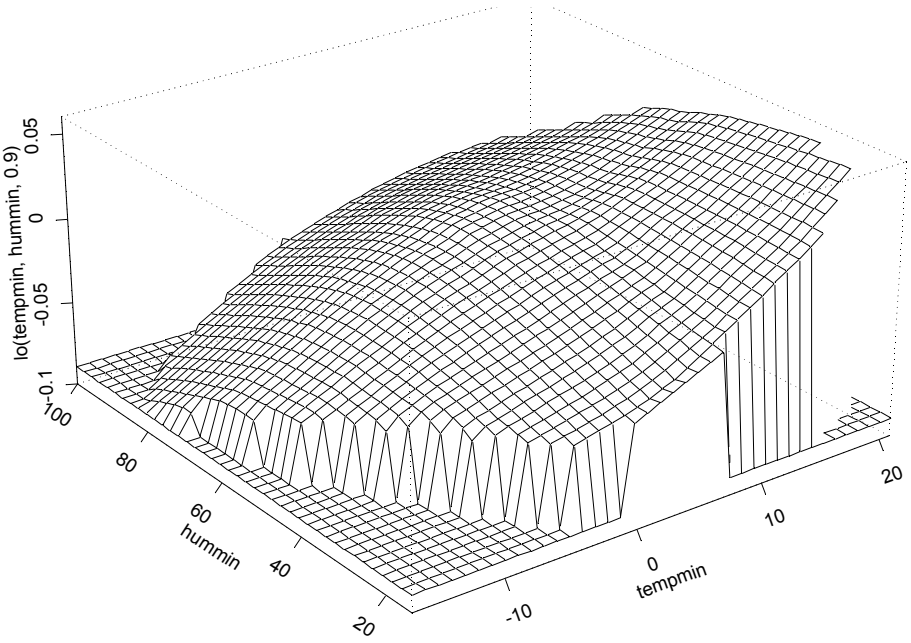
6.4.11. Gestion des autocorrélations persistant dans le modèle

Il se peut qu'en dépit des ajustements successifs et du soin apporté à la prise en compte de l'autocorrélation dans la série des résidus, quelques pics d'**autocorrélation** persistent sur les premiers retards. En théorie, s'il existe une autocorrélation sur un retard, il faut rajouter un terme autorégressif. La prise en compte de cette autocorrélation résiduelle permet d'avoir une inférence valide sur la variance, donc sur les tests de significativité des paramètres. Le modèle étant marginal, la prise en compte de l'autocorrélation résiduelle se fait au niveau de la définition de la matrice de variance-covariance. Ce modèle marginal peut être, sous certaines conditions, *approximé* par un modèle conditionnel. Dans ce cas, les termes autorégressifs sont inclus comme variables explicatives. L'intérêt de cette approche est la simplicité d'estimation. C'est le parti pris dans la fonction développée.

6.4.12. Test de l'interaction température-humidité

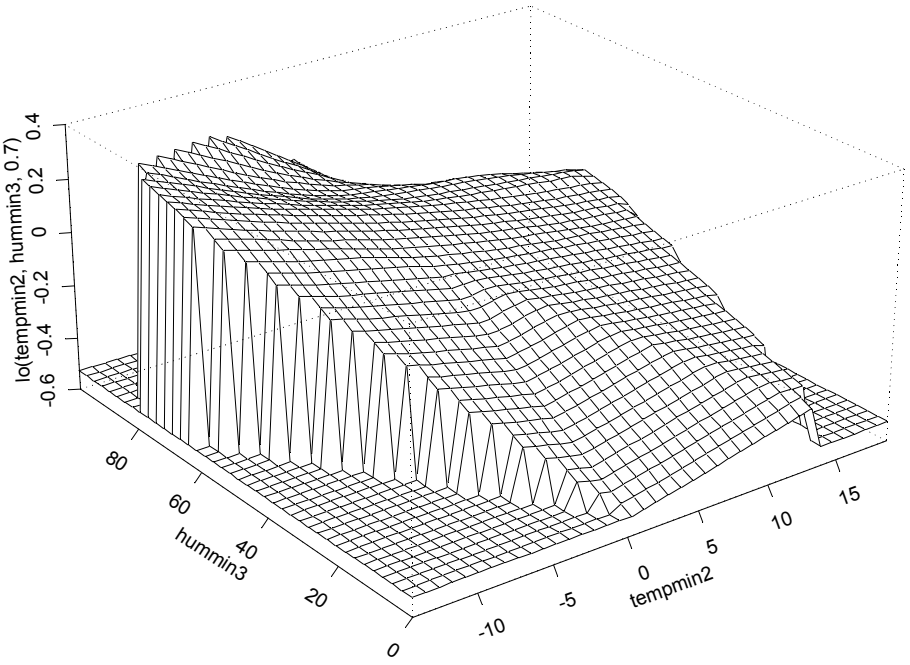
La température et l'humidité peuvent, dans certains cas interagir dans leurs effets respectifs sur la variable sanitaire. Cette interaction doit être prise en compte. Elle est testée sur la base de la minimisation de l'**AIC** ou, en cas d'incertitude quant au choix (peu de diminution de l'AIC), sur l'**aspect du graphique des effets partiels** (la modification de la forme de la surface pour certains intervalles de températures et/ou humidité est le témoin d'une interaction) (figures 97 et 98).

Figure 97. Interaction température humidité : peu d'interaction



Sur ce graphe, on n'observe pas d'interaction majeure entre la température (`tempmin`) et l'humidité (`hummin`).

Figure 98. Interaction température humidité : interaction visible



Ce graphe montre une interaction entre la température (`tempmin`) et l'humidité (`hummin`).

Remarque 1. Ajouter un terme d'interaction augmente le nombre de degrés de liberté ce qui peut-être préjudiciable à la stabilité du modèle ; il faut, par conséquent, être attentif à ne pas rajouter de terme d'interaction si cela n'est pas nécessaire.

Remarque 2. Il semble évident que, lorsque la température et l'humidité sont affectées de décalages différents, elle n'interagissent pas au sens physique du terme ; il est possible, toutefois, d'envisager qu'il y ait interaction au niveau des effets, par exemple que la température ait un effet trois ou quatre jours après une "préparation" par l'humidité ; Il faut avoir une argumentation solide concernant la plausibilité biologique ; cependant, il semble inutile d'étudier l'interaction de la température et de l'humidité avec des décalages différents.

6.4.13. Analyses de sensibilité

Les analyses de sensibilité ont pour objectif de tester la robustesse du modèle.

6.4.13.1. Suppression des valeurs extrêmes du polluant

Quand le modèle est déterminé, les valeurs extrêmes des concentrations de l'indicateur de pollution (celles qui se trouvent en deçà du percentile 5 ou au delà du percentile 95) sont éliminées. Le coefficient du polluant dans le modèle sans valeurs extrêmes est comparé au coefficient du polluant dans le modèle avec valeurs extrêmes. Si une différence apparaît, il faut vérifier que les valeurs extrêmes ne sont pas aberrantes (erreurs de mesures, conditions particulières). Si c'est le cas, il faut supprimer ces valeurs car sinon, la relation trouvée serait alors biaisée.

Les différentes analyses adéquates sont :

Analyse préliminaire : visualisation graphique de la distribution

```
boxplot(morta$so224h)
```

Calcul des paramètres généraux

```
summary(morta$so224h)
 Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
    3      13     18 22.04      28  124   52
```

Détermination des percentiles 5 et 95

```
quantile(morta$so224h, c(.05,.95),na.rm=T)
  5% 95%
  7  49
```

Écriture du modèle sans valeurs extrêmes

```
mortot91p.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h1,.9),family=quasi(log,mu),data=mo
rta,na=na.omit,subset=(mortot<16&so224h1>7&so224h1<49))
```

Calcul des coefficients

```
summary.glm(mortot91p.gam,cor=F)
```

Remarque. Une autre façon de gérer les valeurs aberrantes est le recours à une régression robuste, moins sensible à celles-ci [30])

6.4.13.2. Suppression des valeurs extrêmes de la température

Le procédé est semblable au précédent (éliminations des mesures qui se trouvent à l'extérieur de l'intervalle percentile 5-percentile 95 et comparaison des coefficients du polluant). S'il apparaît une différence, il peut être préférable de conserver le modèle sans valeurs extrêmes.

Les différentes analyses à réaliser sont les suivantes :

Visualisation graphique de la distribution de la température

```
boxplot(morta$tempmin)
```

Calcul des paramètres généraux

```
summary(morta$tempmin)
 Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
    3      13     18 22.04      28  124   52
```

Détermination des percentiles 5 et 95

```
quantile(morta$tempmin, c(.05,.95),na.rm=T)
  5% 95%
  7  49
```

Écriture du modèle

```
mortot9mn.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+
vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+
lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h1,.9),family=quasi(log,mu),data=mo
rta,na=na.omit,subset=(mortot<16&tempmin>7&tempmin<49))
```

Calcul des coefficients

```
summary.glm(mortot9mn.gam)
```

6.4.13.3. Variation des décalages de la température et de l'humidité

L'ensemble des décalages des facteurs météorologiques est à nouveau testé. Si le modèle est robuste, le coefficient du polluant doit rester relativement constant et le sens de la relation polluant-variable sanitaire doit rester le même quel que soit le décalage.

```
summary.glm(mortot10.gam)
```

6.4.13.4. Variation de la fenêtre de la tendance

Là aussi, si le modèle est robuste, le coefficient du polluant doit rester constant et le sens de la relation polluant-variable sanitaire doit rester le même quel que soit la taille de la fenêtre.

```
summary.glm(mortot11.gam)
```

6.4.14. Retards polynomiaux

L'analyse de la littérature montre une grande diversité dans les modalités de prise en compte des effets à court terme, retardés, de la pollution sur la santé. Certaines études sélectionnent le retard brut correspondant à l'effet le plus significatif (la gamme des retards explorés peut aller de 0 à 3 jours, ou 0 à 5 jours, voire plus). D'autres études analysent les effets cumulés (0 à 3 jours, etc.). Or, il est

raisonnable de penser que l'impact sanitaire du niveau de polluant observé un jour donné s'étend sur plusieurs jours et qu'il est très vraisemblablement distribué selon une certaine loi décroissante au cours du temps. Pour répondre à cette problématique et afin d'unifier les approches diverses concernant la prise en compte de ces effets retardés, il a été proposé de construire des modèles à retards échelonnés. Ces modèles, utilisés depuis une vingtaine d'années dans les sciences sociales [50], ont été appliqués plus récemment au domaine de l'épidémiologie de la pollution atmosphérique [51,52]. Il est ainsi possible d'analyser, à l'aide des retards polynomiaux de degré 3, l'effet cumulatif sur 5 jours pour déterminer la répartition des effets cumulés à court terme [40]. Pour l'ozone, l'introduction d'une variable d'interaction ozone/été entraîne des zones de « rupture » dans la série et ne permet pas d'appliquer les modèles à retard échelonnés qui nécessitent, la prise en compte, en continu, des cinq jours précédant l'événement sanitaire. Le modèle à retards échelonnés peut s'écrire de la façon suivante :

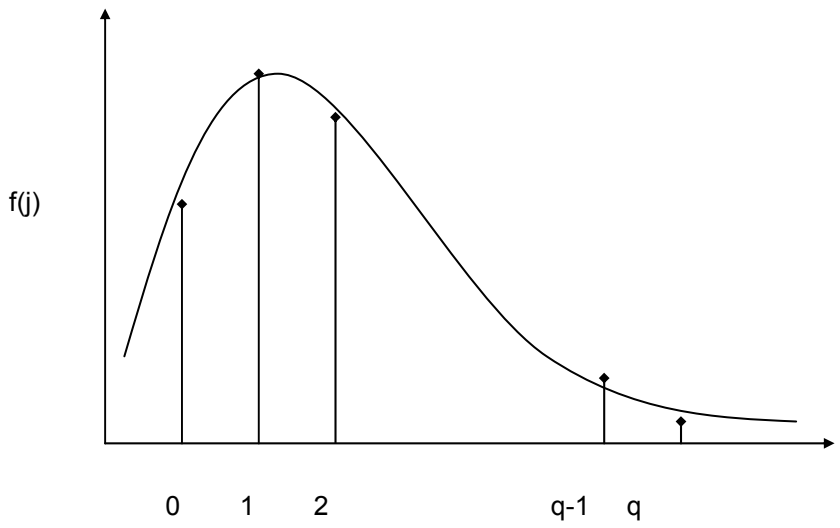
$$\ln\left(E\left[Y_t\right]\right) = a + \sum_{j=1}^{p} f_j\left(X_{jt}\right) + b_0 Z_{0t} + b_1 Z_{1t} + \ldots b_q Z_{qt}$$

La lettre a désigne une constante, Y_t est la variable *nombres journaliers d'événements sanitaires*, $E[Y_t]$ est l'espérance du *nombre journalier d'événements sanitaires*, les X_{it} sont les covariables (autres que le polluant), les f_j sont des fonctions de lissage, Z_{0t} est l'exposition au polluant, contemporaine de l'effet sanitaire, Z_{1t} l'exposition de la veille, Z_{qt} l'exposition q jours avant l'événement sanitaire (jour t-q), les b_j sont des paramètres à estimer. Ce modèle est sans contrainte, c'est-à-dire que les valeurs du polluant aux différents retards sont incluses simultanément dans le modèle. En fait, la présence d'une forte colinéarité entre les variables retardées (*i.e.* les expositions retardées) entraîne une grande instabilité des estimateurs des différents retards. Néanmoins, leur somme est sans biais, même si pour un grand nombre de retards elle devient inefficiente (*i.e.* leur variance augmente excessivement). Pour pallier à ce problème et à l'imprécision qui en résulte, le nombre de paramètres à estimer peut être réduit en imposant des contraintes à la distribution des coefficients b_m, m = 0,1, …, q, au cours du temps. La méthode basée sur le retard d'Almon [53] permet d'estimer les coefficients b_m, indirectement, en supposant qu'ils peuvent être approchés par une fonction polynomiale du décalage (figure 99). Ainsi, b_m est approché par β_m tels que :

$$\beta_m = \beta(m) = \sum_{k=0}^{d} \theta_k m^k \quad \text{pour m = 0,1, …, q}$$

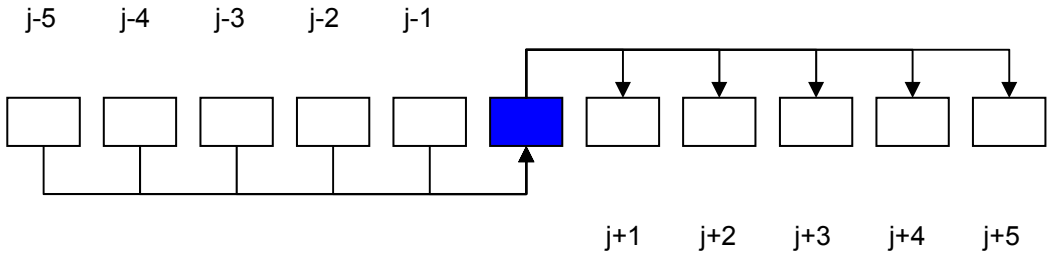
où m est le retard, d le degré du polynôme et les θ_k, k = 0, 1, …, d, des coefficients à estimer. Le choix du degré du polynôme utilisé pour l'approximation représente une certaine difficulté. Imposer une contrainte trop forte (i.e. choisir un polynôme de degré faible) peut biaiser le risque et produire une distorsion de la distribution. À l'opposé, une contrainte trop faible (*i.e.* choisir un polynôme de degré élevé) produit des estimateurs trop bruyants pour être informatifs. Dans cette optique, il apparait pertinent de choisir un polynôme d'ordre 3 pour les retards 0 à 5 jours afin de déterminer la répartition des effets à court terme (voir exemple à l'Annexe 3).

Figure 99. Fonction polynomiale du décalage



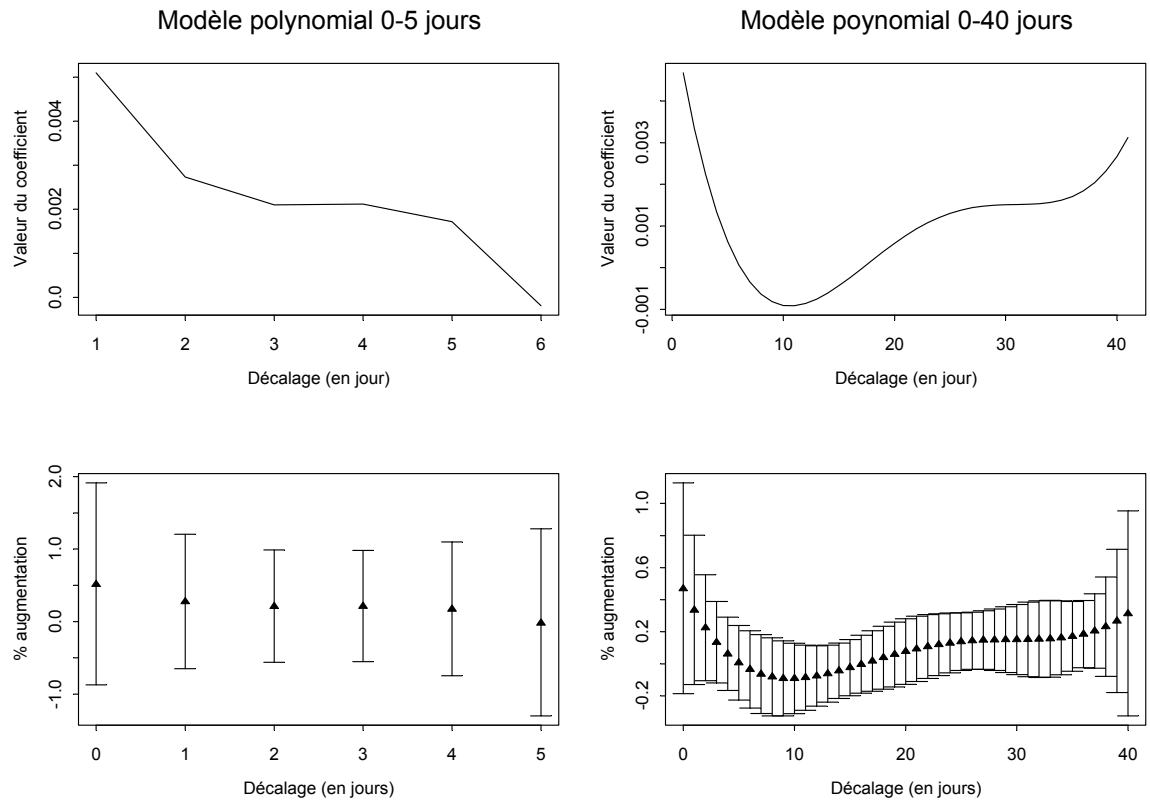
Remarque. La méthode des retards polynomiaux estime l'effet de l'exposition du jour et des 5 jours précédents sur l'indicateur sanitaire du jour et, ce qui revient au même, la distribution de l'effet de l'exposition d'un jour sur l'indicateur sanitaire du jour et des 5 jours suivants (figure 100). En effet, on fait l'hypothèse que l'impact de l'exposition du jour t-Δt sur l'indicateur sanitaire du jour t ne dépend que de Δt ; cet impact est donc le même que celui de l'exposition du jour t sur l'indicateur sanitaire du jour t+Δt.

Figure 100. Effets de l'exposition distribués



Ci-dessous, figure un exemple de représentation des effets cumulés (figure 101).

Figure 101. Modèles à retards échelonnés. Effets du SO_2 sur la période 0-5 jours et 0-40 jours et excès de risque (avec intervalle de confiance) en fonction du décalage



6.4.15. Cas particulier de l'ozone

Il a été vu plus haut (cf. § 6.1) que la variable ozone est introduite sous la forme d'une interaction avec une variable *été* valant 1 en été et 0 en hiver. A cette fin il faut créer une variable indicatrice pour l'été (été : avril à septembre).

La suite de commandes pour créer cette variable est :

```
Création d'une variable mois
mois_as.numeric(months(morta$date.study))

Création d'une variable été
morta$summer_mois>3&mois<10

1°) La variable « été » peut-être numérique
morta$summer.num_as.numeric(morta$summer))

2°) La variable « été » peut-être qualitative à deux classes
morta$summer.fac_as.factor(morta$summer))
```

Si la variable *été* est construite comme **variable numérique**, il est possible alors d'inclure dans le modèle l'interaction sous forme de fonction loess :

```
lo(o38h,summer.num)
```

Mais le graphe est difficile à interpréter, dans ce cas.

Si la variable *été* est qualitative à 2 classes, l'interaction sera incluse dans le modèle sous la forme :

```
summer.fac*o38h
```

Ceci permettra d'étudier l'effet de la variable été, de la variable ozone et de leur interaction.

Remarque. Il est possible de procéder à une modélisation pour l'été uniquement ; le modèle ne contient pas de variable grippe ; la variable ozone est introduite avec une fonction *loess*.

Ouvrages et articles recommandés

Les séries temporelles

Brockwell PJ, Davis RA. Time series in statistics. 1st ed. New York : Springer Verlag, 1987.

Gourieroux C, Monfort A. Séries temporelles et modèles dynamiques. 1st ed. Paris : Economica, 1990.

Les GLM et les GAM

Nelder JA, Wedderburn RWM. Generalized linear models. J R Statist Soc A 1972;135:370-84.

McCullagh P, Nelder JA. Generalized linear models. 2nd ed. London: Chapman & Hall, 1989.

Hastie TJ, Tibshirani RJ. Generalized Additive Models. 1st ed. London: Chapman & Hall, 1990.

La modélisation

Le Tertre A, Quénel P, Medina S et al. Modélisation des liens à court terme entre la pollution atmosphérique et la santé. Un exemple : SO2 et mortalité totale, Paris, 1987-1990 Rev Epidemiol Sante Publique 1998;46:316-28.

Institut de Veille Sanitaire. Programme de Surveillance Air et Santé 9 villes, Surveillance des effets sur la santé liés à la pollution atmosphérique en milieu urbain - Phase II : rapport de l'étude. Institut de Veille Sanitaire. Saint-Maurice, juin 2002.

Schwartz J. The distributed lag between air pollution and daily deaths. Epidemiology 2000;11:320-26.

Ramsay TO, Burnett RT, Krewski D. The effect of concurvity in generalized additive models linking mortality to ambiant particulate matter. Epidemiol 2003;14:18-23.

Le logiciel S-PLUS

Krause A, Olson M. The basics of S and S-PLUS. 1st ed. New York: Springer-Verlag, 1997.

Venables WN, Ripley BD. Modern applied statistics with S. 4th ed. New York: Springer-Verlag, 2002.

Dominici F, McDermott A, Zeger SL, Samet JM. On the use of generalized additive models in time-series studies of air pollution and health. Am J Epidemiol 2002;156;193-203.

Manuels du logiciel S-PLUS.

« R »

Voir le site CRAN (http://cran.r-project.org).

Glossaire

Autoregressive process

Processus autorégressif

Degree of freedom

Degré de liberté

Moving average process

Processus moyenne mobile

Smoother

Fonction de lissage

Système dynamique

Ensemble de variables dépendant du temps, le système dynamique est défini par une ensemble d'équations exprimant la variation de son état (valeurs des variables) en fonction du temps.

Lorsque le temps est pris comme ensemble discret, le système, lorsqu'il est du premier ordre s'exprime de la façon suivante : $X(t) = f[X(t-1)]$.

Exemple : X(t) = a X(t-1) [1 - X(t-1)], ce qui s'exprime, pour les observations par x(t) = a x(t-1) [1 - x(t-1)].

Remarque. Quand il s'agit de temps discret, il est d'usage de préférer la notation : $x_t = a\, x_{t-1}\, (1-x_{t-1})$.

Cet exemple est célèbre [19] : il est utilisé en étude des populations proies-prédateurs et mène à une illustration du chaos déterministe.

Lorsque le temps est considéré comme continu, le système s'écrit : $\frac{dX(t)}{dt} = f[X(t)]$.

Exemple : $\frac{dX(t)}{dt} = aX(t)$

Cet exemple est utilisé en théorie de la désintégration atomique mais est très « présent » en biologie.

La trajectoire de phases ou (trajectoire dans l'espace des phases) est la courbe représentative de tous les états du système, chaque point ayant pour coordonnées les valeurs des différentes variables en fonction du temps.

Index

R

S

T

V

Références

1 Katsouyanni K, Schwartz J, Spix C, Touloumi G, Zmirou D, Zanobetti A et al. Short-term effects of air pollution on health: a European approach using epidemiologic time series data: the Aphea protocol. Journal of Epidemiology and Community Health 1995;50:S12-S18.

2 Schwartz J, Spix C, Touloumi G, Bachárová L, Barumamdzadeh T, Le Tertre A et al. Methodological issues in studies of air pollution and daily counts of deaths or hospital admissions. J Epidemiol Community Health 1996;50(Suppl 1):S3-S11.

3 Schwartz J. Air pollution and hospital admissions for the elderly in Birmingham, Alabama. American Journal of Epidemiology 1994;139:589-98.

4 Schwartz J. Non parametric smoothing in the analysis of air pollution and respiratory illness. The Canadian Journal of Statistics 1994;22:471-87.

5 Daniels MJ, Dominici F, Samet JM, Zeger SL. Estimating particulate matter-mortality dose-responsee curves and threshold levels: an analysis of daily time-series for the largest US cities. Am J Epidemiol 2000;152:397-406.

6 Medina S, Le Tertre A, Quénel P, Le Moullec Y. Evaluation de l'impact de la pollution atmosphérique urbaine sur la santé en Ile-de-France (Etude Erpurs). ORS Ile-de-France 1994;104 pages.

7 Brockwell PJ, Davis RA. Time series in statistics. 1st ed. New York : Springer Verlag, 1987.

8 Droesbeke JJ, Fichet B, Tassi P ed. Séries chronologiques. Théorie et pratique des modèles ARIMA. Paris : Economica, 1989.

9 Gourieroux C, Monfort A. Séries temporelles et modèles dynamiques. 1st ed. Paris : Economica, 1990.

10 Coutrot B, Droesbeke JJ. Les méthodes de prévision. 2nd ed. Paris : Presses universitaires de France, 1990.(Que sais-je ?).

11 Giraud R, Chaix N. Économétrie. 2nd ed. Paris : Presses universitaires de France, 1994.

12 Bresson G, Pirotte A. Économétrie des séries temporelles. Théorie et applications 1st ed. Paris : Presses universitaires de France, 1995.

13 Amegandjin J. Démographie mathématique. Paris : Economica, 1989.

14 Levine B. Fondements théoriques de la radiotechnique statistique Tome III. Moscou : Éditions Mir, 1976. Traduction française, 1979.

15 Gourieroux C, Monfort A. Séries temporelles et modèles dynamiques. 1st ed. Paris : Economica, 1990.

16 Guégan D. Séries chronologiques non linéaires à temps discret. 1st ed. Paris : Economica, 1994.

17 Roberts GO. Markov chain concepts related to sampling algorithm. In: Gilks WR, Richardson S, Spiegelhalter DJ eds. Markov chain Monte Carlo in practice. 1st ed. Boca Raton: Chapman and Hall / CRC, 1996: pp 45-57.

18 Bergé P, Pomeau Y, Vidal C. L'ordre dans le chaos. Vers une approche déterministe de la turbulence. Paris: Hermann,1988.

19 May R M. Simple mathematical models with very complicated dynamics. Nature 1976; 261:459-67.

20 Schaffer WM, Kot M. Nearly one dimensional dynamics in an epidemic. J Theor Biol 1985;112:403-27.

21 Schaffer WM, Olsen LF, Truty GL, Fulmer SL, Graser DJ. 1988. Periodic and chaotic dynamics in childhood infections. In: Markus M, Mueller SC, Nicolis G eds. From chemical to biological organization. Berlin: Springer Verlag, 1988: pp 331-347.

22 Nelder JA, Wedderburn RWM. Generalized linear models. J R Statist Soc A 1972;135:370-84.

23 McCullagh P, Nelder JA. Generalized linear models. 2nd ed. London: Chapman & Hall, 1989.

24 Fahrmeir L, Tutz G. Multivariate statistical modelling based on generalized linear models. 2nd rev ed. New York : Springer Verlag,1996.

25 Lindsey JK. Applying generalized linear models. G Casella, S Fienberg, I Olkin eds. Springer-Verlag, New York,1997.

26 Zeghnoun A. Relation à court terme entre pollution atmosphérique et santé. Quelques aspects statistiques et épidémiologiques. Thèse doctorat Université Paris VII. Spécialité Biostatistiques, 2002.

27 Wedderburn R. Quasi-liquelihood, generalized linear models, and the Gauss-Newton method. Biometrika 1974;61:439-447.

28 McCullagh P. Quasi-likelihood fuctions. Ann Statist 1983;11:59-67.

29 Morris C. Natural exponential families with quadratic variance functions. Ann Statist 1982;10:65-80.

30 Hastie TJ, Tibshirani RJ. Generalized Additive Models. 1st ed. London: Chapman & Hall, 190.

31 Cleveland WS, Devlin SJ. Locally-weighted regression, An approach to regression analysis by local fitting. J Am Statist Assoc 1988;83:597-610.

32 MathSoft, Data Analysis Products Division. S-Plus 4, Guide to statistics. Seattle, Washington: MathSoft, Inc, 1997.

33 Venables WN, Ripley BD. Modern applied statistics with S. 4th ed. New York: Springer.

34 Monfort A. Cours de statistique mathématique. 3rd ed. Paris : Economica, 1997.

35 Akaike H. Information theory and an extension of the maximum likelihood principle, in Second International Symposium on Information Theory. (eds B.N. Petrov and F. Czáki). Akademiai Kiadó, Budapest 1973:267-81.

36 Akaike H. A new look at the Bayes procedure. Biometrika 1978;65:53-9.

37 Akaike H. On the likelihood of a time series model. Statistician 1978;27:215-35.

38 Institut de Veille Sanitaire. Surveillance des effets sur la santé liés à la pollution atmosphérique en milieu urbain : rapport de l'étude. Institut de Veille Sanitaire. Saint-Maurice, mars 1999.

39 Le Tertre A, Quénel P, Medina S et al. Modélisation des liens à court terme entre la pollution atmosphérique et la santé. Un exemple : SO2 et mortalité totale, Paris, 1987-1990 Rev Epidemiol Sante Publique 1998;46:316-28.

40 Institut de Veille Sanitaire. Programme de Surveillance Air et Santé 9 villes, Surveillance des effets sur la santé liés à la pollution atmosphérique en milieu urbain - Phase II : rapport de l'étude. Institut de veille sanitaire. Saint-Maurice, juin 2002.

41 Robert C. L'analyse statistique bayésienne. 1st ed. Paris : Economica, 1992.

42 Marinucci D, Petrella L. A Bayesian proposal for the analysis of stationary and nonstationary AR(1) time series. In: Bernardo JM, Berger JO, Dawid AP, Smith AFM. Bayesian statistics 6. New York: Oxford University Press, 1999:821-38.

43 Rosenberg MA, Young VR. A bayesian approach to understanding time series data. North Am Actuarial J 1999;3:130-43.

44 Droesbecke J, Fine J, Saporta G. Methodes bayésiennes en statistique. Paris: Technip, 2002.

45 Robert CP. Quelques modèles de séries temporelles. In: JJ Droesbecke, J Fine, G Saporta eds. Methodes bayésiennes en statistique. Paris: Technip, 2002:279-94.

46 Gilks WR, Richardson S, Spiegelhalter DJ. Markov Chain Monte Carlo in practice. New York: Chapman & Hall Interdisciplinary Statistics, 1996.

47 Dominici F, McDermott A, Zeger SL, Samet JM. On the use of generalized additive models in time-series studies of air pollution and health. Am J Epidemiol 2002;156;193-203.

48 Samet JM. The National Morbidity, Mortality, and Air Pollution Study. Part 1: Methods and Methodologic Issues. Health Effects Institute, 2000.

49 Samet JM. The National Morbidity, Mortality, and Air Pollution Study. Part 2: National Morbidity, Mortality and Air Pollution in the United States. Health Effects Institute, 2000.

50 Judge GG, Griffiths WE, Hill RC, Lutkepohl H, Lee TC. The theory and practice of econometrics. New York: John Wiley & Sons, 1980.

51 Schwartz J. The distributed lag between air pollution and daily deaths. Epidemiology 2000;11:320-26.

52 Pope CA, Schwartz J. Time series for the analysis of pulmonary health data. American Journal of Respiratory and Critical Care Medicine 1996 ;154:S229-S233.

53 Almon S. The distributed lag between capital appropriations and expenditures. Econometrica 1962;30:407-23.

Annexes

Annexe 1. Données

Tableau 4. Marches aléatoires normale, uniforme et de Poisson

	dnorm1 [a]	y1 [b]	dnorm2	y2	dnorm3	y3	dnorm4	y4
1		0		0		0		0
2	-0,118210365	-0,118210365	0,416617632	0,416617632	-0,667433185	-0,667433185	0,559419244	0,559419244
3	-0,4127843	-0,530994665	1,452998225	1,869615856	-1,720296139	-2,387729324	0,898038181	1,457457425
4	-1,23487036	-1,765865025	-2,296184282	-0,426568426	-0,996965186	-3,384694511	1,826680713	3,284138138
5	0,186767869	-1,579097156	-0,941443548	-1,368011974	-0,696180626	-4,080875136	0,599517084	3,883655222
6	-0,381043767	-1,960140923	-0,324810211	-1,692822185	0,214429022	-3,866446114	0,549893228	4,43354845
7	-0,041880067	-2,00202099	-0,229250888	-1,922073073	1,638945389	-2,227500725	1,736641593	6,170190042
8	-1,151558186	-3,153579176	0,712101032	-1,209972041	-0,079887295	-2,307388021	-1,065868973	5,104321069
9	-0,070774995	-3,224354171	-0,247450652	-1,457422693	-0,172056927	-2,479444948	-1,943064684	3,161256384
10	-1,110114098	-4,334468269	0,820317973	-0,637104719	1,393409902	-1,086035046	1,386965557	4,548221942
11	-0,823208587	-5,157676857	-0,484664327	-1,121769046	-0,584157736	-1,670192783	1,661348256	6,209570198
12	2,759256184	-2,398420672	-1,094435147	-2,216204193	-0,633505691	-2,303698473	0,431173439	6,640743636
13	1,181264182	-1,217156491	-1,416376407	-3,632580601	-0,023495537	-2,32719401	2,325667742	8,966411378
14	-1,835460443	-3,052616934	0,793861735	-2,838718866	-1,127484582	-3,454678592	0,922327032	9,88873841
15	1,067378011	-1,985238923	0,174781361	-2,663937505	1,140915869	-2,313762723	1,857476472	11,74621488
16	0,118100288	-1,867138634	-0,265337417	-2,929274922	-1,777645813	-4,091408536	1,55985527	13,30607015
17	-1,170681832	-3,037820466	-1,870853673	-4,800128596	-1,54271944	-5,634127976	0,740612389	14,04668254
18	-0,230776781	-3,268597248	2,491658747	-2,308469849	1,019283895	-4,614844081	-1,357641174	12,68904137
19	0,478144925	-2,790452323	0,73984866	-1,568621189	-0,722047822	-5,336891903	-0,346768694	12,34227267
20	-1,445047274	-4,235499597	0,21234471	-1,356276478	0,984537513	-4,35235439	1,045015903	13,38728858
21	1,048300233	-3,187199363	-0,063578258	-1,419854737	-0,101070736	-4,453425126	0,049832878	13,43712145
22	-0,469485215	-3,656684579	-0,071565411	-1,491420147	-0,116086739	-4,569511865	-0,127356716	13,30976474
23	0,769870176	-2,886814403	0,819263183	-0,672156965	-1,80168953	-6,371201394	1,284846942	14,59461168
24	0,864072218	-2,022742184	-0,45759047	-1,129747434	1,121080324	-5,25012107	-1,36527299	13,22933869
25	1,226289758	-0,796452426	1,27054205	0,140794616	1,249389218	-4,000731852	1,632532693	14,86187138
26	0,036956823	-0,759495603	0,339507178	0,480301793	0,601784313	-3,398947539	-0,665891397	14,19597999
27	-0,541799383	-1,301294986	1,348071002	1,828372795	-0,641777801	-4,04072534	0,426938773	14,62291876
28	-0,100559909	-1,401854896	-0,646204394	1,182168402	1,091450259	-2,949275081	0,406785705	15,02970446
29	-1,14974149	-2,551596386	-0,242850116	0,939318285	0,635233177	-2,314041904	-0,111635972	14,91806849
30	-1,551535677	-4,103132063	-2,588840782	-1,649522497	1,499349765	-0,814692139	-0,548276085	14,36979241
31	0,048204937	-4,054927126	-2,02172318	-3,671245677	-2,419610124	-3,234302264	-0,631339638	13,73845277
32	1,458171323	-2,596755804	0,207479867	-3,463765811	0,124404064	-3,1098982	2,40137255	16,13982532
33	0,667873838	-1,928881966	-0,715331528	-4,179097339	1,937418677	-1,172479523	1,00436884	17,14419416
34	-0,345342959	-2,274224924	0,365367986	-3,813729353	1,164568194	-0,007911329	0,844619684	17,98881384
35	-0,866150648	-3,140375572	0,633872846	-3,179856507	0,911126819	0,90321549	-0,09052339	17,89829045
36	0,230827348	-2,909548224	-0,665671464	-3,845527971	-0,129122244	0,774093246	0,36471479	18,26300524
37	-0,025644333	-2,935192557	-0,374950967	-4,220478938	0,654303773	1,428397019	-1,794385735	16,46861951
38	-2,521737799	-5,456930357	-0,541627147	-4,762106086	-1,069715241	0,358681778	-1,102802396	15,36581711

39	-1,996585127	-7,453515483	1,407518909	-3,354587176	0,651093583	1,009775361	0,431292128	15,79710924
40	-1,194035773	-8,647551257	-1,965386441	-5,319973617	0,132464073	1,142239434	-1,366613766	14,43049547
41	1,067642092	-7,579909164	1,679750814	-3,640222804	1,733441338	2,875680773	-1,398313579	13,03218189
42	-1,355420885	-8,93533005	-0,852134936	-4,49235774	-0,299111666	2,576569106	-0,629077679	12,40310421
43	-1,066535407	-10,00186546	0,189768369	-4,302589371	-0,636411414	1,940157693	-0,024295392	12,37880882
44	-1,054151632	-11,05601709	0,371845617	-3,930743754	-2,2007467	-0,260589008	-0,127994236	12,25081459
45	0,114689243	-10,94132785	-1,270474296	-5,20121805	1,071826327	0,811237319	0,560424334	12,81123892
46	-0,936277334	-11,87760518	-2,426162423	-7,627380473	-1,880049059	-1,06881174	-0,3816728	12,42956612
47	-0,669786276	-12,54739146	0,773465427	-6,853915046	0,41846182	-0,65034992	-1,536616855	10,89294927
48	-1,852842018	-14,40023347	1,037531877	-5,816383169	1,626870055	0,976520136	-0,210734578	10,68221469
49	-0,552914695	-14,95314817	0,26518165	-5,551201518	0,328546463	1,305066598	0,443250355	11,12546504
50	0,517710477	-14,43543769	-0,190688944	-5,741890462	-0,998468329	0,306598269	-0,213152851	10,91231219
51	-1,053620724	-15,48905842	0,696502616	-5,045387846	0,479348369	0,785946638	1,11348078	12,02579297
52	-0,125971266	-15,61502968	-0,84531611	-5,890703956	1,895482849	2,681429487	-0,667148554	11,35864442
53	0,109034966	-15,50599472	0,284964216	-5,60573974	0,632631772	3,314061259	0,689552166	12,04819658
54	1,571606965	-13,93438775	0,341206131	-5,264533609	-0,509316581	2,804744678	-1,123133212	10,92506337
55	-0,032597499	-13,96698525	0,963148888	-4,301384721	-0,144554439	2,660190238	0,931132051	11,85619542
56	0,665652424	-13,30133282	0,488591563	-3,812793157	0,208128199	2,868318437	-0,329730773	11,52646465
57	0,877837692	-12,42349513	-1,043678293	-4,85647145	1,376325509	4,244643946	-0,800525286	10,72593936
58	-0,598585085	-13,02208022	1,873946677	-2,982524773	-1,515417825	2,729226121	2,14995616	12,87589552
59	-1,334825394	-14,35690561	1,25103098	-1,731493794	-0,208776422	2,520449699	-0,560778408	12,31511712
60	2,412443554	-11,94446206	-1,230029616	-2,96152341	2,853760551	5,37421025	0,995768499	13,31088561
61	0,045400513	-11,89906155	0,615212065	-2,346311344	0,648905132	6,023115382	0,917405651	14,22829127
62	1,347928225	-10,55113332	0,845792677	-1,500518667	1,085049867	7,108165249	-0,740336809	13,48795446
63	-0,756658497	-11,30779182	-0,260922734	-1,761441401	1,601245011	8,709410261	-1,208666915	12,27928754
64	-0,357230937	-11,66502275	0,320713844	-1,440727557	-0,8235063	7,885903961	1,015572066	13,29485961
65	0,130674414	-11,53434834	0,314411454	-1,126316103	0,704791067	8,590695028	-0,973586628	12,32127298
66	0,946541438	-10,5878069	1,625649798	0,499333695	-1,926324678	6,66437035	1,37027395	13,69154693
67	0,027815804	-10,5599911	-0,385359502	0,113974193	0,500557488	7,164927839	-0,060015244	13,63153169
68	1,755810071	-8,804181027	-0,315671201	-0,201697008	0,680373112	7,845300951	0,389808231	14,02133992
69	-1,385535791	-10,18971682	-2,280601813	-2,482298821	-0,292847167	7,552453784	1,573839315	15,59517923

	dunif1	**y1**	**dunif2**	**y2**	**dunif3**	**y3**	**dunif4**	**y4**
1		0		0		0		0
2	-0,474392445	-0,474392445	0,085930927	0,085930927	-0,437804502	-0,437804502	0,968290481	0,968290481
3	0,866172448	0,391780003	-0,88134061	-0,795409683	0,25490505	-0,182899452	0,904498378	1,872788859
4	0,08887417	0,480654173	0,369466113	-0,42594357	0,172835155	-0,010064296	0,806965133	2,679753992
5	0,703826056	1,184480228	-0,704770616	-1,130714186	-0,994913856	-1,004978152	0,815403795	3,495157787
6	0,754109922	1,93859015	-0,175212075	-1,305926261	0,932098316	-0,072879836	-0,635419955	2,859737831
7	-0,233974827	1,704615324	-0,532757982	-1,838684243	0,634393269	0,561513433	-0,552944837	2,306792994
8	-0,70379626	1,000819064	0,207428999	-1,631255244	-0,652932593	-0,091419159	-0,202606604	2,10418639
9	0,364086915	1,364905979	-0,454593214	-2,085848458	-0,405593638	-0,497012798	0,679882281	2,78406867
10	-0,315260212	1,049645767	0,917292613	-1,168555846	-0,248713302	-0,7457261	-0,150865904	2,633202766
11	-0,88867444	0,160971327	-0,300225892	-1,468781738	-0,35212355	-1,09784965	0,575976091	3,209178857
12	-0,821310079	-0,660338752	0,238098003	-1,230683735	0,680721736	-0,417127915	-0,629703733	2,579475123
13	-0,406870111	-1,067208863	0,841457033	-0,389226702	-0,821313949	-1,238441864	-0,659236476	1,920238648

14	-0,838715686	-1,905924548	-0,504518969	-0,893745671	-0,89875481	-2,137196674	0,761718594	2,681957241
15	-0,924870023	-2,830794571	0,431909744	-0,461835927	0,440219838	-1,696976836	-0,401011736	2,280945505
16	-0,846473833	-3,677268404	0,053957468	-0,407878459	-0,716314252	-2,413291088	0,496563563	2,777509068
17	-0,178314094	-3,855582497	0,541924194	0,134045735	0,142731215	-2,270559873	-0,701935842	2,075573226
18	0,596620074	-3,258962424	0,730039014	0,864084749	-0,660654696	-2,931214569	-0,875472628	1,200100599
19	-0,57128579	-3,830248213	-0,552829641	0,311255108	-0,553007271	-3,48422184	-0,160652222	1,039448377
20	-0,207189551	-4,037437764	-0,044007035	0,267248073	0,425611884	-3,058609956	-0,442932175	0,596516201
21	-0,606659749	-4,644097513	-0,240392663	0,02685541	-0,395503688	-3,454113644	-0,128946229	0,467569972
22	-0,888908778	-5,533006291	-0,082776158	-0,055920748	-0,527520939	-3,981634582	0,119660926	0,587230898
23	0,371018222	-5,161988069	-0,013656636	-0,069577384	0,515408976	-3,466225606	-0,167804085	0,419426814
24	-0,717954629	-5,879942698	-0,978363266	-1,04794065	0,081126397	-3,385099209	0,34496996	0,764396774
25	0,415158045	-5,464784654	-0,50577505	-1,5537157	0,488870005	-2,896229204	0,473526273	1,237923047
26	-0,079888768	-5,544673421	0,128483559	-1,425232141	-0,879975142	-3,776204346	0,361795565	1,599718612
27	-0,606698223	-6,151371645	-0,909705513	-2,334937654	0,45770534	-3,318499005	-0,281425253	1,318293359
28	-0,723452193	-6,874823838	-0,15481594	-2,489753595	-0,107315022	-3,425814027	-0,912804356	0,405489003
29	0,974627372	-5,900196466	-0,811355201	-3,301108795	0,311162933	-3,114651094	0,611727161	1,017216165
30	0,030901683	-5,869294783	-0,64309813	-3,944206925	0,950240511	-2,164410583	-0,134020458	0,883195707
31	0,982666945	-4,886627838	0,00345628	-3,940750645	0,319893442	-1,844517141	0,700787801	1,583983508
32	-0,207766814	-5,094394652	0,461713542	-3,479037103	0,230182999	-1,614334142	-0,019489877	1,564493631
33	-0,889369657	-5,983764309	0,451190153	-3,02784695	0,491674395	-1,122659747	-0,219203723	1,345289908
34	0,44542702	-5,538337289	-0,300311488	-3,328158438	0,035870199	-1,086789547	0,379640551	1,72493046
35	-0,101450084	-5,639787373	-0,576790838	-3,904949276	-0,289359045	-1,376148593	-0,467851143	1,257079316
36	-0,030777088	-5,670564461	-0,666737736	-4,571687012	-0,284579958	-1,660728551	-0,310414393	0,946664924
37	-0,829163698	-6,499728159	0,307388918	-4,264298094	0,362655411	-1,298073139	-0,481897784	0,464767139
38	-0,185813954	-6,685542113	0,416185654	-3,848112441	0,043432188	-1,254640951	0,00637314	0,471140279
39	0,99249417	-5,693047944	0,542753572	-3,305358869	0,099639083	-1,155001868	0,845753477	1,316893756
40	-0,673934255	-6,366982198	-0,162411203	-3,467770072	0,785914375	-0,369087492	0,800774197	2,117667953
41	0,878247934	-5,488734264	0,244448715	-3,223321357	-0,992133445	-1,361220937	0,340600275	2,458268228
42	0,02259305	-5,466141214	0,636461318	-2,586860039	-0,48566157	-1,846882507	0,274667782	2,73293601
43	-0,670622599	-6,136763813	0,038054703	-2,548805336	-0,585819052	-2,432701559	-0,87788233	1,85505368
44	-0,689073169	-6,825836982	0,987244158	-1,561561178	-0,072396503	-2,505098062	0,673052662	2,528106342
45	-0,821831306	-7,647668288	-0,403055686	-1,964616864	-0,442717136	-2,947815198	-0,224191502	2,30391484
46	-0,783906847	-8,431575135	0,399951865	-1,564664999	0,993353954	-1,954461244	0,466384826	2,770299667
47	-0,523704642	-8,955279778	-0,87464054	-2,439305539	0,61109525	-1,343365994	0,067216072	2,837515739
48	-0,263112844	-9,218392622	-0,652046548	-3,091352087	-0,39243496	-1,735800955	-0,96888232	1,868633418
49	0,025535552	-9,19285707	-0,932039475	-4,023391563	0,709759107	-1,026041848	-0,522234221	1,346399197
50	-0,197909435	-9,390766505	-0,0653437	-4,088735263	0,120323023	-0,905718825	-0,296699454	1,049699743
51	-0,461146255	-9,85191276	0,527509268	-3,561225994	0,848822393	-0,056896431	-0,626965305	0,422734438
52	0,883648532	-8,968264229	0,860106584	-2,701119411	0,32016113	0,263264699	-0,120301368	0,302433071
53	-0,297088945	-9,265353174	0,535491169	-2,165628241	-0,163306128	0,099958571	0,857235855	1,159668926
54	0,994405156	-8,270948018	-0,342815723	-2,508443965	-0,389421098	-0,289462527	0,892639077	2,052308003
55	-0,250086322	-8,52103434	-0,651745403	-3,160189368	0,49145143	0,201988903	-0,3593751	1,692932904
56	0,575420321	-7,945614019	0,421109729	-2,739079639	-0,50228252	-0,300293617	0,966089009	2,659021913
57	0,153413251	-7,792200768	0,147316376	-2,591763264	-0,167831302	-0,468124919	-0,556796211	2,102225702
58	0,862169635	-6,930031134	0,814707456	-1,777055807	0,943292555	0,475167637	0,118970914	2,221196616
59	-0,368095342	-7,298126476	-0,974269609	-2,751325416	-0,440226627	0,03494101	-0,824062024	1,397134592

60	0,055393586	-7,24273289	-0,218539704	-2,96986512	0,631928857	0,666869868	0,147289302	1,544423894
61	-0,396015388	-7,638748278	-0,584227553	-3,554092673	-0,620850492	0,046019375	-0,56624392	0,978179974
62	0,989371576	-6,649376702	-0,8588937	-4,412986373	0,396581539	0,442600914	0,978034538	1,956214513
63	-0,713892793	-7,363269495	-0,136499346	-4,549485719	0,813472522	1,256073437	0,258673791	2,214888304
64	0,406567579	-6,956701916	-0,46897137	-5,018457089	-0,46414441	0,791929027	0,164183393	2,379071697
65	-0,885440732	-7,842142648	0,034460301	-4,983996788	0,359307449	1,151236476	0,427332642	2,806404339
66	0,762640652	-7,079501996	0,269635533	-4,714361255	0,972262495	2,123498972	-0,218918032	2,587486307
67	0,924665701	-6,154836295	-0,083722018	-4,798083273	0,122986329	2,2464853	0,882851728	3,470338035
68	-0,886640705	-7,041477	-0,381232825	-5,179316098	0,337797554	2,584282855	0,115800621	3,586138656
69	-0,975197873	-8,016674873	-0,102442165	-5,281758263	0,690108697	3,274391552	-0,250028709	3,336109947

	dpois1	**y1**	**dpois2**	**y2**	**dpois3**	**y3**	**dpois4**	**y4**
1		0		0		0		0
2	2	2	2	2	5	5	2	2
3	7	9	5	7	2	7	1	3
4	1	10	4	11	1	8	1	4
5	4	14	2	13	1	9	4	8
6	4	18	3	16	5	14	3	11
7	6	24	1	17	4	18	5	16
8	2	26	4	21	4	22	5	21
9	3	29	5	26	3	25	1	22
10	3	32	2	28	3	28	6	28
11	3	35	2	30	6	34	2	30
12	5	40	5	35	2	36	3	33
13	6	46	10	45	1	37	6	39
14	4	50	2	47	1	38	4	43
15	1	51	4	51	3	41	2	45
16	1	52	5	56	3	44	6	51
17	7	59	5	61	2	46	0	51
18	2	61	1	62	2	48	2	53
19	1	62	4	66	4	52	1	54
20	4	66	4	70	4	56	0	54
21	3	69	6	76	3	59	3	57
22	5	74	5	81	1	60	1	58
23	4	78	3	84	3	63	4	62
24	3	81	2	86	0	63	4	66
25	1	82	4	90	1	64	1	67
26	7	89	1	91	0	64	4	71
27	4	93	2	93	1	65	2	73
28	1	94	5	98	2	67	2	75
29	1	95	5	103	3	70	4	79
30	2	97	4	107	3	73	2	81
31	1	98	4	111	4	77	7	88
32	1	99	2	113	5	82	1	89
33	2	101	2	115	5	87	2	91
34	5	106	3	118	3	90	4	95

35	0	106	9	127	4	94	1	96
36	2	108	1	128	6	100	1	97
37	4	112	4	132	2	102	3	100
38	1	113	6	138	3	105	1	101
39	3	116	5	143	8	113	2	103
40	3	119	1	144	2	115	2	105
41	1	120	3	147	7	122	1	106
42	2	122	3	150	1	123	3	109
43	1	123	2	152	2	125	2	111
44	2	125	4	156	1	126	3	114
45	3	128	2	158	4	130	2	116
46	3	131	3	161	4	134	5	121
47	2	133	2	163	3	137	2	123
48	5	138	7	170	2	139	3	126
49	3	141	2	172	3	142	3	129
50	2	143	2	174	4	146	4	133
51	4	147	2	176	3	149	0	133
52	1	148	5	181	3	152	2	135
53	2	150	2	183	4	156	3	138
54	2	152	4	187	4	160	4	142
55	4	156	2	189	2	162	1	143
56	1	157	5	194	2	164	0	143
57	2	159	2	196	3	167	2	145
58	7	166	2	198	5	172	8	153
59	4	170	3	201	3	175	2	155
60	7	177	2	203	1	176	2	157
61	4	181	4	207	3	179	2	159
62	5	186	3	210	2	181	0	159
63	5	191	2	212	3	184	2	161
64	2	193	2	214	3	187	2	163
65	2	195	6	220	4	191	5	168
66	2	197	4	224	3	194	5	173
67	0	197	3	227	1	195	4	177
68	0	197	1	228	4	199	5	182
69	1	198	1	229	2	201	3	185

[a] Nombres au hasard tirés des distributions normales (dnorm), uniforme (dunif), de Poisson (dpois) ;
[b] Marches aléatoires normales, uniforme, de Poisson.

Annexe 2. Calculs

Tableau 5. Corrélogrammes des séries « valeur » et « valeur sans bruit »

h	Y	y - z	h	y	y - z	h	y	y - z	H	y	y - z
0	1	1	40	0,2122	0,3645	80	0,1237	0,2247	120	0,3928	0,7242
1	0,5755	0,9969	41	0,2307	0,3459	81	0,1324	0,2375	121	0,4286	0,7261
2	0,5589	0,9929	42	0,1948	0,3278	82	0,154	0,2508	122	0,4046	0,7272
3	0,5513	0,9878	43	0,1858	0,3102	83	0,1343	0,2646	123	0,4359	0,7274
4	0,5534	0,9818	44	0,1681	0,2932	84	0,1474	0,2789	124	0,416	0,7268
5	0,5529	0,9748	45	0,1923	0,2768	85	0,1771	0,2935	125	0,4138	0,7253
6	0,528	0,9669	46	0,1591	0,261	86	0,1703	0,3085	126	0,4225	0,7229
7	0,5456	0,958	47	0,1319	0,246	87	0,1886	0,3238	127	0,3998	0,7196
8	0,5511	0,9483	48	0,1652	0,2316	88	0,1755	0,3394	128	0,4039	0,7155
9	0,5333	0,9376	49	0,1145	0,2179	89	0,1966	0,3551	129	0,3844	0,7106
10	0,5043	0,9261	50	0,1293	0,2051	90	0,2017	0,3711	130	0,3858	0,7048
11	0,5077	0,9138	51	0,1072	0,193	91	0,2103	0,3872	131	0,3634	0,6982
12	0,5054	0,9006	52	0,1188	0,1817	92	0,2185	0,4034	132	0,3927	0,6907
13	0,5358	0,8867	53	0,1199	0,1713	93	0,2466	0,4196	133	0,4032	0,6825
14	0,5029	0,8721	54	0,1154	0,1617	94	0,2557	0,4358	134	0,3861	0,6735
15	0,4723	0,8567	55	0,1013	0,153	95	0,2583	0,452	135	0,3947	0,6637
16	0,4808	0,8406	56	0,1048	0,1452	96	0,2821	0,4681	136	0,3876	0,6532
17	0,4558	0,8239	57	0,1151	0,1383	97	0,2743	0,484	137	0,3648	0,6419
18	0,4699	0,8066	58	0,1058	0,1323	98	0,2911	0,4998	138	0,3713	0,63
19	0,4575	0,7888	59	0,0921	0,1273	99	0,2939	0,5154	139	0,3553	0,6174
20	0,4585	0,7704	60	0,0842	0,1231	100	0,2783	0,5306	140	0,3547	0,6042
21	0,4459	0,7516	61	0,1106	0,1199	101	0,3027	0,5456	141	0,3563	0,5904
22	0,409	0,7323	62	0,1031	0,1177	102	0,3116	0,5602	142	0,3207	0,576
23	0,4168	0,7126	63	0,0834	0,1163	103	0,3221	0,5745	143	0,3311	0,5611
24	0,3501	0,6926	64	0,0988	0,1159	104	0,3339	0,5883	144	0,3187	0,5456
25	0,3648	0,6723	65	0,0833	0,1164	105	0,3404	0,6016	145	0,3045	0,5297
26	0,367	0,6517	66	0,0804	0,1179	106	0,3469	0,6144	146	0,3013	0,5134
27	0,3796	0,631	67	0,0877	0,1202	107	0,3489	0,6267	147	0,2772	0,4967
28	0,3453	0,61	68	0,0947	0,1235	108	0,3484	0,6384	148	0,2773	0,4796
29	0,3457	0,589	69	0,077	0,1276	109	0,3679	0,6495	149	0,285	0,4622
30	0,2984	0,5679	70	0,0716	0,1326	110	0,3641	0,66	150	0,2653	0,4445
31	0,3172	0,5468	71	0,0847	0,1384	111	0,3375	0,6698			
32	0,3013	0,5258	72	0,1204	0,145	112	0,3841	0,6789			
33	0,2826	0,5048	73	0,0998	0,1525	113	0,4037	0,6873			
34	0,2823	0,484	74	0,0922	0,1607	114	0,382	0,695			
35	0,2817	0,4634	75	0,1236	0,1697	115	0,4061	0,7019			
36	0,2695	0,4429	76	0,1316	0,1794	116	0,3996	0,708			
37	0,2645	0,4228	77	0,1295	0,1898	117	0,411	0,7133			
38	0,2117	0,403	78	0,1315	0,2008	118	0,4019	0,7177			
39	0,2265	0,3835	79	0,1233	0,2125	119	0,3863	0,7214			

Ce tableau se rapporte au corrélogramme du § 2.4.2 (*Figure 21. Corrélogramme de la série temporelle*).

Annexe 3. Exemple de calcul de retard polynomial

A titre d'exemple, si le polynôme est supposé de degré trois :

$$\beta_j = f(j) = \theta_0 + \theta_1 j + \theta_2 j^2 + \theta_3 j^3$$

Alors :

$$\begin{aligned}
\beta_0 &= f(0) = \theta_0 \\
\beta_1 &= f(1) = \theta_0 + \theta_1 + \theta_2 + \theta_3 \\
\beta_2 &= f(2) = \theta_0 + 2\theta_1 + 4\theta_2 + 8\theta_3 \\
\beta_3 &= f(3) = \theta_0 + 3\theta_1 + 9\theta_2 + 27\theta_3 \\
&\ldots \\
\beta_q &= f(q) = \theta_0 + q\theta_1 + q^2\theta_2 + q^3\theta_3
\end{aligned} \qquad (3)$$

En reportant (3) dans (1), en assimilant les b_j aux β_j, et en factorisant selon les coefficients θ_k, k = 0, …, 3, l'expression du modèle GAM devient :

$$\begin{aligned}
\ln(E[Y]) = a + \sum_{i=1}^{p} S_i(X_i) &+ \theta_0(Z_0 + Z_1 + \ldots + Z_q) \quad + \\
&+ \theta_1(Z_1 + 2Z_2 + \ldots + qZ_q) \quad + \\
&+ \theta_2(Z_1 + 4Z_2 + \ldots + q^2 Z_q) + \\
&+ \theta_3(Z_1 + 8Z_2 + \ldots + q^3 Z_q)
\end{aligned} \qquad (4)$$

En posant $W_0 = Z_0 + Z_1 + \ldots + Z_q, \ldots, W_3 = Z_1 + 8Z_2 + \ldots + q^3 Z_q$, le modèle à retards échelonnés s'écrit :

$$\ln(E[Y]) = a + \sum_{i=1}^{p} S_i(X_i) + \theta_0 W_0 + \theta_1 W_1 + \theta_2 W_2 + \theta_3 W_3$$

Les quatre nouvelles variables explicatives W_0, W_1, W_2, W_3 sont des combinaisons linéaires des variables d'exposition retardées. Le modèle donne des estimations des coefficients θ_k, k = 0, 1, …, 3, et leurs variances qui, à leur tour, permettent d'estimer les coefficients b_j, j=1, 2, …, q, ainsi que leur variance.

Annexe 4. Résumé de la procédure

À chaque étape les 2 tests les plus importants sont la PACF et le graphique des résidus. Ensuite on regarde les valeurs observées et les valeurs prédites. L'AIC ne sert que pour tester les températures et l'humidité ou, dans le but d'avoir un critère "objectif", la taille des fenêtres.

Entre parenthèses, les numéros des chapitres concernés.

1. ANALYSE DESCRIPTIVE (§ 6.3.)

1.1. PARAMÈTRES (§ 6.3.1.)

1.2. GRAPHES (§ 6.3.2.)

1.3. BOXPLOTS (§ 6.3.3/)

1.4. PACF (§ 6.3.4.)

2. ANALYSE PRÉLIMINAIRE

2.1. ÉCRIRE LE MODÈLE COMPLET (§ 6.1.)

Le modèle initial contient à priori :

```
morta.gam_gam(mortot~lo(trend,183/(nbjours))+dowf+j.feries+vac+lo(grip,.7)+
lo(tempmin,.7)+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log
,mu),data=morta,na=na.omit)
```

2.2. ANALYSE (§ 6.2.)

- PACF des résidus

- Graphe des résidus

- Prédites et observées (courbes superposées)

- Plot partiel (plot.gam)

- AIC

- Dispersion

2.3. ANALYSE DE SENSIBILITÉ (§ 6.4.1)

Élimination des valeurs extrêmes des données sanitaires (supérieures à la médiane+2*écarts_interquartiles)

- PACF des résidus

2.4. SI BESOIN, MODIFICATION DE LA VARIABLE GRIPPE (§ 6.4.2.)

- Prédites et observées (courbes superposées)

3. AJUSTER LA TAILLE DES FENÊTRES (§ 6.4.4)

3.1. AUGMENTATION DES FENÊTRES DES VARIABLES AUTRES QUE LA TENDANCE (§ 6.4.4.1.)

- PACF des résidus

S'arrêter quand on a obtenu le minimum de la somme des autocorrélations et le PACF le plus conforme

- Graphe des résidus

- Plot partiel (plot.gam)

- AIC

Pour confirmer le choix de la fenêtre (on privilégie le choix issu de PACF)

3.2. FAIRE VARIER LA FENÊTRE DE LA TENDANCE (§ 6.4.4.2.)

Pas de limite supérieure, limite inférieure : 100 jours

- PACF des résidus

- Graphe des résidus

- AIC

Pour confirmer le choix de la fenêtre (on privilégie le choix issu de PACF)

- Plot partiel (plot.gam)

4. AJUSTER LA GRIPPE A DIFFÉRENTS RETARDS AINSI QUE POUR DIFFÉRENTES FENÊTRES (§ 6.4.5.)

4.X. ON TESTE gripX

4.X.1. On fait varier le décalage

- AIC

- PACF des résidus

- plot.gam

4.X.2. Puis on peut faire varier la fenêtre

- AIC

- PACF des résidus

- plot.gam

5. AJUSTER TEMPMAX (§ 6.4.7.)

tempmin0+hummin0+tempmax à différents lags >=1 et <=3 et leurs moyennes (tempmax1, tempmax2,tempmax3, tempmax12, tempmax23, tempmax123)

- AIC

Pour choisir le décalage

- PACF des résidus

Pour contrôler l'impact si choix par AIC

Pour choix du décalage si choix non fait par AIC

- Plot partiel (plot.gam)

6. AJOUTER ET AJUSTER HUMMIN A DIFFERENTS DECALAGES (§ 6.4.8.)

hummin à différents lags >=1 et <=3 et leurs moyennes

. Si on hésite, regarder la PACF.

- AIC

Si l'AIC ne varie pas beaucoup, on ne garde pas hummin_lag

- PACF des résidus

Si on hésite sur la base de l'AIC

- Plot partiel (plot.gam)

7. TRAITEMENT DE LA VARIABLE JOURS_DE_LA_SEMAINE (§ 6.4.9.)

- AIC

Si l'AIC ne varie pas beaucoup, on ne garde pas la transformation *spline*.

- PACF des résidus

Si on hésite sur la base de l'AIC

8. TRAITEMENT DE LA VARIABLE INDICATEUR DE POLLUTION (§ 6.4.10.)

Deux cas, selon que le modèle initial contient le polluant sous forme paramétrique ou non paramétrique.

8.1. SI LE POLLUANT EST SANS TRANSFORMATION AVEC UN DÉCALAGE 0-1 (§ 6.4.10.1)

`Summary.glm`

8.2. SI LE POLLUANT EST SOUS FORME NON-PARAMÉTRIQUE (§ 6.4.10.2.)

Choix du décalage : coefficients (`summary.glm`) et surtout *plot partiel* (`plot.gam`)

Choix de la relation paramétrique : *plot partiel*

9. GESTION DES AUTOCORRÉLATIONS PERSISTANT DANS LE MODELE (§ 6.4.11.)

10. TEST DE L'INTERACTION TEMPERATURE-HUMIDITE (§ 6.4.12.)

- AIC
- Plot partiel (plot.gam)

11. ANALYSE DE SENSIBILITE (§ 6.4.13.)

11.1. SUPRESSION DES VALEURS EXTRÊMES DU POLLUANT (P5 et P95) (§ 6.4.13.1.)

- coefficients (summary.glm)
- Plot partiel (plot.gam)

11.2. SUPRESSION DES VALEURS EXTREMES DE LA TEMPERATURE (P5 et P95) (§ 6.4.13.2.)

- coefficients (summary.glm)
- Plot partiel (plot.gam)

11.3. VARIATION DES DECALAGES DE LA TEMPERATURE ET DE L'HUMIDITE (§ 6.4.13.3.)

- coefficients (summary.glm)
- Plot partiel (plot.gam)

11.4. VARIATION DE LA FENÊTRE DE LA TENDANCE (§ 6.4.13.4.)

- coefficients (summary.glm)
- Plot partiel (plot.gam)

12. RETARDS POLYNOMIAUX (§ 6.4.14.)

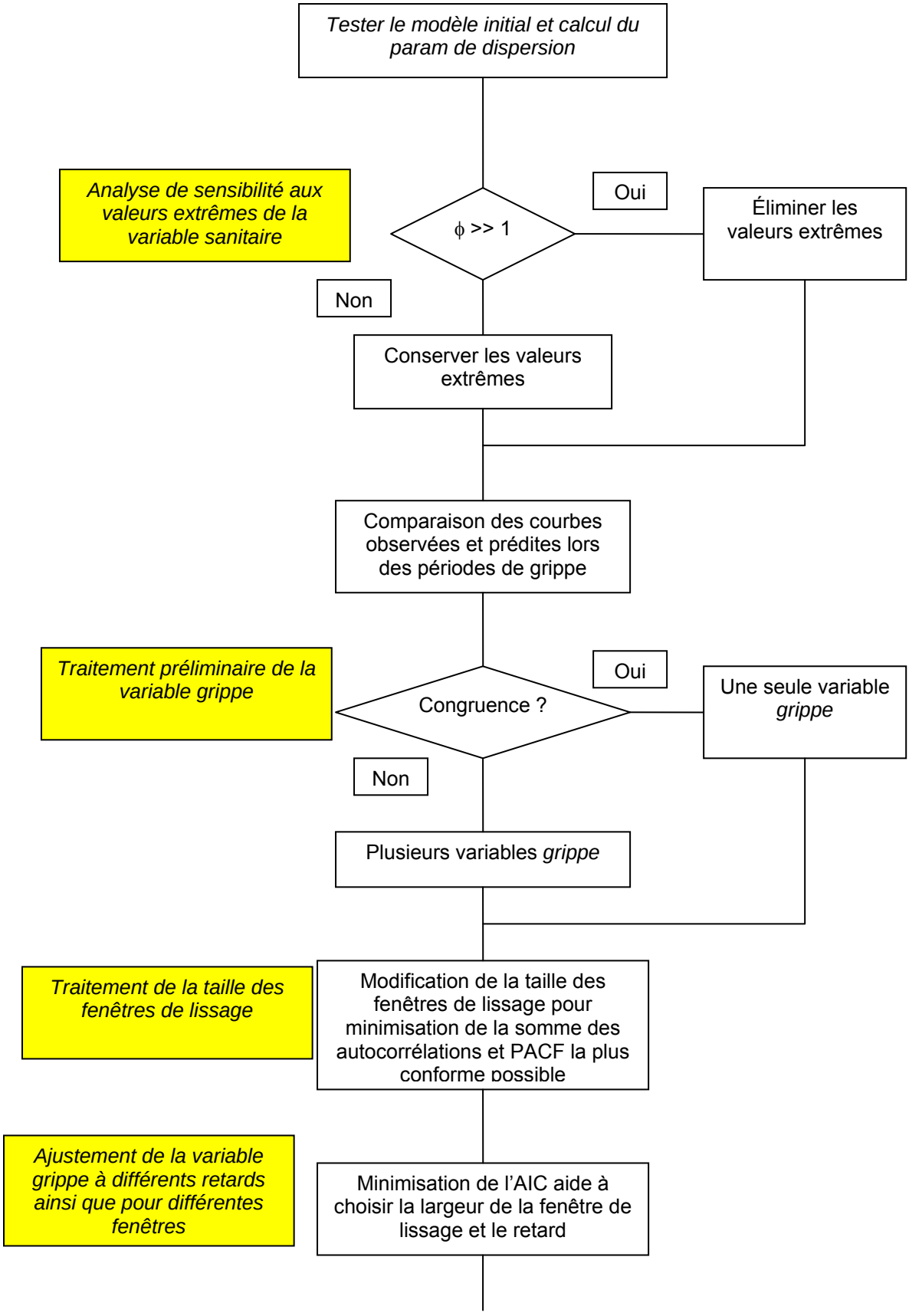
13. CAS PARTICULIER DE L'OZONE (§ 6.4.15.)

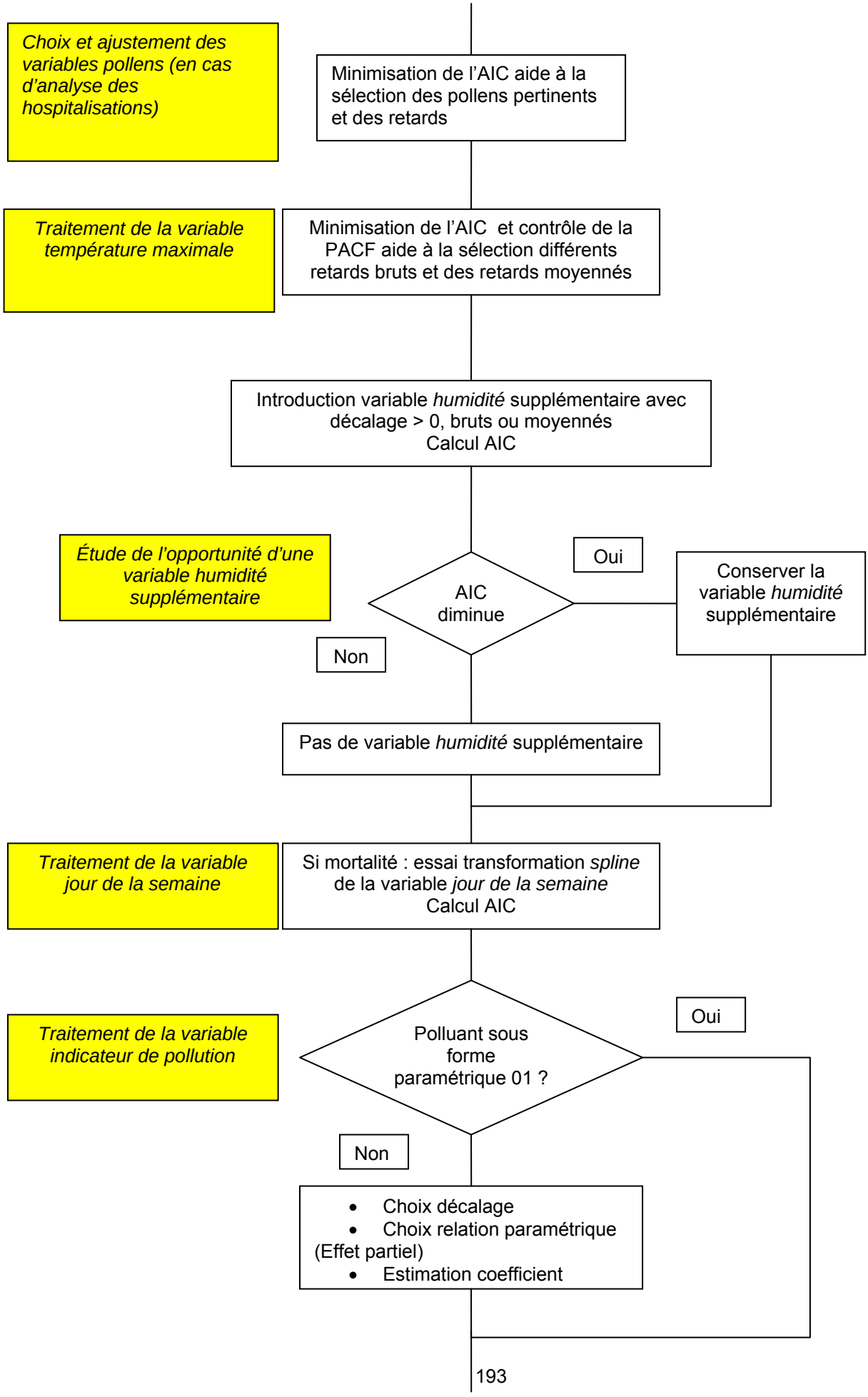
On procédera à trois approches

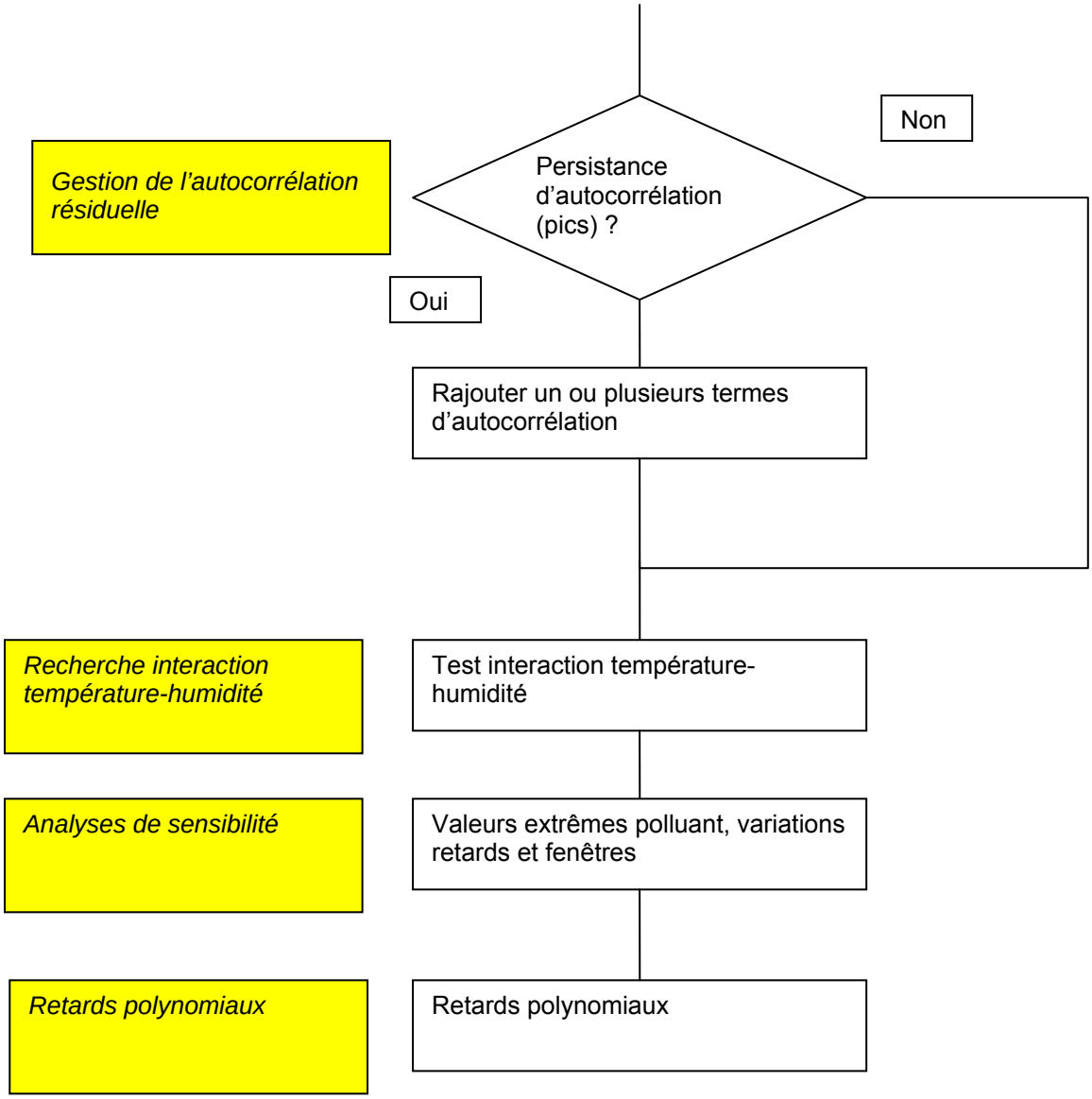
... ...

Annexe 5. Organisation logique de l’analyse

Voir page suivante


Tester le modèle initial et calcul du param de dispersion
Analyse de sensibilité aux valeurs extrêmes de la variable sanitaire
ϕ >> 1
Oui
Éliminer les valeurs extrêmes
Non
Conserver les valeurs extrêmes
Comparaison des courbes observées et prédites lors des périodes de grippe
Traitement préliminaire de la variable grippe
Congruence ?
Oui
Une seule variable grippe
Non
Plusieurs variables grippe
Traitement de la taille des fenêtres de lissage
Modification de la taille des fenêtres de lissage pour minimisation de la somme des autocorrélations et PACF la plus conforme possible
Ajustement de la variable grippe à différents retards ainsi que pour différentes fenêtres
Minimisation de l'AIC aide à choisir la largeur de la fenêtre de lissage et le retard


Choix et ajustement des variables pollens (en cas d'analyse des hospitalisations)
Minimisation de l'AIC aide à la sélection des pollens pertinents et des retards
Traitement de la variable température maximale
Minimisation de l'AIC et contrôle de la PACF aide à la sélection différents retards bruts et des retards moyennés
Introduction variable *humidité* supplémentaire avec décalage > 0, bruts ou moyennés
Calcul AIC
Étude de l'opportunité d'une variable humidité supplémentaire
AIC diminue
Oui
Conserver la variable *humidité* supplémentaire
Non
Pas de variable *humidité* supplémentaire
Traitement de la variable jour de la semaine
Si mortalité : essai transformation *spline* de la variable *jour de la semaine*
Calcul AIC
Traitement de la variable indicateur de pollution
Polluant sous forme paramétrique 01 ?
Oui
Non
• Choix décalage
• Choix relation paramétrique
(Effet partiel)
• Estimation coefficient


Gestion de l'autocorrélation résiduelle
Persistance d'autocorrélation (pics) ?
Non
Oui
Rajouter un ou plusieurs termes d'autocorrélation
Recherche interaction température-humidité
Test interaction température-humidité
Analyses de sensibilité
Valeurs extrêmes polluant, variations retards et fenêtres
Retards polynomiaux
Retards polynomiaux

Annexe 6. Programmes

Ci-dessous figure un exemple d'analyse de données (commandes et sorties).

ANALYSE mortot

1. DESCRIPTION

Paramètres principaux, graphe et boxplot de toutes les variables, lissage pour voir les variations saisonnières,

PACF pour voir l'autocorrélation dans la série

Paramètres

```
summary(morta$mortot)
```

```
ici
 Min. 1st Qu. Median  Mean 3rd Qu. Max.
    1       6      8 8.512      10   23
```

```
var(morta$mortot)
```

```
ici
8.790853
```

```
var(morta$mortot)/mean(morta$mortot)
1.03276
Donc faiblement surdispersé
```

Graphes

```
plot(morta$date.study,morta$mortot,type="l")
```

```
plot(morta$trend,morta$mortot,type="l")
lines(smooth.spline(morta$trend,morta$mortot),col=2)
lines(loess.smooth(morta$trend,morta$mortot),col=3)
lines(supsmu(morta$trend,morta$mortot),col=4)
```

Boxplot

```
boxplot(morta$mortot)
```

Voir pour chaque variable si 🕐 des outliers

si oui : les déterminer en calculant médiane + 2 * écart interquartile

puis prévoir de les ôter lors de l'analyse (en faisant un subset)

```
summary(morta$mortot)
```

ici

```
 Min. 1st Qu. Median  Mean 3rd Qu. Max.
    1       6      8 8.512      10   23
```

donc 8+2*(10-6) = 16

PACF

```
acf(morta$mortot,,type="p")
```

ou (si l'on ne veut pas voir défiler la liste des autocorrélations) :

```
sum(acf(morta$mortot,type="p")$acf)
```

il faut que la somme des PACF soit proche de 0 (~ bruit blanc)

ici

```
[1] 0.6892052
```

2. ÉCRIRE LE MODÈLE COMPLET

Il porte sur toutes les valeurs de la mortalité (avec les outliers). Il doit contenir :

morta.gam_gam(mortot~lo(trend,183/(nb jours))+dowf+j.feries+vac+lo(grip,.7)+lo(tempmin,.7)

+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,mu),data=morta,na=na.omit)

La fenêtre du trend correspond à 6 mois

Il faut tempmin et hummin ainsi que tempmax avec un lag >0 car tempmax est correlé à

tempmin

```
mortot1.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(grip,.7)+lo(tempmin,.7)+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,mu),data=morta,na=na.omit)
```

LES OUTILS

- PACF des résidus pour voir si

. somme autocorrélations proche de 0

. bruit blanc après les 10 1ers retards en veillant à ne pas surmodéliser ces périodes (ce qui se traduirait par une autocorrélation négative)

. diminution de l'autocorrélation pour les 10 1ers retards

- Prédites et observées (courbes superposées)

- Résidus (plot)

- Plot partiel (plot.gam)

- Summary pour voir le paramètre de surdispersion (si > 1 surdispersion = variation extrapoisson, si < 1 surspécification ???)

- Anal de sensiblité (éliminer les valeurs extrêmes des données sanitaires)

```
plot.gam(mortot.gam)

sum(acf(resid(mortot.gam),type="p")$acf)
summary(mortot.gam)$dispersion

plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$gripa7)&!is.na(morta$gripb3)&!is.na(morta$tempmax1)&
   !is.na(morta$so224h)&morta$mortot<16],fitted(mortot.gam),col=2)

plot(resid(mortot.gam))
```

- PACF des résidus

(voir si somme autocorrélations proche de 0 et voir si bruit blanc après les 10 1ers retards et
diminuer l'autocorrélation pour les 10 1ers retards)

```
sum(acf(resid(mortot1.gam),type="p")$acf)
```

ici

```
[1] -0.5286197
```

après le 10ème retard pas encore de bruit blanc (retard 15 et 28 signif)

- Prédites et observées (courbes superposées)

Attention aux VM (les polluants et les variables météo car lag => VM)

Sur un même graphique : la mortalité et le modèle

```
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax1)],fitted(mortot1.gam),
col=2)
```

ici

mortot n'a pas de saisonnalité propre ; celle-ci est due à la grippe

on distingue, vers trend=2700, un double pic sur le fitted (il s'agit d'un point correspondant à un outlier)

Donc on refait les 2 courbes et on trace la grippe en dessous

```
par(mfrow=c(2,1))
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax1)],fitted(mortot1.gam),
col=2)
```

```
plot(morta$grip,type="l")
par(mfrow=c(1,1))
```

- Résidus (plot)

```
plot(resid(mortot1.gam))
```

ici

Pas de saisonnalité apparente sur le graphe des résidus

- Plot partiel (plot.gam)

```
plot.gam(mortot1.gam)
```

ici

le graphe partiel de la tendance est un peu perturbé, le jour max est le mardi, le min est le lundi, l'effet j.feries ets négatif (moins de mortot les jours fériés), l'effet vac est positif (plus de mortot lors des périodes de vacances !!), l'effet grip est croissant et régulier, l'effet tempmin est en cloche (croissant puis décroissant avec un max pour 5°C, l'effet hummin est en cloche également, l'effet tempmax1 est décroissant puis croissant avec un min pour 17-18°C, l'effet so224h est croissant

- Summary

On fait un simple summary pour voir le coefficient de dispersion

```
summary(mortot1.gam)
```

ici

```
Call: gam(formula = mortot ~ lo(trend, 183/2922) + do
wf +
       j.feries + vac + lo(grip, 0.7) + lo(tempmin,
       0.7) + lo(hummin, 0.7) + lo(tempmax1, 0.7) +
       lo(so224h, 0.7), family = quasi(log, mu),
       data = morta, na.action = na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q
 -3.250055 -0.7025755 -0.04783621 0.6102306
      Max
 3.497915
```

(Dispersion Parameter for Quasi-likelihood family tak en to be 0.9710607)

```
    Null Deviance: 2989.714 on 2868 degrees of freedo
m
Residual Deviance: 2808.703 on 2818.577 degrees of fr
eedom
Number of Local Scoring Iterations: 3
DF for Terms and F-values for Nonparametric Effects
                    Df Npar Df   Npar F     Pr(F)
        (Intercept)  1
lo(trend, 183/2922)  1    27.3  1.65283 0.0180695
               dowf  6
           j.feries  1
                vac  1
      lo(grip, 0.7)  1     2.8  1.86820 0.1372048
   lo(tempmin, 0.7)  1     1.3  4.82051 0.0202937
    lo(hummin, 0.7)  1     1.2  1.93117 0.1627858
  lo(tempmax1, 0.7)  1     1.2 14.10616 0.0000587
    lo(so224h, 0.7)  1     1.7  0.85712 0.4113992
```

Il y a sous-dispersion.

- Anal de sensiblité

(éliminer les valeurs extrêmes des données sanitaires :ce sont celles qui sont sup à moyenne+2*écarts_interquartiles

```
ici
 Min. 1st Qu. Median  Mean 3rd Qu. Max.
    1       6      8 8.512      10   23
```

donc 8+2*(10-6) = 16

On recrée le modèle en enlevant les outliers

Ici, ont été été gardées les val strictement inf à 16 ; en fait il n'y a que 18 val égales à 16

```
mortot11.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(grip,.7)+lo(tempmin
,.7)+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,mu),data=morta,na
=na.omit,subset=mortot<16)

sum(acf(resid(mortot11.gam),type="p")$acf)

ici
Ne change pas grand chose aux autocorrél si ce n'est au niveau des lags (16 et 32 :
corrél négat)
[1] -0.4833517

plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax1)&morta$mortot<16],fitt
```

```
ed(mortot11.gam),col=2)

ici
Ne change pas grand chose aux graphes

par(mfrow=c(2,1))
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax1)&morta$mortot<16],fitt
ed(mortot11.gam),col=2)
plot(morta$grip,type="l")
par(mfrow=c(1,1))

ici
Ne change pas grand chose aux graphes

plot(resid(mortot11.gam))

ici
Ne change pas grand chose aux graphes

plot.gam(mortot11.gam)

ici
Ne change pas grand chose aux graphes

summary(mortot11.gam)

ici
Call: gam(formula = mortot ~ lo(trend, 183/2922) + do
wf +
       j.feries + vac + lo(grip, 0.7) + lo(tempmin,
       0.7) + lo(hummin, 0.7) + lo(tempmax1, 0.7) +
       lo(so224h, 0.7), family = quasi(log, mu),
       data = morta, subset = mortot < 16,
       na.action = na.omit)
Deviance Residuals:
     Min         1Q      Median        3Q      Max
 -3.2203 -0.6837754 -0.02708562 0.6223171 2.514939
(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.8845414 )
    Null Deviance: 2737.709 on 2832 degrees of freedo
m
Residual Deviance: 2571.902 on 2782.616 degrees of fr
eedom
Number of Local Scoring Iterations: 3
DF for Terms and F-values for Nonparametric Effects
                   Df Npar Df   Npar F     Pr(F)
```

```
                (Intercept)  1
lo(trend, 183/2922)  1    27.2  1.85828 0.0044881
                       dowf  6
                   j.feries  1
                        vac  1
              lo(grip, 0.7)  1     2.8  1.66908 0.1751979
           lo(tempmin, 0.7)  1     1.3  1.33893 0.2550509
            lo(hummin, 0.7)  1     1.2  1.74396 0.1865028
          lo(tempmax1, 0.7)  1     1.2 11.92689 0.0002336
            lo(so224h, 0.7)  1     1.8  1.01712 0.3535275
```

Donc le coeffic de sursispersion est encore plus petit (encore plus sous dispersé)

3. AJUSTER LA TAILLE DE LA FENÊTRE DE LA SAISON (TREND) PUIS CELLE DES FENÊTRES DES AUTRES VARIABLES

Se servir des graphes partiels.

Alors on change : la fenêtre du trend (augmentation), on crée 2 var gripa et grip b (car double-pic aux environ de trend=2700 voir plus haut), on change les autres fenêtres (0.7 -> 0.9)

gripa est égale à grip en dehors de la grippe de l'hiver 1996-1887

gripb est égale à grip lors de la grippe de l'hiver 1996-1887

```
morta$gripa_ifelse(morta$trend<2466|morta$trend>2677,morta$grip,0)
morta$gripa1_c(rep(NA,1),morta$gripa[1:(length(morta$gripa)-1)])
morta$gripa2_c(rep(NA,2),morta$gripa[1:(length(morta$gripa)-2)])
morta$gripa3_c(rep(NA,3),morta$gripa[1:(length(morta$gripa)-3)])
morta$gripa4_c(rep(NA,4),morta$gripa[1:(length(morta$gripa)-4)])
morta$gripa5_c(rep(NA,5),morta$gripa[1:(length(morta$gripa)-5)])
morta$gripa6_c(rep(NA,6),morta$gripa[1:(length(morta$gripa)-6)])
morta$gripa7_c(rep(NA,7),morta$gripa[1:(length(morta$gripa)-7)])

morta$gripb_ifelse(morta$trend<2466|morta$trend>2677,0,morta$grip)
morta$gripb1_c(rep(NA,1),morta$gripb[1:(length(morta$gripb)-1)])
morta$gripb2_c(rep(NA,2),morta$gripb[1:(length(morta$gripb)-2)])
morta$gripb3_c(rep(NA,3),morta$gripb[1:(length(morta$gripb)-3)])
morta$gripb4_c(rep(NA,4),morta$gripb[1:(length(morta$gripb)-4)])
morta$gripb5_c(rep(NA,5),morta$gripb[1:(length(morta$gripb)-5)])
morta$gripb6_c(rep(NA,6),morta$gripb[1:(length(morta$gripb)-6)])
morta$gripb7_c(rep(NA,7),morta$gripb[1:(length(morta$gripb)-7)])
```

```
mortot12.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb,
.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu)
,data=morta,na=na.omit,subset=mortot<16)
```

- PACF des résidus

(voir si somme autocorrélations proche de 0 et voir si bruit blanc après les 10 1ers retards et
diminuer l'autocorrélation pour les 10 1ers retards)

```
sum(acf(resid(mortot12.gam),type="p")$acf)
```

ici

```
[1] 0.01609538
```

De plus le graphe des pacf montre une seule autocor négative signif après les 10 1ers
points mais 2 autoc positives

remarque les différentes fenêtres testées sont **183**/2922, **300**/2922, **500**/2922, **730**/2922, **800**/2922

```
mortot12a.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb
,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu
),data=morta,na=na.omit,subset=mortot<16)
> sum(acf(resid(mortot12a.gam),type="p")$acf)
[1] -0.5029867
mortot12b.gam_gam(mortot~lo(trend,300/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb
,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu
),data=morta,na=na.omit,subset=mortot<16)
> sum(acf(resid(mortot12b.gam),type="p")$acf)
[1] -0.2005212
mortot12c.gam_gam(mortot~lo(trend,500/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb
,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu
),data=morta,na=na.omit,subset=mortot<16)
> sum(acf(resid(mortot12c.gam),type="p")$acf)
[1] -0.03046872
mortot12d.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb
,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu
),data=morta,na=na.omit,subset=mortot<16)
> sum(acf(resid(mortot12d.gam),type="p")$acf)
[1] 0.01609538
mortot12e.gam_gam(mortot~lo(trend,800/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb
,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu
),data=morta,na=na.omit,subset=mortot<16)
> sum(acf(resid(mortot12e.gam),type="p")$acf)
[1] 0.02954079
```

Le minimum est donc obtenu pour lo(trend,**730**/2922)

---

en résumé :

mortot

sum_pacf : 0.6892052 / disp : 1.03276

fenêtre 183 et .7, grip, avec outliers

```
mortot.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(grip,.7)+lo(tempmin,.
7)+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,mu),data=morta,na=n
a.omit)
```

sum_pacf : -0.5286197 / disp : 0.9710607

fenêtre 183 et .7, grip, **sans** outliers

```
mortot.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(grip,.7)+lo(tempmin,.
7)+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,mu),data=morta,na=n
a.omit,subset=mortot<16)
```

sum_pacf : -0.4833517 / disp : 0.8845414

fenêtre 183 et .7, **gripa et b**, sans outliers

```
mortot.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(gripa,.7)+lo(gripb,.7
)+lo(tempmin,.7)+lo(hummin,.7)+lo(tempmax1,.7)+lo(so224h,.7),family=quasi(log,mu),d
ata=morta,na=na.omit,subset=mortot<16)
```

sum_pacf : -0.493269 / disp : 0.8850095

fenêtre 183 et **.9**, gripa et b, sans outliers

```
mortot.gam_gam(mortot~lo(trend,183/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb,.9
)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),d
ata=morta,na=na.omit,subset=mortot<16)
```

sum_pacf : -0.5029867 / disp : 0.8856106

fenêtre : **300** et .9, gripa et b, sans outliers

```
mortot.gam_gam(mortot~lo(trend,300/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb,.9
)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),d
ata=morta,na=na.omit,subset=mortot<16)
```

sum_pacf : -0.2005212 / disp : 0.8894019

fenêtre : **325** et .9, gripa et b, sans outliers

```
mortot.gam_gam(mortot~lo(trend,325/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb,.9
)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),d
ata=morta,na=na.omit,subset=mortot<16)
```

sum_pacf : -0.1554951 / disp : 0.8903161

fenêtre : **500** et .9, gripa et b, sans outliers

```
mortot.gam_gam(mortot~lo(trend,500/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb,.9
)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),d
ata=morta,na=na.omit,subset=mortot<16)
```

sum_pacf : -0.03046872 / disp : 0.8929081

fenêtre : **730**

```
mortot.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb,.9
)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),d
ata=morta,na=na.omit,subset=mortot<16)
```

sum_pacf : 0.01609538 / disp : 0.8935006

fenêtre : 800

```
mortot.gam_gam(mortot~lo(trend,800/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb,.9
)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),d
ata=morta,na=na.omit,subset=mortot<16)
```

sum_pacf : 0.02954079 / disp : 0.8937276

- Prédites et observées (courbes superposées)

Sur un même graphique : la mortalité et le modèle

```
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax1)&morta$mortot<16],fitt
ed(mortot12.gam),col=2)
```

- Résidus (plot)

```
plot(resid(mortot12.gam))
```

- Plot partiel (plot.gam)

```
plot.gam(mortot12.gam)
```

- Summary

On fait un simple summary pour voir le coefficient de dispersion

```
summary(mortot1.gam)
```

```
ici
Call: gam(formula = mortot ~ lo(trend, 183/2922) + do
wf +
       j.feries + vac + lo(grip, 0.7) + lo(tempmin,
       0.7) + lo(hummin, 0.7) + lo(tempmax1, 0.7) +
       lo(so224h, 0.7), family = quasi(log, mu),
       data = morta, na.action = na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q
 -3.250055 -0.7025755 -0.04783621 0.6102306
       Max
 3.497915

(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.9710607 )

    Null Deviance: 2989.714 on 2868 degrees of freedo
m
```

```
Residual Deviance: 2808.703 on 2818.577 degrees of fr
eedom

Number of Local Scoring Iterations: 3

DF for Terms and F-values for Nonparametric Effects

                    Df Npar Df   Npar F     Pr(F)
        (Intercept)  1
lo(trend, 183/2922)  1    27.3  1.65283 0.0180695
               dowf  6
           j.feries  1
                vac  1
      lo(grip, 0.7)  1     2.8  1.86820 0.1372048
   lo(tempmin, 0.7)  1     1.3  4.82051 0.0202937
    lo(hummin, 0.7)  1     1.2  1.93117 0.1627858
  lo(tempmax1, 0.7)  1     1.2 14.10616 0.0000587
    lo(so224h, 0.7)  1     1.7  0.85712 0.4113992
```

- AIC

```
ici
> AIC(mortot12a.gam)
gam(formula = mortot ~ lo(trend, 183/2922) + dowf +
       j.feries + vac + lo(gripa, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2833
Degrees of Freedom Residual =  2784.27
Residual Deviance =  2576.629
AIC= 2662.942
> AIC(mortot12b.gam)
gam(formula = mortot ~ lo(trend, 300/2922) + dowf +
       j.feries + vac + lo(gripa, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2833
Degrees of Freedom Residual =  2795.228
Residual Deviance =  2596.802
AIC= 2663.99
```

```
> AIC(mortot12c.gam)
gam(formula = mortot ~ lo(trend, 500/2922) + dowf +
       j.feries + vac + lo(gripa, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2833
Degrees of Freedom Residual =  2802.128
Residual Deviance =  2611.311
AIC= 2666.442
> AIC(mortot12d.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2833
Degrees of Freedom Residual =  2805.608
Residual Deviance =  2615.863
AIC= 2664.813
> AIC(mortot12e.gam)
gam(formula = mortot ~ lo(trend, 800/2922) + dowf +
       j.feries + vac + lo(gripa, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2833
Degrees of Freedom Residual =  2806.265
Residual Deviance =  2617.048
AIC= 2664.835
```

Un minimum "local" de l'AIC (pas le vrai minimum) est obtenu pour la fenêtre 730/2922 :

donc bon compromis entre l'AIC et l'autocorrélation

4. AJUSTER LA GRIPPE A DIFFÉRENTS RETARDS AINSI QUE POUR DIFFÉRENTES FENÊTRES

Essayer d'expliquer le max d'augmentation de la va sanitaire (avec AIC): choisir une

fenêtre puis tester les différents retards puis changer de fenêtre, etc.

Il faut que

la courbe représentant l'effet grippe soit le plus lisse possible

Si deux va grip : gripa à différents retard puis gripb à différents retards

```
mortot12.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb,
.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu)
,data=morta,na=na.omit,subset=mortot<16)
```

gripa

```
mortot13a.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa,.9)+lo(gripb
,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu
),data=morta,na=na.omit,subset=mortot<16)
mortot13b.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa1,.9)+lo(grip
b,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<16)
mortot13c.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa2,.9)+lo(grip
b,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<16)
mortot13d.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa3,.9)+lo(grip
b,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<16)
mortot13e.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa4,.9)+lo(grip
b,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<16)
mortot13f.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa5,.9)+lo(grip
b,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<16)
mortot13g.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa6,.9)+lo(grip
b,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<16)
mortot13h.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
b,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,m
u),data=morta,na=na.omit,subset=mortot<16)
```

- AIC

ici

```
> AIC(mortot13a.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2833
Degrees of Freedom Residual =  2805.608
Residual Deviance =  2615.863
AIC= 2664.813
> AIC(mortot13b.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa1, 0.9) + lo(gripb,
```

```
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2833
Degrees of Freedom Residual =  2805.604
Residual Deviance =  2615.947
AIC= 2664.906
> AIC(mortot13c.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa2, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2832
Degrees of Freedom Residual =  2804.606
Residual Deviance =  2615.832
AIC= 2664.799
> AIC(mortot13d.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa3, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2831
Degrees of Freedom Residual =  2803.61
Residual Deviance =  2613.887
AIC= 2662.838
> AIC(mortot13e.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa4, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2830
Degrees of Freedom Residual =  2802.612
Residual Deviance =  2610.32
AIC= 2659.219
> AIC(mortot13f.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
```

```
       j.feries + vac + lo(gripa5, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2829
Degrees of Freedom Residual =  2801.602
Residual Deviance =  2613.476
AIC= 2662.469
> AIC(mortot13g.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa6, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2828
Degrees of Freedom Residual =  2800.606
Residual Deviance =  2606.869
AIC= 2655.748
> AIC(mortot13h.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.609
Residual Deviance =  2606.235
AIC= 2655.127
```

Donc OK pour gripa7

gripb

```
mortot14a.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot14b.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb1,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot14c.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb2,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot14d.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
```

```
b3,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,
mu),data=morta,na=na.omit,subset=mortot<16)

mortot14e.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
b4,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,
mu),data=morta,na=na.omit,subset=mortot<16)

mortot14f.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
b5,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,
mu),data=morta,na=na.omit,subset=mortot<16)

mortot14g.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
b6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,
mu),data=morta,na=na.omit,subset=mortot<16)

mortot14h.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
b7,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,
mu),data=morta,na=na.omit,subset=mortot<16)
```

- AIC

ici

```
> AIC(mortot14a.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.609
Residual Deviance =  2606.235
AIC= 2655.127
> AIC(mortot14b.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb1,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.608
Residual Deviance =  2604.974
AIC= 2653.841
> AIC(mortot14c.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb2,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
```

```
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.597
Residual Deviance =  2604.404
AIC= 2653.274
> AIC(mortot14d.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb3,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.593
Residual Deviance =  2604.382
AIC= 2653.266
> AIC(mortot14e.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb4,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.594
Residual Deviance =  2604.483
AIC= 2653.367
> AIC(mortot14f.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb5,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.585
Residual Deviance =  2603.339
AIC= 2652.22
> AIC(mortot14g.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
```

```
      mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.586
Residual Deviance =  2603.008
AIC= 2651.885
> AIC(mortot14h.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
      j.feries + vac + lo(gripa7, 0.9) + lo(gripb7,
      0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
      lo(tempmax1, 0.9) + lo(so224h, 0.9), family
       = quasi(log, mu), data = morta, subset =
      mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.582
Residual Deviance =  2603.924
AIC= 2652.822
```

Donc gripb6

```
mortot14g.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

5. AJUSTER TEMPMAX

tempmax retardée à différents retards 1-3 j ou moyennée (ne pas modifier tempmin qui doit rester ds le modèle

avec un décalage 0)

```
morta$tempmax12_(morta$tempmax1+morta$tempmax2)/2
morta$tempmax23_(morta$tempmax2+morta$tempmax3)/2
morta$tempmax123_(morta$tempmax1+morta$tempmax2+morta$tempmax3)/3

mortot15a.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot15b.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot15c.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax3,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot15d.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax12,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot15e.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax23,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot15f.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
```

```
b6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax123,.9)+lo(so224h,.9),family=quasi(lo
g,mu),data=morta,na=na.omit,subset=mortot<16)
```

- AIC

```
ici
> AIC(mortot15a.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax1, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.586
Residual Deviance =  2603.008
AIC= 2651.885
> AIC(mortot15b.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax2, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.587
Residual Deviance =  2599.789
AIC= 2648.59
> AIC(mortot15c.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax3, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.582
Residual Deviance =  2606.192
AIC= 2655.124
> AIC(mortot15d.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax12, 0.9) + lo(so224h, 0.9), family
```

```
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.526
Residual Deviance =  2601.022
AIC= 2649.96
> AIC(mortot15e.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax23, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.53
Residual Deviance =  2602.441
AIC= 2651.393
> AIC(mortot15f.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax123, 0.9) + lo(so224h, 0.9),
       family = quasi(log, mu), data = morta,
       subset = mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2799.493
Residual Deviance =  2601.976
AIC= 2650.989
```

Donc tempmax2

```
mortot15b.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
b6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(so224h,.9),family=quasi(log,
mu),data=morta,na=na.omit,subset=mortot<16)
```

- PACF des résidus

```
sum(acf(resid(mortot15b.gam),type="p")$acf)
```

ici

```
[1] -0.03276536 (moins bon que le précédent : 0.01609538)
```

- Prédites et observées (courbes superposées)

```
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax2)&!is.na(morta$gripa7)&
!is.na(morta$grip6)&morta$mortot<16],fitted(mortot15b.gam),col=2)
```

- Résidus (plot)

```
plot(resid(mortot15b.gam))
```

- Plot partiel (plot.gam)

```
plot.gam(mortot15b.gam)
```

- Summary

```
summary(mortot15b.gam)
```

```
ici
Call: gam(formula = mortot ~ lo(trend, 730/2922) + do
wf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax2, 0.9) + lo(so224h, 0.9), family
        = quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q
 -3.233119 -0.7094678 -0.03282252 0.6365732
      Max
 2.296282

(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.8901073 )

    Null Deviance: 2726.071 on 2826 degrees of freedo
m

Residual Deviance: 2599.789 on 2799.587 degrees of fr
eedom

Number of Local Scoring Iterations: 3

DF for Terms and F-values for Nonparametric Effects

                    Df Npar Df   Npar F     Pr(F)
        (Intercept)  1
lo(trend, 730/2922)  1     5.9 1.092308 0.3642075
               dowf  6
           j.feries  1
                vac  1
    lo(gripa7, 0.9)  1     1.7 0.942630 0.3767428
    lo(gripb6, 0.9)  1     1.1 2.144783 0.1406357
   lo(tempmin, 0.9)  1     0.6 0.295071 0.4735099
    lo(hummin, 0.9)  1     0.6 2.536790 0.1212284
```

```
  lo(tempmax2, 0.9)  1     0.6 6.249282 0.0252256
    lo(so224h, 0.9)  1     0.9 1.396469 0.2356043
```

Le param de disp est moins bon

6. AJOUTER ET AJUSTER HUMMIN A DIFFERENTS DECALAGES

(tout en gardant hummin avec un décalage 0) et voir si nécessaire de garder ds le modèle

```
morta$hummin12_(morta$hummin1+morta$hummin2)/2
morta$hummin23_(morta$hummin2+morta$hummin3)/2
morta$hummin123_(morta$hummin1+morta$hummin2+morta$hummin3)/3

mortot16a.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin1,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot16b.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin2,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot16c.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin3,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot16d.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot16e.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin23,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot16f.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin123,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

- AIC

```
> AIC(mortot16a.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax2, 0.9) + lo(hummin1, 0.9) + lo(
       so224h, 0.9), family = quasi(log, mu), data
        = morta, subset = mortot < 16, na.action =
       na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2798.023
Residual Deviance =  2596.25
AIC= 2647.793
> AIC(mortot16b.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
```

```
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax2, 0.9) + lo(hummin2, 0.9) + lo(
       so224h, 0.9), family = quasi(log, mu), data
        = morta, subset = mortot < 16, na.action =
       na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2798.023
Residual Deviance =  2596.337
AIC= 2647.884
> AIC(mortot16c.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax2, 0.9) + lo(hummin3, 0.9) + lo(
       so224h, 0.9), family = quasi(log, mu), data
        = morta, subset = mortot < 16, na.action =
       na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2798.023
Residual Deviance =  2597.366
AIC= 2648.919
> AIC(mortot16d.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax2, 0.9) + lo(hummin12, 0.9) + lo(
       so224h, 0.9), family = quasi(log, mu), data
        = morta, subset = mortot < 16, na.action =
       na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2798.032
Residual Deviance =  2593.68
AIC= 2645.155
> AIC(mortot16e.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax2, 0.9) + lo(hummin23, 0.9) + lo(
       so224h, 0.9), family = quasi(log, mu), data
        = morta, subset = mortot < 16, na.action =
       na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2798.032
```

```
Residual Deviance =  2596.029
AIC= 2647.546
> AIC(mortot16f.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + dowf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
       lo(tempmax2, 0.9) + lo(hummin123, 0.9) + lo(
       so224h, 0.9), family = quasi(log, mu), data
        = morta, subset = mortot < 16, na.action =
       na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2798.011
Residual Deviance =  2593.979
AIC= 2645.492

Donc hummin12

mortot16d.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(grip
b6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),f
amily=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

- PACF des résidus

```
sum(acf(resid(mortot16b.gam),type="p")$acf)

ici
[1] -0.0440752
```

- Prédites et observées (courbes superposées)

```
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax2)&!is.na(morta$gripa7)&
!is.na(morta$grip6)&morta$mortot<16],fitted(mortot16d.gam),col=2)
```

- Résidus (plot)

```
plot(resid(mortot16d.gam))
```

- Plot partiel (plot.gam)

```
plot.gam(mortot16d.gam)
```

- Summary

```
summary(mortot16d.gam)

ici
Call: gam(formula = mortot ~ lo(trend, 730/2922) + do
wf +
       j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) +
```

```
       lo(tempmax2, 0.9) + lo(hummin12, 0.9) + lo(
       so224h, 0.9), family = quasi(log, mu), data
        = morta, subset = mortot < 16, na.action =
       na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q     Max
 -3.205291 -0.7016076 -0.03047159 0.6270401 2.30691

(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.8884598 )

    Null Deviance: 2726.071 on 2826 degrees of freedo
m

Residual Deviance: 2593.68 on 2798.031 degrees of fre
edom

Number of Local Scoring Iterations: 3

DF for Terms and F-values for Nonparametric Effects

                   Df Npar Df   Npar F     Pr(F)
        (Intercept)  1
lo(trend, 730/2922)  1     5.9 1.144185 0.3340019
              dowf  6
          j.feries  1
               vac  1
    lo(gripa7, 0.9)  1     1.7 0.913548 0.3871663
    lo(gripb6, 0.9)  1     1.1 2.215068 0.1339501
   lo(tempmin, 0.9)  1     0.6 0.281757 0.4804259
    lo(hummin, 0.9)  1     0.6 4.087829 0.0627436
  lo(tempmax2, 0.9)  1     0.6 5.386032 0.0354651
  lo(hummin12, 0.9)  1     0.6 5.291709 0.0398217
    lo(so224h, 0.9)  1     0.9 1.357483 0.2420502

Le paramètre de surdispersion diminue encore

mortot17.gam_gam(mortot~lo(trend,730/2922)+dowf+j.feries+vac+lo(gripa7,.9)+lo(gripb
6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),fa
mily=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

7. TRAITEMENT DE LA VARIABLE JOURS_DE_LA_SEMAINE

a. Pour la mortalité : tester spline de jours de la semaine car peu de variation de l'influence de différents jours, de plus permet de gagner des ddl

- créer une va quantitative : dowf_as.numeric(weekdays(morta$date.study)
- tester s(dowf,3) ou s(dowf,4) et calculer l'AIC

b. Pour la morbidité : il faut laisser dowf sous forme de va qualitatives (va 0,1) car les jours

interviennent de façon importante

Vérifier les j avec plot.gam ou un boxplot.

```
morta$dowf.num_as.numeric(weekdays(morta$date.study))
mortot17a.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,3)+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
mortot17b.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

- AIC

```
AIC(mortot17a.gam)
AIC(mortot17b.gam)
```

```
> AIC(mortot17a.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 3) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2801.028
Residual Deviance =  2597.988
AIC= 2644.175
> AIC(mortot17b.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2800.038
Residual Deviance =  2595.864
AIC= 2643.787
>
```

ici

mortot17b.gam (dimin de l'AIC qui était égal à 2645.155 à l'étape précédente)

- PACF des résidus

```
sum(acf(resid(mortot17a.gam),type="p")$acf)
sum(acf(resid(mortot17b.gam),type="p")$acf)
```

- Prédites et observées (courbes superposées)

```
plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax2)&!is.na(morta$gripa7)&
!is.na(morta$grip6)&morta$mortot<16],fitted(mortot17a.gam),col=2)

plot(morta$trend,morta$mortot,type="l")
lines(morta$trend[!is.na(morta$so224h)&!is.na(morta$tempmax2)&!is.na(morta$gripa7)&
!is.na(morta$grip6)&morta$mortot<16],fitted(mortot17b.gam),col=3)
```

- Résidus (plot)

```
plot(resid(mortot17a.gam))
plot(resid(mortot17b.gam))
```

- Plot partiel (plot.gam)

```
plot.gam(mortot17a.gam)
plot.gam(mortot17b.gam)
```

- Summary

```
summary(mortot17a.gam)
summary(mortot17b.gam)
```

ici

```
> summary(mortot17a.gam)
Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 3) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q
 -3.221924 -0.7069217 -0.02540483 0.6382764
      Max
 2.291642
(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.889165 )
    Null Deviance: 2726.071 on 2826 degrees of freedo
m
```

```
Residual Deviance: 2597.988 on 2801.028 degrees of fr
eedom
Number of Local Scoring Iterations: 3
DF for Terms and F-values for Nonparametric Effects
                   Df Npar Df   Npar F     Pr(F)
        (Intercept)  1
lo(trend, 730/2922)  1     5.9 1.153018 0.3290467
     s(dowf.num, 3)  1     2.0 2.183901 0.1126878
           j.feries  1
                vac  1
    lo(gripa7, 0.9)  1     1.7 0.890492 0.3956639
    lo(gripb6, 0.9)  1     1.1 2.192860 0.1360303
   lo(tempmin, 0.9)  1     0.6 0.265936 0.4889897
    lo(hummin, 0.9)  1     0.6 4.365959 0.0561536
  lo(tempmax2, 0.9)  1     0.6 5.482567 0.0341246
  lo(hummin12, 0.9)  1     0.6 5.475264 0.0371872
    lo(so224h, 0.9)  1     0.9 1.466799 0.2245030

> summary(mortot17b.gam)
Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Deviance Residuals:
       Min        1Q     Median        3Q      Max
 -3.221037 -0.704369 -0.0272575 0.6343021 2.295362
(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.888708 )
    Null Deviance: 2726.071 on 2826 degrees of freedo
m
Residual Deviance: 2595.865 on 2800.038 degrees of fr
eedom
Number of Local Scoring Iterations: 3
DF for Terms and F-values for Nonparametric Effects
                   Df Npar Df   Npar F     Pr(F)
        (Intercept)  1
lo(trend, 730/2922)  1     5.9 1.152849 0.3291409
     s(dowf.num, 4)  1     3.0 2.280053 0.0775492
           j.feries  1
                vac  1
    lo(gripa7, 0.9)  1     1.7 0.904150 0.3906069
```

```
    lo(gripb6, 0.9)  1     1.1 2.209601 0.1344606
   lo(tempmin, 0.9)  1     0.6 0.272813 0.4852246
    lo(hummin, 0.9)  1     0.6 4.302431 0.0575874
  lo(tempmax2, 0.9)  1     0.6 5.495811 0.0339436
  lo(hummin12, 0.9)  1     0.6 5.453998 0.0374813
    lo(so224h, 0.9)  1     0.9 1.428083 0.2305317
```

Donc le paramètre de dispersion s'éloigne encore de 1 en diminuant

```
mortot17b.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

8. (INTRODUCTION ET/OU) DECALAGE DU POLLUANT

Pour le polluant, tester lo(Xt-j,) avec décalages 0 à 5 puis moyennes 0-1, 0-2, 0-3 puis 1-2, 1-3, 2-3

Si autocorrélation sur un retard rajouter un terme autorégressif :

fonction gam.ar(mortot~...,data,ar.order=1) ; ne pas préciser la famille car déjà quasi(log,mu) ; ne pas

précisez non plus na.omit car déjà na.action = na.gam.replace

Pour so2, uniquement décalage car déjà dans le modèle

```
mortot18a.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot18b.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h1,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot18c.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h2,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot18d.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h3,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot18e.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h4,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot18f.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h5,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortotso201.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,
.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(s
o224h01,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot18h.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h02,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot.so203.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7
,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(
so224h03,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot18j.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h12,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```

```
mortot.so213.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7
,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(
so224h13,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

mortot18l.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h23,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

AIC(mortot18a.gam)
AIC(mortot18b.gam)
AIC(mortot18c.gam)
AIC(mortot18d.gam)
AIC(mortot18e.gam)
AIC(mortot18f.gam)

ici
> AIC(mortot18a.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2800.038
Residual Deviance =  2595.864
AIC= 2643.787
> AIC(mortot18b.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h1, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2827
Degrees of Freedom Residual =  2800.048
Residual Deviance =  2608.9
AIC= 2656.978
> AIC(mortot18c.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h2, 0.9), family =
```

```
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2826
Degrees of Freedom Residual =  2799.095
Residual Deviance =  2630.648
AIC= 2679.031
> AIC(mortot18d.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h3, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2826
Degrees of Freedom Residual =  2799.103
Residual Deviance =  2622.176
AIC= 2670.39
> AIC(mortot18e.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h4, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2826
Degrees of Freedom Residual =  2799.064
Residual Deviance =  2640.329
AIC= 2688.902
> AIC(mortot18f.gam)
gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h5, 0.9), family =
       quasi(log, mu), data = morta, subset =
       mortot < 16, na.action = na.omit)
Degrees of Freedom Total =  2826
Degrees of Freedom Residual =  2799.05
Residual Deviance =  2630.512
AIC= 2678.951
```

```
> summary.glm(mortot18a.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
      0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h, 0.9),
      family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min        1Q     Median        3Q      Max
 -3.221037 -0.704369 -0.0272575 0.6343021 2.295362

Coefficients:
                          Value  Std. Error     t value
        (Intercept)  2.106642227 0.021317885  98.8204145
lo(trend, 730/2922)  0.042542831 0.380776866   0.1117264
     s(dowf.num, 4)  0.002621429 0.003076058   0.8522039
           j.feries -0.014919618 0.017706078  -0.8426269
                vac  0.014290768 0.006944428   2.0578757
    lo(gripa7, 0.9)  2.087936252 0.350546106   5.9562386
    lo(gripb6, 0.9)  0.949373356 0.340370064   2.7892387
   lo(tempmin, 0.9)  0.957864969 0.749906057   1.2773133
    lo(hummin, 0.9) -0.124733277 0.487766279  -0.2557235
  lo(tempmax2, 0.9) -2.890167288 0.845410640  -3.4186550
  lo(hummin12, 0.9) -1.063855865 0.558477534  -1.9049215
    lo(so224h, 0.9)  0.673387424 0.453138810   1.4860511

(Dispersion Parameter for Quasi-likelihood family taken to be 0.888708 )

    Null Deviance: 2726.071 on 2826 degrees of freedom

Residual Deviance: 2595.865 on 2800.037927 degrees of freedom

Number of Fisher Scoring Iterations: 3
> summary.glm(mortot18b.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
      0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h1, 0.9),
      family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min         1Q      Median       3Q      Max
 -3.195223 -0.6970702 -0.02778507 0.638564 2.317596
```

```
Coefficients:
                           Value  Std. Error     t value
        (Intercept)  2.1082582538 0.021423739  98.4075777
lo(trend, 730/2922)  0.0974422713 0.379815714   0.2565514
     s(dowf.num, 4)  0.0008272978 0.003093774   0.2674073
           j.feries -0.0195755009 0.017784903  -1.1006808
                vac  0.0152964174 0.006949339   2.2011327
    lo(gripa7, 0.9)  2.1205061690 0.351041493   6.0406140
    lo(gripb6, 0.9)  0.9301826340 0.341694189   2.7222665
   lo(tempmin, 0.9)  0.9040925010 0.718804538   1.2577724
    lo(hummin, 0.9) -0.2824593617 0.492081752  -0.5740090
  lo(tempmax2, 0.9) -2.7716031251 0.854518933  -3.2434660
  lo(hummin12, 0.9) -1.0341812804 0.560577178  -1.8448508
   lo(so224h1, 0.9)  0.9852554745 0.428292998   2.3004240

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8919386 )

    Null Deviance: 2743.13 on 2826 degrees of freedom

Residual Deviance: 2608.9 on 2800.0481654 degrees of freedom

Number of Fisher Scoring Iterations: 3
> summary.glm(mortot18c.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
      0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h2, 0.9),
      family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q      Max
 -3.213335 -0.7076039 -0.02907963 0.6428075 2.290581

Coefficients:
                           Value  Std. Error       t value
        (Intercept)  2.1063891345 0.021325049  98.775347358
lo(trend, 730/2922) -0.0005250081 0.378964316  -0.001385376
     s(dowf.num, 4)  0.0018033092 0.003085160   0.584510770
           j.feries -0.0186810640 0.017744522  -1.052779212
                vac  0.0142735214 0.006972398   2.047146641
    lo(gripa7, 0.9)  2.0450615406 0.351799915   5.813138240
    lo(gripb6, 0.9)  0.8339773094 0.344615507   2.420022583
   lo(tempmin, 0.9)  0.5624238902 0.718715057   0.782540848
    lo(hummin, 0.9) -0.1861409857 0.491171644  -0.378973395
  lo(tempmax2, 0.9) -2.7120883606 0.852924564  -3.179751732
```

```
  lo(hummin12, 0.9) -0.9979542722 0.559798423  -1.782702899
   lo(so224h2, 0.9)  0.5911128316 0.414863366   1.424837382

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8991257 )

    Null Deviance: 2759.032 on 2825 degrees of freedom

Residual Deviance: 2630.648 on 2799.0946306 degrees of freedom

Number of Fisher Scoring Iterations: 3
> summary.glm(mortot18d.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h3, 0.9),
       family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q      Max
 -3.220451 -0.7111232 -0.02534688 0.6367609 2.297535


Coefficients:
                          Value  Std. Error     t value
        (Intercept)  2.110828017 0.021143716  99.8324047
lo(trend, 730/2922)  0.170193360 0.377357731   0.4510133
     s(dowf.num, 4)  0.001894327 0.003074602   0.6161210
           j.feries -0.013964834 0.017563879  -0.7950883
                vac  0.012551128 0.006961441   1.8029498
    lo(gripa7, 0.9)  2.067586237 0.350980834   5.8908807
    lo(gripb6, 0.9)  0.830770220 0.341911003   2.4297850
   lo(tempmin, 0.9)  0.700377635 0.717819548   0.9757015
    lo(hummin, 0.9) -0.154595920 0.489042443  -0.3161196
  lo(tempmax2, 0.9) -2.721313245 0.851116896  -3.1973437
  lo(hummin12, 0.9) -1.116575284 0.560192583  -1.9931990
   lo(so224h3, 0.9)  0.970944856 0.406805811   2.3867527

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8962815 )

    Null Deviance: 2755.645 on 2825 degrees of freedom

Residual Deviance: 2622.176 on 2799.1031284 degrees of freedom

Number of Fisher Scoring Iterations: 3
> summary.glm(mortot18e.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
```

```
      0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h4, 0.9),
      family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min         1Q     Median         3Q      Max
 -3.205818 -0.7136819 -0.0183599 0.6430802 2.280987

Coefficients:
                          Value Std. Error     t value
        (Intercept)  2.107719220 0.02114543  99.6773054
lo(trend, 730/2922)  0.061829713 0.37745948   0.1638049
     s(dowf.num, 4)  0.002019168 0.00309369   0.6526730
           j.feries -0.017147988 0.01757037  -0.9759606
                vac  0.014010475 0.00697858   2.0076397
    lo(gripa7, 0.9)  2.032870001 0.35196208   5.7758210
    lo(gripb6, 0.9)  0.821050441 0.34362989   2.3893452
   lo(tempmin, 0.9)  0.560022662 0.72461470   0.7728558
    lo(hummin, 0.9) -0.155711245 0.49079953  -0.3172604
  lo(tempmax2, 0.9) -2.743680472 0.85360681  -3.2142205
  lo(hummin12, 0.9) -0.996225857 0.56342289  -1.7681672
   lo(so224h4, 0.9)  0.567289054 0.40457277   1.4021929

(Dispersion Parameter for Quasi-likelihood family taken to be 0.9016465 )

    Null Deviance: 2768.179 on 2825 degrees of freedom

Residual Deviance: 2640.329 on 2799.0645487 degrees of freedom

Number of Fisher Scoring Iterations: 3
> summary.glm(mortot18f.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
      0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h5, 0.9),
      family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min         1Q      Median         3Q      Max
 -3.216515 -0.7081789 -0.01767632 0.6422318 2.292454

Coefficients:
                          Value  Std. Error     t value
        (Intercept)  2.104406939 0.021153726  99.4816225
lo(trend, 730/2922) -0.033583966 0.376678360  -0.0891582
     s(dowf.num, 4)  0.002718387 0.003114083   0.8729335
```

```
           j.feries -0.016350739 0.017540479  -0.9321718
                vac  0.013773505 0.006970445   1.9759864
    lo(gripa7, 0.9)  2.062546906 0.350899632   5.8778828
    lo(gripb6, 0.9)  0.869628247 0.340393066   2.5547766
   lo(tempmin, 0.9)  0.568450184 0.721302249   0.7880887
    lo(hummin, 0.9) -0.065909361 0.489057566  -0.1347681
  lo(tempmax2, 0.9) -2.794440956 0.850518308  -3.2855741
  lo(hummin12, 0.9) -1.013553129 0.561452568  -1.8052338
   lo(so224h5, 0.9)  0.359944510 0.402552458   0.8941555

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8986866 )

    Null Deviance: 2760.819 on 2825 degrees of freedom

Residual Deviance: 2630.512 on 2799.0500756 degrees of freedom

Number of Fisher Scoring Iterations: 3

so224h1 et so224h3

morta$so224h13_(morta$so224h1+morta$so224h2+morta$so224h3)/3

> summary.glm(mortotso201.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h01, 0.9),
       family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q      Max
 -3.215139 -0.6995901 -0.02876478 0.6367207 2.301519

Coefficients:
                          Value  Std. Error     t value
        (Intercept)  2.108874638 0.021522767  97.9834370
lo(trend, 730/2922)  0.185914929 0.391469693   0.4749153
     s(dowf.num, 4)  0.001759659 0.003098694   0.5678711
           j.feries -0.015446824 0.017844280  -0.8656457
                vac  0.015314507 0.006969947   2.1972202
    lo(gripa7, 0.9)  2.126459662 0.351317777   6.0528097
    lo(gripb6, 0.9)  0.965731418 0.343906162   2.8081248
   lo(tempmin, 0.9)  1.162323593 0.739846616   1.5710332
    lo(hummin, 0.9) -0.262748965 0.492469863  -0.5335331
  lo(tempmax2, 0.9) -2.868704373 0.852766191  -3.3639987
  lo(hummin12, 0.9) -1.011312974 0.561511503  -1.8010548
```

```
  lo(so224h01, 0.9)  0.985837433 0.462905464   2.1296734

(Dispersion Parameter for Quasi-likelihood family taken to be 0.888351 )

    Null Deviance: 2698.185 on 2798 degrees of freedom

Residual Deviance: 2568.625 on 2772.0386754 degrees of freedom

Number of Fisher Scoring Iterations: 3

> summary.glm(mortot.so203.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h03, 0.9),
       family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q      Max
 -3.214284 -0.7040931 -0.03456517 0.6372217 2.284958

Coefficients:
                          Value  Std. Error     t value
        (Intercept)  2.107929569 0.021654604  97.3432496
lo(trend, 730/2922)  0.249216165 0.408085379   0.6106961
     s(dowf.num, 4)  0.002287392 0.003128960   0.7310391
           j.feries -0.015281771 0.017950149  -0.8513451
                vac  0.014430957 0.007031003   2.0524749
    lo(gripa7, 0.9)  2.079834295 0.352738951   5.8962422
    lo(gripb6, 0.9)  0.858973306 0.352747658   2.4350929
   lo(tempmin, 0.9)  0.991008679 0.742515298   1.3346643
    lo(hummin, 0.9) -0.260312599 0.497249106  -0.5235054
  lo(tempmax2, 0.9) -2.770321440 0.863343917  -3.2088272
  lo(hummin12, 0.9) -0.987218944 0.565724467  -1.7450526
  lo(so224h03, 0.9)  1.021314694 0.473539223   2.1567690

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8922953 )

    Null Deviance: 2660.006 on 2744 degrees of freedom

Residual Deviance: 2532.188 on 2718.0434189 degrees of freedom

Number of Fisher Scoring Iterations: 3
> summary.glm(mortot.so213.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
```

```
      0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h13, 0.9),
      family = quasi(log, mu), data = morta, subset = mortot < 16, na.action =
na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q      Max
 -3.206221 -0.7016167 -0.03026931 0.6397941 2.281482

Coefficients:
                          Value  Std. Error     t value
        (Intercept)  2.109975912 0.021603183  97.6696776
lo(trend, 730/2922)  0.164161786 0.397927662   0.4125418
     s(dowf.num, 4)  0.001518077 0.003114063   0.4874907
           j.feries -0.016109711 0.017954025  -0.8972757
                vac  0.014086577 0.007010398   2.0093834
    lo(gripa7, 0.9)  2.050456830 0.352452458   5.8176834
    lo(gripb6, 0.9)  0.811516470 0.350168165   2.3175050
   lo(tempmin, 0.9)  0.816587829 0.728802094   1.1204521
    lo(hummin, 0.9) -0.238649156 0.496191164  -0.4809621
  lo(tempmax2, 0.9) -2.798150915 0.861849732  -3.2466807
  lo(hummin12, 0.9) -1.047979872 0.564242125  -1.8573230
  lo(so224h13, 0.9)  0.994833711 0.447172626   2.2247196

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8951259 )

    Null Deviance: 2697.209 on 2769 degrees of freedom

Residual Deviance: 2567.386 on 2743.0376747 degrees of freedom

Number of Fisher Scoring Iterations: 3
>
```

9. ANALYSE DE SENSIBILITE

```
boxplot(morta$so224h)
boxplot(morta$so224h13)

summary(morta$so224h)
 Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
    3      13     18 22.04      28  124   52
summary(morta$so224h1)
 Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
    3      13     18 22.04      28  124   53
summary(morta$so224h3)
 Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
    3      13     18 22.04      28  124   55
```

```
summary(morta$so224h01)
 Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
    3      13   18.5 22.03    27.5  110   82
summary(morta$so224h03)
 Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
 3.25   13.75  18.75 21.97   27.25   91  138
summary(morta$so224h13)
  Min. 1st Qu. Median Mean 3rd Qu.  Max. NA's
 3.333   13.33  18.67   22      27 102.7  112

quantile(morta$so224h, c(.05,.95),na.rm=T)
  5% 95%
  7  49
quantile(morta$so224h1, c(.05,.95),na.rm=T)
  5% 95%
   7  49
quantile(morta$so224h3, c(.05,.95),na.rm=T)
  5% 95%
   7  49
quantile(morta$so224h01, c(.05,.95),na.rm=T)
 5% 95%
   8  47
quantile(morta$so224h03, c(.05,.95),na.rm=T)
  5%     95%
 8.5 44.7125
quantile(morta$so224h13, c(.05,.95),na.rm=T)
  5%      95%
   8 45.66667
```

```
mortot19a.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h1,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=(mortot<16&so224h1>7&so2
24h1<49))

mortot19b.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h3,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=(mortot<16&so224h3>7&so2
24h3<49))

mortot19c.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h03,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=(mortot<16&so224h13>8.5
&so224h13<44.7))

mortot19d.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h13,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=(mortot<16&so224h13>8&s
o224h13<45.7))

mortot19e.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9
)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so2
24h01,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=(mortot<16&so224h13>8&s
o224h13<47))
```

```
summary.glm(mortot19a.gam,cor=F)
summary.glm(mortot19b.gam,cor=F)
summary.glm(mortot19c.gam,cor=F)
summary.glm(mortot19d.gam,cor=F)

> summary.glm(mortot19a.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h1, 0.9), family =
       quasi(log, mu), data = morta, subset = (
       mortot < 16 & so224h1 > 7 & so224h1 < 49),
       na.action = na.omit)
Deviance Residuals:
       Min        1Q      Median        3Q     Max
 -3.224606 -0.700697 -0.02791914 0.6331195 2.27995

Coefficients:
                          Value  Std. Error
        (Intercept)  2.113957596 0.022788623
lo(trend, 730/2922)  0.232787246 0.403649579
     s(dowf.num, 4)  0.002526024 0.003273030
           j.feries -0.010181057 0.018820497
                vac  0.018430473 0.007388488
    lo(gripa7, 0.9)  1.973864871 0.365953605
    lo(gripb6, 0.9)  1.000770342 0.412316525
   lo(tempmin, 0.9)  0.974645806 0.765075030
    lo(hummin, 0.9) -0.288821408 0.517186510
  lo(tempmax2, 0.9) -2.748612722 0.922461013
  lo(hummin12, 0.9) -1.009484673 0.589451419
   lo(so224h1, 0.9)  1.110699353 0.638495998
                         t value
        (Intercept)  92.7637270
lo(trend, 730/2922)   0.5767063
     s(dowf.num, 4)   0.7717693
           j.feries  -0.5409558
                vac   2.4944851
    lo(gripa7, 0.9)   5.3937571
    lo(gripb6, 0.9)   2.4271895
   lo(tempmin, 0.9)   1.2739219
    lo(hummin, 0.9)  -0.5584473
  lo(tempmax2, 0.9)  -2.9796519
```

```
  lo(hummin12, 0.9)  -1.7125833
   lo(so224h1, 0.9)   1.7395557

(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.8895192 )

    Null Deviance: 2414.861 on 2520 degrees of freedo
m

Residual Deviance: 2313.311 on 2494.1568298 degrees o
f freedom

Number of Fisher Scoring Iterations: 3
> summary.glm(mortot19b.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h3, 0.9), family =
       quasi(log, mu), data = morta, subset = (
       mortot < 16 & so224h3 > 7 & so224h3 < 49),
       na.action = na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q
 -3.218233 -0.7115588 -0.03757061 0.6395295
      Max
 2.297471

Coefficients:
                         Value  Std. Error
        (Intercept)  2.09926316 0.022717674
lo(trend, 730/2922)  0.33722075 0.403521523
     s(dowf.num, 4)  0.00405594 0.003275913
           j.feries -0.01869006 0.018899085
                vac  0.01371346 0.007465173
    lo(gripa7, 0.9)  2.06813857 0.366919158
    lo(gripb6, 0.9)  1.15771307 0.397490103
   lo(tempmin, 0.9)  0.84954763 0.770163626
    lo(hummin, 0.9) -0.36287772 0.519279860
  lo(tempmax2, 0.9) -2.85308027 0.904373931
  lo(hummin12, 0.9) -1.09831175 0.595231783
   lo(so224h3, 0.9)  0.94000885 0.623628030
                        t value
        (Intercept)  92.4066059
lo(trend, 730/2922)   0.8356946
```

```
     s(dowf.num, 4)   1.2381099
           j.feries  -0.9889401
                vac   1.8369920
    lo(gripa7, 0.9)   5.6364966
    lo(gripb6, 0.9)   2.9125582
   lo(tempmin, 0.9)   1.1030742
    lo(hummin, 0.9)  -0.6988095
  lo(tempmax2, 0.9)  -3.1547573
  lo(hummin12, 0.9)  -1.8451833
   lo(so224h3, 0.9)   1.5073230

(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.9066305 )

    Null Deviance: 2478.938 on 2522 degrees of freedo
m

Residual Deviance: 2369.712 on 2496.156614 degrees of
 freedom

Number of Fisher Scoring Iterations: 3

> summary.glm(mortot19c.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
       0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h03, 0.9),
       family = quasi(log, mu), data = morta, subset = (mortot < 16 & so224h13 > 8.5
& so224h13 < 44.7),
       na.action = na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q      Max
 -3.252437 -0.7183125 -0.03384838 0.6362135 2.271321

Coefficients:
                          Value  Std. Error    t value
        (Intercept)  2.109728063 0.023282500  90.614325
lo(trend, 730/2922)  0.438792147 0.440935070   0.995140
     s(dowf.num, 4)  0.004303960 0.003344323   1.286945
           j.feries -0.006475845 0.019251922  -0.336374
                vac  0.015529824 0.007576596   2.049710
    lo(gripa7, 0.9)  2.015232957 0.365190451   5.518307
    lo(gripb6, 0.9)  1.281989045 0.430045482   2.981055
   lo(tempmin, 0.9)  1.108799486 0.795644889   1.393586
    lo(hummin, 0.9) -0.588299508 0.529337995  -1.111387
```

```
  lo(tempmax2, 0.9) -2.730877568 0.921174295  -2.964561
  lo(hummin12, 0.9) -0.725249845 0.599231672  -1.210300
  lo(so224h03, 0.9)  0.888904882 0.675418037   1.316081

(Dispersion Parameter for Quasi-likelihood family taken to be 0.8979975 )

    Null Deviance: 2353.523 on 2432 degrees of freedom

Residual Deviance: 2253.104 on 2406.137934 degrees of freedom

Number of Fisher Scoring Iterations: 3
>

> summary.glm(mortot19d.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(
       dowf.num, 4) + j.feries + vac + lo(gripa7,
       0.9) + lo(gripb6, 0.9) + lo(tempmin, 0.9) +
       lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(
       hummin12, 0.9) + lo(so224h13, 0.9), family
        = quasi(log, mu), data = morta, subset = (
       mortot < 16 & so224h13 > 8 & so224h13 < 45.7
       ), na.action = na.omit)
Deviance Residuals:
       Min         1Q      Median        3Q
 -3.242262 -0.7148296 -0.03080921 0.6389121
      Max
 2.270989

Coefficients:
                          Value  Std. Error
        (Intercept)  2.108111354 0.023026968
lo(trend, 730/2922)  0.304237432 0.425786065
     s(dowf.num, 4)  0.004048291 0.003312972
           j.feries -0.009045669 0.019030445
                vac  0.014170511 0.007474029
    lo(gripa7, 0.9)  1.901515215 0.364089241
    lo(gripb6, 0.9)  1.093672909 0.405638393
   lo(tempmin, 0.9)  0.681499880 0.773381677
    lo(hummin, 0.9) -0.465566221 0.525061473
  lo(tempmax2, 0.9) -2.618133027 0.912014219
  lo(hummin12, 0.9) -0.745011428 0.596123932
  lo(so224h13, 0.9)  0.678556866 0.637173765
                        t value
        (Intercept)  91.5496696
lo(trend, 730/2922)   0.7145312
```

```
     s(dowf.num, 4)   1.2219514
           j.feries  -0.4753262
                vac   1.8959668
    lo(gripa7, 0.9)   5.2226625
    lo(gripb6, 0.9)   2.6961770
   lo(tempmin, 0.9)   0.8811948
    lo(hummin, 0.9)  -0.8866890
  lo(tempmax2, 0.9)  -2.8707151
  lo(hummin12, 0.9)  -1.2497593
  lo(so224h13, 0.9)   1.0649479

(Dispersion Parameter for Quasi-likelihood family tak
en to be 0.9034343 )

    Null Deviance: 2428.209 on 2494 degrees of freedo
m

Residual Deviance: 2328.083 on 2468.122348 degrees of
 freedom

Number of Fisher Scoring Iterations: 3

> summary.glm(mortot19e.gam,cor=F)

Call: gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac
+ lo(gripa7, 0.9) + lo(gripb
6,
      0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12,
0.9) + lo(so224h01, 0.9),
      family = quasi(log, mu), data = morta, subset = (mortot < 16 & so224h13 > 8 &
so224h13 < 47),
      na.action = na.omit)
Deviance Residuals:
       Min         1Q      Median       3Q      Max
 -3.251262 -0.7180203 -0.03539833 0.638891 2.275668

Coefficients:
                          Value  Std. Error     t value
        (Intercept)  2.111812436 0.023097145  91.4317528
lo(trend, 730/2922)  0.495599649 0.426585228   1.1617834
     s(dowf.num, 4)  0.003621527 0.003319446   1.0910033
           j.feries -0.007272287 0.019066452  -0.3814180
                vac  0.013975923 0.007472815   1.8702352
    lo(gripa7, 0.9)  1.968358147 0.364840984   5.3951125
    lo(gripb6, 0.9)  1.178262405 0.407390937   2.8922156
   lo(tempmin, 0.9)  0.917946410 0.789177361   1.1631687
    lo(hummin, 0.9) -0.454003571 0.522519596  -0.8688738
```

lo(tempmax2, 0.9) -2.457501065 0.914581575 -2.6870223

lo(hummin12, 0.9) -0.758523074 0.596690111 -1.2712178

lo(so224h01, 0.9) 1.060233559 0.599307426   1.7690980

(Dispersion Parameter for Quasi-likelihood family taken to be 0.9032229)

Null Deviance: 2420.999 on 2487 degrees of freedom

Residual Deviance: 2319.531 on 2461.0200378 degrees of freedom

Number of Fisher Scoring Iterations: 3

Aucun des modèles ne reste significatif mais tous les décalages donnent un effet positif ; les effets les plus importants sont ceux de so24h1, so224h3 ; celui qui résiste lieux est so224h01 (de plus 01 est combiné de décalage le plus souvent retrouvé dans la littérature en raison de son intérêt pour les comparaisons entre les différentes études).

donc à priori :

mortotso201.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h01,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

INTERACTION TEMPMIN*HUMMIN

mortotso201.inter.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7,.9)+lo(gripb6,.9)+lo(tempmin,hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(so224h01,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)

Comparaison des AIC des modèles avec et sans interaction

> AIC(mortotso201.gam)

gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,

0.9) + lo(tempmin, 0.9) + lo(hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12, 0.9) + lo(so224h01, 0.9),

family = quasi(log, mu), data = morta, subset = mortot < 16, na.action = na.omit)

Degrees of Freedom Total = 2799

Degrees of Freedom Residual = 2772.039

Residual Deviance = 2568.625

AIC= 2616.527

> AIC(mortotso201.inter.gam)

gam(formula = mortot ~ lo(trend, 730/2922) + s(dowf.num, 4) + j.feries + vac + lo(gripa7, 0.9) + lo(gripb6,

0.9) + lo(tempmin, hummin, 0.9) + lo(tempmax2, 0.9) + lo(hummin12, 0.9) + lo(so224h01, 0.9), family =

quasi(log, mu), data = morta, subset = mortot < 16, na.action = na.omit)

Degrees of Freedom Total = 2799

Degrees of Freedom Residual = 2770.703

Residual Deviance = 2566.009

AIC= 2616.263

Très peu de différence ; d'autre part, les graphes ne montrent quasiment pas d'interaction (quasiment plan)

DONC ON EN RESTE AU MODELE SANS INTERACTION

```
mortot.so201.gam_gam(mortot~lo(trend,730/2922)+s(dowf.num,4)+j.feries+vac+lo(gripa7
,.9)+lo(gripb6,.9)+lo(tempmin,.9)+lo(hummin,.9)+lo(tempmax2,.9)+lo(hummin12,.9)+lo(
so224h01,.9),family=quasi(log,mu),data=morta,na=na.omit,subset=mortot<16)
```