



HAL
open science

Surveillance sanitaire à partir de données des services d'urgence : modélisation de séries temporelles et analyse automatique

Nicolas J. Jung

► To cite this version:

Nicolas J. Jung. Surveillance sanitaire à partir de données des services d'urgence : modélisation de séries temporelles et analyse automatique. *Méthodologie [stat.ME]*. 2010. dumas-00516268

HAL Id: dumas-00516268

<https://dumas.ccsd.cnrs.fr/dumas-00516268>

Submitted on 9 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ DE STRASBOURG

MASTER STATISTIQUES ET APPLICATIONS

RAPPORT DE STAGE DE M1

**Surveillance sanitaire à partir de données
des services d'urgence : modélisation de
séries temporelles et analyse automatique**

Auteur :
Nicolas JUNG

Tuteur :
Franck GOLLIOT



Période du stage : Juin-Août 2010

Remerciements

Je souhaiterai tout d'abord remercier Mr. Franck GOLLIOT pour m'avoir accueilli au sein de la Cire Languedoc-Roussillon. Je tiens également à le remercier pour sa disponibilité et pour son appui constant lors de ce stage.

Je tiens également à remercier tous les agents de la Cire avec lesquels j'ai pu travailler pour leurs conseils et leur soutien. Je les remercie également de m'avoir fait profiter de leur expérience professionnelle.

Préambule

Ce stage a eu lieu à la Cire du Languedoc Roussillon à l'adresse suivante :

CIRE
28, Parc Club du Millénaire
1025, Rue Henri Becquerel - CS 30001
34067 MONTPELLIER CEDEX 2

Ce stage a été effectué sous l'autorité du coordonnateur de la Cire, Mr. Franck GOLLIOT de juin à août 2010.

Le présent document a été entièrement rédigé en L^AT_EX.

Tous les calculs ont été réalisés grâce au logiciel R. Les principaux packages utilisés sont : «epitools» pour la gestion des dates, «surveillance» pour les algorithmes de surveillance, et «Tcl/Tk» pour créer des programmes interactifs. Le code source R a été placé en annexe.

Pour toute question a propos de ce rapport, veuillez envoyer un courriel à l'adresse suivante :

nicolas.julien.jung@gmail.com

Synthèse

Avant la canicule de 2003, la surveillance sanitaire en France était essentiellement portée sur des pathologies connues (maladies à déclaration obligatoire, grippe, gastro-entérites...). Mais au vue des conséquences sanitaires importantes de la vague de chaleur de l'été 2003, l'objectif d'une surveillance sanitaire plus générale s'est imposé. A la surveillance spécifique qui était en place est venue s'ajouter une surveillance dite « non-spécifique » dont le but est de détecter tout événement sanitaire inhabituel, afin d'en prévenir suffisamment tôt les effets.

Le nombre important de données à analyser quotidiennement et leur grande diversité nécessitent de disposer de méthodes statistiques permettant en routine de déclencher une alarme lorsque la valeur d'un indicateur s'écarte de son niveau habituel.

Les objectifs de ce stage s'inscrivent dans cette perspective : comparer et évaluer les performances de trois méthodes statistiques pour la surveillance continue de données d'activité hospitalière (Cartes de contrôle cusum décrites par Hutwagner, Algorithme de Farrington, méthode Arima), et mettre en place un système de surveillance non-spécifique qui s'appuie sur les méthodes les plus performantes. En complément de ce travail, un système de surveillance spécifique basé sur la méthode de régression de Serfling a également été proposé pour l'analyse de séries temporelles de cas de Bronchiolite et de gastroentérites afin de renforcer le dispositif existant.

La comparaison a été effectuée au moyen de séries semi-artificielles issues de données des urgences. Plusieurs indicateurs, dont la sensibilité et la spécificité ont été calculés.

Les résultats de la phase de test montrent une prédominance des méthodes à historique long (Farrington et Arima). Néanmoins, compte tenu de la grande diversité des séries à analyser et des objectifs de surveillance, il semble intéressant d'effectuer la surveillance non-spécifique au moyen de plusieurs méthodes.

Abstract

Before the 2003 heat wave, French public health surveillance systems were mainly focused on known diseases (notifiable diseases, influenza, gastroenteritis ...). Considering the important health consequences of this heat wave, the objective of health surveillance systems has become more general. A non-specific surveillance system has been developed to prevent unexpected health events. The goal is to detect any unusual sanitary event, in order to prevent its effects.

The diversity and the large number of count data analysed daily in such system need to implement statistical methods routinely in order to trigger an alarm when the value of the observed count data deviates from its usual level.

The objectives of this training course were to compare and evaluate the performance of three statistical methods for continuous monitoring of hospital activity data (Hutwagner's cusum control charts, Farrington's Algorithm, Arima method), and to establish a non-specific monitoring system based on the most efficient methods. A specific surveillance system based on the Serfling's regression method was also proposed for strengthen the existing control system.

This comparison was made with semi-artificial time series of hospital activity count data. Several indicators, including sensitivity and specificity were calculated.

The study results have showed a predominance of methods using large historical data (Farrington and Arima). However, given the wide variety of time series and surveillance objectives, it seems useful to analyse data using several methods.

Table des matières

Remerciements	iii
Préambule	v
Synthèse	vii
1 Contexte et objectifs	1
1.1 La surveillance sanitaire à partir de données d'activité d'urgences : surveiller pour alerter	1
1.2 Le système de surveillance mis en place en Languedoc-Roussillon . .	2
1.3 Objectifs du stage	3
2 Matériel et méthodes	5
2.1 Analyse des données	5
2.1.1 Données pour l'analyse syndromique ou non-spécifique : SRVA	5
2.1.2 Données pour l'analyse spécifique : Oscour	6
2.1.3 Amélioration de la qualité des données	6
2.2 Méthodes Statistiques Utilisées	7
2.2.1 Une analyse des séries chronologiques : Arima	7
2.2.2 L'algorithme de Farrington	8
2.2.3 Cartes de contrôle : la méthode CUSUM	10
2.2.4 Une méthode pour l'analyse spécifique : Serfling	13
2.3 Comparaison des méthodes	15
2.3.1 Constitution de séries semi-artificielles	15
2.3.2 Procédure d'analyse de résultats	17
3 Résultats	19
3.1 Description des données	19
3.2 Phase de test et adaptation des méthodes	23
3.2.1 Mise en œuvre des différentes méthodes	23
3.2.2 Les séries semi-artificielles	28
3.3 Résultats de la phase de test	29
3.3.1 Interprétation graphique	29
3.3.2 Analyse de la variance	31

4	Automatisation	37
4.1	Système de surveillance syndromique	37
4.2	Système de surveillance spécifique	44
5	Discussions	49
5.1	Rappel des résultats	49
5.2	Comparaison avec les études déjà publiées	50
5.3	Limites de notre étude de simulation et des méthodes statistiques utilisées	51
5.4	Avantages et limites du modèle de Serfling	53
5.5	Automatisation : apports et limites	54
6	Conclusion	57
A	Données et analyses	59
A.1	Données Oscour	59
A.2	Données Srva	59
A.3	Analyse de la variance	60
B	Résultats des tests	63
B.1	Tableaux de résultat	63
B.2	Graphiques	66
B.3	Analyse de variance	71
	B.3.1 Quelques lignes du tableau d'analyse	71
	B.3.2 Diagnostiques du modèle	71
C	Code source R	73
C.1	Mise en forme des données	73
C.2	Phase de test	80
	C.2.1 La fonction pour générer les signaux	80
	C.2.2 Les méthodes choisies pour la phase de test	84
	C.2.3 Appel des fonctions pour les tester	91
	C.2.4 Analyse des résultats	96
C.3	Mise en routine	98
	C.3.1 Amélioration de la qualité des données	98
	C.3.2 Partie syndromique	100
	C.3.3 Partie spécifique	119

Table des figures

2.1	Schématisation du choix de la période de référence pour la méthode de Farrington	10
2.2	Périodes de référence des différentes méthodes	13
2.3	Simulation pic rapide 5-6 (à gauche) et lent 7-8 (à droite)	16
3.1	Première représentation des données journalières SRVA	19
3.2	Première représentation des données hebdomadaires Oscour	20
3.3	Résolution des problèmes de décalage pour les hospitalisations à Carcassonne	21
3.4	Boîtes à moustaches pour le nombre de primo-passages de Montpellier	21
3.5	Fonction d'autocorrélation après transformation logarithmique	24
3.6	Fonction d'autocorrélation partielle après transformation logarithmique	24
3.7	Residus du modèle	25
3.8	Fonction d'autocorrélation des résidus du modèle	26
3.9	Fonction d'autocorrélation partielle des résidus du modèle	26
3.10	Période de référence pour la méthode de Farrington	27
3.11	Les différents signaux que nous avons voulu détecter	28
3.12	Exemple : graphique pour le signal 6	29
3.13	Performance des méthodes Farrington et Arima en fonction de l'indicateur de variance des séries.	34
3.14	Sensibilité et spécificité des méthodes Arima et Farrington	35
4.1	Tableau de contrôle généré par notre outil de surveillance syndromique	39
4.2	Graphique Arima généré par l'outil syndromique	41
4.3	Graphique Farrington généré par l'outil syndromique	42
4.4	Graphique Cusum généré par l'outil syndromique	43
4.5	Seuil en choisissant le choix automatique...	46
4.6	... ou le choix «annuel+semi» !	46
4.7	Seuil sans faire de décalage...	47
4.8	... ou avec un ajustement sur les 25 dernières semaines !	47
A.1	Diagramme quantile-quantile des résidus du modèle	60

B.1	Résultat pic 1	63
B.2	Résultat pic 2	63
B.3	Résultat pic 3	64
B.4	Résultat pic 4	64
B.5	Résultat du pic lent max = 25 %	64
B.6	Résultat du pic lent max = 50 %	64
B.7	Résultat du pic rapide max = 25 %	65
B.8	Résultat du pic rapide max = 50 %	65
B.9	Graphiques des résultats pour Carcassonne	66
B.10	Graphiques des résultats pour Montpellier	67
B.11	Graphiques des résultats pour Nimes	68
B.12	Graphiques des résultats pour Perpignan	69
B.13	Graphiques des résultats pour Mende	70
C.1	Diagramme quantile-quantile des résidus du modèle	80

Liste des tableaux

2.1	Méthodologie de l'analyse des résultats	17
3.1	Résultats de l'analyse de variance	32
3.2	Choix de la méthode en fonction de l'intensité du pic	33
3.3	Indicateur de variance des séries	33
3.4	Tableau de choix	35

Chapitre 1

Contexte et objectifs

1.1 La surveillance sanitaire à partir de données d'activité d'urgences : surveiller pour alerter

La surveillance sanitaire a pour but de détecter tout évènement pouvant représenter un risque pour la santé publique : maladies infectieuses, exposition environnementale, bioterrorisme... Cette surveillance est le résultat d'un processus continu : collecte de données, analyse et interprétation de ces données, communication des résultats aux décideurs qui ont la possibilité de prendre des mesures spécifiques de contrôle en fonction des risques relevés.

C'est un instrument essentiel permettant d'éviter et de limiter les épidémies. Les déplacements fréquents des populations favorisent la diffusion rapide des maladies infectieuses augmentant ainsi fortement le risque épidémique. Nous avons par ailleurs pu constater ces dernières années la rapidité de diffusion de certaines pathologies : grippe H1N1, grippe aviaire... Il faut donc pouvoir réduire au maximum le délai entre le début de l'évènement sanitaire et sa détection afin d'avoir le temps de mettre en place les mesures permettant de limiter l'impact sanitaire.

Jusqu'en 2003, le système de surveillance en France était essentiellement basé sur la surveillance des épidémies saisonnières : grippe, bronchiolite, gastro-entérite... et sur le système des déclarations obligatoires de certaines maladies infectieuses. Cette surveillance est «spécifique» ne porte que sur certaines pathologies ciblées. Mais après la canicule de 2003, l'objectif a été de mettre en place un système de surveillance dit «non-spécifique», ou «syndromique», capable de détecter de nouvelles menaces pour la santé publique d'origines diverses. En effet, les conséquences sanitaires de cette canicule de forte intensité ont été majeures : le phénomène a entraîné d'une part un recours massif au système de soins d'urgences et d'autre, une augmentation brutale de la mortalité. Cette crise a montré que de tels phénomènes, dans leur origine, leur ampleur géographique et leur conséquences pouvaient

exister, et donc, se reproduire. Cela ne signifie pas que les outils de surveillance spécifique manquent de pertinence, c'est simplement qu'ils n'étaient pas prévus pour répondre à un objectif de surveillance globale. Les deux outils de surveillance, syndromique et spécifique, sont donc complémentaires. Dans ce cadre que l'Institut de Veille Sanitaire (InVS) a mis en place une surveillance syndromique qui recueillant des informations quotidiennement a pour but d'informer de l'état de la santé de la population au jour le jour.

La surveillance sanitaire s'appuie donc sur la surveillance syndromique d'une part, qui cherche à détecter des événements inhabituels et la surveillance spécifique d'autre part, qui permet de suivre l'évolution de pathologies connues. La statistique a dans ce contexte deux rôles différents puisque les objectifs ne sont pas les mêmes. Dans le premier cas, le but est de créer des seuils permettant de détecter, avec un risque de se tromper a priori, un événement sanitaire inhabituel par son importance ou sa gravité. Dans le second cas, il s'agit de déterminer le début d'une épidémie attendue et connue, dont on connaît, dans la plupart des cas, la forme et l'amplitude.

1.2 Le système de surveillance mis en place en Languedoc-Roussillon

Le service d'accueil dans lequel j'ai travaillé était la Cire du Languedoc-Roussillon. On donne ici la description d'une Cire :

«L'InVS dispose d'un réseau régional, les Cellules de l'InVS en région (Cire), qui relaient son action et exercent une partie de ses missions au niveau régional. Elles sont placées sous la responsabilité scientifique de la directrice générale de l'InVS et sont localisées au sein des Agences régionales de santé (ARS), au plus près de l'autorité sanitaire. Les Cire fournissent aux ARS un appui méthodologique et une expertise indépendante des signaux d'alerte sanitaire. Elles animent la veille en région.¹ »

Les Cires développent en région les systèmes d'information pour la surveillance sanitaire. Elles ont pour objectif la mise en place de l'animation et de la structuration du système de surveillance au niveau régional.

La Cire recueille journalièrement plusieurs fichiers de données. Dans le cadre de mon stage, il m'a été demandé de travailler sur deux types de fichiers : les fichiers du serveur régional de veille et d'alerte (SRVA), et les fichiers de l'organisation

1. Présentation extraite du site de l'InVS : http://www.invs.sante.fr/regions/cire_presentation.htm.

de la surveillance coordonnée des urgences (réseau Oscour). Les deux types de données sont recueillis auprès des services hospitaliers de la région, mais par deux réseaux différents. Pour chacune de ces deux sources, il y a un fichier par hôpital ou clinique. L'analyse de ces données est traitée dans les chapitres suivants.

1.3 Objectifs du stage

Dans le cadre de ce stage, plusieurs objectifs m'étaient proposés :

1. Développer une procédure de test permettant de juger de manière objective de l'efficacité de méthodes statistiques
2. Automatiser les méthodes statistiques et créer un outil de surveillance syndromique
3. Créer un outil pour la définition de seuil dans le cadre de la surveillance spécifique

Chapitre 2

Matériel et méthodes

Dans ce chapitre, nous nous attacherons à présenter les données sur lesquelles nous avons travaillé, les méthodes statistiques que nous avons utilisées ainsi que la procédure de test pour lesdites méthodes que nous avons appliquées.

2.1 Analyse des données

Les données proviennent de deux base de données : Oscour et SRVA. Les données Oscour correspondent à la partie spécifique de mon analyse, puisqu'elles contiennent les passages aux urgences pour différentes pathologies hivernales : grippe, bronchiolites et gastro-entérites. Les données SRVA sont quant à elles liées à la partie syndromique de mon analyse. Elles sont constituées de quatre indicateurs : les admissions dans les services d'urgence, les admissions dans les services d'urgence des personnes de plus de 75 ans, les admissions dans les urgences des enfants de moins de 1 an et les hospitalisations suite à une admission aux urgences. Les pôles de chaque département recueillent de surcroît le nombre d'affaires traitées par les services du SAMU.

2.1.1 Données pour l'analyse syndromique ou non-spécifique : SRVA

Les données Oscour sont, pour chaque pathologie, une agrégation de 7 hôpitaux et cliniques¹ de la région Languedoc-Roussillon. Ces établissements ont été choisis pour la qualité et la régularité de la transmission de leurs données. En effet, le système n'étant pas encore entièrement rodé, il arrive que certains établissements ne parviennent pas à fournir les données, ou qu'ils se trompent dans le codage informatique de celles-ci. L'agglomération des données a été effectuée afin de palier

1. Liste des 7 établissements pour Oscour : CH Ales - CH Bagnols - CH Carcassonne - CHRU Montpellier - Clin St Louis - Poly St Roch - Poly St Jean

à des problèmes d'effectif. Le pas de temps de ces données est hebdomadaire. Quelques lignes des données sont données en annexe A.1.

2.1.2 Données pour l'analyse spécifique : Oscour (Résumé des passages aux urgences)

Les données SRVA sont recueillies auprès des 10 plus gros centres hospitaliers de la région². Ces données sont établies de manière quotidienne. Pour chacun des établissements nous avons donc une série pour chacun des indicateurs défini ci-dessus. Les séries diffèrent essentiellement par la taille des effectifs : certains hôpitaux ont des admissions plus conséquentes que d'autres et les indicateurs ont des effectifs très variables. On se rendra facilement compte de ces différences en regardant les données mises dans l'annexe A.2.

2.1.3 Amélioration de la qualité des données

Les transmissions des données étant assujetties à des erreurs produisant des outliers et à des «oublis» conduisant à des valeurs manquantes, nous avons décidé d'établir une procédure permettant d'améliorer la qualité des données. En effet, nous présumons que des données trop faussées conduiront à des résultats biaisés. Nous décidons tout d'abord de supprimer toute valeur visiblement fautive : elle sera donc considérée comme manquante. Cette décision pourra également être prise en concertation avec les agents du service qui travaillant régulièrement avec ces données ont une idée précise de la validité d'une valeur. Il s'agit alors de remplacer les valeurs manquantes. Nous le ferons selon la procédure suivante :

- La valeur manquante est isolée (i.e. : si X_t est manquante, ni X_{t-1} ni X_{t+1} le sont) : nous avons simplement remplacé la valeur par la moyenne des deux observations environnantes.
- La valeur manquante n'est pas isolée (i.e. : si X_t est manquante, X_{t-1} ou X_{t+1} l'est) : nous retirons temporairement toutes les valeurs manquantes contiguës, calculons les estimations à moyenne mobile des deux valeurs entre lesquelles se trouvent les valeurs manquantes, et répartissons uniformément nos valeurs manquantes sur la droite ainsi créée.

Nous nous attacherons également à analyser précisément les données, afin de déterminer leurs caractéristiques. Grâce à des graphiques pertinents et à des analyses de variance, nous tenterons de mettre en l'évidence les possibles effets «jour de la semaine», mois ou année.

2. Liste des 10 centres hospitaliers : Carcassonne, Mende, Montpellier, Beziers, Ales, Perpignan, Narbonne, Bagnols, Nîmes, Sète

2.2 Méthodes Statistiques Utilisées

Dans cette partie nous allons décrire une à une les méthodes statistiques utilisées ; nous nous attacherons à donner les hypothèses sous-jacentes à l'utilisation de ces méthodes ainsi que les diagnostics a posteriori permettant de valider l'utilisation de la méthode.

Les trois premières méthodes seront Arima, Farrington, et Cusum. Elles seront destinées à l'analyse syndromique et nous les testerons dans cet optique. La dernière méthode, Serfling, sera uniquement destiné à l'analyse spécifique.

2.2.1 Une analyse des séries chronologiques : Arima

Rappel de la définition d'un processus Arma :

Definition 1. *Un processus linéaire stationnaire $\{X_t\}$ est appelé ARMA(p, q), p et q étant positifs, s'il existe des constantes $a_1 \dots a_p$ et $\theta_1 \dots \theta_q$ avec $a_p \neq 0$ et $\theta_q \neq 0$ et ε_t un bruit blanc gaussien tels que :*

$$X_t = \sum_{k=1}^p a_k X_{t-k} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t$$

Le processus $\{\varepsilon_t\}$ est appelé processus des innovations.

On notera par ailleurs que si $p = 0$, X_t sera appelé processus à moyenne mobile (Moving Average : MA) et que si $q = 0$, X_t sera appelé processus AutoRégressif (AR)

L'analyse des données a suivi rigoureusement la procédure de Box et Jenkins. Une référence complète sera trouvée dans [1]³. En utilisant des modèles de type ARIMA, des prévisions avec leur intervalle de confiance pourront être établies. Sitôt qu'une valeur dépassera les intervalles de confiance calculés, une alerte statistique sera déclenchée. La procédure de Box-Jenkins se déroule généralement en trois étapes résumées ici :

- **Stationnarisation de la série** : un processus $\{X_t\}$ est dit stationnaire si la covariance entre X_t et X_{t+h} ne dépend que de h . La non-stationnarité d'un tel processus peut être assignée à deux causes. La première est que la partie déterministe (i.e. la tendance et la saisonnalité) n'est pas constante. Dans ce cas on peut appliquer des opérateurs de différentiation. La deuxième peut provenir d'une non-stationnarité de la variance, auquel cas il faut envisager d'effectuer une transformation stabilisante (logarithmique, racine-carrée...).

3. Un cours très complet par Rainer von Sachs et Sébastien Van Bellegem peut être trouvé à l'adresse suivante : http://www.math.univ-montp2.fr/~ribereau/matos/S-TAT2414_syllabus4ed.pdf

- **Identification et estimation des paramètres** : il faut d’abord parvenir à identifier le modèle stochastique adéquat. Pour cela nous disposons principalement de deux outils : la fonction d’autocorrélation (ACF) et la fonction d’autocorrélation partielle (PACF). Tandis que la fonction PACF nous donne une indication sur l’ordre de la partie autorégressive, la fonction ACF nous permet de choisir correctement l’ordre de la partie à moyenne mobile. L’estimation de l’écart-type devant être minimum, cela peut nous permettre de choisir l’ordre de la différentiation. L’estimation des paramètres peut ensuite être réalisé par maximum de vraisemblance. Notons également qu’il est possible de se servir d’indicateurs comme AIC pour choisir le meilleur modèle possible.
- **Diagnostic** : les résidus de notre modèle, c’est à dire la différence entre la valeur observée et la valeur estimée, doivent satisfaire trois conditions, qui expriment le fait que les résidus doivent être un bruit «blanc gaussien» :
 - Leur moyenne ne doit pas être significativement différente de zéro. En effet, nous avons enlevé toute composante déterministe.
 - Leur distribution doit être gaussienne. Pour vérifier cette condition, nous pourrons faire un test de Kolmogorov-Smirnov ou établir le tracé d’un diagramme quantile-quantile («qq-plot»).
 - Ils ne doivent pas contenir d’autocorrélation résiduelle. Pour vérifier cette condition, nous pourrons faire un test de Box-Ljung.
 Si le diagnostic du modèle est bon, nous pouvons l’utiliser pour faire de la prévision statistique.

2.2.2 L’algorithme de Farrington

Cet algorithme a été proposé pour la première fois par Farrington [4]. Pour cette méthode l’objectif est tout d’abord de déterminer les périodes de référence à utiliser. Elles sont définies génériquement par la taille d’une demi-fenêtre w et par la taille de l’historique en année b . Si le jour actuel est le x^{eme} de l’année y , la période de référence sera définie par les jours $x - w$ à $x + w$ des années $y - 1$ à $y - b$. Finalement, le nombre de jour considéré pour la période de référence à chaque étape est $b(2w + 1)$. On notera qu’il est recommandé de prendre un historique d’au moins deux-trois ans pour avoir suffisamment de données à chaque pas.

On suppose alors que la variable d’intérêt est distribuée, sur la période de référence γ_i considérée, selon une densité de probabilité de moyenne μ_i et de variance $\phi\mu_i$. Le modèle linéaire généralisé suivant est alors donné :

$$\log(\mu_i) = \alpha + \beta t_i$$

avec α une constante et β une tendance linéaire.

Les paramètres du modèles sont estimés par une méthode de quasi-vraisemblance. L'estimation de la variance se fera par la formule corrigée suivante :

$$\hat{\phi} = \max \left(\frac{1}{n-p} \sum_{i=1}^n \omega_i \frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i}, 1 \right)$$

avec p valant 1 ou 2 selon que l'on décide d'inclure ou non β dans le modèle. La tendance β sera incluse dans le modèle si et seulement si les trois conditions suivantes sont réalisées :

- L'historique des données est d'au moins trois ans
- Le coefficient est significatif à un seuil $\alpha = 5\%$
- Pour éviter des aberrations nous rajoutons la condition : $\hat{\mu}_0 \leq \max(y_i, i \in 1 \dots n)$

avec $\hat{\mu}_0$ l'estimation à un instant donné, que l'on peut réécrire sous la forme :

$$\hat{\mu}_0 = e^{\hat{\alpha} + \hat{\beta}t_0}$$

Par hypothèse, les données devront suivre une loi de Poisson lorsque la taille de l'effectif sera petit, ou une loi normale lorsque l'effectif sera plus conséquent. Afin de symétriser nos données une transformation à la puissance 2/3 sera généralement appliquée.

Les observations seront pondérés de manière spécifique, de manière à donner moins d'importance aux observations qui correspondent à des pics passés. En effet, ceux-ci diminueraient fortement la sensibilité de la méthode. La régression ci-dessus est appliquée une première fois, et les résidus suivants sont alors établis :

$$s_i = \frac{3}{2\hat{\phi}^{\frac{1}{2}}} \frac{y_i^{\frac{2}{3}} - \hat{\mu}_i^{\frac{2}{3}}}{\mu_i^{\frac{1}{6}} (1 - h_{ii})^{\frac{1}{2}}}$$

avec h_{ii} les éléments diagonaux de la Hat-Matrice. Les pondérations sont alors définies de la manière suivante :

$$\omega_i = \begin{cases} \eta s_i^{-1} & \text{si } s_i < 1 \\ \eta & \text{sinon} \end{cases}$$

avec η choisi tel que $\sum \omega_i = 1$. Ceci est donc une manière de s'affranchir d'un traitement potentiel des données, puisque cette méthode permet de prendre en compte les pics passés en leur donnant moins d'importance.

Nous terminerons cette partie en faisant quelques remarques sur cette méthode. Nous donnons ici un schéma permettant d'appréhender visuellement la période de référence considérée :

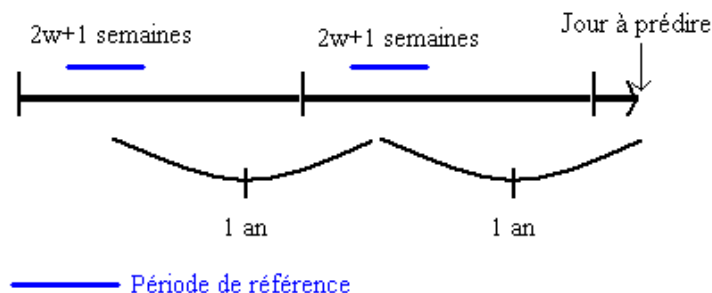


FIGURE 2.1 – Schématisation du choix de la période de référence pour la méthode de Farrington

Toutes les observations de la dernière année en cours ne sont pas prises en compte dans le calcul du seuil ; cela conduit à une remarque très importante sur cette méthode. Si dans l'année courante, une modification du comportement du phénomène étudié devait avoir lieu, il ne serait plus possible d'utiliser cette méthode. Cet algorithme suppose donc une certaine régularité dans le comportement du phénomène étudié.

2.2.3 Cartes de contrôle : la méthode CUSUM

Introduction aux cartes de contrôle

Supposons dans un premier temps que nos observations $x = (x_1, \dots, x_n)$ soient la réalisation d'un processus stochastique $X = (X_1, \dots, X_t)$. Nous nous donnons une statistique y_t construite à chaque temps donné à partir des observations. L'idée des cartes de contrôle est de faire figurer ces points sur une carte en y ajoutant deux limites, appelées limites de contrôle, et au-delà desquelles une alarme est déclenchée. Bien entendu, dans notre cas, nous nous bornerons à considérer la limite supérieure de contrôle, puisque un nombre «trop faible» de cas ne doit pas conduire à une alarme.

On différencie généralement les cartes de contrôle en deux catégories : celle sans mémoire (cartes de Shewart par exemple) et celle qui tiennent compte, avec plus ou moins d'importance des observations passées. La carte *CUSUM* que nous présentons fait partie de cette dernière catégorie.

Les cartes Cusum classiques

Les cartes de contrôle *CUSUM* sont définies par deux paramètres : h et k . Le premier intervient pour définir la zone d'alerte (c'est le seuil) tandis que le

deuxième intervient dans le calcul des quantités *CUSUM* suivantes :

$$S_t = \max \left(0, S_{t-1} + \frac{X_t - \mu_0}{\sigma_0} - k \right)$$

où μ_0 et σ_0 sont les valeurs estimées ou attendues de notre processus, et où on pose $S_0 = 0$. La zone d'alerte se définit alors simplement. On notera qu'à l'observation de cette formule, k peut être considéré comme l'écart que l'on désire mettre en évidence entre la valeur observée et la valeur attendue. Une alerte sera déclenchée sitôt que $S_t > h$. Toute la difficulté réside alors à faire un choix judicieux de ces deux paramètres. Nous allons nous servir des deux indicateurs suivants :

Definition 2. On appelle ARL_0 ⁴ ou «in-control average run length» le temps moyen avant de déclencher une fausse alarme.

Definition 3. On appelle ARL_1 ou «out of control average run length» le temps moyen entre un changement significatif dans notre processus et le déclenchement de l'alarme.

Par exemple, le choix $k = 1/2$ et $h = 2$ qui est souvent utilisé correspond à une ARL_0 de 500.

Le choix $k = \frac{1}{2}$ se révèle souvent être judicieux. Siegmund [16] a calculé l' ARL_0 en fonction de h :

$$ARL_0 = 2(e^a - a - 1)$$

avec $a = h + 1.66$. Et le paramètre h en fonction de l' ARL_0 , avec dans un premier temps $k = \frac{1}{2}$:

$$\begin{aligned} h &\approx \left(\frac{ARL_0 + 4}{ARL_0 + 2} \right) \ln \left(\frac{ARL_0}{2} + 1 \right) - 1.166 \\ &\approx \left(\frac{2k^2 ARL_0 + 2}{2k^2 ARL_0 + 1} \right) \frac{\ln(1 + 2k^2 ARL_0)}{2k} - 1.166 \end{aligned}$$

Puisque les baisses ne sont pas prises en compte, nous préférons considérer la version positive de la carte *CUSUM* :

$$\begin{aligned} S_t &= \max \left(0, S_{t-1} + \frac{X_t - (\mu_0 + k\sigma_0)}{\sigma_0} \right) \\ &= \max \left(0, S_{t-1} + \frac{X_t - \mu_0}{\sigma_0} - k \right) \end{aligned}$$

Et nous déclenchons une alerte sitôt que $S_t > 2$. Dans notre formule, X_t est notre variable d'intérêt, μ_0 et σ_0 la valeur attendue de la moyenne et de l'écart-type, et k est le paramètre de sensibilité.

4. POM_0 dans la littérature française pour Période Opérationnelle Moyenne

Les méthodes Cusum modifiées que nous utiliserons

Un des principal défaut de cette carte Cusum est qu'elle ne prend pas en compte les tendances et les saisonnalités. C'est pourquoi nous avons choisi d'appliquer la méthode Cusum en changeant les périodes de référence, en nous référant aux travaux de Hutwagner [7].

Nous prendrons toujours une période de référence de 7 jours grâce à une fenêtre glissante. Ainsi, toutes les variations journalières sont prises en compte. Il faut néanmoins faire attention dans le cas où nous aurions des saisonnalités plus importantes, mensuelles ou annuelles par exemple.

Cette méthode permet trois niveaux de sensibilité pour autant d'algorithmes : $C1 - MILD$, $C2 - MEDIUM$ et $C3 - ULTRA$; la méthode $C1 - MILD$ est la moins sensible et la méthode $C3 - ULTRA$ est la plus sensible. Nous décrivons maintenant ces trois algorithmes.

$C1 - MILD$

Pour cette méthode nous prendrons les 7 jours précédents le jour de l'observation pour période de référence à chaque pas. C'est le choix de cette fenêtre qui rend cet algorithme aussi peu sensible. En effet, puisque la valeur de la veille est immédiatement prise en compte pour le calcul des paramètres, une augmentation lente ne sera pas prise en compte. Aussi, le risque de déclencher plusieurs alarmes consécutives est plus faible, puisque la forte valeur de la veille est immédiatement prise en compte.

Cet algorithme se simplifie en considérant les valeurs au-dessus de la ligne de la moyenne plus trois fois l'écart-type. On peut ensuite changer la sensibilité, en prenant par exemple les points au-dessus de la moyennes plus 2 fois l'écart-type.

$C2 - MEDIUM$

Cet algorithme est le même que le précédent, sauf qu'il faut changer la période la période de référence. En effet, au lieu de prendre les sept derniers jours, nous prendrons les sept jours compris entre le neuvième et le troisième jour avant le jour de l'observation. Contrairement à la méthode précédente, une forte augmentation ne sera donc pas tout de suite prise en compte, ce qui rend l'algorithme plus sensible.

$C3 - ULTRA$

Pour la méthode $C3 - ULTRA$ nous gardons la même période de référence que ci-dessus. En revanche, pour la statistique de test, nous prendrons la somme des

trois dernières quantités *CUSUM* de cette dernière méthode (i.e. $S_{t-2} + S_{t-1} + S_t$) et nous la comparerons au seuil de référence 2.

Commentaires sur cette méthode

Ces trois méthodes se distinguent avant tout par la période de référence qu'elle considèrent, comme nous le rappelle le schéma suivant :

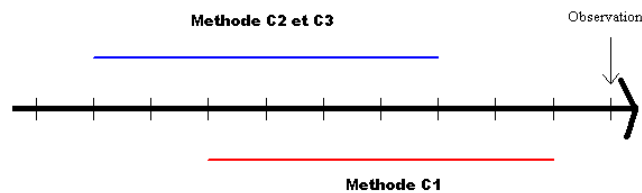


FIGURE 2.2 – Périodes de référence des différentes méthodes

2.2.4 Une méthode pour l'analyse spécifique : Serfling

Cette méthode a été proposée pour la première fois dans [15] par Robert E. Serfling⁵. C'est une méthode que l'on peut uniquement utiliser dans le cadre de la surveillance spécifique, et qui suppose que le phénomène étudié est parfaitement régulier. Il est important d'insister sur la régularité du phénomène : c'est une méthode qui permet de suivre statistiquement l'évolution du phénomène attendu dont la forme et l'amplitude du signal est souvent connu.

L'algorithme de Serfling est le suivant :

1. Estimation de la tendance et de la saisonnalité
2. Régression linéaire intégrant ces tendances et saisonnalités
3. Ecrêtation des données au-delà de la borne supérieure de l'intervalle de confiance
4. Régression linéaire sur les données écrêtées et calcul du seuil.

Nous avons pris la liberté de modifier légèrement cet algorithme que nous expliquerons ensuite point par point :

1. Estimation de la tendance
2. Ecrêtation brute des données selon un quantile
3. Régression linéaire intégrant ces tendances et saisonnalités

5. Il existe une application développée sur internet permettant de mettre en œuvre cette méthode : http://www.u707.jussieu.fr/periodic_regression/. L'application a été développée par les auteurs de l'article [11].

4. Ecrêtation des données au-delà de la borne supérieure de l'intervalle de confiance
5. Régression linéaire sur les données écrêtées et calcul du seuil
6. Ajustement horizontal du seuil par minimisation des résidus au carrés des dernières semaines

Tout d'abord, l'estimation de la tendance a été réalisée grâce à une régression linéaire simple. On teste ensuite, au risque $\alpha = 5\%$ la nullité de la tendance linéaire. Si celle-ci est effectivement présente, on l'enlève. La première écrêtation consiste à choisir un quantile et à «couper» toutes les données au-dessus de ce quantile. Ce choix est fortement dépendant de notre volonté à trouver le seuil final plus ou moins sensible. En effet, plus nous écréterons avec un quantile bas, et plus le seuil sera haut. Le choix de ce quantile doit aussi dépendre des courbes épidémiques. En effet, nous nous interdisons d'écrêter des données correspondant à des périodes non-épidémiques : cela biaiserait la méthode.

Ensuite, nous procédons à une première régression linéaire. Nous posons le modèle complet suivant :

$$\begin{aligned}
 y_t &= \alpha + bt \\
 &+ a_1 \cos\left(\frac{2\pi t}{52}\right) + a_2 \sin\left(\frac{2\pi t}{52}\right) \\
 &+ a_3 \cos\left(\frac{2\pi t}{26}\right) + a_4 \sin\left(\frac{2\pi t}{26}\right) \\
 &+ a_5 \cos\left(\frac{2\pi t}{13}\right) + a_6 \sin\left(\frac{2\pi t}{13}\right) \\
 &+ a_7 \cos\left(\frac{2\pi t}{4}\right) + a_8 \sin\left(\frac{2\pi t}{4}\right)
 \end{aligned}$$

La première ligne correspond à la constante ainsi qu'à la potentielle tendance linéaire. La deuxième ligne correspond à une saisonnalité annuelle, la troisième à une saisonnalité semestrielle, la quatrième à une saisonnalité trimestrielle et la dernière à une saisonnalité mensuelle. Bien entendu, il n'est pas nécessaire de mettre tous les régresseurs dans le modèle ; il est également possible de choisir les régresseurs en faisant une procédure stepwise avec critère AIC.

Nous écréterons ensuite toutes les valeurs dépassant la borne supérieure de l'intervalle de confiance selon un risque α déterminé à l'avance, et recommençons la même régression.

Il est alors possible d'ajuster le modèle aux dernières semaines. En effet, la régression linéaire fournissant un modèle rigide, il se peut que les données soient

décalées par rapport au seuil sur les dernières semaines. C'est pourquoi nous avons choisi de minimiser les résidus au carré sur les dernières semaines. Nous aurions aussi pu pondérer les observations, en donnant un poids plus important aux observations récentes, mais nous ne souhaitons pas changer la forme du seuil, et seul son décalage temporel nous importait.

2.3 Comparaison des méthodes

Notre comparaison se limite à l'étude des méthodes que nous voulons utiliser pour la surveillance syndromique. L'algorithme de Serfling est donc écarté.

2.3.1 Constitution de séries semi-artificielles

Pour tester l'efficacité de nos méthodes, et pour pouvoir les comparer, nous sommes passés par une phase de simulation. Nous décrivons maintenant la procédure de simulation et de comparaison.

Nous avons choisi une période d'une année dans nos données retenues et, pour chacune des méthodes, nous avons simulé 8 signaux différents :

- Signal 1 : augmentation de 100% sur 1 jour
- Signal 2 : augmentation de 50% sur 3 jours
- Signal 3 : augmentation de 20% sur 6 jours
- Signal 4 : augmentation de 10% sur 9 jours
- Signal 5 : augmentation rapide jusqu'à 25% sur 4 jours
- Signal 6 : augmentation rapide jusqu'à 50% sur 5 jours
- Signal 7 : augmentation lente jusqu'à 25% sur 12 jours
- Signal 8 : augmentation lente jusqu'à 50% sur 14 jours

Pour les signaux 1 à 4, il s'agit d'une augmentation brutale et constante. Pour les signaux 5 à 8, nous avons établi nos augmentations en fonction des courbes suivantes⁶ :

6. Ces deux fonctions issues d'une densité gamma se trouvent en annexe C.2.1

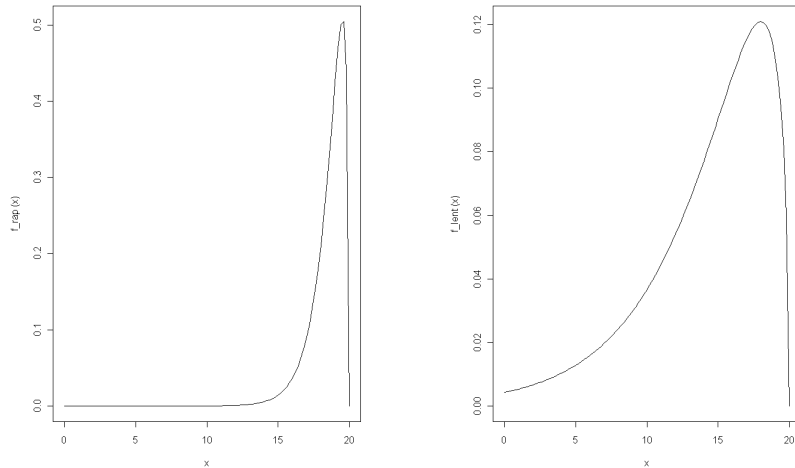


FIGURE 2.3 – Simulation pic rapide 5-6 (à gauche) et lent 7-8 (à droite)

Pour chacune des méthodes ainsi que pour chacune des séries nous avons cherché à détecter ces différents signaux sur toute la période annuelle considérée. Nous avons ensuite calculé quatre indicateurs de performance. Pour certaines définitions à suivre, nous nous sommes fiés à [17]. Le premier indicateur que nous avons considéré est la sensibilité de la méthode. Cet indicateur a été calculé de la manière suivante :

$$\text{Sensibilité} = \frac{\text{Nombre de signaux détectés}}{\text{Nombre de signaux à détecter}}$$

La sensibilité est donc la part de signaux détectés. Nous avons par ailleurs convenu qu'un signal était détecté sitôt qu'un jour dans la période du signal était détecté. Nous avons ensuite pris en compte la spécificité, calculée sur la série non-modifiée, de la manière suivante :

$$\text{Spécificité} = 1 - \frac{\text{Nombre de signaux détectés}}{\text{Nombre de jours du test}}$$

La spécificité est donc la proportion de fausses alarmes. Il est néanmoins important de faire remarquer qu'en écrivant notre définition, nous supposons de fait que toutes les alarmes déclenchées sur la série non-modifiée sont des fausses alarmes. Ceci est un postulat qui se trouve être parfaitement inexact dans la réalité. C'est pourquoi, la spécificité telle que nous la calculons doit plutôt être considérée comme un indicateur de spécificité. Notre but étant la comparaison des méthodes entre elles et l'erreur potentielle sur la spécificité n'étant pas propre à telle ou telle méthode, nous n'aurons pas d'impact sur les résultats finaux.

Nous avons également calculé le délai moyen de détection (pour les signaux s'étendant sur plusieurs jours) et la variance du délai de détection.

2.3.2 Procédure d'analyse de résultats

Dans un premier temps, nous chercherons à présenter les résultats obtenus sous la forme de graphiques astucieux, permettant de juger rapidement de l'efficacité relative d'une méthode selon une série et un signal donnés. Nous espérons ainsi pouvoir dégager une tendance «visuelle» que nous confirmerons par la suite.

La deuxième partie consiste à établir un protocole objectif de choix de méthode selon plusieurs objectifs. Nous nous fixons comme objectif de maximiser la somme *sensibilité* + *spécificité*, et de choisir parmi les potentiels ex-æquo ceux qui minimisent le délai de détection puis la variance. Pour cela nous allons faire une analyse de la variance. Nous posons alors le modèle :

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + \varepsilon_{ijk}$$

où Y_{ijk} représente la somme *sensibilité* + *spécificité* pour la méthode i , pour le pic j et la série k . Les facteurs $(\alpha\beta)_{ij}$ et $(\alpha\gamma)_{ik}$ sont les interactions des facteurs correspondants et ε_{ijk} est l'erreur. Nous devons bien entendu vérifier les conditions d'application classiques d'un tel modèle. Pour l'interprétation des résultats, nous donnons alors la grille de lecture suivante :

Objectif	Méthode
Meilleure méthode en toute généralité	Regarder les α_i
Meilleure méthode en fonction de la série	Regarder $\alpha_i + (\alpha\gamma)_{ik}$
Meilleure méthode en fonction du pic	Regarder $\alpha_i + (\alpha\beta)_{ij}$
Meilleure méthode en fonction du pic et de la série	Regarder $\alpha_i + (\alpha\gamma)_{ik} + (\alpha\beta)_{ij}$

TABLE 2.1 – Méthodologie de l'analyse des résultats

Chapitre 3

Résultats

Dans ce chapitre nous allons présenter tous les résultats que nous avons pu obtenir durant ce stage. Par ailleurs, nous expliquerons comment, en nous basant sur ces résultats, nous avons pu établir un système de surveillance.

3.1 Description des données

Notre premier travail consiste d'abord à étudier les données. Outre la mise en place effective de la procédure visant à améliorer la qualité des données, nous avons cherché tous les défauts potentiels que celles-ci pouvaient avoir. On donne ici une première représentation du nombre d'admissions aux urgences sur quatre ans (2007-2010), pour l'hôpital de Montpellier :

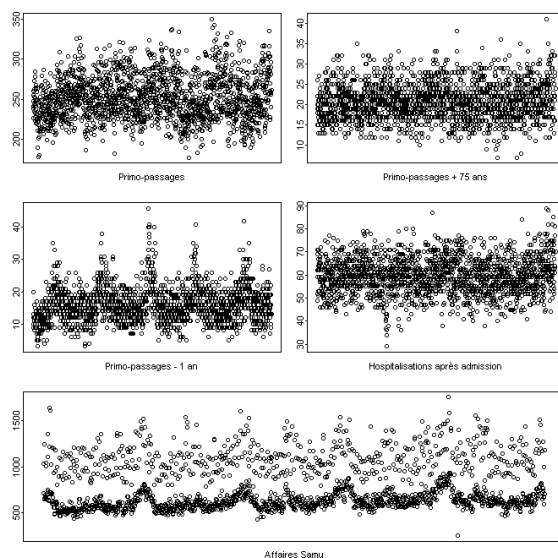


FIGURE 3.1 – Première représentation des données journalières SRVA

On constate que certaines séries semblent plus caractérisées que d'autres : par exemple, le mouvement saisonnier est beaucoup plus visible pour les affaires Samu. On donne également une première représentation des données Oscore, pour la bronchiolite, la grippe et les gastro-entérites :

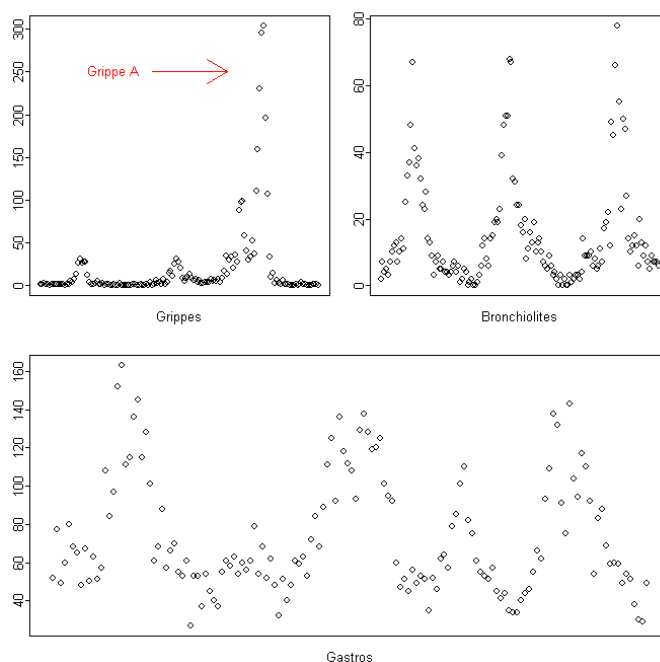


FIGURE 3.2 – Première représentation des données hebdomadaires Oscore

Ces séries sont très régulières, et c'est la raison pour laquelle il est possible de faire de la surveillance spécifique. Néanmoins, il sera difficile d'établir des résultats cohérents pour la série «Grippes». En effet, l'épidémie de grippe A de 2009 constitue un pic beaucoup plus important que les années précédentes. Toutefois, cette différence n'est pas tant le fait d'une augmentation réelle du nombre de cas entre les années, que la modification de la codification de la grippe.

En regardant chacune des séries de nos données SRVA, nous nous sommes aperçus d'un problème assez récurrent. En effet, certaines séries présentent des cassures nettes, représentant selon les cas l'ouverture ou la fermeture d'un service ou d'une clinique, une modification de procédure... Il nous faut donc remédier à ce problème, sous peine de biaiser toutes les analyses que nous allons faire. Nous proposons d'ajuster la différence des moyennes des séries avant et après la cassure, ce qui revient à supposer que nous considérons que la-dite cassure n'entraîne pas de changement dans la forme des données, mais seulement une augmentation moyenne d'un certain nombre de cas. Voilà pour exemple sur la série des hospitalisations à Carcassonne :

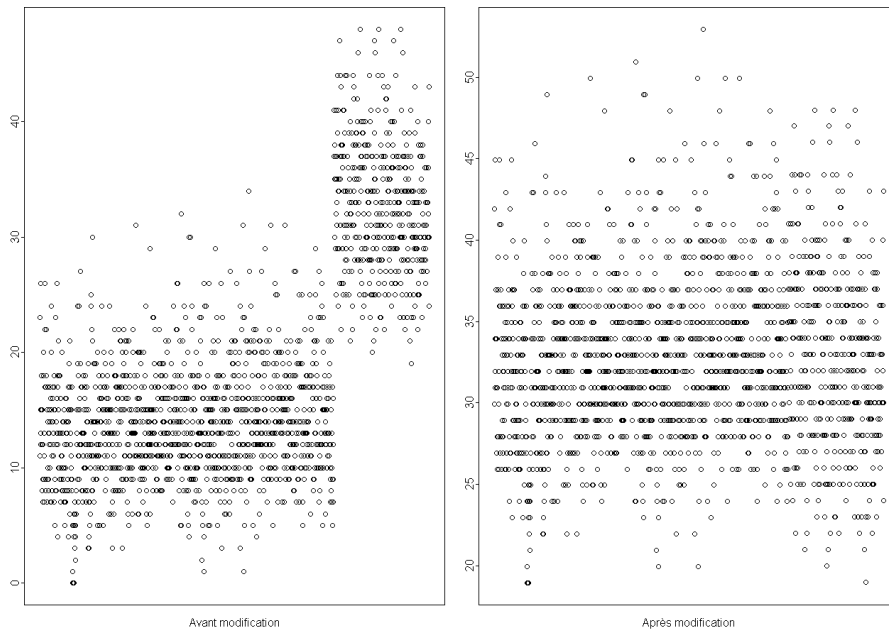


FIGURE 3.3 – Résolution des problèmes de décalage pour les hospitalisations à Carcassonne

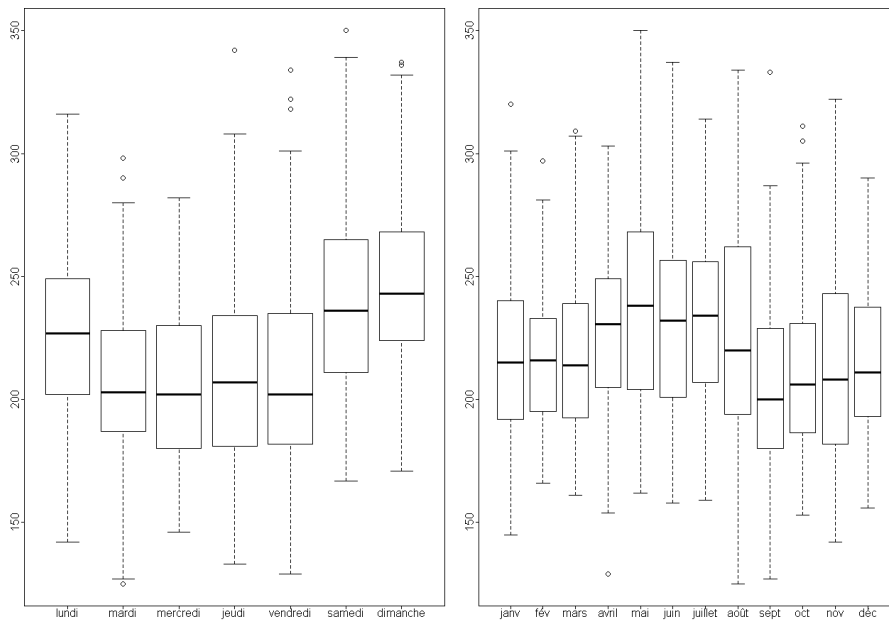


FIGURE 3.4 – Boîtes à moustaches pour le nombre de primo-passages de Montpellier

La régularité des données Oscour et l'objectif d'une surveillance spécifique nous dispensent d'une étude complémentaire de ces données. En revanche, pour les données issues du serveur SRVA, nous devons chercher à déterminer si elles comportent des mouvements saisonniers, journaliers... Nous présentons les résultats pour les admissions aux urgences de Montpellier, mais ceux-ci sont extensibles à l'ensemble des séries SRVA sur lesquels nous avons travaillé. Nous donnons tout d'abord quelques boxplots ci-dessus.

On constate que les week-ends sont des jours de forte affluence dans les services d'urgence. Cette impression correspond à une réalité «pragmatique» : en effet, le week-end, la plupart des cabinets de médecins ne sont pas ouverts, redirigeant ainsi leurs patients vers les services d'urgence. Les mois estivaux (de mai à août) connaissent aussi une augmentation conséquente des passages aux urgences ; cela correspond sans doute à l'augmentation de la population dans le département de l'Hérault pendant cette saison touristique.

Pour confirmer notre impression visuelle, nous allons faire une analyse de variance. Nous donnons ici la table que nous recueillons grâce à R :

Analysis of Variance Table

Response: adm

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
jour	6	45676	7612.7	93.954	< 2.2e-16 ***
mois	11	18195	1654.1	20.414	< 2.2e-16 ***
Residuals	1745	140905	81.0		

Bien entendu, pour donner quelques crédits aux p-valeurs affichées, il nous faut vérifier les hypothèses classiques d'un modèle d'analyse de la variance. Le test de normalité sur les résidus du modèle de Kolmogorov-Smirnov nous donne une p-value satisfaisante ; nous avons mis en annexe A.3 un diagramme quantile-quantile. La normalité des erreurs étant vérifiée, nous pouvons effectuer un test de Bartlett afin de vérifier l'égalité des variances. Ce test se trouve également en annexe A.3. La p-valeur étant supérieur à 0.05, nous décidons de ne pas refuser l'hypothèse nulle d'égalité des variances.

Maintenant que nous avons pu vérifier les conditions d'utilisation du modèle nous pouvons donner du crédit aux résultats de l'analyse de variance. On conclut donc à un effet «jour» et à un effet «mois». Au vue de ces résultats, nous pouvons

envisager de faire des comparaisons multiples grâce à Tuckey. Les résultats (mis en annexe A.3) confirment les impression recueillies grâce aux graphiques établis plus haut.

L'enseignement principal de cette analyse est qu'il nous faudra tenir compte des effets «jour de la semaine» et «mois» pour obtenir des méthodes performantes.

3.2 Phase de test et adaptation des méthodes

Il faut tout d'abord rappeler que la phase de test ne concerne que les trois méthodes suivantes : Farrington, Arima, et les trois différentes cartes CUSUM. Par ailleurs, et ceci afin de limiter un temps de calcul déjà conséquent, nous nous sommes limités à étudier les séries des admissions dans les urgences pour les 5 hôpitaux principaux de chaque département : Carcassonne, Mende, Montpellier, Perpignan, et Nîmes.

3.2.1 Mise en œuvre des différentes méthodes

Nous allons voir dans cette partie comment nous sommes parvenus à adapter et à calibrer nos méthodes à nos jeux de données et à leurs spécificités.

Mise en œuvre de la méthode Arima

Un problème se pose d'entrée ; comment faire pour caler le modèle Arima pour chacune de nos séries ? Nous rappelons que dans la phase de test, chaque série va être modifiée chaque jour au cours d'une année, c'est-à-dire 365 fois. De manière théorique, il nous faudrait donc paramétrer le modèle à chaque fois. Quelques essais nous ont permis de légitimer le fait de garder le même modèle pour toute la période. Par ailleurs - et ceci est en quelques sortes la preuve que les séries se comportent de la même façon - nous avons choisi le même modèle pour les 5 centres hospitaliers. Néanmoins, dans un soucis de concision, nous ne présentons les résultats que dans le cadre du CHRU de Montpellier.

Nous avons suivis scrupuleusement l'algorithme donné dans la partie méthode. Nous avons commencé par appliquer une transformation logarithmique à nos données pour stabiliser la variance. Nous donnons alors la fonction d'autocorrélation et la fonction d'autocorrélation partielle que nous obtenons :

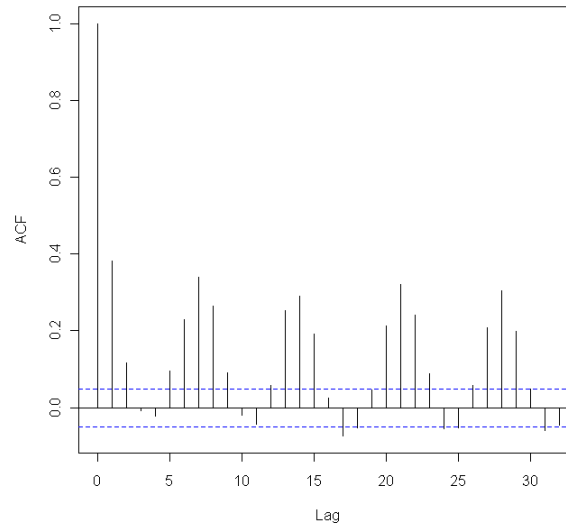


FIGURE 3.5 – Fonction d'autocorrélation après transformation logarithmique

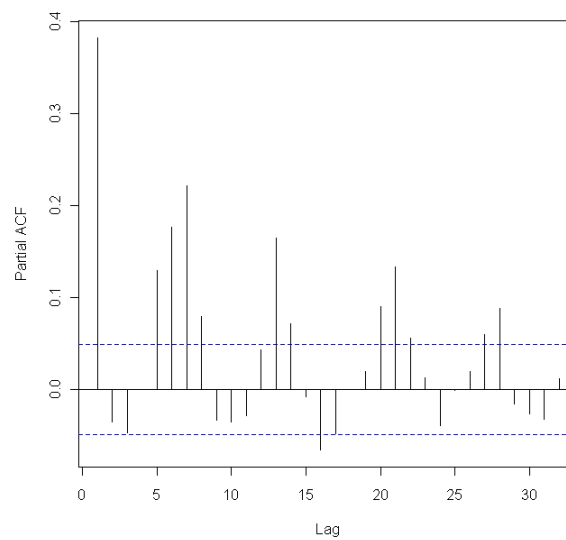


FIGURE 3.6 – Fonction d'autocorrélation partielle après transformation logarithmique

Il y a de fortes corrélations que nous nous devons de corriger. Nous décidons alors - après plusieurs essais - de choisir de le modèle $SARIMA(1, 1, 1) (0, 1, 1)$.

Pour faire ce choix, nous avons cherché à minimiser l'écart-type estimé du modèle. Il nous faut maintenant valider ce modèle. Nous rappelons les trois critères de validation : erreurs centrées, gaussiennes et non-corrélées. Nous ne disposons pas de test pour tester si les erreurs sont centrées, et c'est pourquoi nous fournissons simplement le graphe suivant :

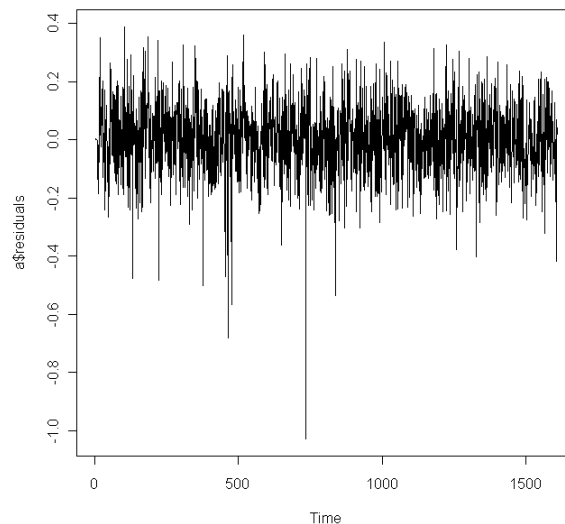


FIGURE 3.7 – Residus du modèle

La normalité nous est assurée par le test de Kolmogorov-Smirnov (hypothèse nulle étant la normalité) :

One-sample Kolmogorov-Smirnov test

```
data: a$residuals
D = 0.0337, p-value = 0.05125
alternative hypothesis: two-sided
```

L'hypothèse de normalité n'est donc pas rejetée. Et nous faisons alors un test de Box-Ljung pour tester la non-corrélation des erreurs :

Box-Ljung test

```
data: a$residuals
X-squared = 0.0129, df = 1, p-value = 0.9096
```

La non-corrélation n'étant pas rejetée, nous pouvons supposer que les conditions de validation sont respectées. Nous présentons pour confirmation les fonctions d'autocorrélation et d'autocorrélation partielles des résidus :

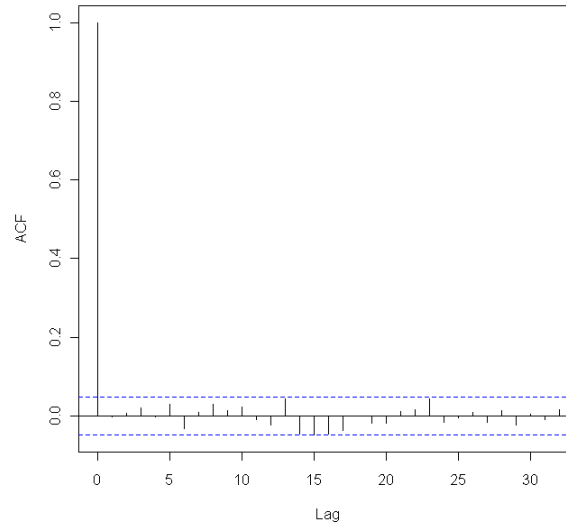


FIGURE 3.8 – Fonction d'autocorrélation des résidus du modèle

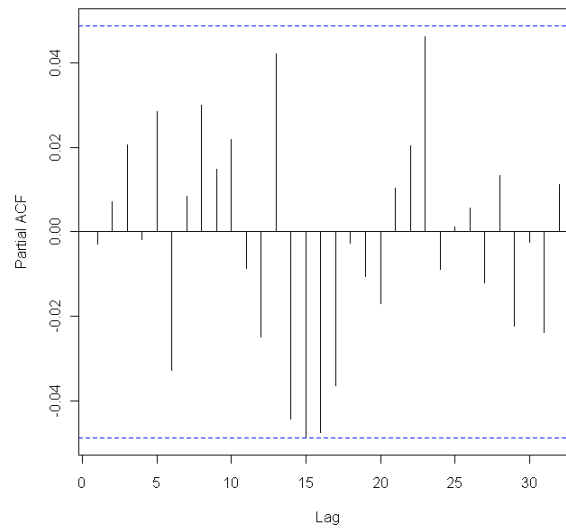


FIGURE 3.9 – Fonction d'autocorrélation partielle des résidus du modèle

Les diagnostics étant bons, nous pouvons donc utiliser ce modèle et le considérer comme bien calibré. Nous avons choisi d'étudier cette méthode en choisissant deux historiques différents : une fenêtre glissante de 365 jours et une fenêtre glissante de 100 jours. Nous avons également choisi de prendre deux risques de première espèce différents. Finalement, voici les trois méthodes retenues pour le test :

- $\alpha = 5\%$ et une fenêtre glissante de 365 jours
- $\alpha = 1\%$ et une fenêtre glissante de 365 jours
- $\alpha = 5\%$ et une fenêtre glissante de 100 jours

Mise en œuvre de la méthode de Farrington

Pour se conformer au fait que nos données ont un effet lié au jour de la semaine important, nous avons dû modifier légèrement la fonction de base¹. En effet, au lieu de prendre exactement les mêmes jours des années précédentes, nous avons choisi d'ajuster nos choix selon les jours de la semaine. Ainsi, si nous voulons faire une prédiction alors que nous sommes un lundi, nous prendrons de l'année passée le lundi tel que la distance entre ces deux jours soit maximale. Nous prendrons les trois lundis des semaines précédentes et suivantes, et cela pour chaque année de l'historique.



FIGURE 3.10 – Période de référence pour la méthode de Farrington

Mise en œuvre de la méthode des cartes CUSUM

Nous avons appliqué les trois méthodes : $C1 - MILD$, $C2 - MEDIUM$ et $C3 - EXTRA$. Pour les deux premières cartes de contrôle, nous avons comparé la valeur observée à la valeur moyenne sur leur période de référence respective ajouté à 3 fois l'écart-type sur cette même période. Pour la dernière carte de contrôle, nous avons calculé les quantités $CUSUM$ sur la même période de référence que pour $C2 - MILD$ et avons additionné à chaque instant les trois dernières valeurs de cette quantité.

Nous avons été conduit lors de nos différentes expériences à corriger la variance. En effet, au lieu de considérer la variance sur la période de référence, nous avons

1. Nous avons en effet modifié la fonction implémentée dans le package «surveillance» de R.

pris le minimum entre cette variance, et la variance sur tout l'historique que nous possédions, et ce, pour chacune des méthodes.

Les raisons de ce choix sont naturelles : nous craignons que l'effet «jour de la semaine» beaucoup plus conséquent certaines semaines augmente de façon démesurée la variance à ces moments précis. Nous espérons donc augmenter la sensibilité, sans toutefois diminuer de façon reprochable la spécificité.

3.2.2 Les séries semi-artificielles

Nous donnons ici un exemple pour chacun des signaux que nous avons cherché à simuler. Cela nous permet d'avoir un aspect visuel de ce que nous cherchons à détecter.

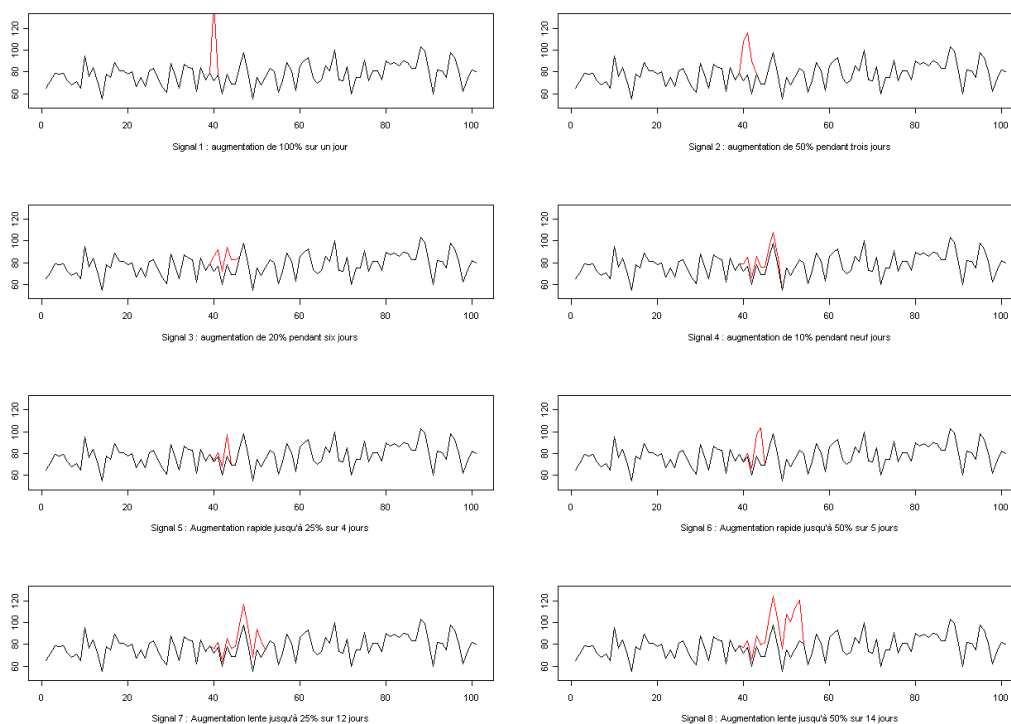


FIGURE 3.11 – Les différents signaux que nous avons voulu détecter

En regardant ces différents graphiques, nous comprenons que certains signaux seront plus difficile à détecter que d'autres. En effet, pour le signal 1 (augmentation de 100% sur un jour), le pic engendré est visible au premier regard. En revanche, détecter le signal 4 sera beaucoup plus délicat.

Une autre constatation nous conforte dans notre idée d'essayer chacune des méthodes sur une période de un an. En effet, plus l'augmentation s'applique à des valeurs élevées, plus le pic engendré est conséquent. La question peut par exemple être de savoir si nous sommes également apte à détecter un évènement qui se produirait dans un jour «creux».

3.3 Résultats de la phase de test

Pour présenter et analyser les résultats obtenus, nous procéderons en deux étapes, comme nous l'avons convenu : d'une part nous ferons des représentations graphiques, et d'autre part, nous ferons une analyse de variance. D'autre part, on trouvera les résultats par signaux en annexe B.1.

3.3.1 Interprétation graphique

Nous avons décidé de prendre une forme particulière de graphique dont nous présentons un exemple ici (l'ensemble des graphiques se trouvant par ailleurs en annexe B.2) :

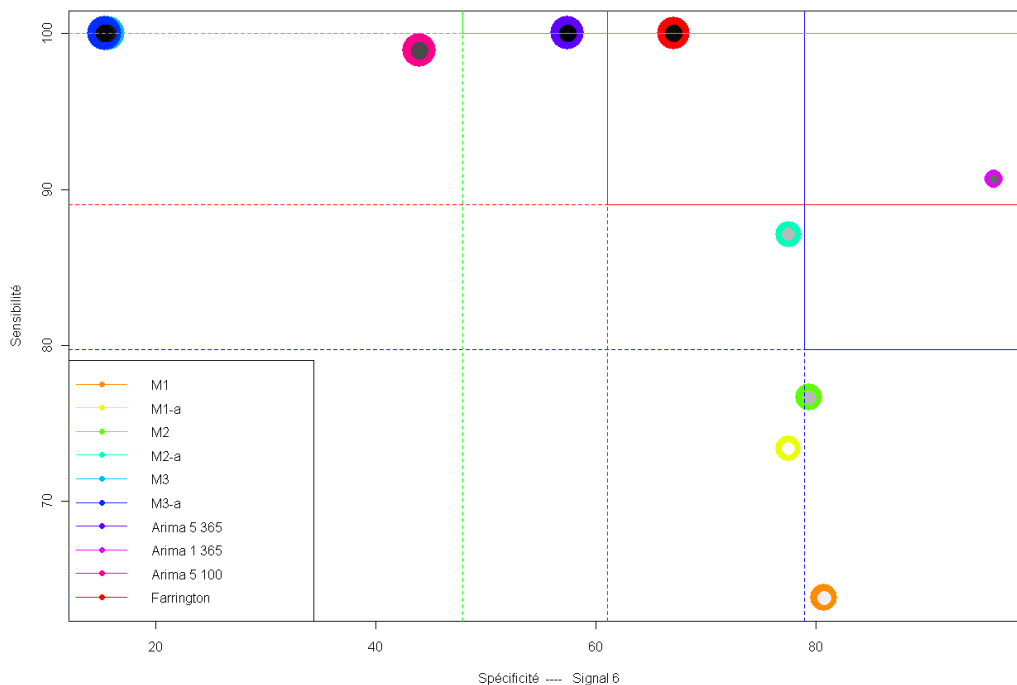


FIGURE 3.12 – Exemple : graphique pour le signal 6

Chaque méthode est donc représentée par un point de couleur spécifique. Il y a un graphique par série et par signal. L'axe des abscisses représentant la spécificité et l'axe des ordonnées représentant la sensibilité, une méthode sera d'autant meilleure qu'elle se trouvera en haut à droite. Ensuite, un point est d'autant plus gros que le délai de détection de la méthode associée est court, et son centre sera d'autant plus foncé que la variance sera faible. On a ensuite tracé plusieurs lignes permettant une meilleure compréhension du graphique :

- En rouge : la droite horizontale représente la moyenne en terme de sensibilité et la droite verticale représente la moyenne en terme de spécificité.
- En vert : on représente le 75% quantile en terme de sensibilité, et le 25% en terme de spécificité.
- En bleu : on représente le 25% quantile en terme de sensibilité, et le 75% en terme de spécificité.

Une méthode «bonne en moyenne» se retrouvera donc placée dans le rectangle rouge supérieur. Une bonne méthode spécifique se trouvera donc à droite la ligne verticale bleue, et une bonne méthode sensible se trouvera au-dessus de la ligne horizontale verte.

S'il s'agissait d'interpréter le graphique donné en exemple, nous dirions que nous possédons deux méthodes globalement intéressantes : Farrington et Arima avec historique de 365 jours et un risque alpha de 1%. S'il nous fallait choisir entre ces deux méthodes, nous choisirions Farrington si nous voulions une méthode très sensible et Arima si nous voulions une méthode très spécifique. Si cette distinction ne nous importait pas, nous choisirions Farrington, car le point le représentant étant plus gros, le délai de détection est meilleur. Le choix de l'une ou l'autre méthode est alors fonction des objectifs de surveillance.

Nous pouvons donner quelques remarques intéressantes sur l'efficacité de nos méthodes. Premièrement, dans une grande majorité des cas, pour les méthodes C1-Mild et C2-Medium, la majoration de la variance a permis d'avoir de meilleurs résultats. En effet, dans certains cas nous sommes parvenus - en comparaison avec la méthode classique - à augmenter la sensibilité de 10% en abaissant la spécificité que de 1% ou 2%. Le gain semble d'autant plus conséquent que le signal que nous cherchons à détecter est fort. En revanche, la méthode C3-extra étant déjà très sensible (c'est par ailleurs dans la majorité des cas une des méthodes les plus sensibles), cette modification perd de son efficacité et de son sens.

Une tendance assez nette se dégage de l'observation de ces graphiques ; en effet, les méthodes de Farrington, Arima avec un historique d'un an et, dans une moindre mesure, celle de C2-medium que nous avons modifiée, semblent être les méthodes les plus performantes. Nous pouvons répartir ces méthodes de la manière

suivante : Farrington et Arima avec historique d'un an et un risque de 5% sont les deux méthodes les plus sensibles, tandis que C2-medium modifiée et Arima à 1% sont les deux méthodes les plus spécifiques.

Une dernière remarque pour dire que la série de l'hôpital de Mende semble se comporter un peu différemment des autres ; nous chercherons des explications à cela dans une analyse plus précise.

3.3.2 Analyse de la variance

Comme nous l'avions prévu dans le chapitre précédent, nous allons faire une analyse de variance afin de pouvoir interpréter plus finement les résultats. Pour analyser les résultats nous nous référerons à la grille de lecture donnée dans le paragraphe 2.3.2. Puisque nous voulons majorer la somme *sensibilite+specificite*, nous compilons tous nos résultats dans un tableau à partir duquel nous ferons l'analyse de variance. Nous rappelons le modèle que nous posons :

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + \varepsilon_{ijk}$$

avec Y_{ijk} représente la somme *sensibilite+specificite* pour la méthode i , pour le pic j et la série k . $(\alpha\beta)_{ij}$, $(\alpha\gamma)_{ik}$, $(\beta\gamma)_{jk}$ sont les interactions des facteurs correspondants et ε_{ijk} est l'erreur.

Nous donnons la table d'analyse de la variance, après quoi nous nous attachons à montrer que les hypothèses classiques d'un tel modèle sont bien vérifiées.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
methode	9	17380	1931.2	84.4677	< 2.2e-16 ***
pic	7	191149	27307.1	1194.3935	< 2.2e-16 ***
serie	4	10800	2700.0	118.0966	< 2.2e-16 ***
methode:pic	63	14288	226.8	9.9195	< 2.2e-16 ***
methode:serie	36	5521	153.4	6.7075	< 2.2e-16 ***
Residuals	280	6402	22.9		

Il nous faut donc vérifier la normalité des résidus ainsi que l'homoscédasticité de la variance. Nous avons fait un test de Kolmogorov suivi d'un test de Bartlett qui nous permettent de valider ces hypothèses. Ils sont donnés en annexe B.3. Nous pouvons donc accorder un crédit au p-valeurs du tableau ci-dessus.

Tous les éléments que nous avons mis dans le tableau sont significatifs. De cela, nous pouvons tirer la remarque suivante : nous ne pourrions pas trouver une

méthode «générale» permettant d'établir une surveillance optimale quelle que soit la série et quel que soit le signal. Ce tableau suggère en fait la construction d'un arbre de décision de choix de méthode, en fonction du type de série sur laquelle doit s'établir la surveillance et du type de signal que l'on souhaite surveiller.

Nous donnons maintenant le tableau-résultat, dans lequel nous avons fait figurer sur chaque ligne, pour chacun des critères que nous avons retenus, les deux meilleures méthodes :

Choix du critère	Résultat 1				Résultat 2			
	Méthode	Spé.	Sens	Délai	Méthode	Spé.	Sens	Délai
Général	Farrington	74.08	89.34	3.06	Arima 365 5	76.8	85.6	3.1
Pic 1	Arima 365 1	99.4	97.3	1	C2-modifiée	98.024	97.584	1
Pic 2	Arima 365 1	98.19	90.88	1.17	Arima 365 5	88.08	98.13	1.10
Pic 3	Farrington	74.18	88.41	2.14	Arima 365 5	77.86	82.09	1.96
Pic 4	Farrington	65.88	73.90	3.4	Arima 365 5	69.18	62.36	1.96
Pic 5	Arima 365 5	84.5	64.18	3.12	Farrington	81.15	67.01	3.02
Pic 6	Arima 365 5	81.16	93.08	3.8	Arima 365 1	96.98	74.40	4.47
Pic 7	Farrington	60.00	94.3	5.3	Arima 365 5	61.37	83.63	5.4
Pic 8	C2-modifiée	74.28	90.16	7.1	Arima 365 1	91.92	67.69	7.9
Carcassonne	Farrington	82.83	87.36	3.4	Arima 365 5	77.20	89.12	3.09
Perpignan	Farrington	78.74	89.80	3.05	Arima 100 5	92.58	58.37	3.63
Nîmes	Arima 365 5	75.93	87.74	3.04	Farrington	72.08	90.45	3.02
Montpellier	Farrington	76.44	93.96	2.77	Arima 365 5	77.75	90.69	2.76
Mende	Arima 365 5	79.64	74.62	3.50	Arima 100 5	100	35.33	3.9

TABLE 3.1 – Résultats de l'analyse de variance

Ce tableau confirme nos premières impressions que nous avons notées en regardant les graphiques. De manière générale, les deux meilleures méthodes sont donc celle de Farrington et Arima avec historique d'un an et un risque de 5%.

Nous décidons de regrouper les pics en deux catégories, en prenant d'une part ceux dont l'intensité est plutôt faible et d'autre part ceux dont l'intensité est plutôt élevée et nous comptons le nombre d'apparitions (en première et deuxième position confondue) dans le tableau ci-dessus. Cela nous conduit au tableau suivant :

	Pic d'intensité faible (<25%)	Pic d'intensité forte (>25%)
Farrington	4	0
Arima 365 5	4	2
Arima 365 1	0	4
C2-modifiée	0	2

TABLE 3.2 – Choix de la méthode en fonction de l'intensité du pic

Ce tableau fait nettement apparaître une tendance : les méthodes que nous avons considérées comme étant sensibles trouvent leur intérêt dans les pics de faible amplitude, tandis que les méthodes que nous avons prises pour spécifiques trouvent leur intérêt dans les pics de forte amplitude. Ce résultat est par ailleurs plutôt logique. En effet, une méthode spécifique va déclencher peu de fausse alerte, mais sera suffisamment sensible pour des signaux importants. Une méthode sensible va avoir tendance à déclencher plus de fausses alarmes, mais de part ce fait, elle sera aussi plus à même de détecter de faibles variations.

Quant on regarde les résultats correspondants au type de série, on peut s'étonner de voir que pour chacune d'entre elles, sont retenues une méthode Arima et la méthode de Farrington, exceptée la série de l'hôpital de Mende. Il est donc légitime d'essayer de trouver une particularité à la série issue de Mende qui pourrait expliquer cette spécificité. Nous pensons que c'est la variabilité inhérente aux séries qui pourraient en être la cause, c'est pourquoi, nous décidons de calculer le rapport entre le quantile à 90% et celui à 10%. Nous donnons cet indicateur sous forme de tableau :

	q_{90}/q_{10}
Monptellier	1.32
Carcassonne	1.45
Perpignan	1.54
Nîmes	1.57
Mendes	1.83

TABLE 3.3 – Indicateur de variance des séries

Nous traçons alors, pour nos deux méthodes principales, c'est-à-dire Farrington et Arima 365 5, l'évolution de la somme *sensibilite* + *specificite* en fonction de ce nouvel indicateur :

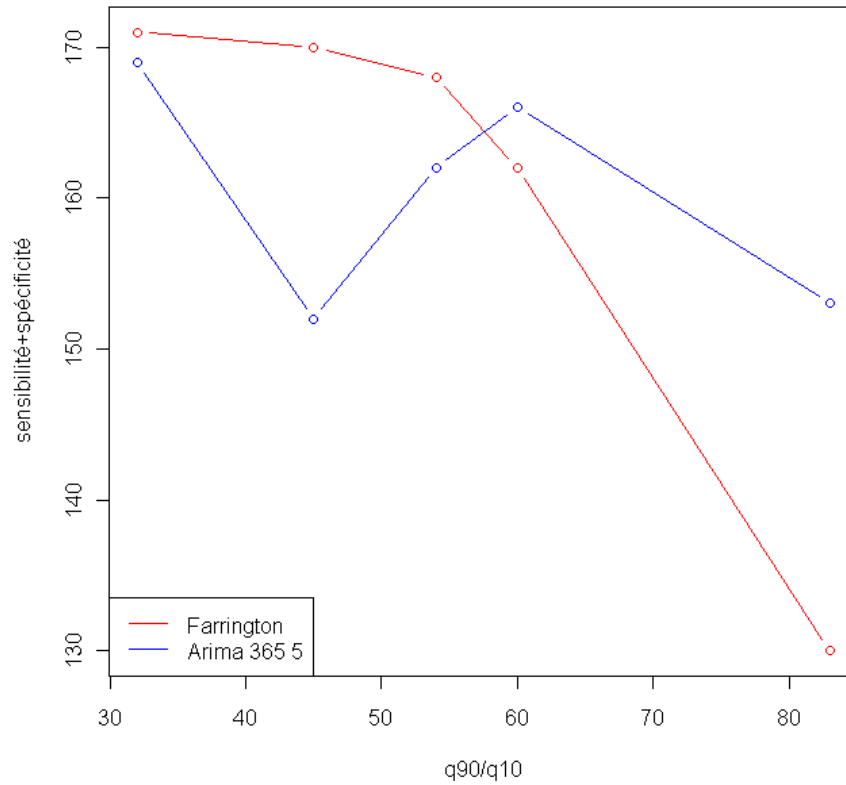


FIGURE 3.13 – Performance des méthodes Farrington et Arima en fonction de l'indicateur de variance des séries.

Il semblerait, au vue de ce graphique, que la méthode de Farrington soit beaucoup plus sensible à notre indicateur que ne peut l'être la méthode Arima. Ainsi, nous pouvons donc penser que c'est la variance inhérente des séries qui rend les données de Mende atypiques.

Pour pouvoir conclure, il nous faut faire une dernière remarque. Pour les pics de faibles intensités, nous retenons systématiquement les méthodes Farrington et Arima 365.5. Il suffit de regarder le graphe ci-dessous pour se rendre compte que Farrington semble être une méthode plus sensible dans ce cas de figure et la méthode Arima semble être plus spécifique.

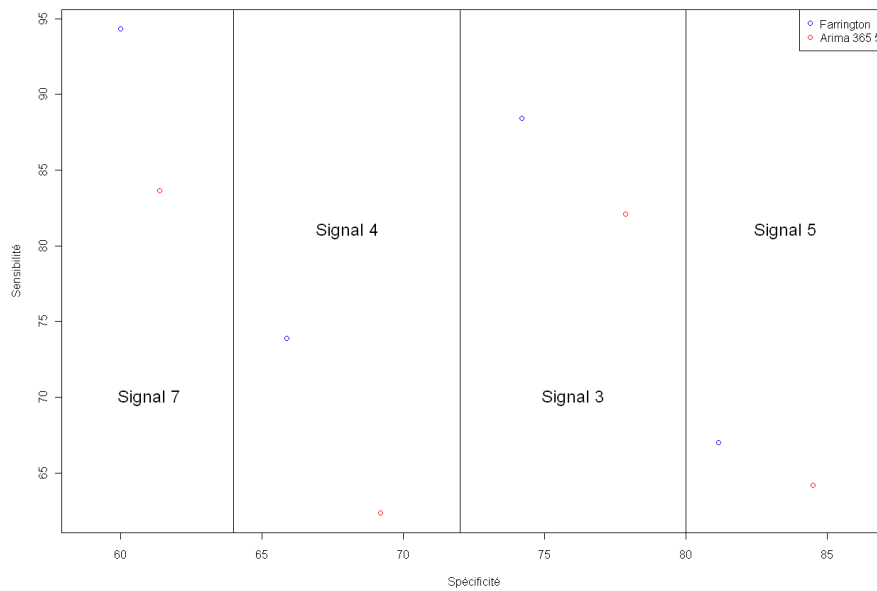


FIGURE 3.14 – Sensibilité et spécificité des méthodes Arima et Farrington

Nous pouvons donc maintenant, de manière tout à fait objective, dresser le tableau de choix suivant :

	Signal fort	Signal faible
Variance forte	Arima 365 1	Arima 365 5
Variance faible	Arima 365 1	Farrington (sensibilité) Arima 365 5 (spécificité)

TABLE 3.4 – Tableau de choix

Nous émettrons quelques réserves sur ces résultats et discuterons de leur portée dans le chapitre suivant. Mais avant, nous allons voir comment il a été possible de mettre œuvre de manière pragmatique ce travail, et comment il a été possible d'implémenter un système de surveillance.

Chapitre 4

Automatisation des méthodes statistiques de surveillance

Dans cette partie nous allons voir comment il a été possible de mettre en place un système de surveillance. Nous devons ici revenir à la distinction que nous avons faite dans les premières pages de ce rapport, c'est-à-dire la distinction entre la surveillance spécifique et la surveillance syndromique ou non-spécifique. Nous développerons tout d'abord la partie syndromique et nous terminerons ensuite par la partie spécifique.

4.1 Mise en place d'un système de surveillance syndromique

Nous avons donc cherché à mettre en place - en nous basant sur les résultats de la phase de test - un système de surveillance syndromique. Nous avons à notre disposition 5 indicateurs : le nombre de primo-passages aux urgences, le nombre de primo-passages aux urgences pour les plus de 75 ans et pour les moins de 1 an, le nombre d'hospitalisations après admission aux services des urgences et les affaires SAMU. Les affaires SAMU n'ayant pas un comportement similaire aux autres indicateurs, et en particulier à l'indicateur primo-passage qui a été la base de notre étude, nous les avons traités à part.

Afin de pouvoir couvrir une majorité d'évènements sanitaires, nous avons décidé d'implémenter les trois méthodes suivantes :

- Arima 365 avec un risque de 1% et de 5%
- Farrington avec un risque de 1%
- C2-Medium avec un risque de 1/2% et 1%

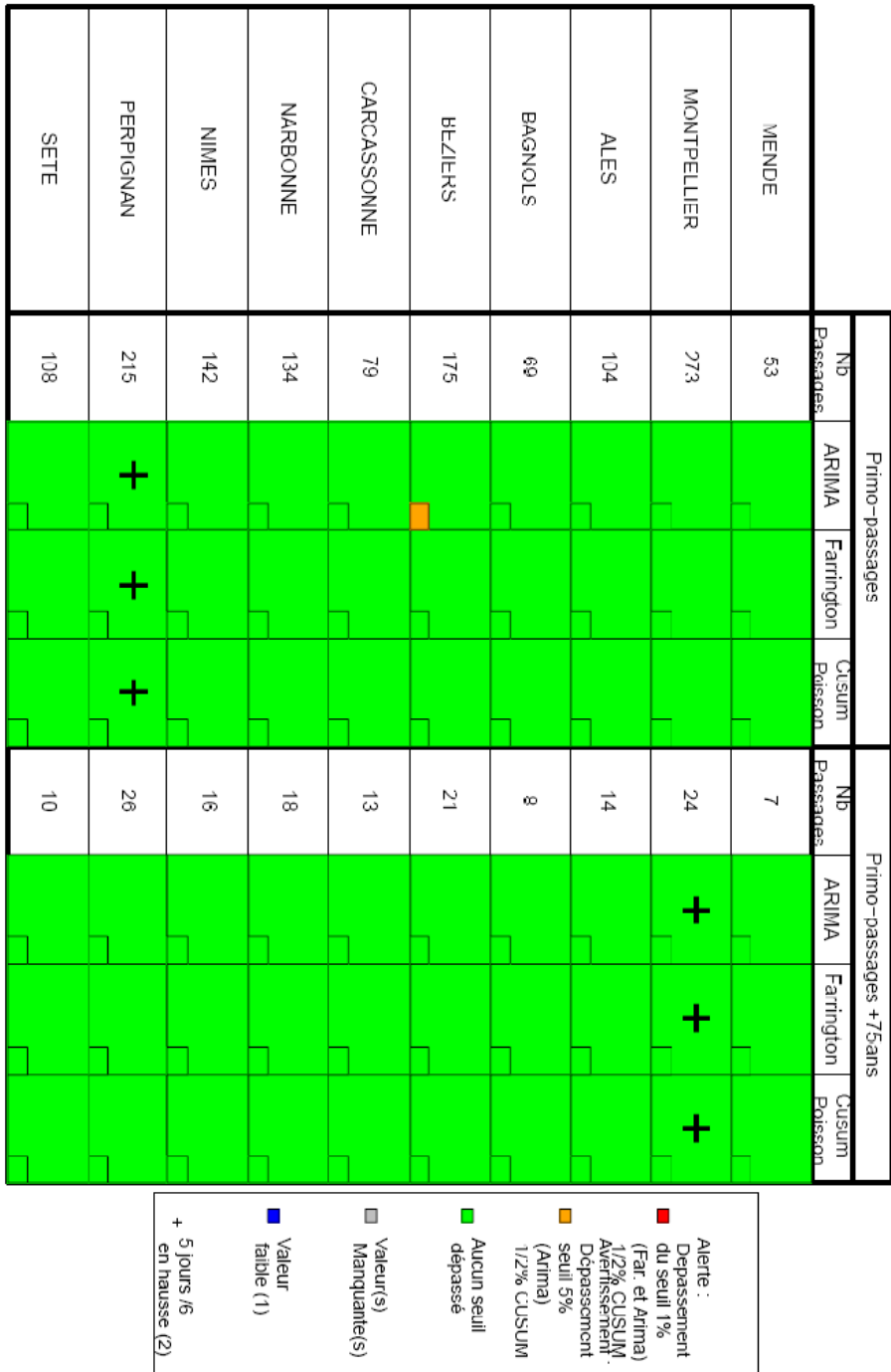
Pour la méthode Cusum, nous avons cependant préféré - au contraire de notre

étude - mettre une loi de poisson sous-jacente, et cela dans le but de pouvoir mieux appréhender les indicateurs ayant des effectifs très restreints. Pour nous conforter dans l'idée de l'efficacité d'un tel procédé, nous nous sommes référés à l'article [14].

Nous avons alors cherché à construire un outil statistique simple d'utilisation. L'idée retenue a été la suivante : un programme R, lancé à partir d'un simple double-clic d'un fichier.bat, qui fournit des graphiques résultats sous forme de pdf dans un dossier déterminé à l'avance. Ainsi, ce programme pourra être utilisé par n'importe qui, pour peu que les graphiques fournis soient compréhensibles.

L'idée a été de construire un tableau récapitulatif dont chaque case renverrait à un graphique complémentaire. Voici le-dit tableau :

Tableau de bord au 05/08/2010



(1) : Dépassement de la borne inférieure de l'intervalle de confiance à 99% de la méthode ARIMA : il se peut qu'il y ait une erreur dans la donnée
 (2) : 5 jours sur 6 en hausse par rapport au même jour de la semaine précédente

FIGURE 4.1 – Tableau de contrôle généré par notre outil de surveillance syndromique

Les colonnes représentent les indicateurs puis les méthodes pour chacun de ces indicateurs (les deux indicateurs manquants se trouvent dans un tableau parfaitement similaire). Quand une méthode a été implémentée avec deux risques différents, une seule colonne a été prévue. Les lignes représentent l'origine des séries. Une case peut ensuite revêtir 5 couleurs différentes :

- Grise : la donnée du jour est manquante
- Bleue : la donnée est peut-être trop faible (dépassement du seuil inférieur d'Arima 365 1%)
- Orange : Dépassement du seuil le plus bas pour la méthode considérée
- Rouge : Dépassement du seuil le plus haut pour la méthode considérée
- Verte : couleur par défaut (aucun problème à signaler)

Ensuite, un petit carré réside dans le coin inférieur droit de chaque case, et il représente l'état du système de la veille. Cela peut être utile pour voir si une alerte se confirme d'un jour à l'autre par exemple. Pour finir, un «+» s'affiche si parmi les 6 derniers jours 5 d'entre eux sont plus élevés que le même jour de la semaine précédente.

Nous avons voulu que ce tableau permette, en un seul regard, une appréciation générale de la situation sanitaire. En effet, cela permet de réduire l'information contenue dans 40 séries (10 hôpitaux * 4 indicateurs) en seulement deux tableaux. Ensuite, un examen plus précis est permis grâce aux graphiques annexes.

Ces graphiques comportent la série représentées sous forme de barres et au-dessus de laquelle figurent les différents seuils en ligne pointillée. Quelques informations complémentaires sont données dans le sous-titre. Et en-dessous de chacun des graphiques, nous avons fait figurer quelques graphiques complémentaires devant être envisagés comme une aide au diagnostic du modèle. Ainsi, pour la méthode Arima, nous avons représenté les fonctions d'autocorrélation et d'autocorrélation partielle; pour la méthode Cusum, nous avons représenté les résidus que nous espérons répartis uniformément autour de zéro ainsi qu'un histogramme de la série donnant une idée de la normalité ou du caractère poissonien des données; pour la méthode de Farrington, nous avons fait apparaître un diagramme quantile-quantile ainsi que le tracé des résidus.

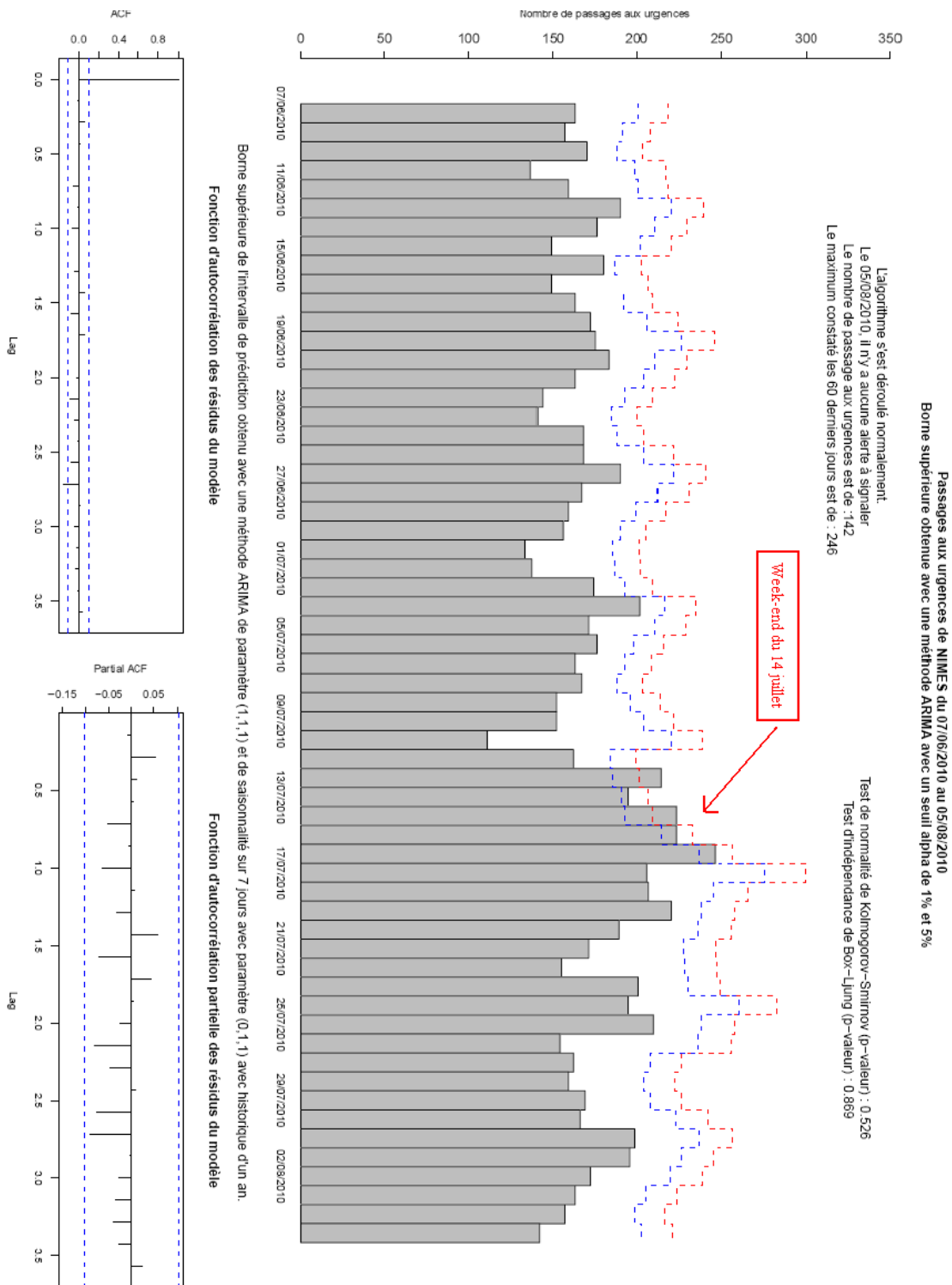


FIGURE 4.2 – Graphique Arima généré par l'outil syndromique

Passages aux urgences de MONTPELLIER du 07/06/2010 au 05/08/2010
 Borne supérieure obtenue grâce à l'algorithme de Farrington avec un seuil alpha de 1%

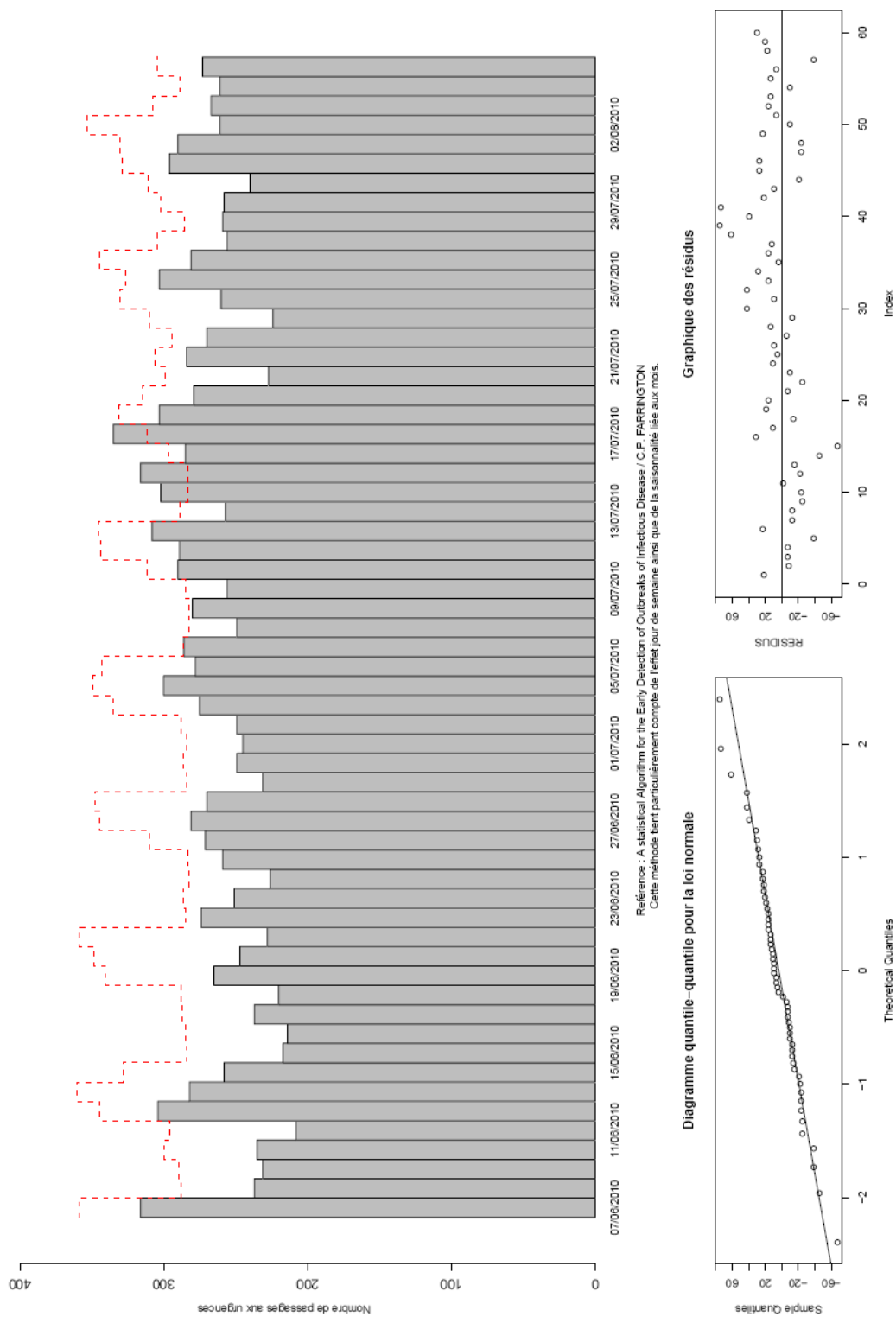


FIGURE 4.3 – Graphique Farrington généré par l'outil syndromique

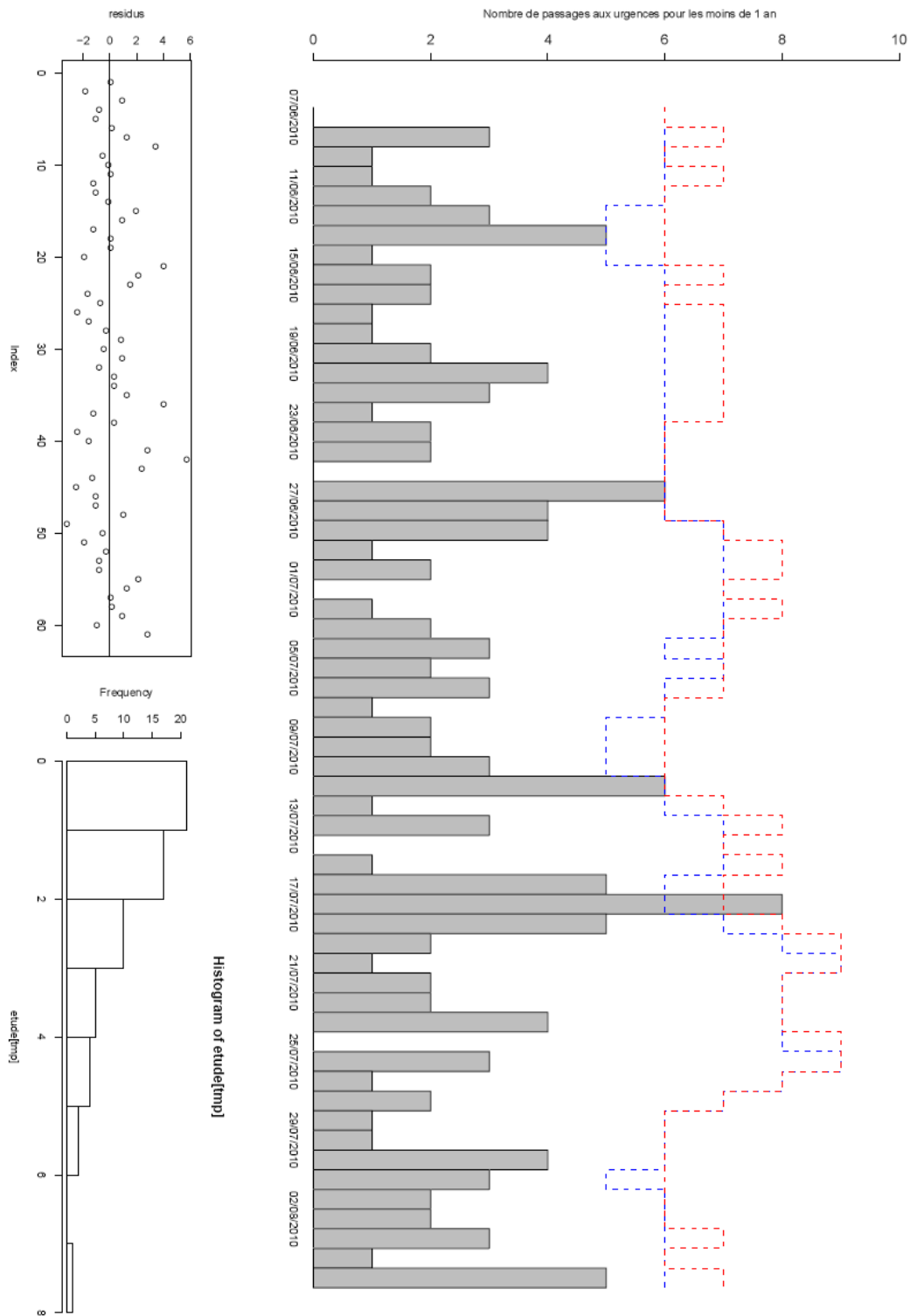


FIGURE 4.4 – Graphique Cusum généré par l’outil syndromique

Nous avons utilisé quotidiennement cet outil pendant un mois, afin de pouvoir en évaluer les capacités en situation concrète. Nous notons ici nos observations.

Il faut d'abord signaler qu'il n'y a pas eu, pendant la période concernée, d'évènement sanitaire notable. Néanmoins, le semaine du 14 juillet a été marqué par une augmentation conséquente du nombre de passage aux urgences. Et il est rassurant de voir que nos différentes méthodes ont détecté cette augmentation.

Les différentes alarmes que nous avons eu à traiter se répartissent en deux catégories : les fausses alarmes et les erreurs de saisie. En effet, grâce à ce système de surveillance, nous avons pu à plusieurs reprises corriger des valeurs erronées. Nous avons donc pu permettre, au-delà de notre objectif principal de surveillance, d'améliorer la qualité des données.

L'observation attentive des résidus pour la méthode de Farrington nous a en outre permis de déceler des «décalages» (comme dans la figure 3.3) que nous n'avions pas su remarquer. En effet, comme aucune observation n'est prise en compte dans l'année en cours pour la période de référence, un tel décalage va entraîner les résidus à se trouver toujours du même côté, positif ou négatif.

4.2 Mise en place d'un système de surveillance spécifique

La surveillance spécifique concerne dans notre cas trois différentes pathologies hivernales : les gastro-entérites, les bronchiolites et les gripes. La méthode utilisée est celle de Serfling. Le but de cet outil sera donc de pouvoir déclarer les périodes épidémiques. Mais un premier constat nous conduit à construire cet outil de manière moins rigide que les deux précédents. En effet, la notion même de période épidémique n'étant pas arrêtée, il nous fallait pouvoir construire un seuil répondant à toutes les possibles définitions. La question est de savoir si la période épidémique correspond à la période couvrant des premiers jusqu'aux derniers balbutiements du pic ou si au contraire, nous préférons considérer comme épidémiques les quelques semaines de très forte activité virale. Comme nous l'avons déjà dit, à chacun de ces objectifs doit correspondre un seuil adapté.

Dans ce cadre, nous avons donc choisi de faire un programme interactif, grâce au package Tcl/Tk de R (voir [2]). Nous avons scrupuleusement suivi les étapes indiquées dans le chapitre 2. Nous nous contentons de les rappeler et d'expliquer l'influence des différents choix sur le seuil final.

La première étape consiste à écrêter les données brutes selon un quantile. C'est

principalement ce choix qui va influencer la hauteur du seuil ; en effet, plus nous choisirons un quantile bas plus notre seuil sera bas. Alors qu'un écrêtement grâce au quantile à 95% sert principalement à se prémunir contre les potentielles valeurs aberrantes, choisir d'écrêter selon un quantile à 30-40% permet dans la plupart des cas de «couper» une partie conséquente des pics épidémiques sans pour autant affecter les données des périodes sans épidémie. Dans ce dernier cas, nous parviendrons donc à détecter les épidémies dès leur balbutiements.

Ensuite, nous suivons la deuxième étape qui consiste à écrêter après avoir posé un modèle de régression linéaire. Le minimum entre la borne supérieure de l'intervalle de confiance à 1% et les données de la série est ensuite pris. Nous laissons à l'utilisateur les choix suivants (pour plus de détails voir 2.2.4) :

- Composante annuelle
- Composante annuelle + Semi-annuelle
- Toutes les composantes
- Choix automatique

Si nous n'avons aucune contrainte à respecter, le choix le plus judicieux semble être «Choix automatique». En effet, un choix de modèle par critère AIC choisira parmi l'ensemble des régresseurs ceux qui permettent d'expliquer au mieux le modèle. Néanmoins, si par des contraintes extérieures - de communication par exemple - il est demandé d'avoir un seuil «simple» à appréhender, il peut être judicieux de choisir, selon la série, un des deux premiers choix. Nous donnons quelques graphiques ci-dessous pour illustration.

Nous avons également pris soin de tracer en rouge en dessous de tous nos graphiques les périodes épidémiques que nous avons défini comme étant deux semaines de suite au-dessus du seuil.

Finalement, nous proposons aussi de décaler le seuil afin de pouvoir l'ajuster aux dernières semaines. Cet ajustement se fait en minimisant la somme des résidus au carré.

Une sortie pdf contenant le graphique résultat final ainsi qu'un zoom sur l'année en cours est fournie ensuite.

Les résultats sont présentés ci-dessous.

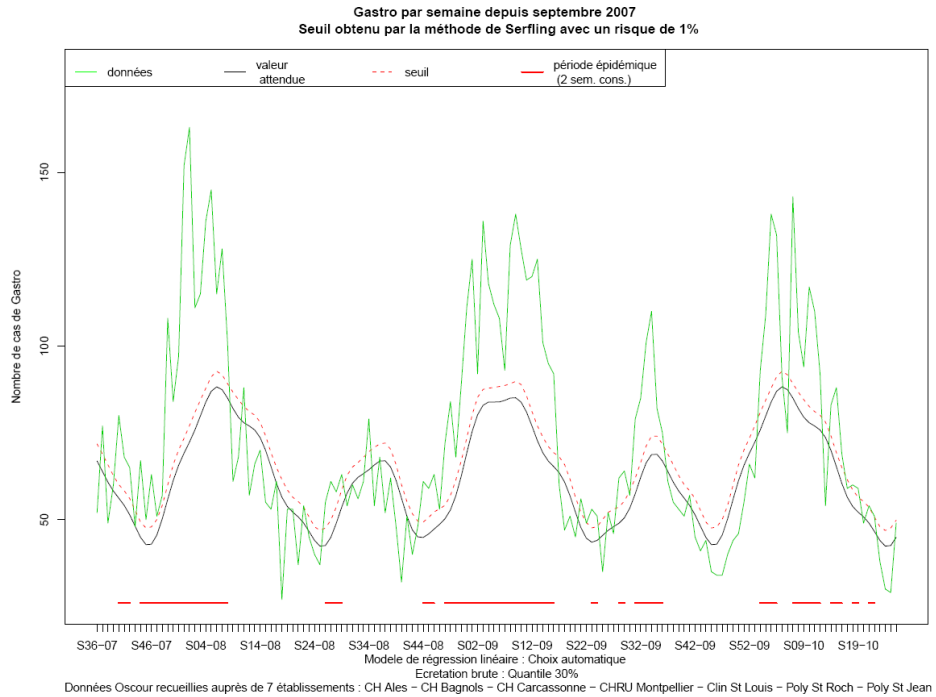


FIGURE 4.5 – Seuil en choisissant le choix automatique...

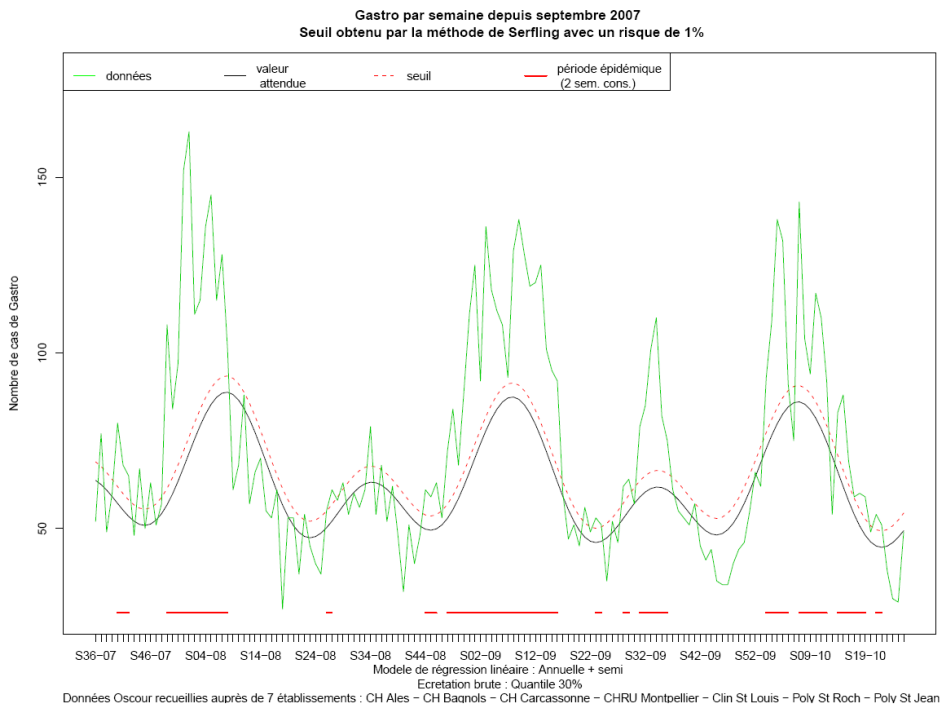


FIGURE 4.6 – ... ou le choix «annuel+semi» !

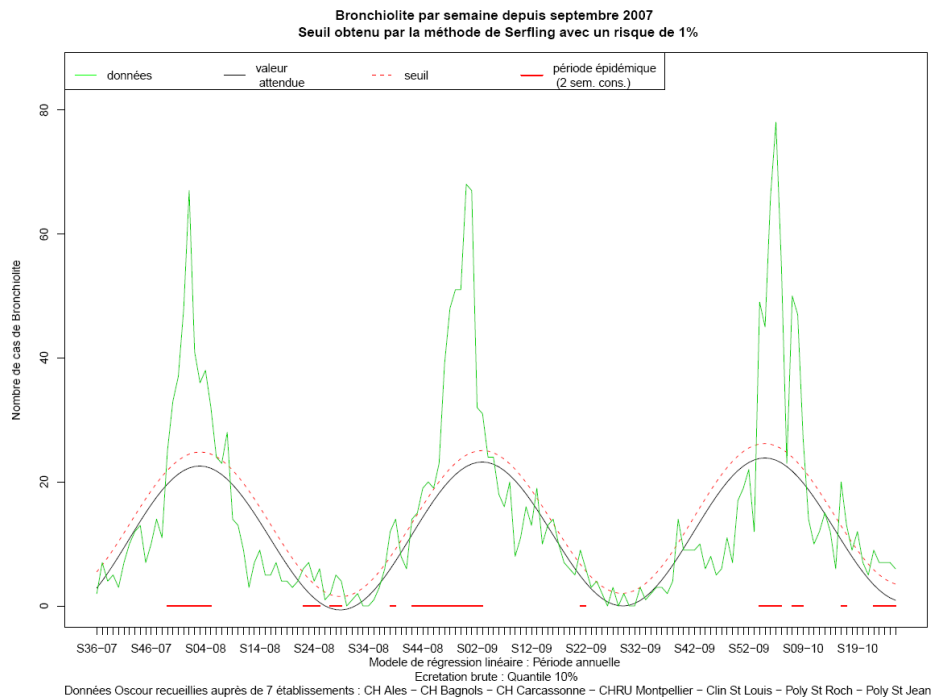


FIGURE 4.7 – Seuil sans faire de décalage...

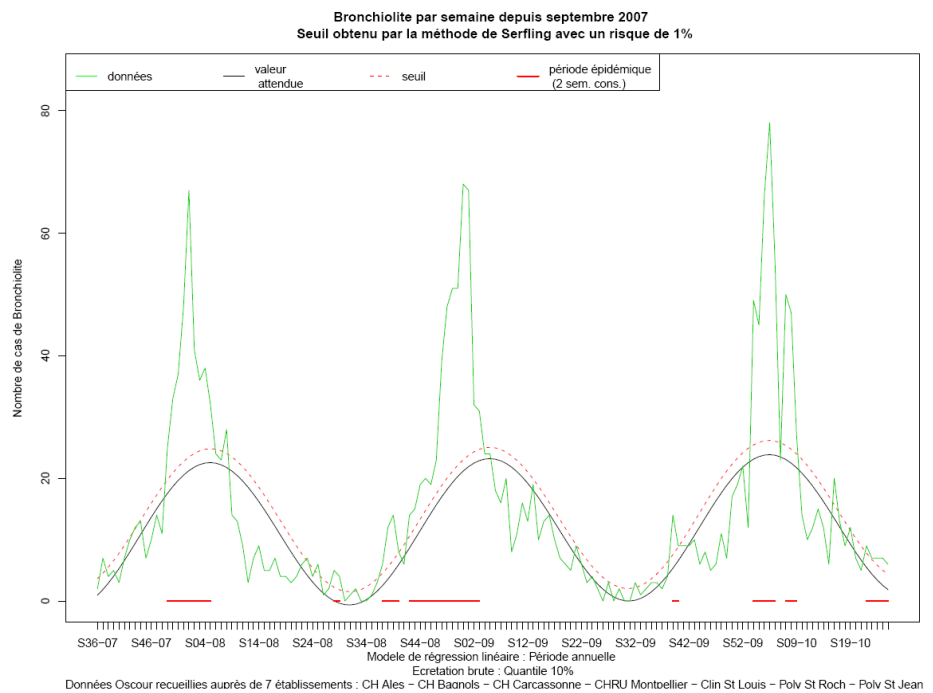


FIGURE 4.8 – ... ou avec un ajustement sur les 25 dernières semaines!

Chapitre 5

Discussions

Dans ce chapitre nous allons tirer un premier bilan de nos résultats et nous allons les discuter. Nous essayerons de dégager les objectifs que nous sommes parvenus à atteindre ainsi que les limites de notre exposé. Par ailleurs, nous comparerons également nos résultats avec ceux trouvés dans divers articles scientifiques. Et pour finir, nous mettrons en avant les différentes perspectives d'amélioration possibles.

5.1 Rappel des résultats

Le premier objectif était de mettre en place une étude scientifique pour tester l'efficacité de méthodes statistiques choisies parmi celles le plus souvent citées dans la littérature scientifique.

Les résultats de cette étude montrent une supériorité générale des méthodes Arima et Farrington sur les méthodes Cusum. Un tableau de choix a été donné dans la partie résultat en prenant les critères qui nous semblaient les plus pertinents : puissance du signal et indicateur de variabilité de la série. Néanmoins, l'ajout d'autres critères auraient peut-être permis de modifier les résultats obtenus ou alors d'établir une table plus précise.

Lors de cette étude, il a également été montré que certaines méthodes étaient plus performantes en terme de sensibilité (C3-extra, Farrington) tandis que d'autres étaient plus à même de répondre à des objectifs de spécificité (Arima, C1-mild, C2-medium). Aussi, il apparait clairement que chaque méthode répond à des objectifs précis et qu'il n'est pas possible de dégager une seule méthode permettant de répondre à tous les objectifs de surveillance. Ainsi, si plusieurs objectifs différents sont demandés, il sera sans doute nécessaire de mettre en place plusieurs méthodes.

5.2 Comparaison avec les études déjà publiées

L'objectif de cette partie est de mettre en parallèle les résultats que nous avons obtenus et ceux qui ont pu être publiés. Nous ne prétendons pas faire un travail exhaustif, mais chercherons à tirer quelques enseignements de nos comparaisons. Toutefois, il faut prendre avec précaution ces comparaisons, étant donné que les séries étudiées d'une étude à l'autre sont différentes et que les signaux utilisés dans les phases de simulation diffèrent dans la plupart des cas.

L'avantage des méthodes Cusum que nous avons étudiées est de pouvoir s'appliquer à des données avec un historique restreint. Nous rappelons que nous avons à la base trois méthodes : C1-mild, C2-medium, et C3-extra tirés de l'article de Hutwagner [7]. Puisque le fait de limiter la variance n'a pas été traité dans les articles que nous avons pu lire, nous n'en parlerons pas. En nous référant à l'article historique de Hutwagner [8], nous constatons que nous obtenons des résultats relativement proches pour les deux premières méthodes C1 et C2. En effet nous avons une spécificité assez haute et relativement constante, et une sensibilité variable en fonction des signaux. En revanche, la méthode C3 est dans notre cas beaucoup plus sensible et beaucoup moins spécifique que pour C1 et C2 ; cette différence importante n'apparaît pas dans Hutwagner [8]. Nous avons peut-être mal choisi les paramètres de notre méthode. Ce sentiment est confirmé en regardant cet autre article du même auteur [6]. L'impression est la même dans l'article de Jackson [9].

Par ailleurs, dans ce dernier article, la comparaison est faite entre les méthodes Cusum énoncées ci-dessus, et un modèle linéaire généralisé. Or, nous savons que la méthode de Farrington est une amélioration d'un tel modèle. Avec précaution, nous pouvons donc essayer de mettre en relation les résultats établis dans cet article et les nôtres. Les résultats obtenus montrent, comme dans notre cas, une supériorité quasi-systématique, du modèle linéaire généralisé en terme de sensibilité.

Nous n'avons hélas pu trouver d'étude comparative pour les méthodes Arima. Mais nos performances pour ces méthodes se rapprochent de celles affichées dans la publication de B.Y.Reis [13]. Malgré les différences de contexte, une des remarques formulées par les auteurs correspond tout à fait à ce que nous avons observé dans notre étude :

«One of the primary benefits of ARIMA models is their ability to correct for local trends in the data - what has happened on the previous day is incorporated into the forecast of what will happen today.»¹

Mais il faut faire attention : si cette capacité à intégrer les augmentations per-

1. Traduction : Un des intérêts majeurs des modèles Arima est leur capacité à intégrer des tendances locales : en effet, ce qui s'est produit le jour précédent est pris en compte le jour suivant.

5.3. LIMITES DE NOTRE ÉTUDE DE SIMULATION ET DES MÉTHODES STATISTIQUES UTILISÉES

met de ne pas déclencher de fausses alarmes en cas de tendances locales, il sera d'autant moins possible de détecter un événement sanitaire s'il s'étend lentement. Ce qui nous conduit à renforcer la remarque que nous avons faite dans le paragraphe 5.5.

Dans notre travail nous nous sommes cantonnés à surveiller l'évolution des séries temporelles. Mais il est possible d'aller plus loin, et de rendre ce travail encore plus intéressant. L'article écrit par A.Earnest [3] montre comment il a été rendu possible, grâce à un modèle Arima, de prédire rétrospectivement le nombre de lit occupés pendant une épidémie de Sras à Singapour. Malgré les différences de contexte, il semble tout à fait envisageable de chercher à prédire - en particulier pendant des périodes épidémiques - le nombre d'admission aux urgences. Ce qui pourrait permettre aux hôpitaux une meilleure prise en charge des patients et une meilleure adaptation. Cela nous conduit aux perspectives sur lesquelles nous aurions pu nous pencher.

5.3 Limites de notre étude de simulation et des méthodes statistiques utilisées

Dans la phase de simulation, nous avons défini a priori les signaux en fonction de ce que nous voulions détecter. Mais il est peut-être regrettable de n'avoir pas choisi des signaux de même amplitude temporelle, ce qui nous aurait permis, comme dans [4] par exemple, d'établir une courbe relatant l'évolution de la sensibilité en fonction de la puissance de signal. Néanmoins il nous semble légitime d'avoir considéré qu'un signal de faible puissance pouvait être détecté avec un délai plus long. Et puis, notre but étant la comparaison de plusieurs méthodes et non pas l'étude précise d'une d'entre elle, ce choix n'avait pas une importance quand au résultat final ; des signaux de même amplitude auraient simplement permis de compléter nos résultats en fournissant une étude de chacune des méthodes séparément, ce qui n'était pas notre objectif.

Le choix des signaux a été réalisé a priori en fonction des objectifs du système de surveillance. Les 4 premiers correspondaient à une augmentation brutale et constante tandis que les quatre autres correspondaient à une augmentation progressive plus ou moins importante. Bien entendu, ces signaux ne recouvrent qu'une faible partie des signaux que l'on aurait voulu pouvoir détecter. Par ailleurs, il aurait été intéressant d'étudier le comportement de nos méthodes face à deux signaux très rapprochés dans le temps, et ainsi évaluer l'influence du premier signal sur les performances des méthodes à détecter le second signal.

Dans l'interprétation des résultats, nous avons également été limité. En effet,

dans le tableau final 3.1 nous aurions aimé mettre tous les «premiers ex-æquo» au sens statistique, c'est-à-dire donner tous ceux qui ne sont pas significativement différents du premier. Pour le premier, une comparaison multiple de Tuckey donnait le résultat donné dans le tableau, mais dans les autres cas, nous ne sommes pas parvenus à trouver de test statistique, du fait de problèmes d'indépendance.

Nous avons aussi pu constater quelques limites à nos méthodes. L'historique très court des cartes Cusum est à la fois un avantage lorsque nous disposons d'un historique très restreint (nous savons que l'historique nécessaire pour Farrington est de trois ans et que le fait de prendre un historique plus court pour la méthode Arima a diminué sa performance dans notre test) mais un inconvénient dans le cas où nous disposons d'un historique plus important parce que l'information non-utilisée empêche de prendre en compte les tendances et les saisonnalités.

Une des principales limites de la méthode de Farrington est que la période de référence considérée ne prend aucune observation de l'année courante en compte. Ainsi, si un changement brutale ou une cassure intervient dans les données, cette méthode prédira inlassablement des valeurs erronées. D'un autre côté, cette particularité nous a permis, grâce à l'observation des résidus, de déceler des anomalies dans les données.

La limite de la méthode Arima est également son intérêt. En effet, comme nous l'avons souligné dans le paragraphe précédent, cette méthode parvient à prendre en compte très rapidement les tendances des données. Néanmoins, le risque est de considérer un signal dont la puissance augmente lentement comme une tendance, et qui ne sera pas, de ce fait, signalé par la méthode.

L'ouvrage [10] donne une ébauche intéressante du panorama des méthodes statistiques pour la surveillance syndromique. Une des idées maîtresses de ce livre est par ailleurs d'associer à la surveillance temporelle une surveillance spatiale. Ceci est particulièrement intéressant dans la mesure où il est entièrement naturel de penser que plus deux centres de soins sont géographiquement proches, plus leurs comportements seront similaires.

Ceci nous amène également à aborder le sujet d'une surveillance multivariée. Dans le paragraphe 5.5, nous projetions de créer une alarme statistique à partir de plusieurs systèmes de surveillance univariés. Mais on se rend compte immédiatement du problème : chaque alarme est défini à un risque α fixé à l'avance, mais nous ne connaissons pas le risque α global que nous prenons. Dans le livre que nous citons plus haut, deux pistes majeures sont explorées : la réduction de données multivariées à des données univariées et une surveillance rigoureuse et parallèle de plusieurs systèmes de surveillance univariés. Un outil qui apparaît être relativement simple d'utilisation et qui a été testé dans un cadre pratique par Griffin [5]

est les cartes de contrôle multivariés MCusum (Multivariate Cusum) ou MEwma (Multivariate Ewma).

Comme pour un même centre hospitalier les quatre indicateurs que nous avons étudiés sont fortement corrélés, il aurait été peut-être intéressant d'envisager la surveillance grâce à des techniques bayésiennes. En effet, en regardant séparément chaque indicateur, l'information apportée par la corrélation est perdue.

5.4 Avantages et limites du modèle de Serfling

Le modèle de Serfling s'est largement généralisé pour la surveillance spécifique en France. Il faut toutefois préciser qu'il n'est pas possible d'étendre l'utilisation de cette méthode pour la surveillance syndromique. En effet, ce modèle ne peut suivre que des épidémies parfaitement connues et attendues.

Le modèle de Serfling est construit à partir de données de périodes non-épidémiques, c'est pourquoi il faut écrêter les pics. On voit là apparaître deux limites. La première est qu'il n'existe pas de définition universelle pour définir les périodes épidémiques, comme nous l'avons déjà vu plus haut : ce choix se fait a priori en fonction des critères fixés par l'épidémiologiste. Le seuil est donc lié à la définition de la «période épidémique». Ensuite, la manière d'écrêter les données est tout à fait «arbitraire». Et comme cet écrêtement a une influence conséquente sur le seuil final, il est légitime de se demander si nous ne pourrions pas améliorer la qualité du seuil en changeant la manière d'écrêter. La question se pose alors de l'utilité d'un tel seuil. Ne serait-il pas possible de suivre tout simplement l'évolution des épidémies, c'est-à-dire en modélisant celles-ci le plus finement possible sans modifier les données ?

Par ailleurs, le modèle initial proposé par Serfling [15] ne prend pas en considération un possible décalage temporel dans l'apparition d'une épidémie attendue. Par exemple, l'apparition de la grippe hivernale peut différer de quelques semaines d'une année à la suivante. C'est pourquoi nous avons proposé de donner la possibilité à l'utilisateur d'ajuster horizontalement le seuil aux dernières semaines en minimisant les résidus au carré sur cette période. En regardant les résultats (un exemple est donné par les graphiques 4.7 et 4.8), l'intérêt de cette manipulation est clair : on ajuste le pic prévu avec le pic observé.

Enfin, pour pouvoir appliquer cette méthode, il faut au moins 2-3 ans d'historiques réguliers. La taille de l'historique est nécessaire à la validité de la régression linéaire que nous effectuons. La régularité des données est nécessaire pour la justesse du seuil produit. Par exemple, en considérant le graphique 3.2, on comprend pourquoi il a été impossible d'appliquer cette méthode pour la grippe. En effet,

l'épidémie de grippe A a considérablement modifié les caractéristiques du pic de grippe hivernal : décalage temporel important, puissance plus importante...

Finalement, un des apports principaux de cette méthode est qu'elle permet une communication aisée. En effet, en choisissant de n'inclure qu'un ou deux paramètres saisonniers, le seuil final obtenu est très régulier et aisément explicable à des personnes non-formées aux statistiques. Il suffit par exemple de mettre en commentaire : «Seuil issu d'une régression linéaire incluant un effet saisonnier annuel ainsi que l'effet été/hiver».

5.5 Automatisation : apports et limites

Notre second objectif était d'implémenter en routine des méthodes de surveillance à partir des résultats de la partie précédente. Nous avons donc, dans ce but, construit deux outils statistiques, un premier pour la surveillance syndromique et un second pour la surveillance spécifique. Nous allons développer un paragraphe pour chacun d'entre eux.

Tout d'abord quelques mots à propos de l'outil de surveillance syndromique. Nous avons eu la possibilité pendant un mois environ de mettre en place quotidiennement cet outil et de l'évaluer sur les données SRVA que la Cire reçoit chaque jour. Le constat est que, malgré quelques fausses alarmes, notre système de surveillance a participé de manière active à l'amélioration de la qualité des données. En effet, à plusieurs reprises nous avons remarqué que des valeurs signalées trop fortes ou trop faibles, étaient simplement des erreurs de saisie que nous avons pu ensuite corriger. D'autre part, le 14 juillet, jour férié en pleine semaine (se comportant donc «comme un dimanche») a fait «sonner» beaucoup d'alarmes. Ceci est un bon signe, parce que ce jour-là l'activité était anormale pour un jour de semaine. Mais ceci apporte une nouvelle limite à notre travail : la prise en compte dans le modèle des jours fériés. C'est un problème d'autant plus épineux que les jours fériés varient d'une année à l'autre ; notre outil ne prenant pas en compte les jours fériés qui ne tombent pas dans un week-end, il faut demeurer vigilant et critique lors de l'analyse.

Nous avons également remarqué que les méthodes ne déclenchaient pas systématiquement des alarmes dans le même temps ; cela nous conduit à penser qu'il y a un intérêt à surveiller nos données avec plusieurs méthodes. En effet, cela permet de diversifier les types de signaux que nous pouvons détecter. Par ailleurs, une bonne connaissance des méthodes permet souvent de comprendre pourquoi une alarme statistique a été déclenchée par telle ou telle méthode. Cela peut aider à juger la pertinence de l'alerte.

Le fait de surveiller plusieurs centres de soins et plusieurs indicateurs peut permettre d'augmenter la spécificité de manière artificielle. En effet, on peut penser que si un évènement sanitaire important apparaît, nous n'aurions pas qu'un indicateur et qu'un centre de soin affecté. De ce fait, nous pouvons poser une règle intuitive stipulant qu'une alerte statistique est donné lorsque nous dépassons des seuils pour n centres ou indicateurs.

Nous n'avons malheureusement pas pu étudier de manière prospective l'outil de surveillance spécifique. En effet, cet outil est principalement destiné à des pathologies hivernales. Néanmoins, comme le montrent les graphiques présentés dans 4.2, le choix des options proposées un effet réel et déterminant sur le résultat final. L'objectif de pouvoir construire un seuil adaptable en fonction de la définition choisie de période épidémique semble donc atteint.

L'apport de nos outils de surveillance est essentiellement de pouvoir apporter une vision rapide de la situation sanitaire. Par ailleurs, la simplicité de ceux-ci permet leur utilisation par des personnes non formées aux statistiques. Néanmoins, pour pouvoir donner une interprétation juste et précise, une connaissance du fonctionnement des méthodes statistique est nécessaire.

Chapitre 6

Conclusion

Globalement, nous sommes parvenus à remplir tous les objectifs que nous nous étions fixés, malgré les limites que nous avons évoquées ci-dessus.

L'évaluation pragmatique de nos méthodes de surveillance syndromique nous a conduit à penser qu'il n'existe pas de méthode universelle permettant la surveillance des données issues des urgences. Ce constat nous a conduit à mettre plusieurs méthodes en parallèle. Il suffit alors de connaître le fonctionnement des méthodes pour que les informations données par ces différentes méthodes deviennent complémentaires.

Pour la surveillance spécifique, nous nous sommes contentés de la méthode de Serfling qui semble être la référence pour ce genre de données en France. Néanmoins, notre idée d'ajustement du seuil aux dernières semaines semble pouvoir tout son intérêt dans la surveillance prospective que malheureusement nous n'avons pas pu tester.

Rendre les outils créés utilisables par des personnes non-formées aux statistiques était pour nous un enjeu majeur. Pour cela nous nous sommes attachés à rendre nos outils intuitifs grâce à des tableaux pour la partie syndromique et grâce à l'interactivité pour la partie spécifique.

D'un point de vue personnel ce stage m'a permis de mettre en application les connaissances théoriques accumulées dans l'année ainsi que de perfectionner mes compétences dans le logiciel R. De plus, j'ai pu être confronté à de réels enjeux statistiques et découvrir par la même occasion l'univers de la surveillance de la santé publique.

Annexe A

Données et analyses

A.1 Données Oscour

Semaine	Nombre pour BRONCHIOLITE (Diag SAU)
S36-07	2
S37-07	7
S38-07	4
S39-07	5
S40-07	3
S41-07	7
S42-07	10

A.2 Données Srva

Données issues de l'hôpital de Montpellier :

date	urg	urg75a	urg1a	hospit	affaires
22/07/2005	226	15	12	52	710
23/07/2005	260	22	20	71	1036
24/07/2005	260	15	9	49	1245
25/07/2005	241	18	12	62	647
26/07/2005	255	26	9	69	659
27/07/2005	247	19	10	59	669
28/07/2005	238	20	13	59	688
29/07/2005	235	21	8	62	667
30/07/2005	244	15	12	51	1001
31/07/2005	273	17	8	64	1037

A.3 Analyse de la variance

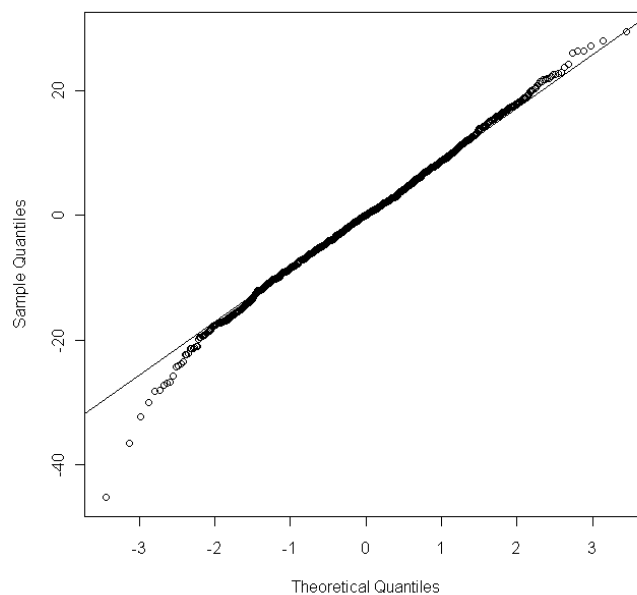


FIGURE A.1 – Diagramme quantile-quantile des résidus du modèle

Bartlett test of homogeneity of variances

data: adm by jour by annee by mois
 Bartlett's K-squared = 4.3699, df = 6, p-value = 0.6268

Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = adm ~ jour + annee + mois, data = donnees)

\$jour

	diff	lwr	upr	p adj
jeudi-dimanche	-12.86840680	-15.2378413	-10.4989723	0.0000000
lundi-dimanche	-6.09761905	-8.4646971	-3.7305410	0.0000000
mardi-dimanche	-12.93333333	-15.3004114	-10.5662553	0.0000000
mercredi-dimanche	-14.10545859	-16.4748931	-11.7360241	0.0000000
samedi-dimanche	-3.29722222	-5.6643003	-0.9301442	0.0008108
vendredi-dimanche	-11.00846561	-13.3755436	-8.6413876	0.0000000
lundi-jeudi	6.77078775	4.4013532	9.1402223	0.0000000

mardi-jeudi	-0.06492654	-2.4343611	2.3045080	1.0000000
mercredi-jeudi	-1.23705179	-3.6088404	1.1347369	0.7207778
samedi-jeudi	9.57118457	7.2017501	11.9406191	0.0000000
vendredi-jeudi	1.85994119	-0.5094933	4.2293757	0.2362459
mardi-lundi	-6.83571429	-9.2027923	-4.4686362	0.0000000
mercredi-lundi	-8.00783954	-10.3772741	-5.6384050	0.0000000
samedi-lundi	2.80039683	0.4333188	5.1674749	0.0088781
vendredi-lundi	-4.91084656	-7.2779246	-2.5437685	0.0000000
mercredi-mardi	-1.17212526	-3.5415598	1.1973093	0.7684211
samedi-mardi	9.63611111	7.2690331	12.0031892	0.0000000
vendredi-mardi	1.92486772	-0.4422103	4.2919458	0.1988843
samedi-mercredi	10.80823637	8.4388019	13.1776709	0.0000000
vendredi-mercredi	3.09699298	0.7275585	5.4664275	0.0022710
vendredi-samedi	-7.71124339	-10.0783214	-5.3441653	0.0000000

\$mois

	diff	lwr	upr	p adj
avril-août	-5.30010367	-8.67399483	-1.9262125	0.0000196
décembre-août	-6.63556881	-9.98169085	-3.2894468	0.0000000
février-août	-7.88883932	-11.31701506	-4.4606636	0.0000000
janvier-août	-6.87048805	-10.21661008	-3.5243660	0.0000000
juillet-août	2.84505999	-0.67309535	6.3632153	0.2546580
juin-août	-2.41018136	-6.00046350	1.1801008	0.5521307
mai-août	-2.00863432	-5.38827427	1.3710056	0.7305451
mars-août	-6.81143810	-10.15756014	-3.4653161	0.0000000
novembre-août	-8.00928522	-11.38317638	-4.6353941	0.0000000
octobre-août	-4.62277019	-7.96889223	-1.2766482	0.0004071
septembre-août	-6.25464573	-9.62853689	-2.8807546	0.0000001
décembre-avril	-1.33546514	-4.70935630	2.0384260	0.9797851
février-avril	-2.58873565	-6.04402114	0.8665498	0.3716127
janvier-avril	-1.57038438	-4.94427554	1.8035068	0.9343088
juillet-avril	8.14516366	4.60058670	11.6897406	0.0000000
juin-avril	2.88992231	-0.72625450	6.5060991	0.2716988
mai-avril	3.29146935	-0.11566655	6.6986053	0.0696772
mars-avril	-1.51133443	-4.88522559	1.8625567	0.9495128
novembre-avril	-2.70918155	-6.11061514	0.6922520	0.2766256
octobre-avril	0.67733348	-2.69655768	4.0512246	0.9999597
septembre-avril	-0.95454206	-4.35597565	2.4468915	0.9989500
février-décembre	-1.25327051	-4.68144625	2.1749052	0.9892897
janvier-décembre	-0.23491924	-3.58104127	3.1112028	1.0000000
juillet-décembre	9.48062880	5.96247346	12.9987841	0.0000000
juin-décembre	4.22538745	0.63510531	7.8156696	0.0067857
mai-décembre	4.62693449	1.24729454	8.0065744	0.0004902
mars-décembre	-0.17586929	-3.52199133	3.1702527	1.0000000
novembre-décembre	-1.37371641	-4.74760757	2.0001747	0.9748805
octobre-décembre	2.01279862	-1.33332342	5.3589207	0.7149263

septembre-décembre	0.38092308	-2.99296808	3.7548142	0.9999999
janvier-février	1.01835127	-2.40982447	4.4465270	0.9982293
juillet-février	10.73389931	7.13761328	14.3301853	0.0000000
juin-février	5.47865796	1.81178162	9.1455343	0.0000704
mai-février	5.88020501	2.41930592	9.3411041	0.0000020
mars-février	1.07740122	-2.35077452	4.5055770	0.9970616
novembre-février	-0.12044590	-3.57573139	3.3348396	1.0000000
octobre-février	3.26606913	-0.16210661	6.6942449	0.0789969
septembre-février	1.63419359	-1.82109190	5.0894791	0.9269078
juillet-janvier	9.71554804	6.19739270	13.2337034	0.0000000
juin-janvier	4.46030669	0.87002455	8.0505888	0.0029170
mai-janvier	4.86185373	1.48221378	8.2414937	0.0001703
mars-janvier	0.05904995	-3.28707209	3.4051720	1.0000000
novembre-janvier	-1.13879717	-4.51268833	2.2350940	0.9945197
octobre-janvier	2.24771786	-1.09840418	5.5938399	0.5510948
septembre-janvier	0.61584231	-2.75804884	3.9897335	0.9999847
juin-juillet	-5.25524135	-9.00637587	-1.5041068	0.0003037
mai-juillet	-4.85369431	-8.40374368	-1.3036449	0.0005037
mars-juillet	-9.65649809	-13.17465343	-6.1383428	0.0000000
novembre-juillet	-10.85434521	-14.39892218	-7.3097682	0.0000000
octobre-juillet	-7.46783018	-10.98598552	-3.9496748	0.0000000
septembre-juillet	-9.09970573	-12.64428269	-5.5551288	0.0000000
mai-juin	0.40154704	-3.21999398	4.0230881	0.9999999
mars-juin	-4.40125674	-7.99153888	-0.8109746	0.0036249
novembre-juin	-5.59910386	-9.21528067	-1.9829270	0.0000288
octobre-juin	-2.21258883	-5.80287097	1.3776933	0.6820026
septembre-juin	-3.84446437	-7.46064118	-0.2282876	0.0257751
mars-mai	-4.80280379	-8.18244374	-1.4231638	0.0002233
novembre-mai	-6.00065090	-9.40778681	-2.5935150	0.0000006
octobre-mai	-2.61413587	-5.99377583	0.7655041	0.3209572
septembre-mai	-4.24601142	-7.65314733	-0.8388755	0.0027698
novembre-mars	-1.19784712	-4.57173828	2.1760440	0.9915932
octobre-mars	2.18866791	-1.15745412	5.5347899	0.5934278
septembre-mars	0.55679237	-2.81709879	3.9306835	0.9999946
octobre-novembre	3.38651503	0.01262387	6.7604062	0.0481529
septembre-novembre	1.75463949	-1.64679410	5.1560731	0.8737965
septembre-octobre	-1.63187554	-5.00576670	1.7420156	0.9153735

Annexe B

Résultats des tests

B.1 Tableaux de résultat

	<i>Spécificité</i>	<i>Sensibilité</i>	<i>Moyenne</i>	<i>Ecart-type</i>
C1	98,63	97,25	1	0
C1 aj	98,35	98,35	1	0
C2	98,08	97,8	1	0
C2 aj	98,08	98,9	1	0
C3	80,49	98,9	1	0
C3 aj	79,95	99,18	1	0
Arima 360 5%	95,88	100	1	0
Arima 360 1%	99,73	99,45	1	0
Arima 100 5%	93,41	99,73	1	0

FIGURE B.1 – Résultat pic 1

	<i>Spécificité</i>	<i>Sensibilité</i>	<i>Moyenne</i>	<i>Ecart-type</i>
C1	95,88	79,95	1,154639175	0,43959714
C1 aj	95,05	89,29	1,233846154	0,533529166
C2	94,78	81,32	1,195945946	0,496206406
C2 aj	93,96	91,76	1,311377246	0,623890647
C3	65,93	99,45	1,20718232	0,486554185
C3 aj	65,38	100	1,175824176	0,435185445
Arima 360 5%	88,19	99,18	1,060941828	0,282150865
Arima 360 1%	99,18	95,88	1,128939828	0,419351263
Arima 100 5%	81,04	99,45	1,05801105	0,2774127

FIGURE B.2 – Résultat pic 2

	<i>Spécificité</i>	<i>Sensibilité</i>	<i>Moyenne</i>	<i>Ecart-type</i>
C1	91,76	40,66	1,891891892	1,330618674
C1 aj	90,11	45,33	2,042424242	1,432852839
C2	90,38	46,98	2,245614035	1,563713925
C2 aj	88,74	52,75	2,395833333	1,614587732
C3	46,43	89,01	2,101851852	1,380580585
C3 aj	46,15	91,21	2,075301205	1,345370792
Arima 360 5%	78,57	86,54	2,066666667	1,411508657
Arima 360 1%	98,35	48,08	2,485714286	1,604079364
Arima 100 5%	67,31	88,19	2,109034268	1,482379633

FIGURE B.3 – Résultat pic 3

	<i>Spécificité</i>	<i>Sensibilité</i>	<i>Moyenne</i>	<i>Ecart-type</i>
C1	87,64	26,65	3,402061856	2,548119379
C1 aj	85,16	28,85	3,419047619	2,518321146
C2	86,26	31,32	3,552631579	2,5524311
C2 aj	84,34	33,79	3,552845528	2,516187762
C3	31,59	83,52	3,078947368	2,255982339
C3 aj	31,32	83,79	3,059016393	2,2528766
Arima 360 5%	69,51	69,51	3,407114625	2,308599515
Arima 360 1%	97,53	21,7	3,746835443	2,539272834
Arima 100 5%	57,14	69,51	3,415019763	2,327733864

FIGURE B.4 – Résultat pic 4

	<i>Spécificité</i>	<i>Sensibilité</i>	<i>Moyenne</i>	<i>Ecart-type</i>
C1	83,52	43,13	6,057324841	3,274301723
C1 aj	80,22	46,43	6,047337278	3,30909758
C2	82,14	50,82	6,518918919	3,193219535
C2 aj	80,22	56,04	6,549019608	3,209061834
C3	19,78	96,98	4,311614731	3,113260851
C3 aj	19,51	97,53	4,222535211	3,068621821
Arima 360 5%	61,81	93,13	5,474926254	3,047230226
Arima 360 1%	96,7	46,43	7,615384615	2,84521319
Arima 100 5%	48,9	89,84	5,085626911	3,069542864

FIGURE B.5 – Résultat du pic lent max = 25 %

	<i>Spécificité</i>	<i>Sensibilité</i>	<i>Moyenne</i>	<i>Ecart-type</i>
C1	80,77	63,74	6,99137931	3,590501393
C1 aj	77,47	73,35	7,262172285	3,63961492
C2	79,4	76,65	7,111111111	3,409442852
C2 aj	77,47	87,09	7,223974763	3,410286836
C3	15,66	100	4,013736264	2,890533289
C3 aj	15,38	100	3,947802198	2,822581239
Arima 360 5%	57,42	100	4,815934066	2,800367736
Arima 360 1%	96,15	90,66	8,354545455	3,141795018
Arima 100 5%	43,96	98,9	4,983333333	3,041907846

FIGURE B.6 – Résultat du pic lent max = 50 %

	<i>Spécificité</i>	<i>Sensibilité</i>	<i>Moyenne</i>	<i>Ecart-type</i>
C1	94,51	32,69	3,285714286	0,931036029
C1 aj	93,41	35,16	3,265625	0,959694628
C2	93,13	35,44	3,333333333	0,912870929
C2 aj	92,03	37,64	3,306569343	0,94372575
C3	59,07	71,7	2,536398467	1,21037854
C3 aj	58,52	73,9	2,550185874	1,216448971
Arima 360 5%	84,62	67,31	3,114285714	1,041751363
Arima 360 1%	98,9	29,67	3,611111111	0,638826806
Arima 100 5%	75,82	66,76	3,049382716	1,070659423

FIGURE B.7 – Résultat du pic rapide max = 25 %

	<i>Spécificité</i>	<i>Sensibilité</i>	<i>Moyenne</i>	<i>Ecart-type</i>
C1	93,13	62,09	4,154867257	1,053205997
C1 aj	91,76	68,96	4,171314741	1,072629416
C2	91,76	67,86	4,198380567	1,046246392
C2 aj	90,38	74,18	4,207407407	1,067229302
C3	52,47	94,78	3,205797101	1,496496301
C3 aj	51,92	96,15	3,197142857	1,492480813
Arima 360 5%	81,59	97,25	3,788135593	1,189079404
Arima 360 1%	98,63	79,67	4,586206897	0,661195383
Arima 100 5%	71,43	95,88	3,727793696	1,240064423

FIGURE B.8 – Résultat du pic rapide max = 50 %

B.2 Graphiques

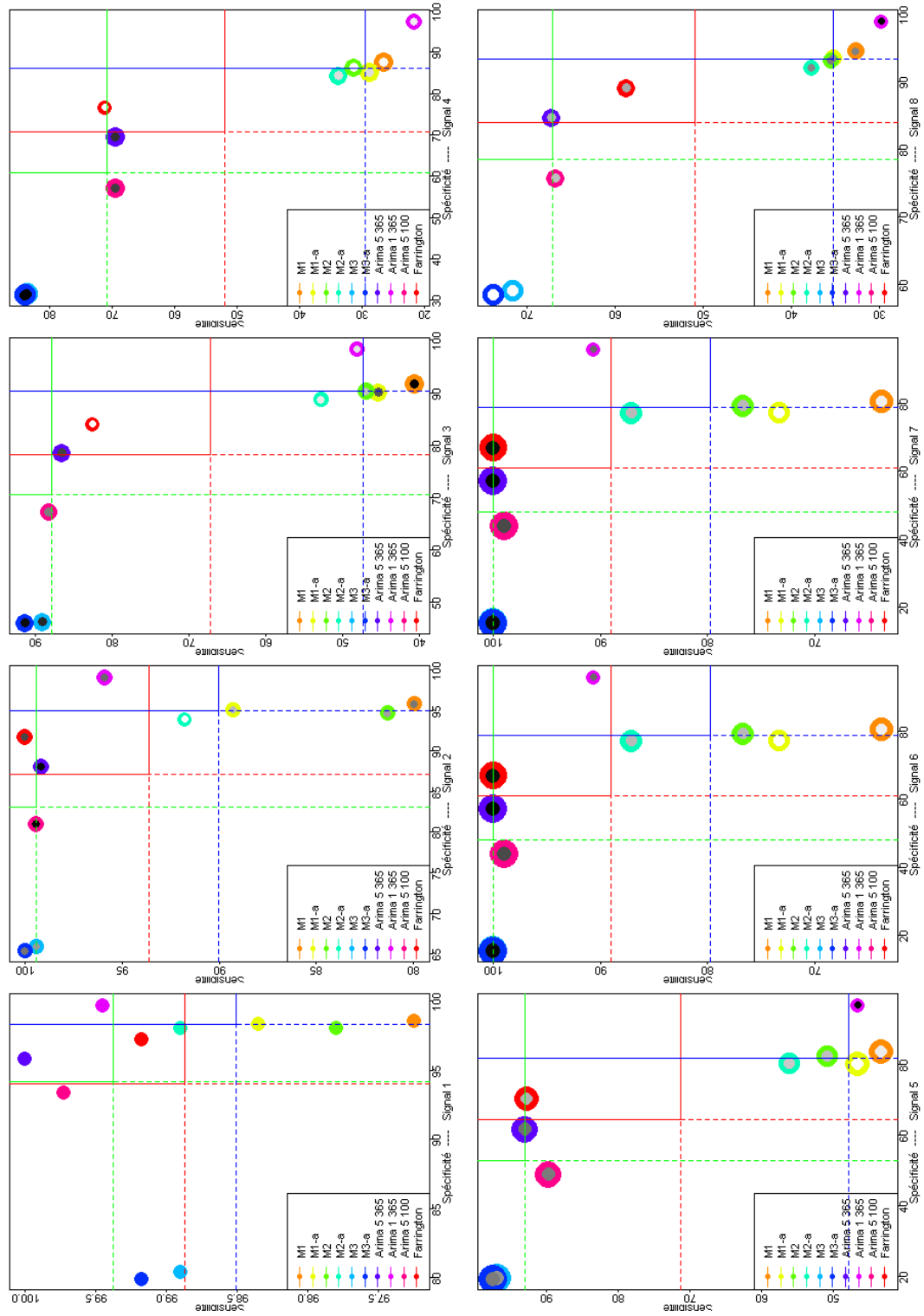


FIGURE B.9 – Graphiques des résultats pour Carcassonne

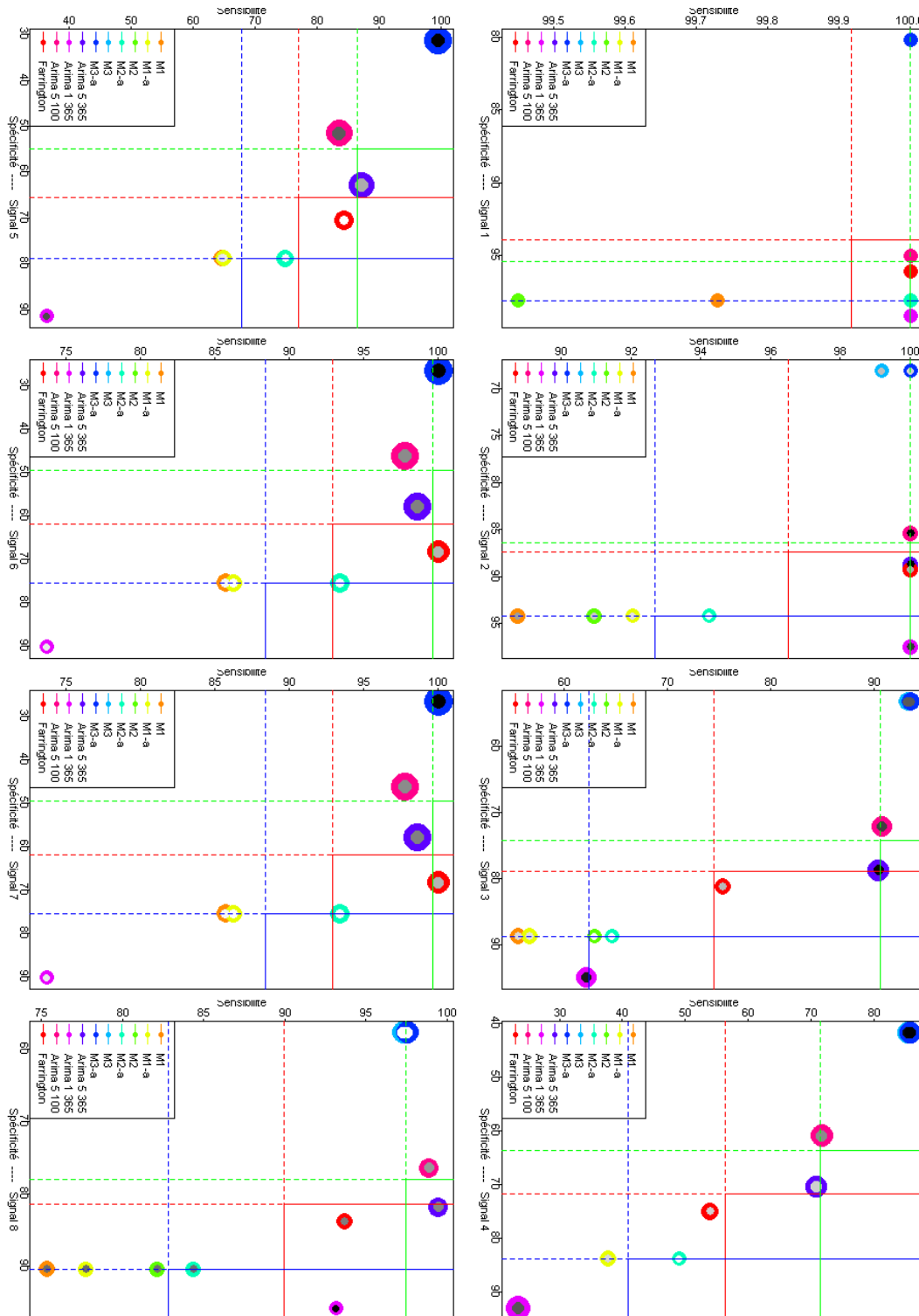


FIGURE B.10 – Graphiques des résultats pour Montpellier

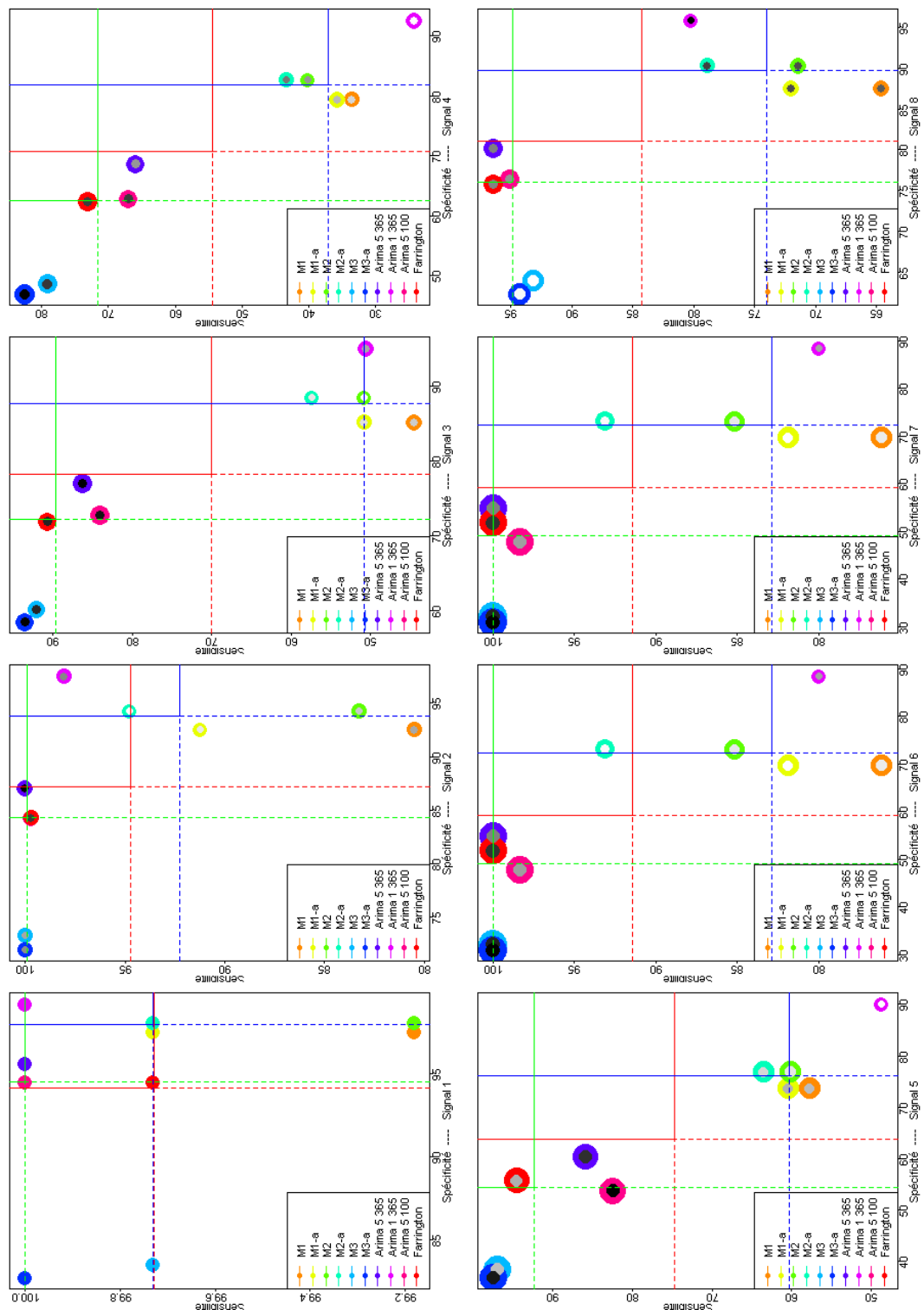


FIGURE B.11 – Graphiques des résultats pour Nimes

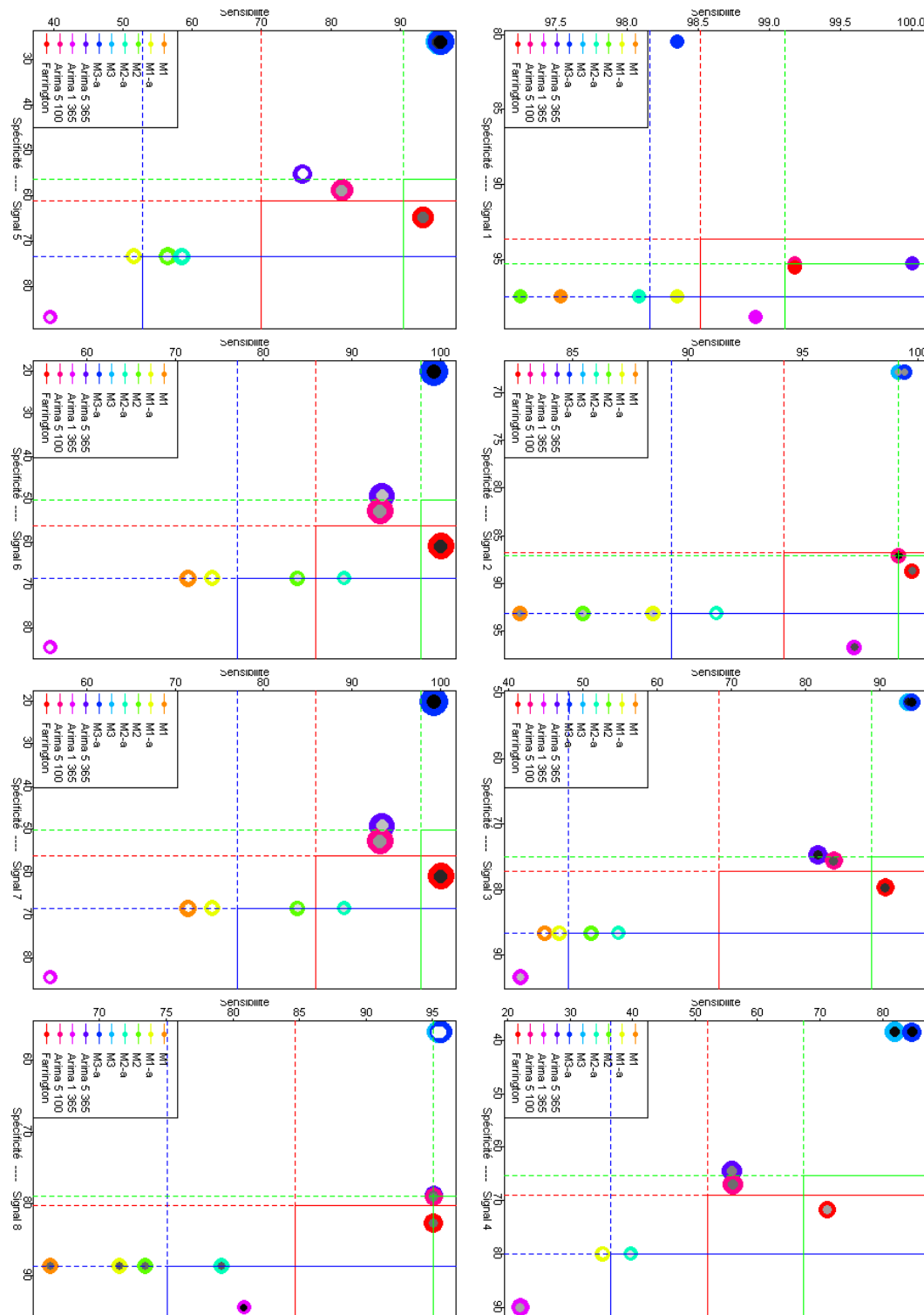


FIGURE B.12 – Graphiques des résultats pour Perpignan

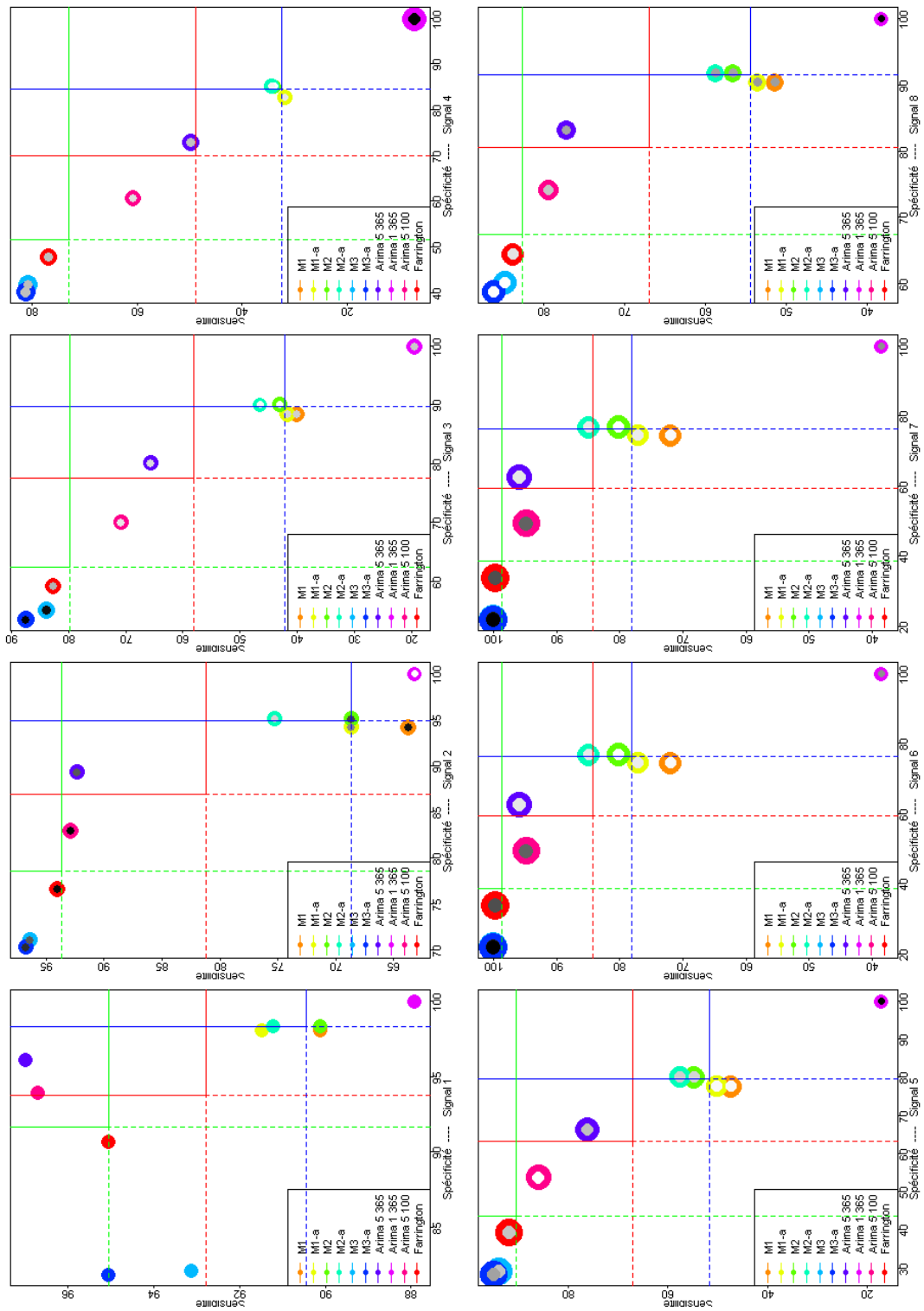


FIGURE B.13 – Graphiques des résultats pour Mende

B.3 Analyse de variance

B.3.1 Quelques lignes du tableau d'analyse

	sens.spe	pic	methode	serie	Spéc	Sens	Moyenne	Ecart.type
1	195.88	1	1	1	98.63	97.25	1.000000	0.00000000
2	196.70	1	2	1	98.35	98.35	1.000000	0.00000000
3	195.88	1	3	1	98.08	97.80	1.000000	0.00000000
4	196.98	1	4	1	98.08	98.90	1.000000	0.00000000
5	179.39	1	5	1	80.49	98.90	1.000000	0.00000000
6	179.13	1	6	1	79.95	99.18	1.000000	0.00000000
7	195.88	1	7	1	95.88	100.00	1.000000	0.00000000
8	199.18	1	8	1	99.73	99.45	1.000000	0.00000000
9	193.14	1	9	1	93.41	99.73	1.000000	0.00000000
10	196.43	1	10	1	97.25	99.18	1.000000	0.00000000
11	175.83	2	1	1	95.88	79.95	1.154639	0.43959714

B.3.2 Diagnostiques du modèle

One-sample Kolmogorov-Smirnov test

data: residus_de_laov
D = 0.0543, p-value = 0.1895
alternative hypothesis: two-sided

Bartlett test of homogeneity of variances

data: sens.spe by methode by pic by dd\$serie by dd\$methode
Bartlett's K-squared = 14.2292, df = 9, p-value = 0.1144

Annexe C

Code source R

Avant de donner les codes sources, nous informons le lecteur que ceux-ci ne sont pas entièrement exhaustifs. En effet, pour des raisons évidentes de concision, nous n'avons pas souhaité placer ici des programmes usuels ou similaires à un autre déjà présent.

C.1 Mise en forme des données

On donne deux exemples :

```
1 #Mise en forme des données
3 donnee<-read.csv("D:\\Administrateur\\Mes documents\\nicolas\\mon.csv",
  ",sep=";",header=TRUE)
5 quantile(donnees[,3],0.90)/quantile(donnees[,3],0.10)
7 donnee
9 #Premiers graphiques
11 plot(donnee$urg,ylab="Nombre de patients",xlab="Jours")
  title("Nombre d'admissions aux urgences par jour")
13
15 plot(donnee$HOSPIT,ylab="Nombre de patients",xlab="Jours")
  title("Nombre d'hospitalisation par jour")
17 #On cherche les valeurs manquantes et les valeurs trop faibles :
19 k<-10
  change<-0
21 for(i in 1:1762){
  if(is.na(donnee[,3])[i] || donnee[i,3]<10)
23 {
  change[k]<-i
25 k<-k+1
```



```

}
27 }
change
29
#On remplace les données isolées
31
for( i in c(156,183,742)){
33 donnee[i,3]<- (donnee[i-1,3]+donnee[i+1,3])/2
donnee[i,4]<- (donnee[i-1,4]+donnee[i+1,4])/2
35 }

37 #Fonction servant à tracer les moyennes mobiles

39 ma3<- function(x) {
y<- numeric(length(x)-2)
41 for(i in 2:(length(x)-1)) {
y[i] <- (x[i-1] + x[i] + x[i+1])/3
43 }
y}
45

#On trace la courbe des moyennes mobiles grâce à notre fonction
47
tm<-ma3(donnee[-c(149:152),3])
49 tm2<-ma3(donnee[-c(149:152),4])

51 #On remplace les quatre valeurs succesives par :

53 for( i in 1:4){
donnee[i+148,3]<- tm[148]+(abs(tm[148]-tm[149]))*i/5
55 donnee[i+148,4]<- tm2[148]+abs(tm2[148]-tm2[149])*i/5
}
57
plot(donnee[,3])
59

61 #On change le format de nos dates

63 library(epitools)
donnees<-donnee
65 donnees <- cbind(donnees,epidate(donnees[,2],format="%d/%m/%Y")$month
)
donnees <- cbind(donnees,epidate(donnees[,2],format="%d/%m/%Y")$year)
67 donnees <- cbind(donnees,epidate(donnees[,2],format="%d/%m/%Y")$
weekday)
n<-length(donnee[,1])
69 colnames(donnees) <- c("hop","date","adm","adm75","hosp","mois","
annee","jour")
for(i in 1:n){
71 if(donnees$jour[i]=="samedi" || donnees$jour[i]=="dimanche")
{
73 donnees$we[i]<- "we"
}
}

```

```

75 else
  {
77 donnees$we[i] <- "se"
  }
79 }

81 colnames(donnees) <- c("hop", "date", "adm", "adm75", "hosp", "mois", "
  annee", "jour", "we")

83 #On fait les boxplot

85 par(mfrow=c(1,1))

87 boxplot(split(donnees$adm, donnees$annee))
  boxplot(split(donnees$adm, donnees$mois))
89 boxplot(split(donnees$adm, donnees$jour))
  boxplot(split(donnees$adm, donnees$we))
91
modele <- lm(adm~jour+annee+mois, data=donnees)
93 anova(lm(adm~jour+annee+mois, data=donnees))

95 shapiro.test(residuals(modele))
  bartlett.test(adm~jour+annee+mois, data=donnees)
97
plot(donnees$hosp)
99 donnees[1,]

101
split(donnees$adm, donnees$jour) -> q
103 boxplot(q$lundi, q$mardi, q$mercredi, q$jeudi, q$vendredi, q$samedi, q$
  dimanche, atxt="n")
axis(1, labels=c("lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi",
  "dimanche"), at=1:7)
105
split(donnees$adm, donnees$mois) -> q
107 boxplot(q$janvier, q$février, q$mars, q$avril, q$mai, q$juin, q$juillet, q$
  août, q$septembre, q$octobre, q$novembre, q$décembre, atxt="n")
axis(1, labels=c("janv", "fév", "mars", "avril", "mai", "juin", "juillet", "
  août", "sept", "oct", "nov", "déc"), at=1:12)

```

mefm.R

et :

```

#Import des données du fichier Carcassonne
2
donnee <- read.csv("D:\\Administrateur\\Mes documents\\nicolas\\a.csv",
  sep=";", header=TRUE)
4
donnee <- donnee[, -6]
donnee
6
#Premier tracé
8

```

```

plot(donnee[,3])
10 plot(donnee[,4])
plot(donnee[,5])
12 abline(v=1328)
donnee[1328,2]
14
#Remarque : Pour les hospitalisations, il y a une forte augmentation
#(cassure) vers avril 2009
16
#On remplace les valeurs manquantes en faisant une moyenne sur les
#valeurs autour, avec une fenêtre de 2.
18
donnee[151,3]<-70/3
20 donnee[151,4]<-20/3

22 donnee[152,3]<-(65+59)/3
donnee[152,4]<-(19+13)/3
24
donnee[156,3]<-(59+48+57+71)/4
26 donnee[156,4]<-(20+38)/4

28 donnee[183,3]<-(63+64+82+105)/4
donnee[183,4]<-(18+15+18+16)/4
30 donnee[183,5]<-(15+12+27+19)/4
donnee
32
#On se donne une idée de la série grâce aux moyennes mobiles
34
#Fonction servant à faire des moyennes mobiles
36
ma3<- function(x) {
38 y<- numeric(length(x)-2)
for(i in 2:(length(x)-1)) {
40 y[i] <- (x[i-1] + x[i] + x[i+1])/3
}
42 y}

44 #On trace la courbe des moyennes mobiles grâce à notre fonction

46 tm<-ma3(donnee[,3])
plot(donnee[,3])
48 lines(tm[2:length(tm-1)])

50
#On différencie les week-end des jours de la semaine, et les jours de
#la semaine entre eux
52
A<-c("s",rep(c("w","w","s","s","s","s","s"),251),"w","w","s","s")
54 B<-c("v",rep(c("s","d","l","ma","me","j","v"),251),"s","d","l","ma")

56 for(i in 1:1762){
donnee$Ind[i]<-A[i]

```

```
58 }
60 for(i in 1:1762){
61   donnee$Ind2[i]<-B[i]
62 }
63 donnee
64
65
66 #Je remarque qu'il n'y a pas de ligne pour le jour 20/01/2006 ; je la
67 rajoute dans le csv en NA
68
69 donnee.sep<-split(donnee[,3],A)
70 donnee.jour<-split(donnee[,3],B)
71
72 #On regarde les deux groupes
73
74 boxplot(donnee.sep)
75 boxplot(donnee.jour)
76
77 #On transforme l'objet en serie chronologique
78
79 serie<-ts(donnee[,3],start=4/7,frequency=7)
80 serie
81
82 #On regarde la fonction d'autocorrélation et la fonction d'
83 autocorrélation partielle avant transformation
84
85 plot(serie)
86 par(mfrow=c(1,2))
87 acf(serie,lag.max=200)
88 acf(serie,lag.max=200,type="partial")
89
90 #On enlève la tendance
91
92 C<-c(rep("a2005",163),rep("a2006",358),rep("a2007",365),rep("a2008",
93   ,366),rep("a2009",365),rep("a2010",145))
94 donnee.annee<-split(donnee[,3],C)
95
96 for(i in 1:1762){
97   donnee$Ind3[i]<-C[i]
98 }
99 donnee
100
101 moy2005<-mean(donnee.annee$a2005)
102 moy2006<-mean(donnee.annee$a2006)
103 moy2007<-mean(donnee.annee$a2007)
104 moy2008<-mean(donnee.annee$a2008)
105 moy2009<-mean(donnee.annee$a2009)
106 moy2010<-mean(donnee.annee$a2010)
```

```

108 for(i in 1: 163) {
donnee$trend[i]<-moy2005
110 donnee$trendl[i]<-donnee[i,3] - donnee$trend[i]
112 }
114 for(i in 164: 521) {
donnee$trend[i]<-moy2006
116 donnee$trendl[i]<-donnee[i,3] - donnee$trend[i]
118 }
120 for(i in 522:886 ) {
donnee$trend[i]<-moy2007
donnee$trendl[i]<-donnee[i,3] - donnee$trend[i]
122 }
124 for(i in 887:1253 ) {
donnee$trend[i]<-moy2008
126 donnee$trendl[i]<-donnee[i,3] - donnee$trend[i]
128 }
130 for(i in 1254:1617 ) {
donnee$trend[i]<-moy2009
donnee$trendl[i]<-donnee[i,3] - donnee$trend[i]
132 }
134 for(i in 1617:1762 ) {
donnee$trend[i]<-moy2010
136 donnee$trendl[i]<-donnee[i,3] - donnee$trend[i]
138 }
#On regarde à nouveau la fonction d'autocorrélation empirique
140 serie2<-ts(donnee$trendl ,start=4/7,frequency=7)
142 acf(serie2)
144 ?cbind
#Gestion de la date : on transforme la date selon les composante que
l'on a
146 library(epitools)
148 donnees<-donnee
donnees <- cbind(donnees ,epidate(donnees [,2] ,format="%d/%m/%Y")$month
)
150 donnees <- cbind(donnees ,epidate(donnees [,2] ,format="%d/%m/%Y")$year)
donnees <- cbind(donnees ,epidate(donnees [,2] ,format="%d/%m/%Y")$
weekday)
152 n<-length(donnee [,1])
colnames(donnees) <- c("hop" ,"date" ,"adm" ,"adm75" ,"hosp" ,"mois" ,"
annee" ,"jour" )
154 for(i in 1:n){

```

```
156 if(donnees$jour[i]=="samedi" || donnees$jour[i]=="dimanche")
158 {
160   donnees$we[i]<-"we"
162 }
164 else
166 {
168   donnees$we[i]<-"se"
170 }
172 }
174 }
176 }
178 }
180 }
182 }
184 }
186 }
188 }
190 }
192 }
194 }
196 }
198 }
200 }
202 }
204 }
206 }
208 }
210 }
212 }
214 }
216 }
218 }
220 }
222 }
224 }
226 }
228 }
230 }
232 }
234 }
236 }
238 }
240 }
242 }
244 }
246 }
248 }
250 }
252 }
254 }
256 }
258 }
260 }
262 }
264 }
266 }
268 }
270 }
272 }
274 }
276 }
278 }
280 }
282 }
284 }
286 }
288 }
290 }
292 }
294 }
296 }
298 }
300 }
302 }
304 }
306 }
308 }
310 }
312 }
314 }
316 }
318 }
320 }
322 }
324 }
326 }
328 }
330 }
332 }
334 }
336 }
338 }
340 }
342 }
344 }
346 }
348 }
350 }
352 }
354 }
356 }
358 }
360 }
362 }
364 }
366 }
368 }
370 }
372 }
374 }
376 }
378 }
380 }
382 }
384 }
386 }
388 }
390 }
392 }
394 }
396 }
398 }
400 }
402 }
404 }
406 }
408 }
410 }
412 }
414 }
416 }
418 }
420 }
422 }
424 }
426 }
428 }
430 }
432 }
434 }
436 }
438 }
440 }
442 }
444 }
446 }
448 }
450 }
452 }
454 }
456 }
458 }
460 }
462 }
464 }
466 }
468 }
470 }
472 }
474 }
476 }
478 }
480 }
482 }
484 }
486 }
488 }
490 }
492 }
494 }
496 }
498 }
500 }
502 }
504 }
506 }
508 }
510 }
512 }
514 }
516 }
518 }
520 }
522 }
524 }
526 }
528 }
530 }
532 }
534 }
536 }
538 }
540 }
542 }
544 }
546 }
548 }
550 }
552 }
554 }
556 }
558 }
560 }
562 }
564 }
566 }
568 }
570 }
572 }
574 }
576 }
578 }
580 }
582 }
584 }
586 }
588 }
590 }
592 }
594 }
596 }
598 }
600 }
602 }
604 }
606 }
608 }
610 }
612 }
614 }
616 }
618 }
620 }
622 }
624 }
626 }
628 }
630 }
632 }
634 }
636 }
638 }
640 }
642 }
644 }
646 }
648 }
650 }
652 }
654 }
656 }
658 }
660 }
662 }
664 }
666 }
668 }
670 }
672 }
674 }
676 }
678 }
680 }
682 }
684 }
686 }
688 }
690 }
692 }
694 }
696 }
698 }
700 }
702 }
704 }
706 }
708 }
710 }
712 }
714 }
716 }
718 }
720 }
722 }
724 }
726 }
728 }
730 }
732 }
734 }
736 }
738 }
740 }
742 }
744 }
746 }
748 }
750 }
752 }
754 }
756 }
758 }
760 }
762 }
764 }
766 }
768 }
770 }
772 }
774 }
776 }
778 }
780 }
782 }
784 }
786 }
788 }
790 }
792 }
794 }
796 }
798 }
800 }
802 }
804 }
806 }
808 }
810 }
812 }
814 }
816 }
818 }
820 }
822 }
824 }
826 }
828 }
830 }
832 }
834 }
836 }
838 }
840 }
842 }
844 }
846 }
848 }
850 }
852 }
854 }
856 }
858 }
860 }
862 }
864 }
866 }
868 }
870 }
872 }
874 }
876 }
878 }
880 }
882 }
884 }
886 }
888 }
890 }
892 }
894 }
896 }
898 }
900 }
902 }
904 }
906 }
908 }
910 }
912 }
914 }
916 }
918 }
920 }
922 }
924 }
926 }
928 }
930 }
932 }
934 }
936 }
938 }
940 }
942 }
944 }
946 }
948 }
950 }
952 }
954 }
956 }
958 }
960 }
962 }
964 }
966 }
968 }
970 }
972 }
974 }
976 }
978 }
980 }
982 }
984 }
986 }
988 }
990 }
992 }
994 }
996 }
998 }
1000 }
```

carcassonne.R

C.2 Phase de test

Avant de donner les codes nous proposons ce schéma permettant de comprendre comment les fonctions interagissent entre elles :

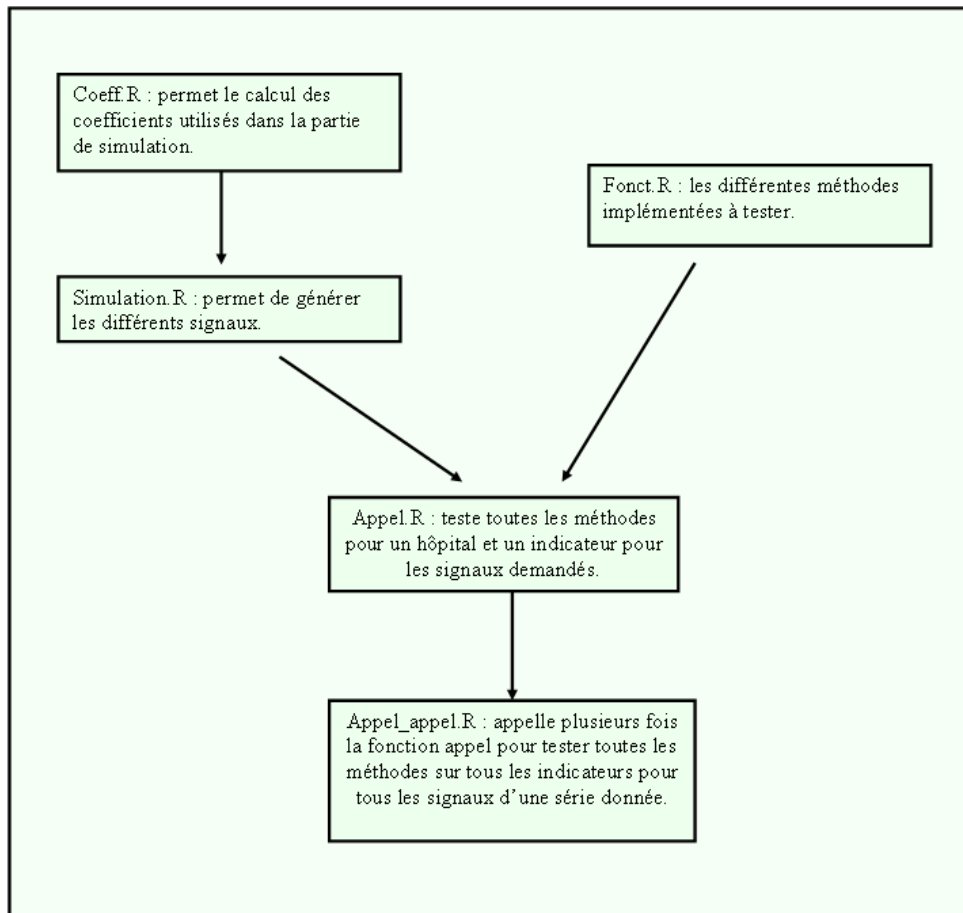
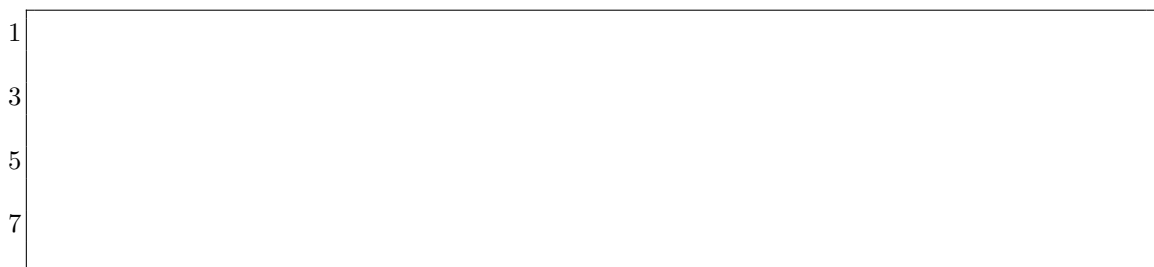


FIGURE C.1 – Diagramme quantile-quantile des résidus du modèle

C.2.1 La fonction pour générer les signaux



```
9 # Générateur d'une série à partir d'un vecteur de données et du
  nombre de jour du pic :
  # Format des données : deux colonnes (date et valeur)
11 # On augmente les valeurs de chaque jours
  generateur2 <- function(donnees ,jour ,J0)
13 {
15
17   es<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\rapport
    \\mypack\\R\\coeff.csv",sep=";")
19   c_1_25<-es[16:19,1]
    c_2_25<-es[8:19,2]
21   c_3_25<-es[5:19,3]
    c_1_50<-es[15:19,4]
23   c_2_50<-es[6:19,5]
    c_3_50<-es[4:19,6]
25
27   data<-donnees
29   # Si 1 jour :
    if(jour==1)
    {
31     # J0 correspond au jour de début du pic :
      data[J0] <- round(data[J0] * 2)
33
    }
35
    #Si 3 jours :
37   if(jour==3)
    {
39     # J0 correspond au jour de début du pic :
41     data[J0] <- round(data[J0] * 1.5)
      data[J0+1] <- round(data[J0+1] * 1.5)
43     data[J0+2] <- round(data[J0+2] * 1.5)
45   }
47   # Si 6 jours :
    if(jour==6)
    {
49     # J0 correspond au jour de début du pic :
51     data[J0] <- round(data[J0] * 1.2)
      data[J0+1] <- round(data[J0+1] * 1.2)
53     data[J0+2] <- round(data[J0+2] * 1.2)
      data[J0+3] <- round(data[J0+3] * 1.2)
55     data[J0+4] <- round(data[J0+4] * 1.2)
      data[J0+5] <- round(data[J0+5] * 1.2)
57   }
```



```
59 | # Si 9 jours :
    | if(jour==9)
61 | {
    |   # J0 correspond au jour de début du pic :
63 |   data[J0] <- round(data[J0] * 1.1)
    |   data[J0+1] <- round(data[J0+1] * 1.1)
65 |   data[J0+2] <- round(data[J0+2] * 1.1)
    |   data[J0+3] <- round(data[J0+3] * 1.1)
67 |   data[J0+4] <- round(data[J0+4] * 1.1)
    |   data[J0+5] <- round(data[J0+5] * 1.1)
69 |   data[J0+6] <- round(data[J0+6] * 1.1)
    |   data[J0+7] <- round(data[J0+7] * 1.1)
71 |   data[J0+8] <- round(data[J0+8] * 1.1)
    | }
73 |
    | #Si pic rapide à 25%
75 |
    | if(jour==4)
77 | {
    |   # J0 correspond au jour de début du pic :
79 |   data[J0:(J0+3)]<-data[J0:(J0+3)]*c_1_25
    | }
81 |
    | #Si pic rapide à 50%
83 |
    | if(jour==5)
85 | {
87 |   # J0 correspond au jour de début du pic :
    |   data[J0:(J0+4)]<-data[J0:(J0+4)]*c_1_50
89 | }
91 |
    | #Si pic lent à 25%
93 |
    | if(jour==12)
95 | {
    |   # J0 correspond au jour de début du pic :
    |   data[J0:(J0+11)]<-data[J0:(J0+11)]*c_2_25
97 | }
99 |
    | #Si pic lent à 50%
101 |
    | if(jour==14)
103 | {
    |   # J0 correspond au jour de début du pic :
    |   data[J0:(J0+13)]<-data[J0:(J0+13)]*c_2_50
105 | }
107 |
    | #Si pic double à 25%
109 |
    | if(jour==15)
    | {
```

```

111     # J0 correspond au jour de début du pic :
      data[J0:(J0+14)]<-data[J0:(J0+14)]*c_3_25
113   }
115   #Si pic double à 50%
117   if(jour==16)
      {
119     # J0 correspond au jour de début du pic :
      data[J0:(J0+15)]<-data[J0:(J0+15)]*c_3_50
121   }
122 data
123 }

```

SIMULATION.R

Les coefficients ayant été calculés grâce à la fonction suivante :

```

1 #Les fonctions servant à calculer les coefficients
3 f_rap<-function(x){
  dgamma(-x+20,1.5,1.05)
5 }
7 f_lent<-function(x){
  dgamma(-x+20,1.5,1/4)
9 }
11 f_double<-function(x){
  dgamma(-x+20,4,0.8)+dgamma(-x+13,4,0.8)
13 }
15 max<-0.50
17 test0<-test
  test_rap<-round(test0*(f_rap(1:30)/max(f_rap(1:20))*max+1))
19 test_lent<-round(test0*(f_lent(1:30)/max(f_lent(1:20))*max+1))
  test_double<-round(test0*(f_double(1:30)/max(f_double(1:20))*max+1))
21 uu<-cbind(f_rap(1:20)/max(f_rap(1:20))*max+1,f_lent(1:20)/max(f_lent
  (1:20))*max+1,f_double(1:20)/max(f_double(1:20))*max+1)
  colnames(uu)<-c("rapide","lent","double")
23
  coef1_50<-uu[,1]
25 coef2_50<-uu[,2]
  coef3_50<-uu[,3]
27
  max<-0.25
29
  test0<-test
31 test_rap<-round(test0*(f_rap(1:30)/max(f_rap(1:20))*max+1))
  test_lent<-round(test0*(f_lent(1:30)/max(f_lent(1:20))*max+1))

```

```

33 test_double<-round(test0*(f_double(1:30)/max(f_double(1:20))*max+1))
uu<-cbind(f_rap(1:20)/max(f_rap(1:20))*max+1,f_lent(1:20)/max(f_lent
(1:20))*max+1,f_double(1:20)/max(f_double(1:20))*max+1)
35 colnames(uu)<-c("rapide","lent","double")

37 coef1_25<-uu[,1]
coef2_25<-uu[,2]
39 coef3_25<-uu[,3]

41 Coefficient<-cbind(coef1_25,coef2_25,coef3_25,coef1_50,coef2_50,coef3
_50)
write.csv2(Coefficient,"D:\\Administrateur\\Mes documents\\nicolas\\
rapport\\coeff.csv")

```

coeff.r

C.2.2 Les méthodes choisies pour la phase de test

```

#Fonctions
2
c1_b<-function(etude,pic){
4
  C1<-0
6  for(i in 1:7){
  C1[i]<-0
8  }

10  for(i in 8:1762){
  m<-mean(etude[(i-7):(i-1)])
12  e<-sd(etude[(i-7):(i-1)])
  C1[i]<-m+3*e
14  }

16
  k<-0
18  u<-0
  for(i in pic){
20  k<-k+1
  if(etude[i]>C1[i]&&u==0){
22  u<-k
  break()
24  }
  }
26  u
  }

28
#-----#
30
c1_m<-function(etude,pic){
32

```

```

C1<-0
34  for(i in 1:7){
    C1[i]<-0
36  }
    ee<-sd(etude)
38  for(i in 8:1762){
m<-mean(etude[(i-7):(i-1)])
40  e<-sd(etude[(i-7):(i-1)])
    C1[i]<-m+3*min(e, ee)
42  }

44
    k<-0
46  u<-0
    for(i in pic){
48  k<-k+1
    if(etude[i]>C1[i]&&u==0){
50  u<-k
    break()
52  }
    }
54  u
    }

56

58 #-----#

60 c2_b<-function(etude , pic){
62  C1<-0
    for(i in 1:9){
64  C1[i]<-0
    }

66  for(i in 10:1762){
68  m<-mean(etude[(i-9):(i-3)])
    e<-sd(etude[(i-9):(i-3)])
70  C1[i]<-m+3*e
    }

72

74  k<-0
    u<-0
76  for(i in pic){
    k<-k+1
78  if(etude[i]>C1[i]&&u==0){
    u<-k
80  break()
    }
82  }
    u
84  }

```

```

86 #-----#
88 c2_m<-function(etude , pic){
90   C1<-0
91   for(i in 1:9){
92     C1[i]<-0
93   }
94   ee<-sd(etude)
95   for(i in 10:1762){
96     m<-mean(etude[(i-9):(i-3)])
97     e<-sd(etude[(i-9):(i-3)])
98     C1[i]<-m+3*min(e , ee)
99   }
100
101
102   k<-0
103   u<-0
104   for(i in pic){
105     k<-k+1
106     if(etude[i]>C1[i]&&u==0){
107       u<-k
108       break()
109     }
110   }
111   u
112   }
113
114 #-----#
116 c3_b<-function(etude , pic){
118   C2<-0
119   for(i in 1:9){
120     C2[i]<-0
121   }
122
123   for(i in 10:1762){
124     m<-mean(etude[(i-9):(i-3)])
125     e<-sd(etude[(i-9):(i-3)])
126     C2[i]<-max(0 ,(etude[i]-m)/e - 0.5 )
127   }
128
129
130   C3<-0
131   C3[1]=0
132   C3[2]=0
133   C3[3]=0
134
135
136

```

```

138   for(i in 4:(length(C2)))
139   {
140   C3[i]<-C2[i]+C2[i-2]+C2[i-1]
141   }

142   u<-0
143   k<-0
144   for(i in pic){
145     k<-k+1
146     if(C3[i]>2 && u==0 ) {
147       u<-k
148       break()
149     }
150   }

152   u
153   }

154 #-----#

156 c3_m<-function(etude , pic){
157
158   C2<-0
159   for(i in 1:9){
160   C2[i]<-0
161   }

162   ee<-sd(etude)
163   for(i in 10:1762){
164   m<-mean(etude[(i-9):(i-3)])
165   e<-min(sd(etude) ,sd(etude[(i-9):(i-3)]))
166   C2[i]<-max(0 ,(etude[i]-m)/e - 0.5 )
167   }

170
171   C3<-0
172   C3[1]=0
173   C3[2]=0
174   C3[3]=0

176
177   for(i in 4:(length(C2)))
178   {
179   C3[i]<-C2[i]+C2[i-2]+C2[i-1]
180   }

182   u<-0
183   k<-0
184   for(i in pic){
185     k<-k+1
186     if(C3[i]>2 && u==0 ) {
187       u<-k
188       break()

```

```

190     }
192     u
194     }
196
198 #-----#
199 AR54<-function(etude , pic){
200     u<-0
202     k<-0
204     for(i in pic){
206         k<-k+1
208         start<-i
210         x0<-etude
212         x0<-x0[max(1,(i-365-1)):(i-1)]
214         x0<-log(x0)
216         x<-ts(x0)
218         a<-arima(x,order=c(1,1,1),seasonal=list(order=c(0,1,1),period=7))
220         prediction<-predict(a,n.ahead=1)
222         pd<-prediction$pred
224         pdse<-prediction$se
226         if(log(etude[i])>pd[1]+qnorm(0.95)*pdse[1] && u==0)
228         {
230             u<-k
232             break()
234         }
236     }
238     u }
240 #-----#
241 AR14<-function(etude , pic){
242     u<-0
244     k<-0
246     for(i in pic){
248         k<-k+1
250         start<-i
252         x0<-etude
254         x0<-x0[max(1,(i-365-1)):(i-1)]

```

```

242  x0<-log(x0)
      x<-ts(x0)

244  a<-arima(x,order=c(1,1,1),seasonal=list(order=c(0,1,1),period=7))

246  prediction<-predict(a,n.ahead=1)
      pd<-prediction$pred
248  pdse<-prediction$se

250  if(log(etude[i])>pd[1]+qnorm(0.99)*pdse[1] && u==0)
      {
252  u<-k
      break()
254  }

256  }
      u }
258

260  #-----#
262

264  AR51<-function(etude , pic){

266  u<-0
      k<-0
268  for(i in pic){
      k<-k+1
270  start<-i

272  x0<-etude
      x0<-x0[max(1,(i-100-1)):(i-1)]
274  x0<-log(x0)
      x<-ts(x0)

276  a<-arima(x,order=c(1,1,1),seasonal=list(order=c(0,1,1),period=7))

278  prediction<-predict(a,n.ahead=1)
      pd<-prediction$pred
280  pdse<-prediction$se

282  if(log(etude[i])>pd[1]+qnorm(0.95)*pdse[1] && u==0)
      {
284  u<-k
      break()
286  }

288  }
      u }
290

292  #-----#

```



```

#-----#
294
296 far_1 <- function(vect_date, etude, pic) {
298   obj<-create.disProg(week=as.Date(vect_date, format="%d/%m/%Y"),
      observed=etude, state=rep(0,1762), freq=365, epochAsDate=TRUE, start=c
      (2005,303))
300   res<-algo.farrington(obj, control=list(range=pic, b=3, w=3, alpha=0.01))
      comp<-0
302   u<-0
      for(j in 1:length(pic)) {
304     if(res$alarm[j]==1){
      u<-comp+1
306     break()
      }
308     comp<-comp+1
      }
310   u
      }
312
314 #-----#
316 far_mod <- function(etude, pic) {
318
      comp<-0
320     uuu<-0
      for(j in 1:length(pic)) {
322
      jour<- rep(0,4)
324     for(k in 4:1){
      jour[5-k]<-etude[pic[j]-k*7]
326     }
328     semaine<- rep(0,4)
      for(k in 4:1){
330     semaine[5-k]<-round(mean(etude[(pic[j]-k*7):(pic[j]-k*7+5)]))
      }
332
      a<-1:4
334     a2<-sin(a)
336     modele<- lm(semaine~a+a2)
      r<-coefficients(modele)
338     pol<-function(x, moy){moy+r[2]*x+r[3]*sin(x)}
      u<-abs((pol(1:4, mean(jour)*r[1]/mean(semaine))-jour)/var(jour))
340
342     for(x in 1:4){

```

```

344 if(u[x]==0) {
    u[x]<-1/10
  }
346 if(u[x]>10){
    u[x]<-10
348 }
  }
350
352 u<-(1/u)^2
352 u<-u*1/(sum(u))
    modele2<- lm(jour~a+a2, weight=u)
354 tt<-as.vector(residuals(modele2))
    var<-t(tt)%*%(u*tt)
356 fitted(modele2)
    rr<-coefficients(modele2)
358 UP<- rr[1]+rr[2]*5+rr[3]*sin(5)+ 3*min(max(var,14),20)

360 comp<-comp+1
    if(etude[pic[j]]>UP){
362 uuu<-comp
    break()
364 }

366 }
    uuu
368 }

```

fonct_t.r

C.2.3 Appel des fonctions pour les tester

```

1 #Detection de pics
3
3 #Donner la série initiale
5
5 base<-donnees[,3]
7
7 #Donner le numéro du générateur
9
9 signal<-3
11
11 #Donner le plus long signal
13
13 max_signal<-30
15
15 #Resultat
17
17 long<-length(base)
19 tmp<-c((long-363)-(max_signal-1):(long-(max_signal-1)))

```

```

date<-donnees [tmp,2]
21 date<-data.frame(date)
date
23
25
#####
27 #####
#####METHODES A LESSAI#####
29 #####
#####
31
33
35 #-----#
37 #Méthode C1-MILD non modifiée
39 comp<-0
for(i in tmp){
41
comp<-comp+1
43 etude<-generateur2(base , signal , i)
ee<-sd(etude)
45
date$C1_b_s [comp]<- c1_b(base , i:( i+(signal -1)))
47 date$C1_b_a [comp]<- c1_b(etude , i:( i+(signal -1)))
cat(i)
49 }
51 #-----#
53 #Méthode C1-MILD modifiée
55 comp<-0
for(i in tmp){
57
comp<-comp+1
59 etude<-generateur2(base , signal , i)
ee<-sd(etude)
61
date$C1_m_s [comp]<- c1_m(base , i:( i+(signal -1)))
63 date$C1_m_a [comp]<- c1_m(etude , i:( i+(signal -1)))
cat(i)
65 }
67
#-----#
69 #Méthode C2-MILD non modifiée
71

```

```

comp<-0
73 for(i in tmp){

75 comp<-comp+1
   etude<-generateur2(base , signal , i)
77 ee<-sd(etude)

79 date$C2_b_s[comp]<- c2_b(base , i:( i+(signal -1)))
date$C2_b_a[comp]<- c2_b(etude , i:( i+(signal -1)))
81 cat(i)
   }
83 #-----#

85 #Méthode C2-MILD modifiée

87 comp<-0
   for(i in tmp ){
89 comp<-comp+1
   etude<-generateur2(base , signal , i)
91 ee<-sd(etude)

93 date$C2_m_s[comp]<- c2_m(base , i:( i+(signal -1)))
date$C2_m_a[comp]<- c2_m(etude , i:( i+(signal -1)))
95 cat(i)
   }
97 #-----#

99 #Méthode C3-EXTRA non modifiée
101 comp<-0
103 for(i in tmp ){
   comp<-comp+1
105 etude<-generateur2(base , signal , i)
   ee<-sd(etude)
107
   date$C3_b_s[comp]<- c3_b(base , i:( i+(signal -1)))
109 date$C3_b_a[comp]<- c3_b(etude , i:( i+(signal -1)))
   cat(i)
111 }

113 #-----#

115 #Méthode C3-EXTRA modifiée

117 comp<-0
   for(i in tmp ){
119 comp<-comp+1
   etude<-generateur2(base , signal , i)
121 ee<-sd(etude)

123 date$C3_m_s[comp]<- c3_m(base , i:( i+(signal -1)))

```

```

date$C3_m_a[comp]<- c3_m(etude , i:( i+(signal -1)))
125 cat( i )
    }
127 #-----#
129 #METHODE ARIMA alpha=5% fenêtre= 365 jours
131 comp<-0
133 for( i in tmp ){
    comp<-comp+1
135 etude<-generateur2( base , signal , i )
    ee<-sd( etude )
137 date$AR54_s[comp]<- AR54( base , i:( i+(signal -1)))
139 date$AR54_a[comp]<- AR54( etude , i:( i+(signal -1)))
    cat( i )
141 }
143 #-----#
145 #METHODE ARIMA alpha=1% fenêtre= 365 jours
147 comp<-0
    for( i in tmp ){
149 comp<-comp+1
    etude<-generateur2( base , signal , i )
151 ee<-sd( etude )
153 date$AR14_s[comp]<- AR14( base , i:( i+(signal -1)))
    date$AR14_a[comp]<- AR14( etude , i:( i+(signal -1)))
155 cat( i )
    }
157 #-----#
159 #METHODE ARIMA alpha=5% fenêtre= 100 jours
161 comp<-0
163 for( i in tmp ){
    comp<-comp+1
165 etude<-generateur2( base , signal , i )
    ee<-sd( etude )
167 date$AR51_s[comp]<- AR51( base , i:( i+(signal -1)))
169 date$AR51_a[comp]<- AR51( etude , i:( i+(signal -1)))
    cat( i )
171 }
173 #-----#
    #-----#
175

```

```

#METHODE far
177 comp<-0
179 for(i in tmp ){
comp<-comp+1
181 etude<-generateur2(base , signal , i)

183
date$far_s[comp]<- far_1(donnees [,2] , base , i:( i+(signal -1)))
185 date$far_a[comp]<- far_1(donnees [,2] , etude , i:( i+(signal -1)))
cat( i)
187 }

189 #-----#
#-----#
191
#METHODE far modifié
193
comp<-0
195 for(i in tmp ){
comp<-comp+1
197 etude<-generateur2(base , signal , i)

199
date$farm_s[comp]<- far_mod(donnees [,2] , base , i:( i+(signal -1)))
201 date$farm_a[comp]<- far_mod(donnees [,2] , etude , i:( i+(signal -1)))
cat( i)
203 }

205 #-----#

```

appel.r

Puis :

```

appel_appel<-function(donnees , signals ,max_signal , methodes , nom_donnees
){
2
  long<-length( donnees [,1])
4  tmp<-c((( long -363)-(max_signal -1)):( long -(max_signal -1)))
date<-donnees[tmp,2]
6  date<-data.frame( date)
  temps<-date
8  comp<-1

10 for(j in signals){
  nom<-c(" date ")
12 temps<-date

14 for(i in methodes) {
16  comp<-comp+1

```

```

18  nom<-c(nom, paste("methode", i, "_base", sep=""))
    nom<-c(nom, paste("methode", i, "_signal", sep=""))
20  r<-appel(donnees[,3], donnees[,j], max_signal, i)
    temps<-cbind(temps, r[,2], r[,3])
22  comp<-comp+1
    cat("\n")
24  }
26
28  colnames(temps)<-nom
    nom3<-paste("D:\\Administrateur\\Mes documents\\nicolas\\nou\\", "
        signal", j, nom_donnees, ".csv", sep="")
30  write.csv2(temps, nom3)
32  }

```

appel_appel.r

C.2.4 Analyse des résultats

```

1  nb.met<-10
    M1<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\resultats_
        finaux\\syn_carc_signal1.csv", sep=";", header=TRUE)
3  M2<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\resultats_
        finaux\\syn_carc_signal3.csv", sep=";", header=TRUE)
    M3<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\resultats_
        finaux\\syn_carc_signal6.csv", sep=";", header=TRUE)
5  M4<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\resultats_
        finaux\\syn_carc_signal9.csv", sep=";", header=TRUE)

7  M5<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\resultats_
        finaux\\syn_carc_signal12.csv", sep=";", header=TRUE)
    M6<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\resultats_
        finaux\\syn_carc_signal14.csv", sep=";", header=TRUE)
9  M7<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\resultats_
        finaux\\syn_carc_signal14.csv", sep=";", header=TRUE)
    M8<-read.csv2("D:\\Administrateur\\Mes documents\\nicolas\\resultats_
        finaux\\syn_carc_signal5.csv", sep=";", header=TRUE)
11
    resultat<-c(rep("signal 1", nb.met), rep("signal 2", nb.met), rep("signal
        3", nb.met), rep("signal 4", nb.met), rep("signal 5", nb.met), rep("
        signal 6", nb.met), rep("signal 7", nb.met), rep("signal 8", nb.met))
13  signalitique<-rep(c("M1", "M1-a", "M2", "M2-a", "M3", "M3-a", "Arima 5 365"
        , "Arima 1 365", "Arima 5 100", "Farrington"), 8)
    resultat<-data.frame(resultat)
15  resultat[,2] <- signalitique

17  centre<-function(x) {(x-mean(x))/sd(x)}

```

```

19 for(i in 2:3){
  resultat[,1+i]<-c(centre(M1[,i]),centre(M2[,i]),centre(M3[,i]),centre
    (M4[,i]),centre(M5[,i]),centre(M6[,i]),centre(M7[,i]),centre(M8[,i]
    )))
21 }
  for(i in 2:5){
23 resultat[,1+i]<-c((M1[,i]),(M2[,i]),(M3[,i]),(M4[,i]),(M5[,i]),(M6[,i]
    ),(M7[,i]),(M8[,i]))
  }
25
27
  colnames(resultat) <- c("Signal", "Methode", "Spécificité", "
    Sensibilité", "Moyenne", "Ecarttype")
29 resultat[1:10,5]<-1
  resultat[1:10,6]<-0
31
  p<-1
33
35 for(p in 0:7){
37 x11()
  vect<-(1+10*p):(p*10+10)
39
  color<-rainbow(11)[c(2,3,4,6,7,8,9,10,11,1)]
41 plot(resultat$Spécificité[vect],resultat$Sensibilité[vect],cex=1/(1+
    exp(resultat$Moyenne[vect])/max(exp(resultat$Moyenne[vect])))*6,
    col=color,pch=16,xlab=paste("Spécificité ——— Signal ",p+1,sep=
    ""),ylab="Sensibilité")
  if(p != 0){
43 par(new=TRUE)
45 plot(resultat$Spécificité[vect],resultat$Sensibilité[vect],cex=1/(1+
    exp(resultat$Moyenne[vect])/max(exp(resultat$Moyenne[vect])))*3,
    col=grey((resultat$Ecarttype[vect]-min(resultat$Ecarttype[vect]))/
    max(resultat$Ecarttype[vect]-min(resultat$Ecarttype[vect])),pch
    =16,xlab=paste("Spécificité ——— Signal ",p+1,sep=""),ylab="
    Sensibilité")
  }
47
  segments(mean(resultat$Spécificité[vect]),mean(resultat$Sensibilité[
    vect]),mean(resultat$Spécificité[vect]),120,col="red")
49 segments(mean(resultat$Spécificité[vect]),mean(resultat$Sensibilité[
    vect]),120,mean(resultat$Sensibilité[vect]),col="red")
  abline(h=mean(resultat$Sensibilité[vect]),col="red",lty=2)
51 abline(v=mean(resultat$Spécificité[vect]),col="red",lty=2)

53 j<-0.75
  jj<-0.255
55

```



```

segments(quantile((resultat$Spécificité [vect]), j), quantile((resultat$
  Sensibilité [vect]), jj), quantile((resultat$Spécificité [vect]), j)
  ,120, col="blue")
57 segments(quantile((resultat$Spécificité [vect]), j), quantile((resultat$
  Sensibilité [vect]), jj), 120, quantile((resultat$Sensibilité [vect]),
  jj), col="blue")
abline(h=quantile(resultat$Sensibilité [vect], jj), col="blue", lty=2)
59 abline(v=quantile(resultat$Spécificité [vect], j), col="blue", lty=2)

61 j<-0.255
  jj<-0.75
63
segments(quantile((resultat$Spécificité [vect]), j), quantile((resultat$
  Sensibilité [vect]), jj), quantile((resultat$Spécificité [vect]), j)
  ,120, col="green")
65 segments(quantile((resultat$Spécificité [vect]), j), quantile((resultat$
  Sensibilité [vect]), jj), 120, quantile((resultat$Sensibilité [vect]),
  jj), col="green")
abline(h=quantile(resultat$Sensibilité [vect], jj), col="green", lty=2)
67 abline(v=quantile(resultat$Spécificité [vect], j), col="green", lty=2)

69 #segments(mean(resultat$Sensibilité [vect]), mean(resultat$Sensibilité [
  vect]), mean(resultat$Sensibilité [vect])+100, mean(resultat$
  Sensibilité [vect])+100)
legend("bottomleft", legend=as.character(resultat [vect, 2]), col=color,
  lty=1, pch=16)
71
73
}

```

dessin.R

C.3 Mise en routine

C.3.1 Amélioration de la qualité des données

```

2 remp<-function( ville , endroit , cond){
  chemin_des_donnees<- "Z:\\\\SURSAUD\\\\SURV_NON_SPEC\\\\Resultat_
  syndromique\\\\Data\\\\SRVA_data_"
4  donnees<-read.csv(paste(chemin_des_donnees, ville, ".csv", sep=""),
  sep=";")
6
  etude<-donnees[, cond]
8
  erreur<-0
10 while(is.na(etude[length(etude)])) {

```

```
12  erreur<-erreur+1
    etude<-etude[-length(etude)]
    }
14
16  v<-0
18  for( i in 1:length(etude)){
    if(is.na(etude[i])){
20  for(f in 1:50){
    if(is.na(etude[i+f])){
22  v<-v+1
    }
24  else{
    v<-f
26  break()
    }
28  }
30  for(u in 0:(v-1)){
    etude[i+u]<-round((etude[i-1]+etude[i+v])/2)
32
34  }
    }
36  }
38
40  m1<-mean(etude[1:endroit])
    m2<-mean(etude[(endroit+1):length(etude)])
    m3<-round(m2-m1)
42
44  if(ville=="MENDE"){
    for(i in 1:endroit){
46  if(etude[i]<=0 && cond==6){
    etude[i]<-round(m2)
    }
48  if(etude[i]>quantile(etude,0.92)){
    etude[i]<-round(quantile(etude,0.92))
50  }
    if(etude[i]<quantile(etude,0.07)){
52  etude[i]<-round(quantile(etude,0.07))
    }
54  }
    }
56  etude<-as.numeric(etude)
    etude[1:endroit]<-etude[1:endroit]+m3
58
60  etude<-c(etude,rep(NA,erreur))
62
    donnees[,cond]<-etude
```

```

        write.csv2(donnees , paste(chemin_des_donnees , ville , ".csv" , sep="") ,
        row.names=FALSE)
64
66 }
68 remp("CARCASSONNE" , 1328 , 5)
    remp("CARCASSONNE" , 400 , 6)
70 remp("CARCASSONNE" , 1480 , 6)
    remp("MENDE" , 335 , 6)
72 remp("MENDE" , 1490 , 5)
    remp("MONTPELLIER" , 1008 , 5)
74 remp("MONTPELLIER" , 1008 , 2)
    remp("MONTPELLIER" , 1008 , 3)
76 remp("MONTPELLIER" , 1008 , 4)
    remp("PERPIGNAN" , 1549 , 5)
78 remp("BAGNOLS" , 1570 , 5)
    remp("NIMES" , 1392 , 6)

```

remplacement.R

C.3.2 Partie syndromique

```

1 #####
  #A CHANGER LORS DU DEPLACEMENT DES FICHIERS SOURCES !!#
3 #####
5
7 #####
  #Ceci n'a pour but que de donner les endroits où sont
  #stockés les données et où les résultats doivent être
  #placés. Il suffit de changer les chemins d'accès ci-
  #dessous pour modifier le code tout entier
11 #####
13
    chemin_des_donnees<- "Z:\\SURSAUD\\SURV_NON_SPEC\\Resultat_
      syndromique\\Data\\SRVA_data_"
15 chemin_graphiques <- "Z:\\SURSAUD\\SURV_NON_SPEC\\Resultat_
      syndromique\\Resultats\\graphiques_1.pdf"
    chemin_cartes      <- "Z:\\SURSAUD\\SURV_NON_SPEC\\Resultat_
      syndromique\\Resultats\\tableau_1.pdf"
17 chemin_tests       <- "Z:\\SURSAUD\\SURV_NON_SPEC\\Resultat_
      syndromique\\Resultats\\tests_1.pdf"
19
21 #####
  #Ici, l'on choisit les villes sur lesquelles nous voulons
  #effectuer la surveillance.
23 #####

```

```

25 villes<-c("MENDE", "MONTPELLIER", "ALES", "BAGNOLS", "BEZIERS", "
    CARCASSONNE", "NARBONNE", "NIMES", "PERPIGNAN", "SETE")
27 #villes<-c("MENDE")

29 #####
31 #Attention, la librairie biosurv nécessite l'installation de
31 #la librairie surveillance.
33 #####

35 library(biosurv)

37 #####
37 #Ici, la fonction servant à générer les différents graphiques;
37 #Nous expliquons maintenant l'intérêt des différentes options :
39 #ville : vecteurs des villes à surveiller
39 #histo : sur combien de jours faire la surveillance
41 #cond : quelle colonne de la matrice doit-on analyser ?
41 #color : graphique en gris (0) ou en couleur (1) ?
43 #####

45

47 surveillance<-function(ville, histo, cond, color) {
49
51 #####
51 #layout sert à définir l'espace graphique. Il sera donc de
51 #la forme :
53 # 1 1
53 # 1 1
55 # 1 1
55 # 2 3
57 #####

59 layout(matrix(c(1,1,1,2,1,1,1,3),4))

61 #####
63 #On lit les données à partir du chemin indiqué ci-dessus
63 #####

65

67 donnees<-read.csv(paste(chemin_des_donnees, ville, ".csv", sep=""), sep="
    ;")

69 #####
69 #etude représente le vecteur de la matrice auquel nous nous
71 #intéressons.
71 #####

73

```

```

etude<-donnees[,cond]
75
v<-0
77 g<-0

79 #####
#Ici, une procédure pour vérifier que la dernière donnée disponible
81 #ne soit pas manquante. si tel est le cas, erreur est mis à 1
#et toutes les dernières lignes avec données manquantes sont
supprimées.
83 #####

85 erreur<-0

87 while(is.na(etude[length(etude)])) {
    erreur<-1
89    etude<-etude[-length(etude)]
    }

91
#####
93 #Ici, on définit le message que l'on veut donner à l'utilisateur
#en fonction de l'issue de la procédure précédente
95 #####

97 message<-"L'algorithme s'est déroulé normalement. \n"

99 if(erreur==1) {
    message<-"Il manque le ou les derniers jours. \n"
101 }

103
#####
105 #Ici, on remplace les données manquantes. On trace la droite
#entre les deux valeurs non manquantes autour des valeurs manquantes
cibles
107 #et on répartit régulièrement les valeurs manquantes.
#####
109

111 for( i in 1:length(etude)) {
    if(is.na(etude[i])) {
113        for(f in 1:50) {
            if(is.na(etude[i+f])) {
115                v<-v+1
            }

117            else {
                v<-f
119                break()
            }
121        }
    }

123    for(u in 0:(v-1)) {

```

```

125         etude [ i+u ] <- round ( ( etude [ i-1 ] + etude [ i+v ] ) / 2 )
126             }
127     }
129 #####
130 #####
131 ##### METHODE ARIMA #####
132 #####
133 #####
135 #####
136 #Puisque nous voulons faire une transformation logarithmique
137 #nous rajoutons 1 à toutes nos données afin d'éviter log(0)
138 #####
139
140 etude <- etude + 1
141
142 #####
143 #seuil correspond à la borne supérieure de l'intervale de confiance à
144   1%
145 #seuil5 correspond à la borne supérieure de l'intervale de confiance à
146   5%
147 #temp est la période sur laquelle nous effectuons la surveillance
148 #couleur contient les couleurs pour la méthode à 1%
149 #couleur5 contient les couleurs pour la méthode à 5%
150 #alarme contient les alarmes pour la méthode à 1%
151 #alarme5 contient les alarmes pour la méthode à 5%
152 #####
153
154 seuil <- 0
155 seuil5 <- 0
156 tmp <- ( length ( etude ) - histo + 1 ) : length ( etude )
157 couleur <- ( "green " )
158 couleur5 <- ( "green " )
159 alarme <- 0
160 alarme5 <- 0
161
162 #####
163 #Ici commencent les calculs pour la méthodes Arima :
164
165 #la boucle se déroule naturellement sur la période voulue ,
166 #c'est à dire tmp !
167 #####
168
169 for ( i in tmp ) {
170
171     u <- 0
172     start <- i

```

```

175 x0<-etude
177 #On prend pour historique pour la méthode un an ou toutes les
177 #données si on a moins d'un an d'historique.
179
181 x0<-x0[max(1,(i-365-1)):(i-1)]
181 #Transformation logarithmique des données puis transformation en
181 série
183 #chronologique
185 x0<-log(x0)
185 x<-ts(x0,freq=7)
187
187 #On applique la procédure ARIMA
189
189 a<-arima(x,order=c(1,1,1),seasonal=list(order=c(0,1,1),period=7))
191
191 #On fait une prédiction pour le dernier jour
193 #pd : prédiction
193 #pdse : écart-type de la prédiction
195
195 prediction<-predict(a,n.ahead=1)
197 pd<-prediction$pred
197 pdse<-prediction$se
199
199 #On calcule les alarmes et les couleurs en conséquence
201 #pour la méthode à 1%
203 if(exp(pd[1]+qnorm(0.99)*pdse[1])>etude[i]) {
203     alarme<-c(alarme,0)
205     couleur<-c(couleur,"green")
207     }
207 else {
207     couleur<-c(couleur,"red")
209     alarme<-c(alarme,1)
209     }
211
213 #On répète le calcul pour vérifier si l'on n'est pas
213 #en dessous du seuil.
215
215 if(exp(pd[1]-qnorm(0.99)*pdse[1])>etude[i]) {
217
217     alarme<-alarme[-length(alarme)]
219     alarme<-c(alarme,-1)
219     couleur<-couleur[-length(couleur)]
221     couleur<-c(couleur,"blue")
221     }
223
223 #Mêmes calculs pour 5%

```

```

225
227  if(exp(pd[1]+qnorm(0.95)*pdse[1])>etude[i]) {
229    alarme5<-c(alarme5,0)
231    couleur5<-c(couleur5,"green")
233  }
235
237  else{
239    couleur5<-c(couleur5,"red")
241    alarme5<-c(alarme5,1)
243  }
245  }      #Fin de la boucle
247
249  #On enlève les valeurs d'initialisation
251  seuil5<-seuil5[-1]
253  alarme5<-alarme5[-1]
255  seuil<-seuil[-1]
257  alarme<-alarme[-1]
259  #On calcule les abscisses des points servant à tracer la ligne
261  #des seuils
263  repere<-0
265  for(i in 1:(histo-1)) {
267    repere<-c(repere,i,i)
269  }
271  repere<-c(repere,histo)
273  #On calcule quelques indicateurs
275  y<-max(seuil)
277  z<-max(y,max(etude[tmp]))
279  #On bascule l'affichage en gris si tel est le choix de l'utilisateur
281  couleur<-couleur[-1]
283  if(color==0){
285    couleur<-"grey"

```



```

277     }
279 #Il y a trois possibilités d'avertissements pour l'utilisateur :
280 #1- il y a une alarme : ie le seuil à 1% est dépassé
281 #2- on est entre les deux seuil
282 #3- aucune alerte
283
284 #Ici on genre les cas 2 et 3
285
286 avertissement<-" , il n'y a aucune alerte à signaler "
287 if(alarme5[length(alarme5)]==1) {
288     avertissement<-" , avertissement : le seuil au risque 5% est dépassé
289     "
290     }
291 #Maintenant que tous nos calculs sont effectués , il faut revenir aux
292 données de base
293 #on baisse les seuils et les données étudiées de 1.
294
295 seuil<-seuil-1
296 seuil5<-seuil5-1
297 etude2<-etude-1
298
299 #On enregistre les résidus de notre modèle
300 #On enchaîne avec un test de normalité
301 #Puis par le test d'indépendance
302
303 res<-residuals(a)
304 test1<-ks.test(res , "pnorm" ,mean=mean(res) ,sd=sd(res))
305 p.test1<-test1$p
306 p.test2<-Box.test(res)$p.value
307
308 #On passe maintenant dans la partie graphique a proprement parler
309 #On sépare les cas alarme/non-alarme
310 #On sépare les cas selon les conditions (cond)
311
312 length(etude)->n
313
314 if(alarme[length(alarme)]==1){
315     if(cond==2) {
316         barplot(etude2[tmp] ,ylim=c(0 ,z*1.2) ,space=0,offset=0,names.arg=
317             donnees[tmp,1] ,axisname=TRUE,col=couleur ,ylab="Nombre de
318             passages aux urgences" ,cex.axis=1.2)
319         lines(repere ,seuil ,lty=2,col="red")
320         lines(repere ,seuil5 ,lty=2,col="blue")
321         title(paste("Passages aux urgences de " ,ville , " du " ,donnees[
322             length(donnees[,1])-histo+1,1] , " au " , donnees[length(etude)
323             ,1] ,"\n Borne supérieure obtenue avec une méthode ARIMA avec
324             un seuil alpha de 1% et 5%" ,sep=""))
325     }
326 }

```

```

323   if(cond==3) {
      barplot(etude2[tmp], ylim=c(0, z*1.2), space=0, offset=0, names.arg=
        donnees[tmp,1], axisname=TRUE, col=couleur, ylab="Nombre de
        passages aux urgences de plus de 75 ans", cex.axis=1.2)
325   lines(repere, seuil, lty=2, col="red")
      lines(repere, seuil5, lty=2, col="blue")
      title(paste("Passages aux urgences de ", ville, " pour les plus
        de 75 ans du ", donnees[length(donnees[,1])-histo+1,1], " au "
        , donnees[length(etude),1], "\n Borne supérieure obtenue avec
        une méthode ARIMA avec un seuil alpha de 1% et 5%", sep=""))
327   }

329   text(histo/4, z*1.1, paste(message, "Le ", donnees[length(etude),1],
    ", il y a une alerte. Le dépassement est de ", round((etude[
    length(etude)]/exp(pd[1]) - 1)*100), "% \n Le nombre de passage
    aux urgences est de :", etude2[length(etude)], "\n Le maximum
    constaté les ", histo, " derniers jours est de :", max(etude2[(n
    -histo+1):n]), sep=" " ), cex=1.3)
    text(3*histo/4, z*1.1, paste("Test de normalité Kolmogorov-Smirnov
    (p-valeur) : ", round(p.test1, digits=3), "\n Test d'indépendance
    de Box-Ljung (p-valeur) : ", round(p.test2, digits=3), sep=" " ),
    cex=1.3)
331   }
333
335   else{
      if(cond==2) {
        barplot(etude2[tmp], ylim=c(0, z*1.2), space=0, offset=0, names.arg=
          donnees[tmp,1], axisname=TRUE, col=couleur, ylab="Nombre de
          passages aux urgences", cex.axis=1.2)
337   lines(repere, seuil, lty=2, col="red")
          lines(repere, seuil5, lty=2, col="blue")
339   title(paste("Passages aux urgences de ", ville, " du ", donnees[
            length(donnees[,1])-histo+1,1], " au ", donnees[length(etude)
            ,1], "\n Borne supérieure obtenue avec une méthode ARIMA avec
            un seuil alpha de 1% et 5%", sep=""))
          }
341
          if(cond==3) {
343   barplot(etude2[tmp], ylim=c(0, z*1.2), space=0, offset=0, names.arg=
            donnees[tmp,1], axisname=TRUE, col=couleur, ylab="Nombre de
            passages aux urgences des plus de 75 ans", cex.axis=1.2)
            lines(repere, seuil, lty=2, col="red")
345   lines(repere, seuil5, lty=2, col="blue")
            title(paste("Passages aux urgences de ", ville, " pour les plus
              de 75 ans du ", donnees[length(donnees[,1])-histo+1,1], " au "
              , donnees[length(etude),1], "\n Borne supérieure obtenue avec
              une méthode ARIMA avec un seuil alpha de 1% et 5%", sep=""))
347   }

349   text(histo/4, z*1.1, paste(message, "Le ", donnees[length(etude),1],
    avertissement, " \n Le nombre de passage aux urgences est de :",

```

```

    etude2[length(etude)], "\n Le maximum constaté les ", histo, "
    derniers jours est de : ", max(etude2[(n-histo+1):n]), sep=" " ),
    cex=1.3)
text(3*histo/4, z*1.1, paste("Test de normalité de Kolmogorov-Smirnov
    (p-valeur) : ", round(p.test1, digits=3), "\n Test d'indépendance
    de Box-Ljung (p-valeur) : ", round(p.test2, digits=3), sep=" " ), cex
351   =1.3)
    }

353 title(sub="Borne supérieure de l'intervalle de prédiction obtenu avec
    une méthode ARIMA de paramètre (1,1,1) et de saisonnalité sur 7
    jours avec paramètre (0,1,1) avec historique d'un an.", cex.sub
    =1.2)

355 #On trace pour diagnostics les fonctions d'autocorrélation et d'
    autocorrélation partielle

357 acf(residuals(a), main="Fonction d'autocorrélation des résidus du
    modèle")
pacf(residuals(a), main="Fonction d'autocorrélation partielle des
    résidus du modèle")
359

#On remet notre vecteur d'étude comme au départ
361 etude<-etude2
363 #####
365 #####

367 #On récupère les dates, puis on créé un Disprog object du package
    surveillance, puis on lance l'algo farrington
    #Attention : la période de référence a été changée de l'algorithmme en
    appelant le package biosurv
369
    vect_date<-donnees[,1]
371 obj<-create.disProg(week=as.Date(vect_date, format="%d/%m/%Y"),
    observed=etude, state=rep(0, length(etude)), freq=365, epochAsDate=
    TRUE, start=c(2005,303))
    res<-algo.farrington(obj, control=list(range=(length(etude)-59):length
    (etude), b=3, w=3, alpha=0.01))
373
    limite.sup<-0
375 comp<-0

377 #On change les limites supérieures : on demande à les mettre au
    minimum à deux :
    #il n'y a pas de sens à faire sonner une alarme pour un cas !
379 #Il faut alors changer le vecteur d'alarme en conséquence
    #On calcule tout de suite le vecteur limite.sup pour la ligne de
    seuil
381 for(i in res$upperbound){

```

```

383   comp<-comp+1
385   t<-i
      if(t==0 || t==1){
387     t<-1
      res$upperbound [comp]<-2
389     }

391   if (res$upperbound [comp]<=etude [ res$control$range [comp]]) {
      res$alarm [comp]<-1
393     }

      else {
395     res$alarm [comp]<-0
      }

397   t<-res$upperbound [comp]
399   limite .sup<-c(limite .sup , t , t)
      }

401

403 limite .sup<-limite .sup [-1]

405 #Partie graphique : même chose que pour ARIMA

407 if(cond==2) {
      barplot (etude [tmp] , ylim=c (0 ,max(limite .sup)*1.2) , space=0 ,offset=0 ,
      names .arg=donnees [tmp,1] , axisname=TRUE , col=couleur , ylab="Nombre
      de passages aux urgences " , cex .axis=1.2)
409   lines (repere , limite .sup , lty=2 , col="red ")
      title (paste ("Passages aux urgences de " , ville , " du " , donnees [
      length (donnees [,1]) - histo +1,1] , " au " , donnees [length (etude) ,1] ,
      "\n Borne supérieure obtenue grâce à l'algorithme de Farrington
      avec un seuil alpha de 1% " , sep="") , sub="Référence : A
      statistical Algorithm for the Early Detection of Outbreaks of
      Infectious Disease / C.P. FARRINGTON \n Cette méthode tient
      particulièrement compte de l'effet jour de semaine ainsi que de
      la saisonnalité liée aux mois.")
411   }

413 if(cond==3) {
      barplot (etude [tmp] , ylim=c (0 ,max(limite .sup)*1.2) , space=0 ,offset=0 ,
      names .arg=donnees [tmp,1] , axisname=TRUE , col=couleur , ylab="Nombre
      de passages aux urgences pour les plus de 75 ans" , cex .axis=1.2)
415   lines (repere , limite .sup , lty=2 , col="red ")
      title (paste ("Passages aux urgences de " , ville , " pour les plus de
      75 ans du " , donnees [length (donnees [,1]) - histo +1,1] , " au " ,
      donnees [length (etude) ,1] , "\n Borne supérieure obtenue grâce à l'
      algorithme de Farrington avec un seuil alpha de 1% " , sep="") , sub
      ="Référence : A statistical Algorithm for the Early Detection of
      Outbreaks of Infectious Disease / C.P. FARRINGTON \n Cette
      méthode tient particulièrement compte de l'effet jour de semaine
      ainsi que de la saisonnalité liée aux mois.")

```

```

417     }
419
421 #On finit en tracant des graphes supplémentaires
423 RESIDUS<-res$disProgObj$observed[ res$control$range]-res$control$pd
      [,2]
qqnorm(RESIDUS,main="Diagramme quantile-quantile pour la loi normale"
      )
425 qqline(RESIDUS)
plot(RESIDUS,main="Graphique des résidus")
427 abline(0,0)
429
431
433 #####
435 #####
437 #####
439 sonnette<-0
      residus<-0
441 cusum.lim<-0
      cusum.lim2<-0
443
      #Boucle sur la période de surveillance voulue
445 #La limite est juste le quantile de la loi de poisson
      #On prend pour paramètre la moyenne sur les observation des 7
      derniers
447 #jours si on saute les deux derniers.
449 for(i in tmp) {
      z<-mean(etude[(i-2):(i-9)])
451 residus<-c(residus ,etude[i]-z)
      t<-qpois(0.99 ,z)
453 t2<-qpois(0.995 ,z)
455 if(t>etude[i]) {
      sonnette<-c(sonnette ,0)
457     }
      else{
459 sonnette<-c(sonnette ,1)
      }
461
      if(t2<=etude[i]){
463 sonnette [length(sonnette)]<- -1
      }
465

```

```

cusum.lim<-c(cusum.lim,t,t)
467 cusum.lim2<-c(cusum.lim2,t2,t2)
469     }
471
472 cusum.lim<-cusum.lim[-1]
473 cusum.lim2<-cusum.lim2[-1]
474 sonnette<-sonnette[-1]
475
476 #Partie graphique
477
478 if(cond==2){
479   barplot(etude[tmp],ylim=c(0,max(cusum.lim)*1.2),space=0,offset=0,
           names.arg=donnees[tmp,1],axisname=TRUE,col=couleur,ylab="Nombre
           de passages aux urgences",cex.axis=1.2)
           lines(repere,cusum.lim,lty=2,col="blue")
481   lines(repere,cusum.lim2,lty=2,col="red")
           title(paste("Passages aux urgences de ",ville," du ",donnees[
           length(donnees[,1])-histo+1,1]," au ",donnees[length(etude),1],
           "\n Borne supérieure obtenue grâce à l'algorithme Cusum avec un
           seuil alpha de 1/2% et 1% ",sep=""))#,sub="Insérer peut-être une
           référence")
483 }
484 if(cond==3){
485   barplot(etude[tmp],ylim=c(0,max(cusum.lim)*1.2),space=0,offset=0,
           names.arg=donnees[tmp,1],axisname=TRUE,col=couleur,ylab="Nombre
           de passages aux urgences pour les plus de 75 ans",cex.axis=1.2)
           lines(repere,cusum.lim,lty=2,col="blue")
487   lines(repere,cusum.lim2,lty=2,col="red")
           title(paste("Passages aux urgences de ",ville," pour les plus de
           75 ans du ",donnees[length(donnees[,1])-histo+1,1]," au ",
           donnees[length(etude),1],"\n Borne supérieure obtenue grâce à l'
           algorithme Cusum avec un seuil alpha de 1/2% et 1% ",sep=""))#,
           sub="Insérer peut-être une référence")
489 }
491
492 plot(residus)
493 abline(0,0)
494 hist(etude[tmp])
495
496 #Partie pour construire le "+"
497
498 etude[seq(from=length(etude),by=-1,length.out=6)]->w
499 etude[seq(from=length(etude)-7,by=-1,length.out=6)]->ww
501 compt<-0
502 for(i in 1:6) {
503   if((w[i]-ww[i])>0) {
504     compt<-compt+1
505   }

```

```

507     }
509
510 #On renvoie un vecteur pour sortie de la fonction qui contient, dans
511   l'ordre :
512 #La dernière observation non manquante
513 #Alarme arima 1 % -1 ou 0 ou 1
514 #Alarme arima 5 % 0 ou 1
515 #Alarme farrington 5% 0 ou 1
516 #La dernière donnée était-elle manquante 0 ou 1
517 #Combien de jours successifs en hausse sur 6 jours
518 #Alarme Cusum -1 ou 0 ou 1
519 c<-c(etude[length(etude)], alarme[length(alarme)], alarme5[length(
520   alarme5)], res$alarm[length(res$alarm)], erreur, compt, sonnette[
521   length(sonnette)], alarme[length(alarme)-1], alarme5[length(alarme5)
522   -1], res$alarm[length(res$alarm)-1], sonnette[length(sonnette)-1])
523 }
524
525 #PARTIE GRAPHIQUES#####
526 #Il suffit d'appeler plusieurs fois la fonction créée ci-dessus !
527 pdf(file=paste(chemin_graphiques, sep=" "), width=13.5, height=10)
528
529
530
531 nb.villes<-length(villes)
532 comp<-0
533 a<-matrix(rep(0, nb.villes*11), nb.villes)
534 b<-matrix(rep(0, nb.villes*11), nb.villes)
535 for( i in villes ){
536   comp<-comp+1
537   for(j in 2:3){
538     if(j==2){
539       a[comp,]<-surveillance(i,60,j,0)
540     }
541     if(j==3){
542       b[comp,]<-surveillance(i,60,j,0)
543     }
544   }
545 }
546 }
547 }
548 }
549 dev.off()
550
551
552
553 #PARTIE TABLEAU#####

```

```

555 pdf( file=paste(chemin_cartes , sep=" " ) , width=13.5 , height=10)
557
559 plot.new()
plot.window(xlim=c( -0.2 , 1.2 ) , ylim=c( 0 , 1 ) )

561 nb.lignes<-nb.villes+1
nb.col<-9
563 segments(1/nb.col , 1 , 1 , 1 , lwd=3)
segments(1/nb.col , 1-1/(nb.lignes*2) , 1 , 1-1/(nb.lignes*2) , lwd=3)
565 segments(-0.2 , 0 , -0.2 , 1-1/nb.lignes , lwd=3)
i<-1
567 segments(-0.2 , 1-i/nb.lignes , 1 , 1-i/nb.lignes , lwd=3)

569 for( i in 1:nb.lignes ){
segments(-0.2 , 1-i/nb.lignes , 1 , 1-i/nb.lignes )
571 }

573 for( i in 1:nb.col ){
segments(i/nb.col , 0 , i/nb.col , 1-1/(nb.lignes*2) )
575 }

577
segments(1/nb.col , 0 , 1/nb.col , 1 , lwd=3)
579 segments(((nb.col+1)/2)/nb.col , 0 , ((nb.col+1)/2)/nb.col , 1 , lwd=3)
segments(-0.2 , 0 , 1 , 0 , lwd=3)
581
segments(1/nb.col , 0 , 1/nb.col , 1 , lwd=3)
583 segments(((nb.col+1)/2)/nb.col , 0 , ((nb.col+1)/2)/nb.col , 1 , lwd=3)
segments(1 , 0 , 1 , 1 , lwd=3)
585

587
text(3/nb.col , 1-1/(nb.lignes*4) , "Primo-passages " )
589 text(7/nb.col , 1-1/(nb.lignes*4) , "Primo-passages +75ans ")
text(1.5/nb.col , 1-3/(nb.lignes*4) , "Nb \n Passages ")
591 text(2.5/nb.col , 1-3/(nb.lignes*4) , "ARIMA ")
text(3.5/nb.col , 1-3/(nb.lignes*4) , "Farrington ")
593 text(4.5/nb.col , 1-3/(nb.lignes*4) , "Cusum \n Poisson ")
text(5.5/nb.col , 1-3/(nb.lignes*4) , "Nb \n Passages ")
595 text(6.5/nb.col , 1-3/(nb.lignes*4) , "ARIMA ")
text(7.5/nb.col , 1-3/(nb.lignes*4) , "Farrington ")
597 text(8.5/nb.col , 1-3/(nb.lignes*4) , "Cusum \n Poisson ")

599

601

603
for( i in 1:nb.villes ){
605 text((1/(nb.col)-0.2)/2 , 1-1/(nb.lignes*2)-i*1/nb.lignes , villes [ i ])

```



```

}
607 legend(1.01,0.85,pch=c(""," "," "," "," "+" ),c("Alerte : \n Depassement
      \n du seuil 1% \n (Far. et Arima) \n 1/2% CUSUM ", "Avertissement
      : \n Dépassement \n seuil 5% \n (Arima) \n 1/2% CUSUM", "Aucun
      seuil \n dépassé", "Valeur(s) \n Manquante(s)", "Valeur \n faible
      (1)", "5 jours /6 \n en hausse (2)"),fill=c("red","orange","green"
      ,"grey","blue","white"),border=c("black","black","black","black",
      "black","white"),cex=0.9)
609
611
613 #Remplissage du tableau
615 for( i in 1:nb.villes){
617 text(1.5/nb.col,1-(i+1/2)/nb.lignes ,a[i,1])
619 text(5.5/nb.col,1-(i+1/2)/nb.lignes ,b[i,1])
621 }
623 for( i in 1:nb.villes){
625 if(a[i,3]==0){
627 rect(2/nb.col,1-(i+1)/nb.lignes ,3/nb.col,1-(i)/nb.lignes ,col="green")
629 }
631 if(b[i,3]==0){
633 rect(6/nb.col,1-(i+1)/nb.lignes ,7/nb.col,1-(i)/nb.lignes ,col="green")
635 }
637 if(a[i,3]==1){
639 rect(2/nb.col,1-(i+1)/nb.lignes ,3/nb.col,1-(i)/nb.lignes ,col="orange"
641 )
643 }
645 if(b[i,3]==1){
647 rect(6/nb.col,1-(i+1)/nb.lignes ,7/nb.col,1-(i)/nb.lignes ,col="orange"
649 )
651 }
653 if(a[i,2]==1){
655 rect(2/nb.col,1-(i+1)/nb.lignes ,3/nb.col,1-(i)/nb.lignes ,col="red")
657 }
659 if(b[i,2]==1){
661 rect(6/nb.col,1-(i+1)/nb.lignes ,7/nb.col,1-(i)/nb.lignes ,col="red")
663 }
665 if(a[i,4]==0){
667 rect(3/nb.col,1-(i+1)/nb.lignes ,4/nb.col,1-(i)/nb.lignes ,col="green")
669 }
671 else
673 {
675 rect(3/nb.col,1-(i+1)/nb.lignes ,4/nb.col,1-(i)/nb.lignes ,col="red")
677 }
679 if(b[i,4]==0){

```

```
651 rect(7/nb.col,1-(i+1)/nb.lignes,8/nb.col,1-(i)/nb.lignes,col="green")
652 }
653 else
654 {
655   rect(7/nb.col,1-(i+1)/nb.lignes,8/nb.col,1-(i)/nb.lignes,col="red")
656 }
657 if(a[i,2]==-1){
658   rect(2/nb.col,1-(i+1)/nb.lignes,3/nb.col,1-(i)/nb.lignes,col="blue")
659 }
660 if(b[i,2]==-1){
661   rect(6/nb.col,1-(i+1)/nb.lignes,7/nb.col,1-(i)/nb.lignes,col="blue")
662 }
663
664 if(a[i,7]==1){
665   rect(4/nb.col,1-(i+1)/nb.lignes,5/nb.col,1-(i)/nb.lignes,col="orange"
666   )
667 }
668
669 else{
670   rect(4/nb.col,1-(i+1)/nb.lignes,5/nb.col,1-(i)/nb.lignes,col="green")
671 }
672
673 if(a[i,7]==-1){
674   rect(4/nb.col,1-(i+1)/nb.lignes,5/nb.col,1-(i)/nb.lignes,col="red")
675 }
676
677 if(b[i,7]==1){
678   rect(8/nb.col,1-(i+1)/nb.lignes,9/nb.col,1-(i)/nb.lignes,col="orange"
679   )
680 }
681
682 else{
683   rect(8/nb.col,1-(i+1)/nb.lignes,9/nb.col,1-(i)/nb.lignes,col="green")
684 }
685
686 if(b[i,7]==-1){
687   rect(8/nb.col,1-(i+1)/nb.lignes,9/nb.col,1-(i)/nb.lignes,col="red")
688 }
689
690 }
691
692
693
694
695
696
697 for( i in 1:nb.villes){
698
699   if(a[i,6]>4){
```

```

points(2.5/nb.col,1-(i+1/2)/nb.lignes ,pch="+",cex=3)
701 points(3.5/nb.col,1-(i+1/2)/nb.lignes ,pch="+",cex=3)
points(4.5/nb.col,1-(i+1/2)/nb.lignes ,pch="+",cex=3)
703 }
if(b[i,6]>4){
705 points(6.5/nb.col,1-(i+1/2)/nb.lignes ,pch="+",cex=3)
points(7.5/nb.col,1-(i+1/2)/nb.lignes ,pch="+",cex=3)
707 points(8.5/nb.col,1-(i+1/2)/nb.lignes ,pch="+",cex=3)
}
709
}
711
713 for( i in 1:nb.villes){
715 if(a[i,5]==1){
rect(2/nb.col,1-(i+1)/nb.lignes ,3/nb.col,1-(i)/nb.lignes ,col="grey")
717 rect(3/nb.col,1-(i+1)/nb.lignes ,4/nb.col,1-(i)/nb.lignes ,col="grey")
rect(4/nb.col,1-(i+1)/nb.lignes ,5/nb.col,1-(i)/nb.lignes ,col="grey")
719 }
if(b[i,5]==1){
721 rect(7/nb.col,1-(i+1)/nb.lignes ,8/nb.col,1-(i)/nb.lignes ,col="grey")
rect(6/nb.col,1-(i+1)/nb.lignes ,7/nb.col,1-(i)/nb.lignes ,col="grey")
723 rect(8/nb.col,1-(i+1)/nb.lignes ,9/nb.col,1-(i)/nb.lignes ,col="grey")
}
725
727 }
729
trans<-as.Date("01/01/2001",format="%d/%m/%Y")
731
compteur<-0
733 compteur1<-0
735
for( i in villes){
737 donnees<-read.csv2(paste(chemin_des_donnees,i,".csv",sep=""),sep=";")
trans<-c(trans,as.Date(donnees[length(donnees[,1]),1))
739 }
trans<-trans[-1]
741 transmax<-max(trans)
743 Tr<-rep(1,nb.villes)
for(i in 1:nb.villes){
745 compteur<-compteur+1
if(trans[i]==transmax){
747 Tr[i]<-0
compteur1<-compteur
749 }
}
751

```

```

for(i in 1:nb.villes){
753 if(Tr[i]==1){
rect(2/nb.col,1-(i+1)/nb.lignes,3/nb.col,1-(i)/nb.lignes,col="grey")
755 rect(3/nb.col,1-(i+1)/nb.lignes,4/nb.col,1-(i)/nb.lignes,col="grey")
rect(4/nb.col,1-(i+1)/nb.lignes,5/nb.col,1-(i)/nb.lignes,col="grey")
757 rect(6/nb.col,1-(i+1)/nb.lignes,7/nb.col,1-(i)/nb.lignes,col="grey")
rect(7/nb.col,1-(i+1)/nb.lignes,8/nb.col,1-(i)/nb.lignes,col="grey")
759 rect(8/nb.col,1-(i+1)/nb.lignes,9/nb.col,1-(i)/nb.lignes,col="grey")
}
761 }

763 donnees<-read.csv2(paste(chemin_des_donnees, villes[compteur1], ".csv",
sep=""), sep=";")
date<-donnees[length(donnees[,1]),1]
765 title(paste("Tableau de bord au ",date,sep=""),sub="(1) : Dépassement
de la borne inférieure de l'intervalle de confiance à 99% de la
méthode ARIMA : il se peut qu'il y ait une erreur dans la donnée \
n (2) : 5 jours sur 6 en hausse par rapport au même jour de la
semaine précédente")

767

769 for(j in c(2,3,4,6,7,8)){
for(i in 1:nb.villes){
771 segments((j*4+3)/(4*nb.col),(i-1)/nb.lignes,(j*4+3)/(4*nb.col),(1+4*(
i-1))/(4*nb.lignes))
segments((j*4+3)/(4*nb.col),(1+4*(i-1))/(4*nb.lignes),(j+1)/(nb.col)
,(1+4*(i-1))/(4*nb.lignes))
773 }
775 }

777
y<-3/(4*nb.col)
779 h<-3/(4*nb.lignes)

781 for( i in 1:nb.villes){

783
if(a[i,3+6]==0){
785 rect(2/nb.col+y,1-(i+1)/nb.lignes,3/nb.col,1-(i)/nb.lignes-h,col="
green")
}
787 if(b[i,3+6]==0){
rect(6/nb.col+y,1-(i+1)/nb.lignes,7/nb.col,1-(i)/nb.lignes-h,col="
green")
789 }
if(a[i,3+6]==1){
791 rect(2/nb.col+y,1-(i+1)/nb.lignes,3/nb.col,1-(i)/nb.lignes-h,col="
orange")
}
793 if(b[i,3+6]==1){

```

```

rect(6/nb.col+y,1-(i+1)/nb.lignes,7/nb.col,1-(i)/nb.lignes-h,col="
orange")
795 }
if(a[i,2+6]==1){
797 rect(2/nb.col+y,1-(i+1)/nb.lignes,3/nb.col,1-(i)/nb.lignes-h,col="red
")
}
799 if(b[i,2+6]==1){
rect(6/nb.col+y,1-(i+1)/nb.lignes,7/nb.col,1-(i)/nb.lignes-h,col="red
")
801 }
if(a[i,4+6]==0){
803 rect(3/nb.col+y,1-(i+1)/nb.lignes,4/nb.col,1-(i)/nb.lignes-h,col="
green")
}
805 else
{
807 rect(3/nb.col+y,1-(i+1)/nb.lignes,4/nb.col,1-(i)/nb.lignes-h,col="red
")
}
809 if(b[i,4+6]==0){
rect(7/nb.col+y,1-(i+1)/nb.lignes,8/nb.col,1-(i)/nb.lignes-h,col="
green")
811 }
else
813 {
rect(7/nb.col+y,1-(i+1)/nb.lignes,8/nb.col,1-(i)/nb.lignes-h,col="red
")
815 }
817 if(a[i,2+6]==-1){
rect(2/nb.col+y,1-(i+1)/nb.lignes,3/nb.col,1-(i)/nb.lignes-h,col="
blue")
819 }
if(b[i,2+6]==-1){
821 rect(6/nb.col+y,1-(i+1)/nb.lignes,7/nb.col,1-(i)/nb.lignes-h,col="
blue")
}
823
825 if(a[i,7+4]==1){
rect(4/nb.col+y,1-(i+1)/nb.lignes,5/nb.col,1-(i)/nb.lignes-h,col="
orange")
827 }
829 else{
rect(4/nb.col+y,1-(i+1)/nb.lignes,5/nb.col,1-(i)/nb.lignes-h,col="
green")
831 }
833 if(a[i,7+4]==-1){

```

```

835 rect(4/nb.col+y,1-(i+1)/nb.lignes,5/nb.col,1-(i)/nb.lignes-h,col="red
      ")
837
839 if(b[i,7+4]==1){
      rect(8/nb.col+y,1-(i+1)/nb.lignes,9/nb.col,1-(i)/nb.lignes-h,col="
        orange")
841 }
      else{
843 rect(8/nb.col+y,1-(i+1)/nb.lignes,9/nb.col,1-(i)/nb.lignes-h,col="
        green")
845 }
      if(b[i,7+4]==-1){
847 rect(8/nb.col+y,1-(i+1)/nb.lignes,9/nb.col,1-(i)/nb.lignes-h,col="red
        ")
849 }
851 }
853 dev.off()

```

surveillance_adm_adm75.R

C.3.3 Partie spécifique

```

1
3 #####
3 #####
5 #####Paramètres modifiables#####
5 #####
7
7 #Endroit ou sont stockés les .csv
9 chemin<-"Z:\\SURSAUD\\Stage Nicolas\\Nicolas J\\Données_Oscour\\"
11
11 #Endroit ou est envoyé le pdf
11 chemin_des_sorties<-"Z:\\SURSAUD\\SURV_NON_SPEC\\Resultat_syndromique
13   \\Resultats\\"
13
13 #Donner les établissement
15
17 #nom des maladies
19
19 #####

```

```

21 |
22 | require(tcltk)
23 | require(tkrplot)
    | tt<-tktoplevel()
25 | tl<-tklistbox(tt,height=4,selectmode="single",background="white")
    | tkgrid(tklabel(tt,text="Quelle série voulez-vous étudier?"))
27 | tkgrid(tl)
    | fruits <- c("Aucune","Grippe","Bronchiolite","Gastro")
29 | for (i in (1:4))
    | {
31 |     tkinsert(tl,"end",fruits[i])
    | }
33 | tkselection.set(tl,0) # Default fruit is Banana. Indexing starts at
    |     zero.
35 | OnOK <- function()
    | {
37 |     fruitChoice <- fruits[as.numeric(tkcurselection(tl))+1]
    |     while(fruitChoice == "Aucune"){
39 |     fruitChoice <- fruits[as.numeric(tkcurselection(tl))+1]
    |     }
41 |     tkdestroy(tt)
    |     msg <- paste("Vous avez choisi la série ",fruitChoice,sep="")
43 |     tkmessageBox(message=msg)
    |     fruitChoice
45 |
    | }
47 | tkfocus(tt)
49 | a<-OnOK()
    | maladie<-a
51 | donnees<-read.csv2(paste(chemin,a,".csv",sep=""),sep=";")
    | etude<-donnees[,1]
53 |
    | Myhscale <- 1.5 # Horizontal scaling
55 | Myvscale <- 1.5 # Vertical scaling
57 | plotFunction <- function()
    | {
59 |     params <- par(bg="white")
    |     z5<-quantile(etude,0.95)
61 |     z9<-quantile(etude,0.99)
    |     z10<-quantile(etude,0.90)
63 |     z25<-quantile(etude,0.75)
    |     z50<-quantile(etude,0.50)
65 |     z40<-quantile(etude,0.60)
    |     z30<-quantile(etude,0.70)
67 |     plot(donnees[,1],main="Voici une première représentation de la
    |         série choisie :",type="l")
    |     abline(h=c(z9,z5,z10,z25,z30,z40,z50),lty=2,col=c("red","pink","
    |         orange","yellow","green","blue","purple") )
69 | par(params)

```

```

}
71 tt2 <- tktoplevel()
tkwm.title(tt2,"Série")
73 img <- tkrplot(tt2,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
tkgrid(img)
75
77 tt<-tktoplevel()
tl<-tklistbox(tt,height=9,selectmode="single",background="white")
79 tkgrid(tklabel(tt,text="Ecreter la série de manière brute ?"))
tkgrid(tl)
81 fruits <- c("Attente","Pas de modifications","Quantile 1%","Quantile
5%","Quantile 10%","Quantile 25%","Quantile 30%","Quantile 40%","
Quantile 50%")
for (i in (1:9))
83 {
tkinsert(tl,"end",fruits[i])
85 }
tkselection.set(tl,0) # Default fruit is Banana. Indexing starts at
zero.
87
OnOK <- function()
89 {
fruitChoice <- fruits[as.numeric(tkcurselection(tl))+1]
91 while(fruitChoice == "Attente"){
fruitChoice <- fruits[as.numeric(tkcurselection(tl))+1]
93 }
95
97
tkdestroy(tt)
99 tkdestroy(tt2)
fruitChoice
101 }
103
tkfocus(tt)
105 b<-OnOK()
quantile<-b
107 if(b=="Quantile 1%"){
z<-quantile(etude,0.99)
109 for(i in 1:length(etude)){
if(etude[i]>z){etude[i]<-z}
111 }
}
113
if(b=="Quantile 30%"){
115 z<-quantile(etude,0.70)
for(i in 1:length(etude)){
117 if(etude[i]>z){etude[i]<-z}
}
}

```



```
119 }
121
122   if(b=="Quantile 5%"){
123 z<-quantile(etude,0.95)
124 for(i in 1:length(etude)){
125   if(etude[i]>z){etude[i]<-z}
126   }
127 }
128
129   if(b=="Quantile 10%"){
130 z<-quantile(etude,0.90)
131 for(i in 1:length(etude)){
132   if(etude[i]>z){etude[i]<-z}
133   }
134 }
135
136   if(b=="Quantile 25%"){
137 z<-quantile(etude,0.75)
138 for(i in 1:length(etude)){
139   if(etude[i]>z){etude[i]<-z}
140   }
141 }
142
143   if(b=="Quantile 50%"){
144 z<-quantile(etude,0.50)
145 for(i in 1:length(etude)){
146   if(etude[i]>z){etude[i]<-z}
147   }
148 }
149
150   if(b=="Quantile 40%"){
151 z<-quantile(etude,0.60)
152 for(i in 1:length(etude)){
153   if(etude[i]>z){etude[i]<-z}
154   }
155 }
156
157 Myhscale <- 1.5    # Horizontal scaling
158 Myvscale <- 1.5    # Vertical scaling
159
160 plotFunction <- function()
161 {
162   params <- par(bg="white")
163   plot(etude,main="Voici la série écrétée de manière brute :",type="l")
164   par(params)
165 }
```

```

tt2 <- tkoplevel()
171 tkwm.title(tt2,"Série")
img <- tkrplot(tt2,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
173 tkgrid(img)

175 msg <- paste("Nous allons maintenant écréter la série en la lissant
par une fonction de régression linéaire ",sep="")
tkmessageBox(message=msg)
177 tkdestroy(tt2)

179

181
tt<-tkoplevel()
183 t1<-tklistbox(tt,height=5,selectmode="single",background="white")
tkgrid(tklabel(tt,text="Quel choix des variables de la régression ?")
)
185 tkgrid(t1)
fruits <- c("Attente","Choix automatique","Période annuelle","
Annuelle + semi","Toutes")
187 for (i in (1:5))
{
189 tkinsert(t1,"end",fruits[i])
}
191 tkselection.set(t1,0) # Default fruit is Banana. Indexing starts at
zero.

193 OnOK <- function()
{
195 fruitChoice <- fruits[as.numeric(tkcurselection(t1))+1]
while(fruitChoice == "Attente"){
197 fruitChoice <- fruits[as.numeric(tkcurselection(t1))+1]
}
199

201

203 tkdestroy(tt)
fruitChoice
205
}
207
tkfocus(tt)
209 c<-OnOK()

211 n<-length(donnees$Cas)

213 t <- 1:length(donnees$Cas)
t2 <- t^2
215 t3 <- cos(2*pi*t/52)
t4 <- sin(2*pi*t/52)
217 t5 <- cos(4*pi*t/52)

```

```

t6 <- sin(4*pi*t/52)
219 t7 <- cos(6*pi*t/52)
t8 <- sin(6*pi*t/52)
221 t9 <- cos(13*pi*t/52)
t10 <- sin(13*pi*t/52)
223
#####
225 #####
227
229 if(c=="Choix automatique"){
231
modele <- lm( etude ~ t + t2 + t3 + t4 + t5 + t6 + t7 + t8 + t9 + t10
) #, weights=donnees$poids)
233 modele2 <- step(modele, dir="both")
upper <- predict(modele2, interval="confidence", level=0.85)[,3]
235
Myhscale <- 2 # Horizontal scaling
237 Myvscale <- 2 # Vertical scaling
239
plotFunction <- function()
241 {
params <- par(bg="white")
243 layout(matrix(c(1,1),2))
matplot(matrix(c(predict(modele2), upper, etude), n), type="l", lty=c
(1,2,1), main="Avant", ylab="Nb de cas")
245 par(params)
}
247 tt2 <- tktoplevel()
tkwm.title(tt2, "Série")
249 img <- tkrplot(tt2, fun=plotFunction, hscale=Myhscale, vscale=Myvscale)
tkgrid(img)
251
253
255 upper <- as.vector(upper)
257 etude2 <- etude
259 for(i in t){
etude[i] <- min(etude[i], upper[i])
261 }
263
Myhscale <- 2 # Horizontal scaling
265 Myvscale <- 2 # Vertical scaling
267 plotFunction <- function()

```

```

269 {
      params <- par(bg="white")
matplot(matrix(c(predict(modele2),upper,etude),n),type="l",lty=c
      (1,2,1),main="Après",ylab="Nb de cas")
271 par(params)
      }
273 tt4 <- tktoplevel()
      tkwm.title(tt4,"Série")
275 img <- tkrplot(tt4,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
      tkgrid(img)
277
279 }
281
283 if(c=="Toutes"){
285
286 modele <- lm( etude ~ t + t2 + t3 + t4 + t5 + t6 + t7 + t8 + t9 + t10
      )
287 modele2 <- modele
upper<- predict(modele2,interval="confidence",level=0.99)[,3]
289
291 plotFunction <- function()
293 {
      params <- par(bg="white")
295 layout(matrix(c(1,1),2))
matplot(matrix(c(predict(modele2),upper,etude),n),type="l",lty=c
      (1,2,1))
297 par(params)
      }
299 tt2 <- tktoplevel()
      tkwm.title(tt2,"Série")
301 img <- tkrplot(tt2,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
      tkgrid(img)
303
305
307 upper<-as.vector(upper)
309 etude2<-etude
311 for(i in t){
      etude[i] <- min(etude[i],upper[i])
313 }
315

```

```

317 | plotFunction <- function()
319 | {
      |   params <- par(bg="white")
321 | matplot(matrix(c(predict(modele2),upper,etude),n),type="l",lty=c
      |   (1,2,1))
      | par(params)
323 | }
      | tt4 <- tktoplevel()
325 | tkwm.title(tt4,"Série")
      | img <- tkrplot(tt4,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
327 | tkgrid(img)
      |
329 |
      | }
331 |
333 |
335 | if(c=="Annuelle + semi"){
337 | modele <- lm( etude ~ t + t3 + t4 + t5 + t6 )#,weights=donnees$poids)
      | modele2 <- modele
339 | upper<- predict(modele2,interval="confidence",level=0.99)[,3]
341 |
343 | plotFunction <- function()
      | {
345 |   params <- par(bg="white")
      | layout(matrix(c(1,1),2))
347 | matplot(matrix(c(predict(modele2),upper,etude),n),type="l",lty=c
      |   (1,2,1))
      | title("Avant")
349 | par(params)
      | }
351 | tt2 <- tktoplevel()
      | tkwm.title(tt2,"Série")
353 | img <- tkrplot(tt2,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
      | tkgrid(img)
355 |
357 |
359 | upper<-as.vector(upper)
361 | etude2<-etude
363 | for(i in t){
      | etude[i] <- min(etude[i],upper[i])
365 | }

```

```

367
369 plotFunction <- function()
371 {
371     params <- par(bg="white")
matplot(matrix(c(predict(modele2),upper,etude),n),type="l",lty=c
373     (1,2,1))
title("Après")
par(params)
375 }
375 tt4 <- tktoplevel()
377 tkwm.title(tt4,"Série")
img <- tkrplot(tt4,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
379 tkgrid(img)

381
383
385 if(c=="Période annuelle"){
387
387 modele <- lm( etude ~ t + t3 + t4 )#,weights=donnees$poids)
389 modele2 <- modele
upper<- predict(modele2,interval="confidence",level=0.99)[,3]
391
393 plotFunction <- function()
395 {
395     params <- par(bg="white")
layout(matrix(c(1,1),2))
397 matplot(matrix(c(predict(modele2),upper,etude),n),type="l",lty=c
399     (1,2,1))
par(params)
401 }
401 tt2 <- tktoplevel()
401 tkwm.title(tt2,"Série")
img <- tkrplot(tt2,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
403 tkgrid(img)

405
407
407 upper<-as.vector(upper)
409
409 etude2<-etude
411
411 for(i in t){
413 etude[i] <- min(etude[i],upper[i])
415 }

```

```

417
419 plotFunction <- function()
421 {
422     params <- par(bg="white")
423     matplot(matrix(c(predict(modele2),upper,etude),n),type="l",lty=c
424         (1,2,1))
425     par(params)
426 }
427 tt4 <- tktoplevel()
428 tkwm.title(tt4,"Série")
429 img <- tkrplot(tt4,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
430 tkgrid(img)
431 }
432
433
434
435 msg <- paste("Nous allons tracer le seuil ",sep="")
436 tkmessageBox(message=msg)
437 tkdestroy(tt2)
438 tkdestroy(tt4)
439
440
441 #####
442 #####
443 #####
444
445
446 Myhscale <- 2    # Horizontal scaling
447 Myvscale <- 2    # Vertical scaling
448
449
450
451 donnees.sauv<-donnees
452
453 if(c=="Choix automatique"){
454
455
456 modele <- lm( etude ~ t + t^2 + cos(2*pi*t/52) + sin(2*pi*t/52) + cos
457     (4*pi*t/52) + sin(4*pi*t/52) + cos(6*pi*t/52) + sin(6*pi*t/52) +
458     cos(13*pi*t/52) + sin(13*pi*t/52))
459 modele2 <- step(modele,dir="both")
460 upper<- predict(modele2,interval="confidence",level=0.99)[,3]
461
462 plotFunction <- function()
463 {
464     params <- par(bg="white")

```

```

465 matplot(matrix(c(predict(modele2),upper,donnees$Cas),n),type="l",lty=
      c(1,2,1),main="Seuil Serfling",ylab="Nb de cas")
467 z<-min(donnees.sauv$Cas)
469 for(i in t[-1]){
471 if(donnees.sauv$Cas[i]>upper[i] && donnees.sauv$Cas[i-1]>upper[i-1]){
      lines(c(i,i+1),c(z,z),lwd=2.5,col="red")
473 }
475 }
477 par(params)
      }
479 tt2 <- tktoplevel()
      tkwm.title(tt2,"Série")
481 img <- tkrplot(tt2,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
      tkgrid(img)
483
485 ttt<-tktoplevel()
      tl<-tklistbox(ttt,height=4,selectmode="single",background="white")
487 tkgrid(tklabel(ttt,text="Doit-on décaler la série?"))
      tkgrid(tl)
489 fruits <- c("Attente","Pas de décalage","Histo : 10 semaines","Histo
      : 25 semaines","Histo : 50 semaines")
      for (i in (1:5))
491 {
      tkinsert(tl,"end",fruits[i])
493 }
      tkselection.set(tl,0) # Default fruit is Banana. Indexing starts at
      zero.
495 OnOK <- function()
497 {
      fruitChoice <- fruits[as.numeric(tkcurselection(tl))+1]
499 while(fruitChoice == "Attente"){
      fruitChoice <- fruits[as.numeric(tkcurselection(tl))+1]
501 }
503
505 tkdestroy(ttt)
507 tkdestroy(tt2)
      fruitChoice
509 }
511 tkfocus(ttt)
513 d<-OnOK()

```



```

515 if(d=="Pas de décalage") {d<-0}
      if(d=="Histo : 10 semaines") {d<-10}
517 if(d=="Histo : 25 semaines") {d<-25}
      if(d=="Histo : 50 semaines") {d<-50}
519 a<-0

521 if(d!=0){

523 vect<-(n-d):n

525 new<- data.frame(t = vect)
      pred<-predict(modele2,new, se.fit = TRUE)
527 pred2<-pred$fit
      diff<-donnees$Cas[vect]-pred2
529 diff2<-sum(diff^2)
      residus<-diff2

531
533 i<-0

535 for(i in -5:5){
      new<- data.frame(t = vect+i)
537 pred<-predict(modele2,new, se.fit = TRUE)
      pred2<-pred$fit
539 diff<-donnees$Cas[vect]-pred2
      diff2<-sum(diff^2)

541
      if(min(residus , diff2)==diff2){
543 residus<-diff2
          a<-i
545 }

547 }
      }
549

551 plotFunction <- function()
      {
553     params <- par(bg="white")

555 new<- data.frame(t = t+a)

557 #—5—# Dessin final
      b<-predict(modele2,new,interval="confidence",level=0.99)
559 b1<-b[,1]
          b3<-b[,3]
561 matplot(matrix(c(b1,b3,donnees.sauv$Cas),n),type="l",main="Gastro-
          entérites par semaine depuis septembre 2007 \n Seuil obtenu par la
          méthode de Serfling avec un risque de 1%",
          sub="Modele de régression linéaire : nb.cas ~ cos(2*Pi*t/52)+sin(2*Pi
          *t/52)+cos(4*Pi*t/52)+sin(4*Pi*t/52) \n Ce modèle inclut donc :

```

```

    une saisonnalité annuelle et semi-annuelle",
563 ylab="Nombre de cas de Gastro-entérites")

565 alarme<-0
    for(i in 1:length(b1)){
567 if(donnees$Cas[i]>b3[i]){
    alarme<-c(alarme,1)
569 }
    else{
571 alarme<-c(alarme,0)
    }
573 }
575 alarme<-alarme[-1]
577 z<-min(donnees$Cas)
579 for(i in 2:length(b1)){
581 if(alarme[i]+alarme[i-1]==2){
    lines(c(i-1,i),c(z,z))
583 }
    }
585
587
589 pdf(file=paste(chemin_des_sorties,maladie,".pdf",sep=""),width=13.5,
    height=10)
591 z2<-max(donnees$Cas)
    z<-max(min(donnees$Cas)-1,0)
593 matplot(matrix(c(b1,b3,donnees.sauv$Cas),n),ylim=c(z,z2*1.1),type=c("
    l","l","l"),lty=c(1,2,1),main=paste(maladie," par semaine depuis
    septembre 2007 \n Seuil obtenu par la méthode de Serfling avec un
    risque de 1%",sep=""),
    sub=paste("Modele de régression linéaire : ",c," \n Ecretation brute
    : ",quantile," \n Données Oscour recueillies auprès de 7
    établissements : CH Ales - CH Bagnols - CH Carcassonne - CHRU
    Montpellier - Clin St Louis - Poly St Roch - Poly St Jean",sep="")
    ,xaxt="n", xlab="",ylab=paste("Nombre de cas de ",maladie,sep=""))
595 axis(1,labels=donnees[,2],at=1:length(donnees[,2]))
    for(i in 2:length(b1)){
597 if(alarme[i]+alarme[i-1]==2){
    lines(c(i-1,i),c(z,z),lwd=2,col="red")
599 }
    }
601
603 legend("topleft",col=c("green","black","red","red"),legend=c("données
    ", "valeur \n attendue","seuil","période épidémique \n (2 sem. cons
    .)"),lty=c(1,1,2,1),lwd=c(1,1,1,2),horiz=TRUE)

```

```

605 zoom<-52
    nnn<-length(donnees.sauv$Cas)
607 deb<-nnn-zoom

609 matplot(matrix(c(b1[deb:nnn], b3[deb:nnn], donnees.sauv$Cas[deb:nnn]),
    zoom+1), type=c("l", "l", "l"), lty=c(1,2,1), main=paste(maladie, " par
    semaine, zoom sur la dernière année \n Seuil obtenu par la méthode
    de Serfling avec un risque de 1%", sep=""),
    sub=paste("Modele de régression linéaire : ", c, " \n Ecretation brute
    : ", quantile, sep=""), xaxt="n", xlab="", ylab=paste("Nombre de cas
    de ", maladie, sep=""))
611 axis(1, labels=donnees[deb:nnn, 2], at=1:length(deb:nnn))
    for(i in deb:nnn){
613 if(alarme[i]+alarme[i-1]==2){
        lines(c(i-1-deb+2, i-deb+2), c(z, z), lwd=2, col="red")
615 }
    }
617

619

621 dev.off()
    par(params)
623 if(alarme[length(alarme)]+alarme[length(alarme)-1]==2){
    msg <- paste("Il y a une alarme cette semaine")
625     tkmessageBox(message=msg)
    }
627 else{
    msg <- paste("Il n'y a pas d'alarme cette semaine")
629     tkmessageBox(message=msg)

631 }
    }
633

635

637 tt2 <- tkoplevel()
639 tkwm.title(tt2, "Série")
    img <- tkrplot(tt2, fun=plotFunction, hscale=Myhscale, vscale=Myvscale)
641 tkgrid(img)

643 }

645 #####
647 #####
649

```

```

651 if(c=="Période annuelle"){
653
654 modele <- lm( etude ~ t + cos(2*pi*t/52) + sin(2*pi*t/52) )
655 modele2 <- modele
656 upper<- predict(modele2, interval="confidence", level=0.99)[,3]
657
658
659 plotFunction <- function()
660 {
661     params <- par(bg="white")
662 matplot(matrix(c(predict(modele2), upper, donnees$Cas), n), type="l", lty=
663     c(1,2,1))
664 z<-min(donnees.sauv$Cas)
665
666 for(i in t[-1]){
667
668     if(donnees.sauv$Cas[i]>upper[i] && donnees.sauv$Cas[i-1]>upper[i-1]){
669 lines(c(i, i+1), c(z, z), lwd=2.5, col="red")
670     }
671 }
672
673 par(params)
674 }
675 tt2 <- tktoplevel()
676 tkwm.title(tt2, "Série")
677 img <- tkrplot(tt2, fun=plotFunction, hscale=Myhscale, vscale=Myvscale)
678 tkgrid(img)
679
680
681 ttt<-tktoplevel()
682 t1<-tklistbox(ttt, height=4, selectmode="single", background="white")
683 tkgrid(tklabel(ttt, text="Doit-on décaler la série?"))
684 tkgrid(t1)
685 fruits <- c("Attente", "Pas de décalage", "Histo : 10 semaines", "Histo
        : 25 semaines", "Histo : 50 semaines")
686 for (i in (1:5))
687 {
688     tkinsert(t1, "end", fruits[i])
689 }
690 tkselection.set(t1, 0) # Default fruit is Banana. Indexing starts at
        zero.
691
692 OnOK <- function()
693 {
694     fruitChoice <- fruits[as.numeric(tkcurselection(t1))+1]
695     while(fruitChoice == "Attente"){
696     fruitChoice <- fruits[as.numeric(tkcurselection(t1))+1]
697     }
698 }
699

```

```
701
703   tkdestroy(ttt)
704   tkdestroy(tt2)
705   fruitChoice
707 }
709 tkfocus(ttt)
710 d<-OnOK()
711 if(d=="Pas de décalage") {d<-0}
713 if(d=="Histo : 10 semaines") {d<-10}
714 if(d=="Histo : 25 semaines") {d<-25}
715 if(d=="Histo : 50 semaines") {d<-50}
716 a<-0
717 if(d!=0){
719   vect<-(n-d):n
721   new<- data.frame(t = vect)
723   pred<-predict(modele2,new, se.fit = TRUE)
724   pred2<-pred$fit
725   diff<-donnees$Cas[vect]-pred2
726   diff2<-sum(diff^2)
727   residus<-diff2
729   i<-0
731   for(i in -5:5){
733     new<- data.frame(t = vect+i)
734     pred<-predict(modele2,new, se.fit = TRUE)
735     pred2<-pred$fit
736     diff<-donnees$Cas[vect]-pred2
737     diff2<-sum(diff^2)
739     if(min(residus , diff2)==diff2){
740       residus<-diff2
741       a<-i
742     }
743   }
745 }
747 plotFunction <- function()
749 {
750   params <- par(bg="white")
751
```

```

new<- data.frame(t = t+a)
753 #—5—# Dessin final
755 b<-predict(modele2,new,interval="confidence",level=0.99)
b1<-b[,1]
757 b3<-b[,3]
matplot(matrix(c(b1,b3,donnees.sauv$Cas),n),type=c("l"),main="Gastro-
entérites par semaine depuis septembre 2007 \n Seuil obtenu par la
méthode de Serfling avec un risque de 1%",
759 sub="Modele de régression linéaire : nb.cas ~ cos(2*Pi*t/52)+sin(2*Pi
*t/52)+cos(4*Pi*t/52)+sin(4*Pi*t/52) \n Ce modèle inclut donc :
une saisonnalité annuelle et semi-annuelle",
ylab="Nombre de cas de Gastro-entérites")
761
alarme<-0
763 for(i in 1:length(b1)){
if(donnees$Cas[i]>b3[i]){
765 alarme<-c(alarme,1)
}
767 else{
alarme<-c(alarme,0)
769 }
771 }
773 alarme<-alarme[-1]
775 z<-min(donnees$Cas)
777 for(i in 2:length(b1)){
if(alarme[i]+alarme[i-1]==2){
779 lines(c(i-1,i),c(z,z))
}
781 }
783 pdf(file=paste(chemin_des_sorties,maladie,".pdf",sep=""),width=13.5,
height=10)
z2<-max(donnees$Cas)
785 z<-max(min(donnees$Cas)-1,0)
matplot(matrix(c(b1,b3,donnees.sauv$Cas),n),ylim=c(z,z2*1.1),type=c("
l","l","l"),lty=c(1,2,1),main=paste(maladie," par semaine depuis
septembre 2007 \n Seuil obtenu par la méthode de Serfling avec un
risque de 1%",sep=""),
787 sub=paste("Modele de régression linéaire : ",c," \n Ecretation brute
: ",quantile," \n Données Oscour recueillies auprès de 7
établissements : CH Ales - CH Bagnols - CH Carcassonne - CHRU
Montpellier - Clin St Louis - Poly St Roch - Poly St Jean",sep="")
,xaxt="n",xlab="",ylab=paste("Nombre de cas de ",maladie,sep=""))
axis(1,labels=donnees[,2],at=1:length(donnees[,2]))
789 for(i in 2:length(b1)){
if(alarme[i]+alarme[i-1]==2){
791 lines(c(i-1,i),c(z,z),lwd=2,col="red")

```

```

}
793 }
795
legend("topleft",col=c("green","black","red","red"),legend=c("données
", "valeur \n attendue", "seuil", "période épidémique \n (2 sem. cons
.)"),lty=c(1,1,2,1),lwd=c(1,1,1,2),horiz=TRUE)
797
zoom<-52
799 nnn<-length(donnees.sauv$Cas)
deb<-nnn-zoom
801
matplot(matrix(c(b1[deb:nnn],b3[deb:nnn],donnees.sauv$Cas[deb:nnn]),
zoom+1),type=c("l","l","l"),lty=c(1,2,1),main=paste(maladie," par
semaine, zoom sur la dernière année \n Seuil obtenu par la méthode
de Serfling avec un risque de 1%",sep=""),
803 sub=paste("Modele de régression linéaire : ",c," \n Ecretation brute
: ",quantile,sep=""),xaxt="n",xlab="",ylab=paste("Nombre de cas
de ",maladie,sep=""))
axis(1,labels=donnees[deb:nnn,2],at=1:length(deb:nnn))
805 for(i in deb:nnn){
if(alarme[i]+alarme[i-1]==2){
807 lines(c(i-1-deb+2,i-deb+2),c(z,z),lwd=2,col="red")
}
809 }
811
813
dev.off()
815 if(alarme[length(alarme)]+alarme[length(alarme)-1]==2){
msg <- paste("Il y a une alarme cette semaine")
817 tkmessageBox(message=msg)
}
819 else{
msg <- paste("Il n'y a pas d'alarme cette semaine")
821 tkmessageBox(message=msg)
823 }
par(params)
825 }
827
829
831 tt2 <- tktoplevel()
tkwm.title(tt2,"Série")
833 img <- tkrplot(tt2,fun=plotFonction,hscale=Myhscale,vscale=Myvscale)
tkgrid(img)
835

```

```

837 }
839
841 #####
842 #####
843 #####
844 #####
845 #####
847
848 if(c=="Toutes"){
849
851 modele <- lm( etude ~ t + t^2 + cos(2*pi*t/52) + sin(2*pi*t/52) + cos
      (4*pi*t/52) + sin(4*pi*t/52) + cos(6*pi*t/52) + sin(6*pi*t/52) +
      cos(13*pi*t/52) + sin(13*pi*t/52))
      modele2 <- modele
853 upper<- predict(modele2, interval="confidence", level=0.99)[,3]
855
856 plotFunction <- function()
857 {
      params <- par(bg="white")
859 matplot(matrix(c(predict(modele2), upper, donnees$Cas), n), type="l", lty=
      c(1,2,1))
      z<-min(donnees.sauv$Cas)
861
862
863 for(i in t[-1]){
865 if(donnees.sauv$Cas[i]>upper[i] && donnees.sauv$Cas[i-1]>upper[i-1]){
      lines(c(i, i+1), c(z, z), lwd=2.5, col="red")
867 }
869 }
871 par(params)
      }
873 tt2 <- tktoplevel()
      tkwm.title(tt2, "Série")
875 img <- tkrplot(tt2, fun=plotFunction, hscale=Myhscale, vscale=Myvscale)
      tkgrid(img)
877
878
879 ttt<-tktoplevel()
      t1<-tklistbox(ttt, height=4, selectmode="single", background="white")
881 tkgrid(tklabel(ttt, text="Doit-on décaler la série?"))
      tkgrid(t1)
883 fruits <- c("Attente", "Pas de décalage", "Histo : 10 semaines", "Histo
      : 25 semaines", "Histo : 50 semaines")
      for (i in (1:5))

```



```

885 {
      tkinsert(tl,"end",fruits[i])
887 }
tkselection.set(tl,0) # Default fruit is Banana. Indexing starts at
      zero.
889
OnOK <- function()
891 {
      fruitChoice <- fruits[as.numeric(tkcurselection(tl))+1]
893 while(fruitChoice == "Attente"){
      fruitChoice <- fruits[as.numeric(tkcurselection(tl))+1]
895 }
897
899
      tkdestroy(ttt)
901 tkdestroy(tt2)
      fruitChoice
903
}
905
tkfocus(ttt)
907 d<-OnOK()

909 if(d=="Pas de décalage") {d<-0}
if(d=="Histo : 10 semaines") {d<-10}
911 if(d=="Histo : 25 semaines") {d<-25}
if(d=="Histo : 50 semaines") {d<-50}
913 a<-0

915 if(d!=0){
917 vect<-(n-d):n

919 new<- data.frame(t = vect)
pred<-predict(modele2,new, se.fit = TRUE)
921 pred2<-pred$fit
diff<-donnees$Cas[vect]-pred2
923 diff2<-sum(diff^2)
residus<-diff2
925
i<-0
927

929 for(i in -5:5){
new<- data.frame(t = vect+i)
931 pred<-predict(modele2,new, se.fit = TRUE)
pred2<-pred$fit
933 diff<-donnees$Cas[vect]-pred2
diff2<-sum(diff^2)
935

```

```
  if(min(residus , diff2)==diff2){
937 residus<-diff2
    a<-i
939 }
  }
941 }
943
945 plotFunction <- function()
  {
947     params <- par(bg="white")
949 new<- data.frame(t = t+a)
951 #---5---# Dessin final
    b<-predict(modele2 ,new, interval="confidence" , level=0.99)
953 b1<-b[,1]
    b3<-b[,3]
955 matplot(matrix(c(b1,b3,donnees$Cas),n),type="l",main="Gastro-
      entérites par semaine depuis septembre 2007 \n Seuil obtenu par la
      méthode de Serfling avec un risque de 1%",
    sub="Modele de régression linéaire : nb.cas ~ cos(2*Pi*t/52)+sin(2*Pi
      *t/52)+cos(4*Pi*t/52)+sin(4*Pi*t/52) \n Ce modèle inclut donc :
      une saisonnalité annuelle et semi-annuelle",
957 ylab="Nombre de cas de Gastro-entérites")
959 alarme<-0
    for(i in 1:length(b1)){
961 if(donnees$Cas[i]>b3[i]){
      alarme<-c(alarme,1)
963 }
      else{
965 alarme<-c(alarme,0)
      }
967 }
969 alarme<-alarme[-1]
971 z<-min(donnees$Cas)
973 for(i in 2:length(b1)){
975 if(alarme[i]+alarme[i-1]==2){
      lines(c(i-1,i),c(z,z))
977 }
      }
979
981
```

```

983 pdf( file=paste(chemin_des_sorties , maladie , ". pdf" , sep="" ) , width=13.5 ,
      height=10)
      z2<-max(donnees$Cas)
985 z<-max(min(donnees$Cas) -1,0)
      matplot(matrix(c(b1 , b3 , donnees . sauv$Cas) , n) , ylim=c(z , z2*1.1) , type=c("
      l" , "l" , "l" ) , lty=c(1,2,1) , main=paste(maladie , " par semaine depuis
      septembre 2007 \n Seuil obtenu par la méthode de Serfling avec un
      risque de 1%" , sep="" ) ,
987 sub=paste("Modele de régression linéaire : " , c , " \n Ecretation brute
      : " , quantile , " \n Données Oscour recueillies auprès de 7
      établissements : CH Ales - CH Bagnols - CH Carcassonne - CHRU
      MontPELLIER - Clin St Louis - Poly St Roch - Poly St Jean" , sep="" )
      , xaxt="n" , xlab="" , ylab=paste("Nombre de cas de " , maladie , sep="" ))
      axis(1 , labels=donnees [ , 2] , at=1:length(donnees [ , 2]))
989 for(i in 2:length(b1)){
      if(alarme [ i]+alarme [ i-1]==2){
991 lines(c(i-1 , i) , c(z , z) , lwd=2 , col="red" )
      }
993 }

995 legend("topleft" , col=c("green" , "black" , "red" , "red" ) , legend=c("données
      " , "valeur \n attendue" , "seuil" , "période épidémique \n (2 sem. cons
      .)" ) , lty=c(1,1,2,1) , lwd=c(1,1,1,2) , horiz=TRUE)
997 zoom<-52
999 nnn<-length(donnees . sauv$Cas)
      deb<-nnn-zoom
1001 matplot(matrix(c(b1 [ deb : nnn ] , b3 [ deb : nnn ] , donnees . sauv$Cas [ deb : nnn ] ) ,
      zoom+1) , type=c("l" , "l" , "l" ) , lty=c(1,2,1) , main=paste(maladie , " par
      semaine , zoom sur la dernière année \n Seuil obtenu par la méthode
      de Serfling avec un risque de 1%" , sep="" ) ,
1003 sub=paste("Modele de régression linéaire : " , c , " \n Ecretation brute
      : " , quantile , sep="" ) , xaxt="n" , xlab="" , ylab=paste("Nombre de cas
      de " , maladie , sep="" ))
      axis(1 , labels=donnees [ deb : nnn , 2] , at=1:length(deb : nnn))
1005 for(i in deb : nnn){
      if(alarme [ i]+alarme [ i-1]==2){
1007 lines(c(i-1-deb+2 , i-deb+2) , c(z , z) , lwd=2 , col="red" )
      }
1009 }

1011
1013
1015 dev.off()

1017
1019 if(alarme [ length(alarme) ]+alarme [ length(alarme) -1]==2){

```

```

msg <- paste("Il y a une alarme cette semaine")
1021 tkmessageBox(message=msg)
}
1023 else{
msg <- paste("Il n'y a pas d'alarme cette semaine")
1025 tkmessageBox(message=msg)
1027 }
par(params)
1029 }
1031
1033
1035 tt2 <- tktoplevel()
tkwm.title(tt2,"Série")
1037 img <- tkrplot(tt2,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
tkgrid(img)
1039
1041 }
1043
1045 #####
1047 #####
1049 #####
1051 if(c=="Annuelle + semi"){
1053
modele <- lm( etude ~ t + t^2 + cos(2*pi*t/52) + sin(2*pi*t/52) + cos
(4*pi*t/52) + sin(4*pi*t/52) )
1055 modele2 <- step(modele,dir="both")
upper<- predict(modele2,interval="confidence",level=0.99)[,3]
1057
1059 plotFunction <- function()
{
1061 params <- par(bg="white")
matplot(matrix(c(predict(modele2),upper,donnees$Cas),n),type="l",lty=
c(1,2,1))
1063 z<-min(donnees.sauv$Cas)
1065
for(i in t[-1]){
1067
if(donnees.sauv$Cas[i]>upper[i] && donnees.sauv$Cas[i-1]>upper[i-1]){
1069 lines(c(i,i+1),c(z,z),lwd=2.5,col="red")

```

```

}
1071 }
1073 }
1075 par(params)
1077 }
1077 tt2 <- tktoplevel()
1077 tkwm.title(tt2,"Série")
1079 img <- tkrplot(tt2,fun=plotFunction,hscale=Myhscale,vscale=Myvscale)
1079 tkgrid(img)

1081
1081 ttt<-tktoplevel()
1083 t1<-tklistbox(ttt,height=4,selectmode="single",background="white")
1083 tkgrid(tklabel(ttt,text="Doit-on décaler la série?"))
1085 tkgrid(t1)
1085 fruits <- c("Attente","Pas de décalage","Histo : 10 semaines","Histo
: 25 semaines","Histo : 50 semaines")
1087 for (i in (1:5))
{
1089 tkinsert(t1,"end",fruits[i])
}
1091 tkselection.set(t1,0) # Default fruit is Banana. Indexing starts at
zero.

1093 OnOK <- function()
{
1095 fruitChoice <- fruits[as.numeric(tkcurselection(t1))+1]
1095 while(fruitChoice == "Attente"){
1097 fruitChoice <- fruits[as.numeric(tkcurselection(t1))+1]
}
1099
1101
1103 tkdestroy(ttt)
1103 tkdestroy(tt2)
1105 fruitChoice
1107 }

1109 tkfocus(ttt)
1109 d<-OnOK()
1111
1111 if(d=="Pas de décalage") {d<-0}
1113 if(d=="Histo : 10 semaines") {d<-10}
1113 if(d=="Histo : 25 semaines") {d<-25}
1115 if(d=="Histo : 50 semaines") {d<-50}
1115 a<-0
1117
1117 if(d!=0){
1119

```

```

1121 vect<-(n-d):n
1122 new<- data.frame(t = vect)
1123 pred<-predict(modele2,new, se.fit = TRUE)
1124 pred2<-pred$fit
1125 diff<-donnees$Cas[vect]-pred2
1126 diff2<-sum(diff^2)
1127 residus<-diff2

1129 i<-0

1131 for(i in -5:5){
1132 new<- data.frame(t = vect+i)
1133 pred<-predict(modele2,new, se.fit = TRUE)
1134 pred2<-pred$fit
1135 diff<-donnees$Cas[vect]-pred2
1136 diff2<-sum(diff^2)

1139 if(min(residus , diff2)==diff2){
1140 residus<-diff2
1141 a<-i
1142 }
1143 }
1144 }
1145 }

1147 plotFunction <- function()
1148 {
1149     params <- par(bg="white")
1150 }
1151 new<- data.frame(t = t+a)
1152 #---5---# Dessin final
1153 b<-predict(modele2,new,interval="confidence",level=0.99)
1154 b1<-b[,1]
1155 b3<-b[,3]
1156 matplot(matrix(c(b1,b3,donnees.sauv$Cas),n),type=c("l","l","l"),lty=c
1157 (1,2,1),main=paste(maladie," par semaine depuis septembre 2007 \n
1158 Seuil obtenu par la méthode de Serfling avec un risque de 1%",sep=
1159 ""),
1160 sub=paste("Modele de régression linéaire : ",c," \n Ecretation brute
1161 : ",quantile,sep=""),xaxt="n", xlab="",ylab=paste("Nombre de cas
1162 de ",maladie,sep=""))
1163 axis(1,labels=donnees[,2],at=1:length(donnees[,2]))
1164
1165 alarme<-0
1166 for(i in 1:length(b1)){
1167     if(donnees$Cas[i]>b3[i]){

```

```

1167 alarme<-c(alarme,1)
    }
1169 else{
    alarme<-c(alarme,0)
1171 }

1173 }

1175 alarme<-alarme[-1]

1177 z<-min(donnees$Cas)

1179 for(i in 2:length(b1)){
    if(alarme[i]+alarme[i-1]==2){
1181 lines(c(i-1,i),c(z,z))
    }
1183 }

1185 pdf(file=paste(chemin_des_sorties,maladie,".pdf",sep=""),width=13.5,
    height=10)
    z2<-max(donnees$Cas)
1187 z<-max(min(donnees$Cas)-1,0)
    matplot(matrix(c(b1,b3,donnees.sauv$Cas),n),ylim=c(z,z2*1.1),type=c("
    l","l","l"),lty=c(1,2,1),main=paste(maladie," par semaine depuis
    septembre 2007 \n Seuil obtenu par la méthode de Serfling avec un
    risque de 1%",sep=""),
1189 sub=paste("Modele de régression linéaire : ",c," \n Ecretation brute
    : ",quantile," \n Données Oscour recueillies auprès de 7
    établissements : CH Ales - CH Bagnols - CH Carcassonne - CHRU
    Montpellier - Clin St Louis - Poly St Roch - Poly St Jean",sep="")
    ,xaxt="n", xlab="",ylab=paste("Nombre de cas de ",maladie,sep=""))
    axis(1,labels=donnees[,2],at=1:length(donnees[,2]))
1191 for(i in 2:length(b1)){
    if(alarme[i]+alarme[i-1]==2){
1193 lines(c(i-1,i),c(z,z),lwd=2,col="red")
    }
1195 }

1197 legend("topleft",col=c("green","black","red","red"),legend=c("données
    ", "valeur \n attendue", "seuil", "période épidémique \n (2 sem. cons
    .)"),lty=c(1,1,2,1),lwd=c(1,1,1,2),horiz=TRUE)
1199 zoom<-52
1201 nnn<-length(donnees.sauv$Cas)
    deb<-nnn-zoom
1203 matplot(matrix(c(b1[deb:nnn],b3[deb:nnn],donnees.sauv$Cas[deb:nnn]),
    zoom+1),type=c("l","l","l"),lty=c(1,2,1),main=paste(maladie," par
    semaine, zoom sur la dernière année \n Seuil obtenu par la méthode
    de Serfling avec un risque de 1%",sep=""),

```

```

1205 sub=paste("Modele de régression linéaire : ",c," \n Ecretation brute
      : ",quantile,sep=""),xaxt="n", xlab="",ylab=paste("Nombre de cas
      de ",maladie,sep=""))
      axis(1,labels=donnees[deb:nnn,2],at=1:length(deb:nnn))
1207 for(i in deb:nnn){
      if(alarme[i]+alarme[i-1]==2){
1209 lines(c(i-1-deb+2,i-deb+2),c(z,z),lwd=2,col="red")
      }
1211 }

1213

1215
1215 dev.off()
1217

1219

1221
1221 if(alarme[length(alarme)]+alarme[length(alarme)-1]==2){
1223 msg <- paste("Il y a une alarme cette semaine")
      tkmessageBox(message=msg)
1225 }
      else{
1227 msg <- paste("Il n'y a pas d'alarme cette semaine")
      tkmessageBox(message=msg)
1229 }
1231 par(params)
      }

1233

1235

1237
1237 tt2 <- tktoplevel()
1239 tkwm.title(tt2,"Série")
      img <- tkrplot(tt2,fun=plotFonction,hscale=Myhscale,vscale=Myvscale)
1241 tkgrid(img)

1243
1245 }

1247 msg <- paste("Ce graphique, ainsi qu'un zoom sur la dernière année a
      été enregistré au format pdf à l'adresse : \n ",chemin_des_sorties
      , "\n \n          Merci d'avoir utilisé cette application et à
      bientôt !", sep="")
      tkmessageBox(message=msg)

```


Bibliographie

- [1] P.J. Brockwell and R.A. Davis. *Time series : theory and methods*. Springer Verlag, 2009.
- [2] P. Dalgaard. A primer on the R-Tcl/Tk package. *Porting R to Darwin/X11 and Mac OS X*, page 27.
- [3] A. Earnest, M.I. Chen, D. Ng, and L.Y. Sin. Using autoregressive integrated moving average (ARIMA) models to predict and monitor the number of beds occupied during a SARS outbreak in a tertiary hospital in Singapore. *BMC Health Services Research*, 5(1) :36, 2005.
- [4] CP Farrington, NJ Andrews, AD Beale, and MA Catchpole. A statistical algorithm for the early detection of outbreaks of infectious disease. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, pages 547–563, 1996.
- [5] B.A. Griffin, A.K. Jain, J. Davies-Cole, C. Glymph, G. Lum, S.C. Washington, and M.A. Stoto. Early detection of influenza outbreaks using the DC Department of Health’s syndromic surveillance system. *BMC Public Health*, 9(1) :483, 2009.
- [6] L. Hutwagner, T. Browne, G.M. Seeman, and A.T. Fleischauer. Comparing aberration detection methods with simulated data. *journal.shouxi.net*, 11(2), 2005.
- [7] LC Hutwagner, EK Maloney, NH Bean, L. Slutsker, and SM Martin. Using laboratory-based surveillance data for prevention : an algorithm for detecting Salmonella outbreaks. *Emerging Infectious Diseases*, 3(3) :395, 1997.
- [8] LC Hutwagner, WW Thompson, GM Seeman, and T. Treadwell. A simulation model for assessing aberration detection methods used in public health surveillance for systems with limited baselines. *Statistics in Medicine*, 24(4) :543–550, 2005.
- [9] M.L. Jackson, A. Baer, I. Painter, and J. Duchin. A simulation study comparing aberration detection algorithms for syndromic surveillance. *BMC Medical Informatics and Decision Making*, 7(1) :6, 2007.
- [10] A.B. Lawson and K. Kleinman. *Spatial and syndromic surveillance for public health*. John Wiley & Sons Inc, 2005.

- [11] C. Pelat, P.Y. Boelle, B.J. Cowling, F. Carrat, A. Flahault, S. Ansart, and A.J. Valleron. Online detection and quantification of epidemics. *BMC Medical Informatics and Decision Making*, 7(1) :29, 2007.
- [12] R Development Core Team. *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.
- [13] B.Y. Reis and K.D. Mandl. Time series modeling for syndromic surveillance. *BMC Medical Informatics and Decision Making*, 3(1) :2, 2003.
- [14] P.A. Rogerson and I. Yamada. Approaches to syndromic surveillance when data consist of small regional counts. *Syndromic Surveillance*, page 79, 2004.
- [15] R.E. Serfling. Methods for current statistical analysis of excess pneumonia-influenza deaths. *Public Health Reports*, 78(6) :494, 1963.
- [16] D. Siegmund. *Sequential analysis : tests and confidence intervals*. Springer, 1985.
- [17] A. Thierry. Statistique épidémiologie. *Editions Maloine, Collection «Sciences fondamentales*, 2002.