Statistical analysis of road accidents in the region Franche-Comté : risk factors for accident injuries and spatial modelling for accident occurrences

Cécile Spychala

Statistical analysis of road accidents in the region Franche-Comté: risk factors for accident injuries and spatial modelling for accident occurrences

Analyse statistique des accidents routiers dans la région Franche-Comté: facteurs de risque pour la gravité de l'accident et modélisation spatiale de l'occurrence de l'accident

Thèse de doctorat de l'établissement Université Bourgogne Franche-Comté

préparée à l'Université de Franche-Comté

École doctorale n 553 CARNOT-PASTEUR

présentée et soutenue publiquement le 09 décembre 2022
à l'Université de Franche-Comté en vue de l'obtention du grade de

Docteur de l'Université de Bourgogne Franche-Comté

(mention Mathématiques)

par

Cécile Spychala

Composition du jury:

Anne Gégout-Petit	Université de Lorraine	*Présidente du jury*
Marie Kratz	ESSEC Business School	*Rapporteure*
Anne Ruiz-Gazen	Université Toulouse 1 Capitole	*Rapporteure*
Benjamin Taylor	Lancaster University	*Examinateur*
Joël Armand	École de Gendarmerie de Dijon	*Invité*
Manon Blandin	Grand Besançon Métropole	*Invitée*
Camelia Goga	Université de Franche-Comté	*Directrice de thèse*
Clément Dombry	Université de Franche-Comté	*Directeur de thèse*

Je dédie cette thèse à mes parents, Maryan et Ilma, pour la force dont ils font preuve quotidiennement, notamment face à la maladie. Na Zdrowie, Salute et Živjeli !

Remerciements

Je tiens dans un premier temps à témoigner ma profonde reconnaissance à Camelia et Clément. Je ne pouvais rêver mieux comme encadrants et collègues. Cela a été un grand plaisir de travailler avec eux. Ils ont toujours été à l'écoute et de très bons conseils, et ce, depuis bien avant le début de la thèse. J'ai beaucoup appris grâce à eux de part leurs connaissances, leur rigueur, leur expérience... Je m'estime chanceuse d'avoir pu partager ma passion des mathématiques avec eux. Enfin, ils ont été bien plus que mes directeurs de thèse, ils ont été également un soutien constant durant ces trois années. J'ai découvert à quel point tout deux faisaient preuve d'une gentillesse incroyable envers moi. Ils m'ont tout de suite épaulé lorsque cela a été difficile pour moi, au cours de la deuxième année de thèse, sur le plan personnel. Et cette relation, bien plus que professionnelle, n'a pas de prix et je ne les remercierai jamais assez. Merci Camelia. Merci Clément.

Enfin dans un deuxième temps, la personne ayant également joué un rôle important dans cette thèse est Ben. Je n'aurais jamais pensé, le jour où je l'ai contacté innocemment concernant ses travaux, faire cette collaboration me permettant de lui rendre visite pendant 4 semaines à Lancaster en Angleterre et de réaliser étroitement avec lui une partie des travaux de cette thèse. J'ai rencontré une personne formidable et d'une gentillesse énorme. Merci Ben.

Je tiens également à adresser mes sincères remerciements aux membres de mon jury de thèse. Je remercie Marie Kratz et Anne Ruiz-Gazen qui ont accepté d'être les rapporteures de cette thèse. J'ai été heureuse de rencontrer Marie Kratz lors d'une conférence à Lyon en Juin 2022, qui m'a tout de suite exprimé son intérêt dans les travaux que j'avais présenté ce jour là. Quant à Anne, j'ai également été heureuse de la rencontrer en tant que chair de ma session lors d'une conférence à Besançon en Décembre 2021 et de partager un repas convivial avec elle, quelques jours suivants, à Dijon. J'ai apprécié les commentaires que vous avez porté sur mon manuscrit.

Je tiens aussi à remercier Anne Gégout-Petit qui a accepté de faire partie de mon jury de thèse. Ce fut un plaisir de la revoir à Lyon en Juin 2022 lors d'une conférence, elle qui avait gentiment accepté l'année précédente d'être dans mon comité de suivi de thèse. Merci pour les retours et les conseils exprimés à ces occasions.

Merci à Joël d'être présent afin de représenter la Gendarmerie Nationale et qui, malgré le COVID, a toujours exprimé un très grand intérêt pour cette thèse. Merci aussi à Manon Blandin d'avoir tout de suite été intéressée pour faire partie du jury et de représenter le Grand Besançon.

Je tiens à remercier vivement tous ceux du Laboratoire de Mathématiques de Besançon (LmB) qui, de près ou de loin, ont participé à cette thèse. Je pense notamment à Julien qui s'est arraché les cheveux avec moi sur le plan informatique, il m'a accordé un temps précieux et n'a jamais refusé de discuter longuement avec moi de nos passions pour la programmation. Merci à Charlène et Pascaline pour leur disponibilité et leur service, sans qui, aucune conférence ni aucun voyage n'aurait été

possible. Elles sont également toujours présentes et prêtes à aider pour quoi que ce soit. Merci à Ulrich pour mon comité de thèse, ses retours et ses conseils. Merci à Christophe le directeur du LmB et Louis le directeur de l'école doctorale Carnot Pasteur. Et merci à tous les autres, Romain, Mathilde, Marsault, Jean-Jil, Davit, Yacouba, ...

Je tiens à adresser des remerciements particuliers à deux membres du LmB, enfin, un ancien membre et un actuel, Mehdi et Valentin. D'abord collègues doctorants puis amis. Ils m'ont tout de suite soutenu quand ça n'allait pas lors de la deuxième année de thèse. Ils ont tout pris sur eux pour moi et m'ont protégé. Et outre cette mauvaise période de la thèse, ils m'ont fait rire tout le reste du temps. Nous avons ajouté "é" ou "able" à certains mots, j'ai été arbitre de leurs matchs de ping pong, j'ai parfois participé même si je me prenais une raclée, et j'ai partagé de nombreuses bières avec eux. Merci les gars, ça n'aurait pas été pareil sans vous.

Enfin je tiens à remercier mes amis et ma famille. Merci à mes parents Maryan et Ilma qui m'ont donné le goût des Mathématiques et sans qui je ne serais pas là aujourd'hui en vue de l'obtention d'un doctorat en Mathématiques, ils m'ont toujours poussé à donner le meilleur de moi-même et m'ont toujours soutenu. Merci spécialement à ma mère qui a tout pris sur elle lors de ma deuxième année de thèse, elle a fait disparaître nombreuses de mes inquiétudes et a tout porté pour moi et mes soeurs.

Merci à mes soeurs Marie et Laurine qui m'ont toujours répété que j'étais la "meilleure". Une pensée particulière pour Laurine et Micka, je ne vous remercierai jamais assez pour tout ce que avez fait. C'est difficile aujourd'hui d'avoir quitté Besançon car depuis huit ans ma soeur et mon beau frère m'accueillaient toutes les semaines chez eux, ils ont tout vécu avec moi.

Merci à Claire et Marie, mes meilleures amies, qui m'ont toujours soutenu et encouragé. Les études des Mathématiques ont commencé avec elles, dans un quatre-vingt mètres carrés à dix minutes du campus. Merci au Y qui m'a énormément soutenu et également encouragé pour cette thèse. Il m'a toujours conforté dans ce que je faisais, même dans mes périodes de stress extrêmes. Merci à Coraline, Jean-Baptiste et Arthur, mes collègues du master, devenus amis aujourd'hui. Une pensée particulière pour Coraline qui a été avec moi jusqu'au bout du master et que j'attends impatiemment en Haute-Savoie. Merci à mes amies proches Camille et Agathe qui m'ont toujours témoigné du soutien et qui ont tout de suite exprimé le souhait de partager cet événement important avec moi et être présentes le jour J. Et enfin merci à tous les autres, Lucie, Léane, Amandine, Pierre, Mich, ...

Merci à tous.

Contents

List of Figures

List of Tables

Introduction

Over the very last years, the Gendarmerie Nationale is interested in using *Machine Learning* and *Artificial Intelligence* methods in order to effectively find solutions to different issues of national interest such as the prevention of the cybercrime, the airport security as it is stipulated in the document of the Gendarmerie Nationale *"Plan stratégique de la recherche et de l'innovation 2017-2022"*. The prevention of road crashes is not clearly stated in this document but it obviously stands in this innovative research direction and may be of great interest for the Gendarmerie Nationale. The origin of the work presented in this report is a small study asked in 2019 by the *Gendarmerie Nationale de Besançon*, more precisely by colonel Joël Armand, focused on the multivariate analysis of road crash data from Franche-Comté. I had the opportunity to be part of the 2nd Master degree student team in charge of carrying out this project and the results obtained after one month work were presented to the members of the *Gendarmerie Nationale de Besançon* including the colonel Joël Armand. The *Gendarmerie Nationale de Besançon* was very interested in continuing this preliminary statistical analysis and funding from the *Grand Besançon Métropole (CAGB - Communauté d'Agglomération du Grand Besançon)* enabled this project to continue in the form of a thesis at the Mathematics Laboratory of Besançon (LmB - *Laboratoire de Mathématiques de Besançon*) from the University of Franche-Comté and started in 2019. The first year of this thesis project faced many difficulties due to COVID-19 and impacted the rest of the thesis. First of all, road crashes data from 2020 and 2021 could not be used because the numerous lockdowns made them atypical. In addition, meetings and communications with the Gendarmerie Nationale, the Police Nationale and the Doubs prefecture in order to collect data and to discuss their interests in terms of accidentology were impacted and thus caused an important delay in the progress of the work. Finally, it was impossible to present the work at statistics conferences in 2020 as almost all national and international conferences have been cancelled.

This thesis is structured into two parts entitled *Injuries analysis of road accidents* (one chapter) and *Space and time analyses of road accidents* (three chapters). I give below a short presentation of each part and chapters and I finish with some personal feelings about this thesis.

Part I: Injuries analysis of road accidents

Data used in the first part of the thesis are extracted from the French census of road crashes BAAC files (*Bulletin d'Analyse des Accidents Corporels*) available on the open-source French government website *www.data.gouv*. These data-sets have been enriched with supplementary information concerning the consumption of alcohol and drugs obtained under confidentiality agreements with the *Gendarmerie Nationale de Besançon*. More specifically, our final datasets contain injury road accident characteristics filled in by the security forces present on the accident scene (such as collision type, hurt obstacle, ...) reported on 4 950 accidents that occurred between 2005 and 2018 in the French region of Franche-Comté. The first part of the thesis aims at giving a multivariate statistical analysis of road crash data with special attention to road crash gravity.

The first step of this multivariate analysis was to perform Multiple Correspondence Analysis (MCA) in order to assess associations between the road crash injury and several important accident related factors. Several multiple correspondence analysis have been performed by considering separately the temporal type variables such as the time of the week or the day; the spatial type variables such as the canton and finally, the spatio-temporal type variables. These analyses highlighted the fact that associations may exist between the severity of the accident (*slight*, *serious* or *fatal*), the department of the region (*Doubs*, *Jura*, *Haute-Saône* or *Territoire de Belfort*) and the alcohol and/or drug consumption of the drivers. Besides, it was noted that it may exist possible associations between the accident severity and several temporal type variables such as the season of the year, the day of the week and the hour of the day.

Log-linear models are used next in order to detect which associations between road crash severity and related factors such as alcohol/drug consumption or spatial crash locations are significant. Based on the associations revealed in a descriptive and geometric way with the MCA, we consider a hierarchical and a non-hierarchical log-linear model. The two models allow us to conclude that the considered associations are significant and to quantify the risks by estimating the associated odds ratios. An important finding, for us as well as for the *Gendarmerie Nationale de Besançon*, is the quantification of the estimated risks related to the accident severity. For example, the risk that the accident is fatal is multiplied by six if the driver has consumed alcohol and drugs. The Jura department also stands out as the riskiest department. These results may encourage the *Gendarmerie Nationale de Besançon* to take preventive measures to reduce the alcohol and drug consumption among drivers.

Finally, ordinal logistic regression was used in order to test the influence of each factor on the accident severity. Eight factors, such as the alcohol and/or drug consumption or the time of the day, were found to be highly influential on crash severity. Odds ratios estimated from this model allowed to quantify the risks of the accident to be "serious" (i.e. the accident has at least one hospitalized person) or "fatal" (i.e. the accident has at least one dead person) and confirmed the results obtained with the log-linear modelling. They also revealed new results such as the fact there is a significant risk that an accident will occur within the time slot 4pm - 7:59pm.

The results presented in the first part of the thesis are published in the article entitled *Multivariate statistical analysis for exploring road crash related factors in the Franche-Comté region of France* in *Communications in Statistics: Case Studies, Data Analysis and Applications* (Spychala et al., 2021). I presented them to several members of the *Gendarmerie Nationale de Besançon* in 2020, 2021 and I have planned to present them to several conferences in statistics during 2020 and 2021 which have been unfortunately cancelled due to COVID-19 crisis.

Part II: Space and time analyses of road accidents

While the first part of this thesis is focused on the statistical modelling of the injury accidents, so after the occurrence of the road accidents, the objective of the second part of this thesis is dedicated now to predict the occurrence of road accidents, so before an accident happens. More precisely, the main goal is now to spatially (eventually spatio-temporally) predict the occurrence of the accidents in order to identify the critical geographical zones and to build maps indicating the risky areas. Our study is motivated, in a second time, by determining the riskiest factors. In order to meet these goals, the geographic coordinates of the road accidents (recorded after the occurrence of the accident) are used in specific spatial statistic models.

The data considered in the second part of the thesis concerns road accidents that occurred in the *Communauté d'Agglomeration du Grand Besançon* (CAGB) between 2017 and 2019. Road crash data are also extracted from the French government website *www.data.gouv*. In order to conclude effectively on the critical areas of the CAGB and to identify the risk factors, additional information was brought to the statistical analysis through covariates. The covariates in question are now different from those used previously in the first part of the thesis as the objectives are different: we wish to predict the occurrence on an accident, while in the first part the accident had already happened. The covariates will be represented by environmental factors that can potentially influence the occurrence of road accidents such as the population density of an area, the number of shops in an area or the length of a given road in an area. Gathering all this information required considerable effort and encountered many obstacles. A major difficulty met on collecting such pertinent auxiliary information was the lack of measures on the traffic density at a fine scale such that the number of cars or taxi per day or week, information that generally turns out to be a relevant factor in road accident statistical analysis. The traffic measures are actually collected by meters installed at several locations from Besançon, every year, during one week of September. This poor information of the traffic density in Besançon greatly limits the choice of statistical models. We suggest in Chapter 4 a first statistical analysis based on such traffic data. To use high-performance machine-learning models would be desirable but requires very rich traffic data which is hard to collect without smart meters, devices that are not intended to be installed in the city of Besançon as far as we know.

The second part of the thesis is divided into three chapters. In addition to the presentation of the statistical analyses carried out in these chapters, particular attention is paid also to the implementation in `R` (R Core Team, 2021) of all these methods. This detailed

description aims at providing in an educational way both technical and practical elements and it was inspired from reference books in statistics with applications in `R` such as Baddeley et al. (2015).

Spatial point process analysis

Chapter 2 concerns the data preparation in order to be used next in the statistical methods developed in Chapter 3. As mentioned above, environmental factors will be incorporated into the analysis. Focus is on gathering rich socio-demographic data and road information concerning the CAGB by using open-data collected on numerous sites such as *www.data.gouv, www.insee.fr, www.openstreetmap.fr*. This data collection required merging several databases on a single support. Usually, spatial analysis is done on a computational grid and significant data wrangling work has been carried out by using interpolation methods in order to standardize different information sources on the chosen grid. This chapter gives also numerous guidelines for using specific `R` packages dedicated to manipulation of spatial data such as `sf` (Pebesma et al., 2022), `spatstat` (Baddeley et al., 2021b) and `lgcp` (Taylor et al., 2021).

The statistical modelling considered in this second part of the thesis treats the road crashes as the realization of an underlying stochastic process called *point process*. Road crash data are considered then as a *spatial point pattern*. The essential element in the study of point processes is the expected number of events per unit area or the *intensity* of the point process. The context of road accidents directly implies the consideration of the intensity of the process as a random process. We suggest in Chapter 3 to model the road crash data by a Cox log-Gaussian process (Moller et al., 1998), namely a Cox process with the log-intensity modelled by a latent Gaussian process and linearly related on several covariates previously prepared in Chapter 2. To estimate the intensity of the process, and therefore the average number of road accidents per unit area, we used Bayesian inference methods implemented by using MCMC computation tools. A second concern focused on obtaining a powerful intensity process model in terms of explanation and prediction while having a reasonable computation time. To reduce the computational burden, we select first the most important covariates by considering a variable selection ensemble method inspired from the random forest algorithm and an importance variable criterion based on variable permutation as suggested in (Breiman, 2001). However, the variable selection method is not applied directly on LGCP models, as they are very time-consuming, but on Poisson regression models since we considered that the Poisson regression is the model that is the closest to the LGCP model while being very fast. Several sets of covariates are thus selected and fitted in LGCP type models. The best LGCP model (chosen according to a weighted mean square criterion) is used to create risk areas maps for the CAGB and to identify the risk factors. The riskiest area of the CAGB, for example, corresponds to the area with center given by the intersection of roads $D673$ and $N57$ known to be a high traffic density area.

The results presented in Chapter 3 are the subject of an article in preparation. I have also presented them in several national and international conferences in statistics.

Spatio-temporal semiparametric analysis

While chapters 2 and 3 were concerned by the spatial component of the road accidents, Chapter 4 focuses on temporal coordinates as well. The main objective is now to build maps showing the risky areas relative to space and time. The road accidents considered here occurred in the city of Besançon between 2017 and 2019 and it was extracted as usual from *BAAC* files.

The statistical modelling considered in Chapter 4 is inspired by epidemiological studies, more particularly by *case-control studies*. However, to carry out a spatio-temporal case-control study in our situation, we only have the cases (road accidents). A major difficulty in setting up this study is the absence of the control group. To overcome this issue, we proposed to generate the control-cases by simulating realizations of a point process on the road network (Baddeley et al., 2021a) having as intensity the traffic density. As detailed in the general description of the part II of the thesis, the traffic density information is measured only at several locations from Besançon during one week of September. We propose then to extrapolate the traffic density to any location of the city of Besançon by using the method of kriging used in geostatistics (Cressie and Wikle, 2011). Finally, a semiparametric modelling is proposed in order to estimate the probability that a geolocated point is an accident relative to time and to identify the risk factors.

This work is very recent and is the result of a collaboration with Benjamin Taylor (the main author of the package `lgcp` Taylor et al. (2021)) from the University of Lancaster. I visited the University of Lancaster for a month last May and started a research collaboration with Benjamin Taylor.

Personal considerations

Finally, I wish to give at the end of the introduction chapter some personal thoughts. In my opinion, a statistical analysis can be defined as a *Brainstorming* or *MindMapping* which means that, starting from a study goal, the ideas and the steps are assembled around this specific objective by sharing with specialists of the domain, theoreticians and/or practitioners. The methods used in order to give answers to the issues may vary according to the study goal and especially to the available data. What is magical with the statistics is that there exist several strategies and methods available in order to achieve our goals (if possible). The reader may find in this thesis the solutions that I proposed in order to fulfill the goals that we fixed at the beginning of the thesis, however several different statistical methods were potentially also suitable to meet the same expectations. The subject of "Road accidents" actually weaves a large statistical web.

This thesis represented a huge opportunity for me. At the end of the Master degree, I intended to learn more on the field of statistics and applied statistics and this is why I wished to continue with a PhD. During the last three years, I have gained even more valuable statistical knowledge, especially in spatial statistics which was an unknown field for me. I have also improved my skills in terms of computational implementations as spatial tools represent a very large sphere to explore. Besides, the thesis also enabled me to become more

open even if, unfortunately, the years 2020 and 2021 have been difficult due to the COVID-19. I had the chance to participate to many conferences and above all to go to Lancaster in England several weeks. This was a great experience for me to live in Lancaster and to meet people from the University of Lancaster.

The major part of what seemed very important for me during this thesis work such as the specific literature review, the statistical methods and the computational tools necessary to realize the analyses, has been structured and presented in the following of this report. I hope that the reader will appreciate what took me three years to produce.

Publications

Spychala, C., Armand, J., Dombry, C. and Goga, C. (2021). "Multivariate Statistical Analysis for Exploring Road Crash Related Factors in the Franche-Comté region of France", *Communications in Statistics - Case Studies and Data Analysis*, 7, 442 - 474.

Works in progress

Spychala, C., Dombry, C. and Goga, C. (2022). "Log-Gaussian Cox Processes and Variable Selection Methods for Accident Data: a Case-Study on the urban community of Besançon data".

Communications in conferences

1. Spychala, C., Dombry, C. and Goga, C. (2022). "Spatial modelling road accidents in Besançon using Log-Gaussian Cox Processes", *24th International Conference on COMPUTATIONAL STATISTICS*, 23-26 august, 2022, Bologna, Italy.

2. Spychala, C., Dombry, C. and Goga, C. (2022). "Spatial modelling road accidents in Besançon using Log-Gaussian Cox Processes", *53es Journées de Statistique de la SFdS*, 13-17 juin, 2022, Lyon.

3. Spychala, C., Dombry, C. and Goga, C. (2021). "Spatial modelling road accidents in Besançon using Log-Gaussian Cox Processes", *Forum des Jeunes Mathématicien.ne.s*, 8-10 décembre, 2021, Besançon)

4. Spychala, C., Dombry, C. and Goga, C. (2021). "Spatial modelling road accidents in Besançon using Log-Gaussian Cox Processes", *Journée de la Fédération de Bourgogne Franche-Comté Mathématiques*, 19 novembre, 2021, Dijon.

Communications in seminars

1. Spychala, C., Dombry, C. and Goga, C. (2022). "Spatial modelling road accidents in Besançon using Log-Gaussian Cox Processes", *CHICAS (Centre for Health Informatics, Computing And Statistics) seminar*, May the 11^{th}, 2022, University of Lancaster, England.

2. Spychala, C., Dombry, C. and Goga, C. (2021). "Spatial modelling road accidents in Besançon using Log-Gaussian Cox Processes", *Séminaire de l'équipe Probabilités et Statistiques du LmB*, 8 novembre, 2021, Université de Franche-Comté, Besançon.

3. Spychala, C., Dombry, C. and Goga, C. (2021). "Spatial modelling road accidents in Besançon using Log-Gaussian Cox Processes", *Séminaire doctorant du LmB*, 14 octobre, 2021, Université de Franche-Comté, Besançon.

I Injuries analysis of road accidents

Chapter 1

Multivariate statistical analysis for exploring road crash related factors in the Franche-Comté region of France [1]

Multiple Correspondence Analysis, log-linear models and ordinal logistic regression

Understanding and modelling road crash data is crucial in fulfilling safety goals by helping national authorities to take necessary measures to reduce crash frequency and severity. This work aims at giving a multivariate statistical analysis of road crash data from the French region of Franche-Comté with special attention to road crash gravity. The first step for this multivariate analysis was to perform Multiple Correspondence Analysis in order to assess associations between the road crash injury and several important accident related factors and circumstances. Log-linear models are used next in order to detect associations between road crash severity and related factors such as alcohol/drug consumption or spatial crash locations. The effects of each factors have been also evaluated on the road crash gravity by using ordinal logistic regression. Data used in this study are extracted from BAAC files, the French census of road crashes.

1.1 Introduction

Over the last decade, the number of road crashes has continuously been decreasing in France. Indeed, 61 224 accidents have been recorded in 2017 instead of 58 352 in 2018, a decrease of 4,7% (ONISR, 2019). However, road accidents still happen and important efforts and means are developed to prevent them. Among these, modern statistical methods are

[1] This chapter leads to an article that has been accepted for publication in Communications in Statistics: Case Studies, Data Analysis and Applications.

efficient prevention tools used to describe and model accident data. This paper is concerned about road accidents that occurred in the Franche-Comté region of France (see Fig. 1.1). This region from the east of France is split up into four departments called Doubs, Jura, Haute-Saône and Territoire de Belfort. Regarding to the mortality rate from 2017 to 2018, this rate has globally decreased for this region. However, the situation is quite different within each department. Indeed, the death rate has increased by 3% from 2017 in the Doubs department while it has decreased in the Haute-Saône, Jura and Territoire de Belfort departments by 45%, 65% and respectively by 50% (ONISR, 2019). Understanding and modelling accident data is crucial in fulfilling safety goals by helping national authorities to undertake necessary measures to reduce crash frequency and severity.

This paper focuses on accidents in Franche-Comté involving casualties. An accident refers to a road crash with casualty needing hospital care and can involve several cars and several people. One of the main goals of the National Gendarmerie of Besançon (Doubs, France) is to reduce the number of accidents in Franche-Comté. More precisely, the National Gendarmerie of Besançon plans to be able in the near future to anticipate road crashes by using time and spatial modelling of accident data. This study aims at giving a multivariate statistical analysis of the road crashes in Franche-Comté. A first multivariate descriptive study of French accident data was conducted by Bièvre (2017) in an unpublished technical report. We intend in this work to give a deeper analysis of Franche-Comté accident data.

The main goal of this research work is to explain the variable giving the severity or the gravity of the accidents by using several covariates such as spatial location, time period, weather conditions, road type, alcohol/drug consumption... Our multivariate statistical analysis starts with a Multiple Correspondence Analysis (MCA). The MCA as suggested by Benzécri (Benzécri, 1973, 1982) is the generalization of the Correspondence Analysis (CA) for analysing jointly more than two categorical variables. This method is widely used in categorical data analyses because it allows detecting similarities between individuals and assessing associations between categories. Geometric representations of data clouds in smaller dimension spaces allow identifying clusters of similar individuals and of associated categories or variables. Many applications of MCA and related methods in various fields such as social, demographic, economic are given in Greenacre and Blasius (2006). The goal here is to determine the accident factors mostly related to road crash severity. In the literature concerning the accident analysis and prevention, several studies used MCA in various contexts but different from our framework. For example, Das and Sun (2015) used eight years of pedestrian crash data and MCA to identify key associations between risk factors and Das and Sun (2016) used MCA to identify crash-prone factors producing fatal run-off-road crashes; Das et al. (2018) investigated the wrong way driving crash patterns by using MCA while Fort et al. (2019) tried to explain working conditions and risk exposure of employees whose occupations require driving on public roads.

The MCA analysis conducted on the Franche-Comté accident data set allows us to identify several variables associated with the road crash severity. A more in-depth analysis of these variables is next considered by log-linear modelling (Agresti, 2013). The log-linear model belongs to the class of generalized linear model (McCullagh and Nelder, 1989). In the

case of categorical data, the cell counts of the contingency table are modelled by a Poisson distribution and a log link function is used for the mean. More precisely, the log-linear model specifies how the expected counts depend on the levels of the categorical variables and it allows to quantify the associations and interactions between those variables. Unlike MCA, log-linear models allow getting insight into complex dependence patterns such as conditional or marginal dependence which may exist between several categorical variables. In our framework, we will use log-linear models in order to detect conditional or marginal associations between road crash severity and other variables such as alcohol/drug consumption and spatial location. In a similar way, Abdel-Aty et al. (1998) used log-linear models to explain associations between the driver age and several important factors and circumstances related to the accident. Also, Yannis et al. (2005) performed a log-linear analysis in order to test the significance of first- and second-order effects among various combinations of driver age and engine size categories in relation to two-wheeler accident severity and at-fault risk rates. Then, Abdel-Aty and Abdelwahab (2000) used log-linear models to investigate whether there are associations between the different driver characteristics and alcohol involvement and also in order to identify the high-risk group within each driver factor.

Finally, we propose ordinal logistic regression (Agresti, 2013) to model the gravity level probabilities as a function of explicative covariates such as alcohol/drug consumption, time period and spatial locations. This method widely used in accident data analysis is a popular supervised learning method for analysing dependencies between a binary or multiclass response categorical variable and several explanatory variables. It allows in particular to separate and identify the effects of each explanatory variable on the response variable. Rezapour and Ksaibati (2018) used ordinal logistic regression to investigate the contributory factors that increased the odds of severe single-truck and multiple-vehicle crashes such as characteristics related to driver or vehicle for instance. Then, Mekonnen (2018) has also performed ordinal logistic regression in order to identify the risk factors among driver age, speed record or alcohol consumption for example for severity levels of road traffic accident.

The paper is structured as follows. We first describe in Section 1.2 our data set as well as the analysis methods: MCA is described briefly in Section 1.2.2, log-linear modelling in Section 1.2.2 and ordinal logistic regression in Section 1.2.2. Section 1.3 contains the main results of our study and, lastly, Section 1.4 concludes and proposes several recommendations and perspectives.

1.2 Material and methods

1.2.1 Franche-Comté accident data

Data used in this study concern the Franche-Comté road crashes between 2005 to 2018 which are extracted from the French national analysis bulletin of road traffic injury accidents called BAAC [2] (*Bulletin d'Analyse des Accidents Corporels*). The BAAC data are filled in by the security forces present on the accident scene and next, data are treated, analysed

[2]The reader can find the BAAC open data on the government website https://www.data.gouv.fr/fr/

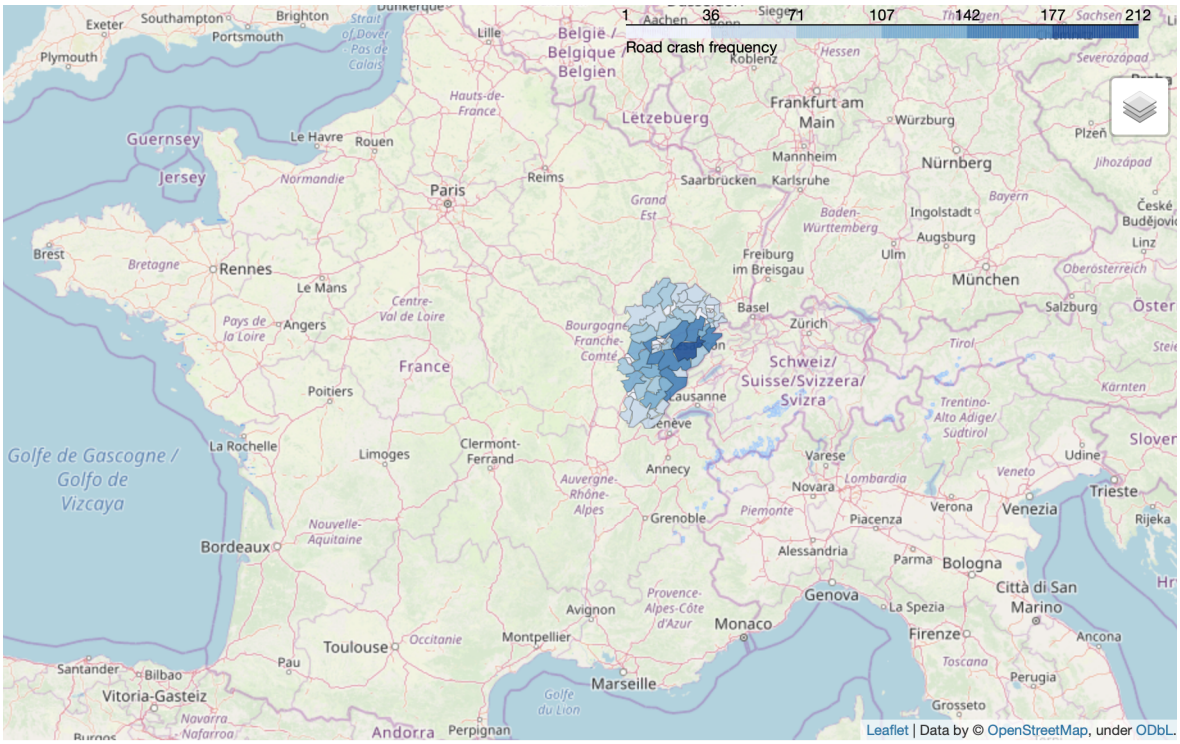

FIG 1.1: France map with road crash frequency of Franche-Comté region. Each small division corresponds to a canton.

and put online by the national interdepartmental observatory of road safety (*Observatoire National Interministériel de la Sécurité Routière*).

The BAAC files contained more than 50 variables from which 15 new categorical variables have been created and/or reclassified. The Franche-Comté accident dataset has 11 776 casualties registered in 4 950 accidents. The study focuses only on the accident itself and not on each casualties. The region Franche-Comté is situated in the east of France and neighboring Switzerland as we can see from FIG. (1.1). The counties situated on the west of Franche-Comté are mountainous and entirely deserved by national and departmental roads. A daily intensive border activity between Switzerland and France is also present in these counties.

The analysis emphasizes the accident severity, denoted by *type_acc*, classified into three ordered levels: “slight_safe”, “serious” and “fatal”. An accident is considered as ”slight_safe” (11,47% of accidents) if all passengers were safe or had minor injuries; the label ”serious” was attributed to accidents involving at least one casualty needing hospital care for more than 24 hours (69,82% of accidents) and lastly, an accident is considered as ”fatal” (18,71% of accidents) if at least one casualty involved died.

The alcohol/drug consumption by car drivers is one of the main accident causes and has a great impact on their severity. The categorical variable *substance* describing the alcohol/drug consumption by the drivers involved in a road crash has the following levels:

- ”alcohol_drug” when at least one of the involved drivers has consumed both alcohol and drugs (2.69%);

- "drug" when at least one of the involved drivers has consumed drugs (2.73%);
- "alcohol" when at least one of the involved drivers has consumed alcohol but not drugs (16.22%);
- "none" is associated with accidents involving only sober drivers (78.36%).

If the accident involves only one driver, the variable *substance* concerns the unique driver.

As mentioned above, the goal of this study is the statistical analysis of accidents and an accident may involve several drivers and casualties. Variables such as *age* or *sex* refer to individuals and are not straightforward to recode for an accident involving several persons. For this reason, *age* or *sex* do not appear in our multivariate analysis.

In order to give a more thorough statistical analysis, we considered further 7 categorical variables giving supplementary information about the weather, the type of the road and of the collision:

- *weather* with two categories: normal and other kind (such as rainy, cloudy or snowy weather);
- *area* with two categories: unurban and urban;
- *intersection* with two categories: intersection and out_of_intersection;
- *obstacle* corresponding to a mobile obstacle with four categories: vehicle, pedestrian, other_kind (such as animals) and none;
- *shape_road* with two categories: curve and straight;
- *collision* with three categories: usual (such as frontal or rear-end collisions), other_kind and none;
- *type_road* with five categories: communal, departmental, national, highway and other_kind (such as parking).

The above variables will be denoted in the paper as *general features*.

In order to conduct the temporal analysis of the Franche-Comté road crashes, we used the following categorical variables related to the time period when the accident occurred, denoted in the rest of the paper as *temporal features*:

- *season* with four categories: spring, summer, autumn and winter;
- *week* with two categories: weekday and week_end;
- *daytime* with two categories: day and night;
- *time* with five categories: 7am_10am, 11am_3pm, 4pm_7pm, 8pm_11pm and midnight_6am. Note that the category 7am_10am means from 7:00 am to 10:59 am. It is also the case for the other categories of *time*.

In our accident data, each accident is located by the *commune* (town or village) and the *department* (Doubs, Jura, Haute-Saône, Territoire de Belfort) where the accident took place. The variable *commune* was used to build the variable *canton* (district) by regrouping the 1176 communes into 50 cantons. In fact, the region of Franche-Comté is splitted up into 62 cantons, however, some cantons have been grouped together as for instance "Belfort-1", "Belfort-2" and "Belfort-3" into "Belfort". This reclassification allows to smooth the variability of *cantons* categories. Hence, Jura department is divided into 15 cantons, Haute-Saône and Doubs both into 14 cantons and Territoire de Belfort into 7 cantons. The categorical variables *department* and *canton* have been used for the spatial analysis, whereas the variable *commune* was dropped due to too many categories. We give in FIG. (1.1) the division of Franche-Comté into cantons with their road crash frequencies; Jura department is the department with the highest road crash frequency. These two variables will be denoted in the rest of the paper as *spatial features*.
TAB 1.1 gives the cross-tabulation of the accident severity (*type_acc*) with the different categorical variables.

TAB 1.1: Contingency tables crossing the accident severity *type_acc* with the other categorical variables. The variable *canton* has too many levels and only "Arbois" and "Villersexel" are described. The numbers in parentheses are, for each categorical variable, the relative frequencies of its different levels and, within each level, the distribution of the accident severity.

Attribute		Type of accident			Attribute		Type of accident		
		slight_safe (11,47%)	serious (69,81%)	fatal (18,71%)			slight_safe (11,47%)	serious (69,81%)	fatal (18,71%)
substance					*department*				
alcohol_drug	133 (2,69%)	10 (7,52%)	61 (45,86%)	62 (46,62%)	Doubs	2060 (41,62%)	255 (12,38%)	1433 (69,56%)	372 (18,06%)
drug	135 (2,73%)	8 (5,93%)	86 (63,70%)	41 (30,37%)	Haute_Saone	1210 (24,44%)	138 (11,40%)	859 (70,99%)	213 (17,60%)
alcohol	803 (16,22%)	5 (7,22%)	542 (67,50%)	203 (25,28%)	Jura	1373 (27,74%)	103 (7,50%)	972 (70,79%)	298 (21,70%)
none	3879 (78,36%)	192 (12,68%)	2767 (71,33%)	620 (15,98%)	Terr_Belfort	307 (6,20%)	72 (23,45%)	192 (62,54%)	43 (14,01%)
season					*canton*				
spring	1199 (24,22%)	151 (12,59%)	810 (67,56%)	238 (19,85%)	Arbois	131 (2,65%)	11 (8,40%)	99 (75,57%)	21 (16,03%)
summer	1580 (31,92%)	167 (10,57%)	1142 (72,28%)	271 (17,15%)	...	...	...	...	...
autumn	1223 (24,71%)	119 (9,73%)	874 (71,46%)	230 (18,81%)	Villersexel	93 (1,88%)	10 (10,75%)	60 (64,52%)	23 (24,73%)
winter	948 (19,15%)	131 (13,82%)	630 (66,46%)	187 (19, 73%)					
					obstacle				
week					vehicle	2484 (42,10%)	321 (12,92%)	1734 (69,81%)	429 (17,27%)
weekday	3133 (63,29%)	395 (12,61%)	2155 (68,78%)	583 (18,61%)	pedestrian	410 (8,28%)	58 (14,15%)	280 (68,29%)	72 (17,56%)
week_end	1817 (36,71%)	173 (9,52%)	1301 (71,60%)	343 (18,88%)	other_kind	146 (2,95%)	21 (14,38%)	98 (67,12%)	27 (18,49%)
					none	1910 (38,59%)	168 (8,80%)	1344 (70,37%)	398 (20,84%)
daytime									
day	3634 (73,41%)	440 (12,11%)	2581 (71,02%)	613 (16,87%)	*shape_road*				
night	1316 (26,59%)	128 (9,73%)	875 (66,49%)	313 (23,78%)	curve	1996 (40,32%)	183 (9,17%)	1427 (71,49%)	386 (19,34%)
					straight	2954 (59,68%)	385 (13,03%)	2029 (68,69%)	540 (18,28%)
time									
7am_10am	785 (15,86%)	79 (10,06%)	566 (72,10%)	140 (17,83%)	*collision*				
11am_3pm	1408 (28,44%)	171 (12,14%)	992 (70,45%)	245 (17,40%)	usual	2995 (60,51%)	407 (13,59%)	2056 (68,65%)	532 (17,76%)
4pm_7pm	1587 (32,06%)	212 (13,36%)	1132 (71,33%)	243 (15,31%)	*other_kind*	1271 (25,68%)	106 (8,34%)	887 (69,79%)	278 (21,87%)
8pm_11pm	618 (12,48%)	63 (10,19%)	410 (66,34%)	145 (23,46%)	none	684 (13,82%)	55 (8,04%)	513 (75,00%)	116 (16,96%)
midnight_6am	552 (11,15%)	43 (7,79%)	356 (64,49%)	153 (27,72%)					
					type_road				
weather					communal	503 (10,16%)	63 (12,52%)	378 (75,15%)	62 (12,33%)
normal	3599 (72,71%)	409 (11,36%)	2550 (70,85%)	640 (17,78%)	departmental	3563 (71,98%)	400 (11,23%)	2488 (69,83%)	675 (18,94%)
other_kind	1351 (37,29%)	159 (11,77%)	906 (67,06%)	286 (21,17%)	national	667 (13,47%)	66 (9,90%)	449 (67,32%)	152 (22,79%)
					highway	92 (1,86%)	20 (21,74%)	51 (55,43%)	21 (22,83%)
area					other_kind	125 (2,53%)	19 (15,20%)	90 (72,00%)	16 (12,80%)
unurban	3460 (69,90%)	337 (9,74%)	2355 (68,06%)	768 (22,20%)					
urban	1490 (30,10%)	231 (15,50%)	1101 (73,89%)	158 (10,60%)	*intersection*				
					intersection	575 (11,62%)	76 (13,22%)	421 (73,22%)	78 (13,57%)
					out_of_intersection	4375 (88,38%)	492 (11,25%)	3035 (69,37%)	848 (19,38%)

1.2.2 Statistical analysis

Multiple Correspondence Analysis

The Multiple Correspondence Analysis (MCA) is an efficient unsupervised method for exploring multivariate categorical data. The aim of MCA is to study the similarities between the individuals, to assess the relationships between the variables and to examine the associations between the categories. For a thorough description of MCA as well as of related methods, the reader is referred to the book of Greenacre and Blasius (2006). This method allows, if appropriate, to corroborate a strong link between categorical variables. In some cases, MCA enables to cluster categories and to reduce the data dimension allowing multivariate data to be analyzed more easily. Indeed, a graphical representation of individuals and variables is built in an orthogonal system similarly as in Correspondence Analysis (CA). This statistical tool is powerful for understanding, visualizing and simplifying the data.

MCA can be derived in several ways. One way is to apply CA on the indicator matrix $\mathbf{X} = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \dots \quad \mathbf{X}_p]$ derived from the original data $Individuals \times Categorical\ variables$ of p categorical variables recorded on n individuals. Each indicator matrix $\mathbf{X}_j$ is obtained by column concatenation of K_j dummy variables where K_j is the number of categories of the jth categorial variable, $j = 1, \dots, p$. Hence, $\mathbf{X}$ is a respondents-by-categories matrix having n rows, corresponding to individuals, and $K = \sum_{j=1}^{p} K_j$ columns, corresponding to variable categories. An element of this table, denoted by x_{ik}, is equal to 1 if the individual i has the category k and 0 otherwise, $i = 1, \dots, n$ and $k = 1, \dots, K$. The indicator matrix $\mathbf{X}$ has row sums equal to the constant p and column sums equal to n_k, the marginal frequency of the kth category, namely the number of individuals having the category k.

This kind of data implies the study of three kinds of objects: the individuals, the variables but also their categories. The scheme of MCA is to compare individuals and evaluate variables characteristics by providing row typologies, column typologies and the relationships between these typologies (Escofier and Pagès, 2008).

From a technical point of view, MCA uses as CA the χ^2 distance in order to assess similarity or dissimilarity between different columns or lines contained in $\mathbf{X}$. The indicator matrix $\mathbf{X}$ is transformed in order to obtain row profiles by dividing each element of a row by the row frequency as well as column profiles by dividing each element of a column by its frequency. In the case of MCA, row and column profiles are very simple. The ith row profile is given by $(x_{ik}/p)_{k=1}^{K}$: the elements of a row profile have only zero and $1/p$ values, the non-zero value being recorded if the individual i possesses the category k. So, row profiles will be different only for ith and i'th individuals having mismatching category levels. The kth column profile is given by $(x_{ik}/n_k)_{i=1}^{n}$: the elements of the column profile are zero and $1/n_k$ values.

The χ^2-distance between two individuals i and i' is a weighted sum of squared distances between the ith and i'th row profiles with weights given by the inverse of the average row

profile given by $(n_k/np)_{k=1}^{p}$:

$$d^2_{i,i'} = \sum_{k=1}^{K} \frac{np}{n_k} \left(\frac{x_{ik}}{p} - \frac{x_{i'k}}{p} \right)^2 = \frac{n}{p} \sum_{k=1}^{K} \frac{(x_{ik} - x_{i'k})^2}{n_k}, \qquad 1 \leq i, i' \leq n. \tag{1.1}$$

Hence, the terms from the above sum will be all zero for coincident zero values and coincident $1/p$ values meaning that these squared differences will not contribute to the distance measure. Only differences between noncoincident categories will contribute to the distance $d^2_{i,i'}$ and this contribution is proportional to $(1/p)^2$ with weight equal to the inverse of the marginal frequency n_k. The χ^2 distance between row profiles can be interpreted as a weighted mismatching dissimilarity coefficient: small distance $d_{i,i'}$ means that individuals i and i' have many categories in commun, so they are very similar and on the contrary, large distance $d_{i,i'}$ means that i and i' have few categories in commun, so they are very different. Moreover, a rare category (small n_k) has a large contribution to the final distance and moves its owner or owners far away from the others individuals.

While the interpretation of the χ^2 distance between individuals is similar to the one given in the CA, the χ^2 distance interpretation for variable analysis is quite different and more difficult to justify (Greenacre, 2006). Information contained in a variable can be studied through its categories, thus, MCA focuses mostly on variable categories. As for row-profiles, the distance between categories k and k' is defined as the weighted sum of squared distances between the kth and k'th column profiles with weights given by the inverse of the average column profile which has in this case all elements equal to $1/n$:

$$d^2_{k,k'} = n \sum_{i=1}^{n} \left(\frac{x_{ik}}{n_k} - \frac{x_{ik'}}{n_{k'}} \right)^2 = \frac{1}{p_k} + \frac{1}{p_{k'}} - \frac{2p_{kk'}}{p_k p_{k'}}, \qquad 1 \leq k, k' \leq K, \tag{1.2}$$

where $p_k = n_k/n$ is the relative frequency of the category k and $p_{kk'}$ the relative frequency of occurence of categories k and k'. If k and k' are different categories of the same variable, then $p_{kk'} = 0$. As it is defined, the distance between column profiles is a decreasing function with respect to the relative frequencies p_k and joint relative frequencies $p_{kk'}$. Two categories are close one to each other with respect to this χ^2 distance if they have many individuals in common. Again, rare categories are far away from the others. In brief, it is important to take the frequency of each category into account. However, as remarked by Greenacre (1989) and Greenacre (2006), the terms $1/p_k$ present in the χ^2 distance are hard to interpret.

Once that distances between objects (individuals and variables) have been defined, the next step in a MCA is to represent individuals and variables in new orthogonal systems and to make the geometric data analysis on smaller dimension sets (Le Roux and Rouanet, 2004). As in principal component or correspondence analysis, new orthogonal systems are built such that they maximise the projected inertia of the individual cloud or variables on these new orthogonal axis, the inertia being defined as usual as the weighted sum of squared distance of individuals or variables to their barycenter. Each axis represents a certain percentage from the total inertia. However, these percentages in MCA are lower than in CA and more dimensions are needed to interpret properly the analysis. Transition relations link the cloud

of individuals with the cloud of categories and a biplot representation is usually used as a joint map of individuals and variable categories. The contribution of each individual to each axis as well as the quality of its representation on each axis are obtained in a similar way to CA. For more details about the graphical representation and all matters connected therewith, see for example Greenacre (2006), Escofier and Pagès (2008, chapter 4), Husson et al. (2016, chapter 3).

Log-linear model

Multivariate categorical data as multidimensional contingency tables (with an order greater than two-way) display relationships between categorical variables. This kind of data can me model by a log-linear model, that is a generalized linear model for Poisson regression. The Poisson distribution is the simplest distribution for count data. The model describes association and interaction among categorical variables and its purpose is to establish dependence patterns between variables. There is no distinction between explanatory or response variables since only the cell counts are considered. The reader may find a comprehensive description in Agresti (2013, chapter 9).

For the sake of simplicity, we present the method for three categorical variables X_1, X_2 and X_3 respectively with K_1, K_2 and K_3 categories. The most general log-linear model for the three-way table $K_1 \times K_2 \times K_3$ is written as

$$\log \mu_{k_1k_2k_3} = \lambda + \lambda_{k_1}^{X_1} + \lambda_{k_2}^{X_2} + \lambda_{k_3}^{X_3} + \lambda_{k_1k_2}^{X_1X_2} + \lambda_{k_1k_3}^{X_1X_3} + \lambda_{k_2k_3}^{X_2X_3} + \lambda_{k_1k_2k_3}^{X_1X_2X_3}, \tag{1.3}$$

where $\mu_{k_1k_2k_3}$ is the expected frequency of the cell with $X_1 = k_1$, $X_2 = k_2$ and $X_3 = k_3$. The model-parameters are interpreted as follows: λ is the overall effect; $\lambda_{k_j}^{X_j}$ is the effect of the level $X_j = k_j$, $j = 1, 2, 3$; $\lambda_{k_jk_{j'}}^{X_jX_{j'}}$ is the interaction effect of levels $X_j = k_j$ and $X_{j'} = k_{j'}$, $1 \leq j, j' \leq 3$; finally $\lambda_{k_1k_2k_3}^{X_1X_2X_3}$ is the interaction effect between the levels $X_1 = k_1$, $X_2 = k_2$ and $X_3 = k_3$. The model (1.3) is called the saturated model, it includes all possible main effects and interactions between the variables. Some constraints between the parameters ensure model identifiability and the number of free parameters in the saturated model is equal to the number of cells $K_1K_2K_3$, which is why the saturated model fits the data perfectly. It reproduces exactly the observed cell frequencies and does not provide much relevant information.

The aim is to find the simplest model that fits the data adequately, that is, a more parsimonious model with less parameters. An unsaturated model is obtained by imposing the nullity of some coefficients in (1.3) and may be more appropriate due to simpler interpretations. Validation is performed thanks to goodness-of-fit assessment comparing the expected cell frequencies to the observed frequencies. The goodness-of-fit can be tested with the likelihood-ratio statistic:

$$G^2 = 2 \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \sum_{k_3=1}^{K_3} n_{k_1k_2k_3} \log \left(\frac{n_{k_1k_2k_3}}{\hat{\mu}_{k_1k_2k_3}} \right),$$

where $n_{k_1k_2k_3}$ and $\hat{\mu}_{k_1k_2k_3}$ are respectively the cell frequencies and the fitted values from model (1.3) taking into account the nullity constraint (Agresti, 1990). The G^2 statistic is used to determine the rejection or acceptance of a model. The larger the value of G^2, the more evidence there is against that the related model does fit the data adequately, hence it should not be kept.

TAB 1.2: Different structures of log-linear models corresponding to different dependence structures. The third column "Symbol" corresponds to model notations, that is, the higher-order model term represented of each variable used in the model.

Log-linear model	Interpretation	Symbol
$\lambda + \lambda^{X_1}_{k_1} + \lambda^{X_2}_{k_2} + \lambda^{X_3}_{k_3}$	mutual independence	(X_1, X_2, X_3)
$\lambda + \lambda^{X_1}_{k_1} + \lambda^{X_2}_{k_2} + \lambda^{X_3}_{k_3} + \lambda^{X_2X_3}_{k_2k_3}$	independence of X_1 and (X_2, X_3)	(X_1, X_2X_3)
$\lambda + \lambda^{X_1}_{k_1} + \lambda^{X_2}_{k_2} + \lambda^{X_3}_{k_3} + \lambda^{X_1X_3}_{k_1k_3} + \lambda^{X_2X_3}_{k_2k_3}$	independence of X_1 and X_2 given X_3	(X_1X_3, X_2X_3)
$\lambda + \lambda^{X_1}_{k_1} + \lambda^{X_2}_{k_2} + \lambda^{X_3}_{k_3} + \lambda^{X_1X_2}_{k_1k_2} + \lambda^{X_1X_3}_{k_1k_3} + \lambda^{X_2X_3}_{k_2k_3}$	homogeneous association	(X_1X_2, X_2X_3,X_1X_3)

Different types of unsaturated log-linear models correspond to different type of dependence between the variables X_1, X_2 and X_3. We will consider here only hierarchical models, meaning that if variables are involved in high order interactions, all the lower-order interaction term must also appear. For example, if the model contains $\lambda^{X_1X_2}_{k_1k_2}$, then it also must contain $\lambda^{X_1}_{k_1}$ and $\lambda^{X_2}_{k_2}$. Table 1.2 summarizes the different types of resulting models which are ordered with increasing complexity. The simplest model, noted (X_1, X_2, X_3), assumes the nullity of all the interaction effects and corresponds to the mutual independence of X_1, X_2 and X_3. The model with no interaction of order 3 and no interaction of second order between X_1, X_2 and X_1, X_3 is noted (X_1, X_2X_3) and corresponds to the independence of X_1 and (X_2, X_3). The model with no interaction of order 3 and no interaction of order 2 between X_1 and X_2 is noted (X_1X_3, X_2X_3) and corresponds to the conditional independence of X_1 and X_2 given X_3. Finally, the model (X_1X_2, X_2X_3, X_1X_3) has all interactions of order 2 but no interaction of order 3 and corresponds to homogeneous association that we will explain below. One goal of the analysis of the log-linear model is to find out which is the simplest model suitably fitting the data.

We now discuss marginal and conditional association of variables. A two-way contingency table can be obtained by marginalizing out the third variable, obtaining the so-called marginal table. Associations in this table are summarized by the marginal odd ratios. The marginal odds ratio of a 2×2 table (of X_1 and X_2) is defined by

$$\theta_{X_1X_2} = \frac{\mu_{11+}\mu_{22+}}{\mu_{12+}\mu_{21+}}.$$

where $\mu_{ij+} = \sum_{k_3} \mu_{ijk_3}$ are the expected marginal frequencies with $i, j = 1, 2$ and k_3 a fixed category of X_3.

The distribution of the two variables X_1 and X_2 can be displayed conditionally on different levels of X_3 using cross sections of the three-way contingency table. The associations in these cross-sections (also called partial tables) are called conditional associations and summarized by conditional odds ratios: for instance the ratio of the odds of a $2 \times 2 \times K_3$ table

is defined by

$$\theta_{X_1X_2(k_3)} = \frac{\mu_{11k_3}\mu_{22k_3}}{\mu_{12k_3}\mu_{21k_3}}.$$

On the other hand, the absence of interaction of order 3 in the model (X_1X_2, X_2X_3, X_1X_3) implies that the conditional odds ratios do not depend on the category of the third conditioning variable (Agresti, 2013). This property explains the term homogeneous association.

In practice, often data sets contain a large number of categorical variables which may have a large number of levels. Hence, using log-linear models as described before would require a large number of higher order interactions. The estimation and interpretation of parameters may be difficult in such situations. To cope with this difficulty, one can restrict the interaction parameters to have some predefined form, for example a product form as suggested by Andersen (1980), Goodman (1986). The resulting model, known as the multidimensional row-column or the RC association model is log-multiplicative rather than log-linear since it contains multiplicative terms for the interactions. The number of parameters from log-multiplicative models to be interpreted are considerably reduced in this way. Coefficients used in these multiplicative terms are closely related to elements from the singular value decomposition associated to the correspondence analysis of the contingency table as described in Van der Heijden et al. (1989). The simple or multiple correspondence analysis may be also used to detect groups of variables or categories of variables which are mostly related. Then, one can fit a log-linear model using only these groups of selected variables/categories of variables. The resulting log-linear model is no longer hierarchical but the number of interactions is considerably reduced.

Ordinal regression model

The logistic regression is a popular supervised learning method for analysing dependencies between a response categorical variable Y (binary or multiclass) and explanatory variables denoted by $\mathbf{X} = (X_1, \ldots, X_p)$. More precisely, the logistic regression is used in order to separate the effects of each variable, that is, identify the effects of an explanatory variable X_j, $j = 1, \ldots, p$, on the response variable Y. The logistic regression for a binary or multiclass response variable will be presented briefly below, for more details see for example (McCullagh and Nelder, 1989, chapter 5), (Agresti, 1990, chapter 9) or (Hothorn and Everitt, 2014, chapter 7).

Let $Y \in \{0, 1\}$ be a binary response variable. The logistic regression model is written as

$$\mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = F(\beta_0 + \boldsymbol{\beta}^T\mathbf{x}),$$

where $\mathbf{x} \in \mathbb{R}^p$, $\beta_0 \in \mathbb{R}$, $\boldsymbol{\beta} \in \mathbb{R}^p$ and $F(t) = e^t/(1 + e^t)$, $t \in \mathbb{R}$, is the inverse logistic link function. The coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are estimated by maximum likelihood method. Equivalently, the log odds of the event $\{Y = 1\}$ given $\mathbf{X} = \mathbf{x}$ is linear in $\mathbf{x}$:

$$\log \text{odds}(Y = 1 \mid \mathbf{X} = \mathbf{x}) = \log \frac{\mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x})}{1 - \mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x})} = \boldsymbol{\beta}^T\mathbf{x}.$$

Finally, a variable X_j reveals to have an effect on the response variable if the result of the nullity coefficient test for β_j is significant, that means, β_j not equal to 0 (several nullity tests exist such as Wald test for instance).

Now, in this study, the focus lies on a categorical variable with more than two categories. Let Y be a multiclass response variable. The logistic regression for a multiclass response variable is an extension of the logistic regression for a binary one. When the categories $\{m_1, \ldots, m_q\}$ of the response variable Y are hierarchically ordered as $m_1 \prec \ldots \prec m_q$, a way to model Y is to suppose that there exists a latent unobserved continuous variable denoted $Y^* \in \mathbb{R}$, with logistic distribution F, such that

$$Y = m_k \quad \text{if and only if} \quad c_{k-1} < Y^* \leq c_k,$$

where $-\infty = c_0 < c_1 < \ldots < c_{q-1} < c_q = +\infty$ and $k = 1, 2, \ldots, q$. Then, the ordinal regression model is written as

$$\mathbb{P}(Y \preceq m_k \mid \mathbf{X} = \mathbf{x}) = F(c_k - \boldsymbol{\beta}^T \mathbf{x}), \tag{1.4}$$

where $k = 1, 2, \ldots, q-1$. Note that the general intercept β_0 is replaced by the set of ordered intercept paramaters c_k mentioned before. The unknown coefficients $c_1, \ldots, c_{q-1}, \beta_1, \ldots, \beta_p$ are estimated by maximimum likelihood.

Model (1.4) is also called the proportional-odds model due to the following property: the log odds ratio of $\{Y \preceq m_k\}$ at $\mathbf{X} = \mathbf{x_1}$ and $\mathbf{X} = \mathbf{x_2}$ is given by

$$\log \frac{\text{odds}(Y \preceq m_k \mid \mathbf{X} = \mathbf{x_1})}{\text{odds}(Y \preceq m_k \mid \mathbf{X} = \mathbf{x_2})} = -\boldsymbol{\beta}^T(\mathbf{x_1} - \mathbf{x_2}),$$

and does not depend on the category m_k.

1.3 Results

This section aims at giving a multivariate analysis of Franche-Comté road crash data by using the above described methods. Our analysis begins by performing several MCA analyses on road crash variables to provide insights multivariate road crash related variables by using geometric data visualization. This method is a powerful tool to distinct non-trivial category associations if it is the case. More exactly, MCA analyzes were performed by considering temporal and spatial features separately and next, a global MCA analysis based on temporal, spatial features and variables which revealed to be most related one to another from separate MCA analysis. General features such as crash gravity, alcohol/drug consumption, road type and so on were included in the three performed MCA analyzes. These MCA analyzes revealed several association and interaction patterns among the set of categorical variables related to road crash gravity and hierarchical and non-hierarchical log-linear models were fitted in second time to describe more thoroughly these associations. Finally, the ordinal regression model allows to quantify the effects of each explanatory variable on the ordered response variable road crash gravity.

This study used open-source R software packages `FactoMineR` (Lê et al., 2008) and `factoextra` (Kassambara and Mundt, 2019) to perform MCA, `glm` function to perform log-linear models, then packages `MASS` (Venables and Ripley, 2002) and `ordinal` (Christensen, 2019) to perform ordinal logistic regression. Graphics were plotted with `ggplot2` package (Wickham et al., 2021).

1.3.1 MCA of road crash temporal variables

We conducted a MCA temporal analysis by considering general and temporal features as described in Section (1.2.1). Spatial features are not included in MCA analysis. Fig (1.2) gives the percentages of variance explained by each of the first ten axes built by the MCA analysis. The first three-factorial axes explain 22,09% of the total variance and only these axes were kept for further analysis. Two-dimensional geometrical representations are given in Fig (1.3)-(1.5) and interpreted below. Each time, only the 25 best represented categories have been plotted.

The two-dimensional map in Fig (1.3) gives the representation of categories on the plane made by axis 1 and axis 2 and it accounts for 16,07% of the total inertia. The more categories a variable has, the more it contributes to the inertia. The variables *season* and *weather* are not represented on the first factorial plane since they are very poorly represented on this plane. Next, categories with the greatest contribution to the axis 1 are "night" (13,26%), "none" from *obstacle* (10,57%) and "midnight_6am" (9,26%) and respectively, "pedestrian" (28,45%), "urban" (19,04%) and "other_kind" from *collision* (11,46%) for axis 2.

In this first factorial plane, axis 1 shows the contrast between weekday accidents (accidents occurring during the week) and weekend ones (accidents occurring during the weekend). Weekday accidents are more frequent during the day and mostly around lunch time, in urban areas, on communal roads and are not associated to alcohol or drug consumption. These accidents are more likely to happen on straight roads, at intersections and caused by collisions between several vehicles. Weekend accidents, instead, are more frequent during the night and mostly between 8 pm and 6 am, outside urban areas and involve more frequently drug consumers. These accidents occur mainly on curve roads and no external factors seem to impact (out of intersections, no bumped mobile obstacles or no collisions). To sum up, this first factorial axis is related to fatal accidents.

Axis 2 provides a similar information as axis 1: it opposes accidents occurred during weekday time, in urban area, at intersection and on straight road to accidents occurred during weekend time, outside urban area, out of intersection and on curve road. However, axis 2 stresses the fact that fatal road crash are more likely to occur between midnight and 6 am especially when alcohol and drugs have been consumed.

Geometrical representations derived in MCA plot closely associated categories and unassociated ones further apart. The first factorial plane reveals several strong associations among categories: categories "straight", "intersection", "weekday", "substance_none", "11am_3pm", "collision_usual" are strongly associated to categories "obstacle_vehicle"; categories "curve", "out_of_intersection, "week_end", "unurban" are strongly associ-

ated to "fatal"; "night", "8pm_11pm", "midnight_6am", "obstacle_none", "alcohol" and "alcohol_drug" in the same way.

On the other hand, this factorial plane also shows that some categories are far from the others, this results from their lower frequencies. Indeed, as it is given in TAB 1.1, "obstacle_pedestrian" and "type_road_other_kind" represents respectively only 8,28% and 2,53% of road accidents.

FIG (1.4) gives the two-dimensional map of axis 1 and axis 3 and it explains 15, 32% of the total inertia. The variable *area* has been omitted from this geometrical representation due to its poor representation quality. Categories that contribute the most to axis 3 are categories "night" (11,82%), "none" (11,08%) of the variable *collision* and category "winter" (10,96%) of the variable *season*.

In this factorial plane, axis 3 suggests that summer accidents tend to differentiate from winter ones. Summer accidents are more likely to occur during the day, around lunch time and on week-end time. They are globally associated to no alcohol or drug consumption, happening on curve roads, out of intersections and other kind of collision. Winter accidents instead occur more frequently during the night, the week time and on national roads. They are also mostly associated with substances consumed, happening on straight roads, at intersections and collisions with vehicles. The associations with straight roads, intersections and collisions with vehicles seem to be caused by weather ("other_kind") which is generally snowy in winter. In addition, the axis 3 specifies that winter accidents are more likely to be fatal.

Two groups of strongly associated categories stand out in the second factorial plane: the group formed by "obstacle_vehicle", "collision_usual", "intersection", "straight", "weekday", "substance_none", "day", "11am_3pm" and the other group formed by "summer"; "night", "8pm_11pm", "midnight_6am", "alcohol_drug". These groups resonates to those mentioned previously.

FIG (1.5) gives the two-dimensional map of axis 2 and axis 3 and it explains 12,79% of the total inertia. From the thirteen variables used, the variables *substance*, *week* and *intersection* are very poorly represented on this map and are omitted from this geometrical representation. This plot emphasizes the differences between serious and fatal accidents which tend to be strongly associated with lunch time and respectively night time. We distinguish two groups of close categories: "winter", "night", "fatal", "8pm_11pm", "obstacle_vehicle", "collision_usual" and "weather_other_kind"; "type_road_departmental", "serious", "weather_normal", "spring", "curve", "day", "summer", "11am_3pm" and "obstacle_none".

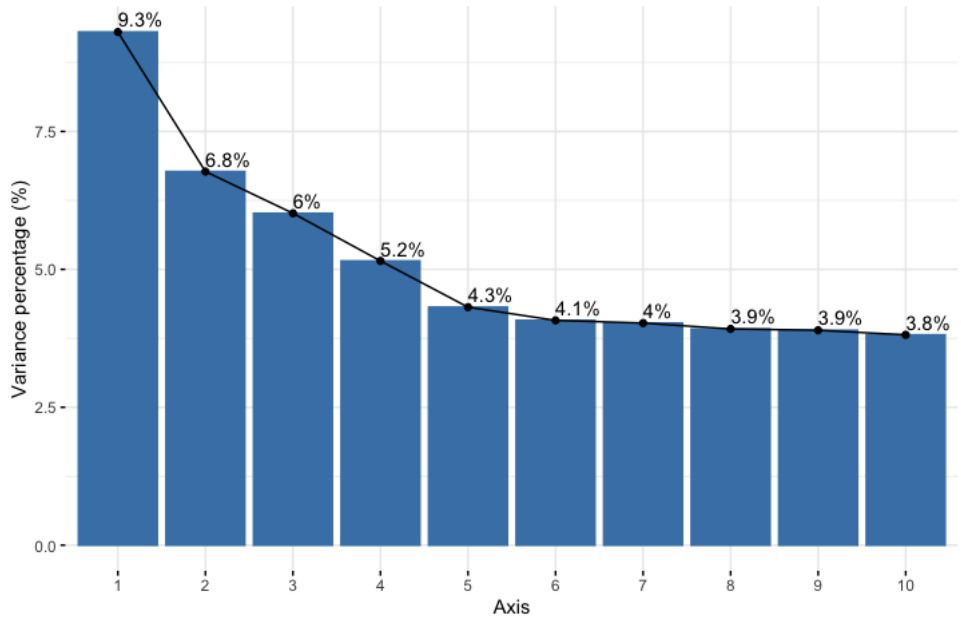

FIG 1.2: MCA temporal analysis: variance percentage explained by the first 10 axes.

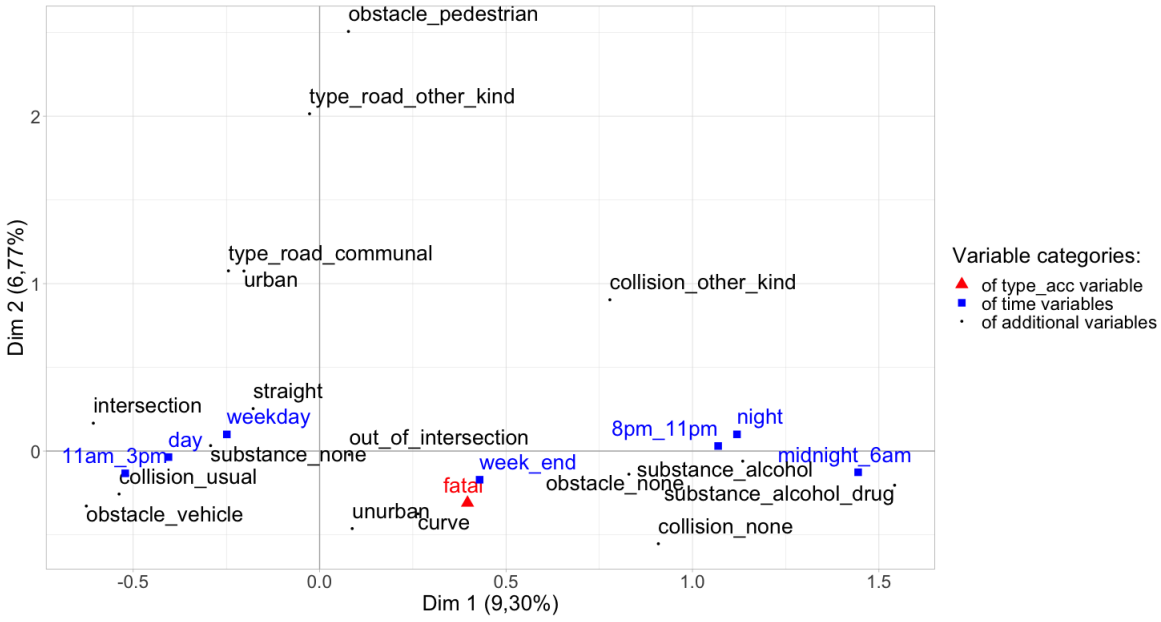

FIG 1.3: MCA temporal analysis: factorial plane made by axis 1 and axis 2.

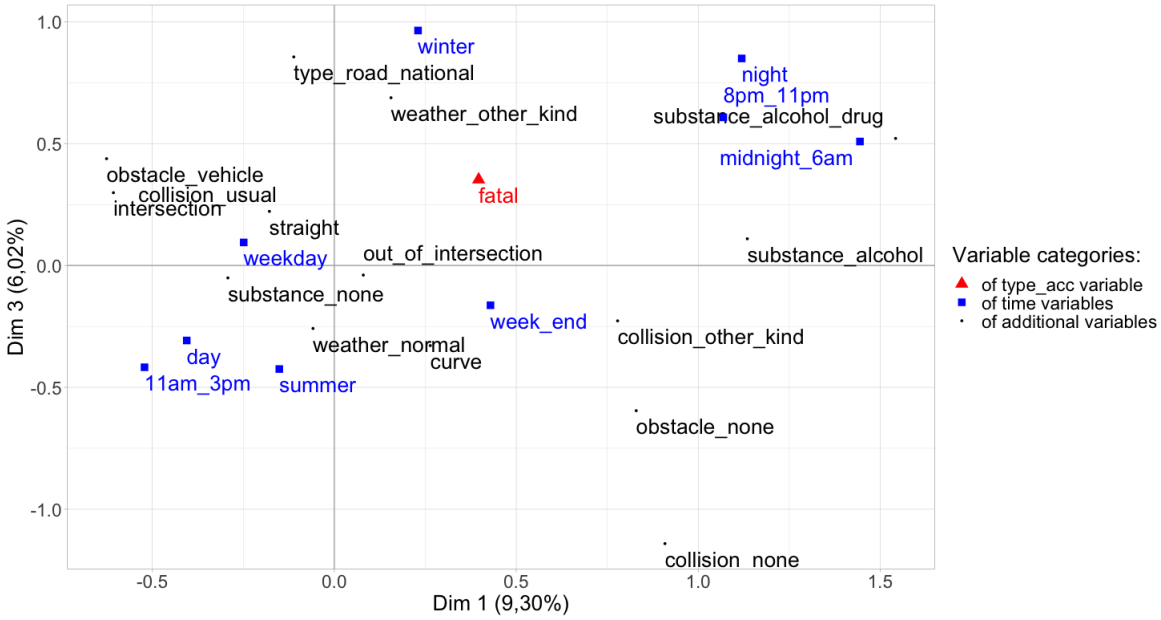

FIG 1.4: MCA temporal analysis: factorial plane made by axis 1 and axis 3.

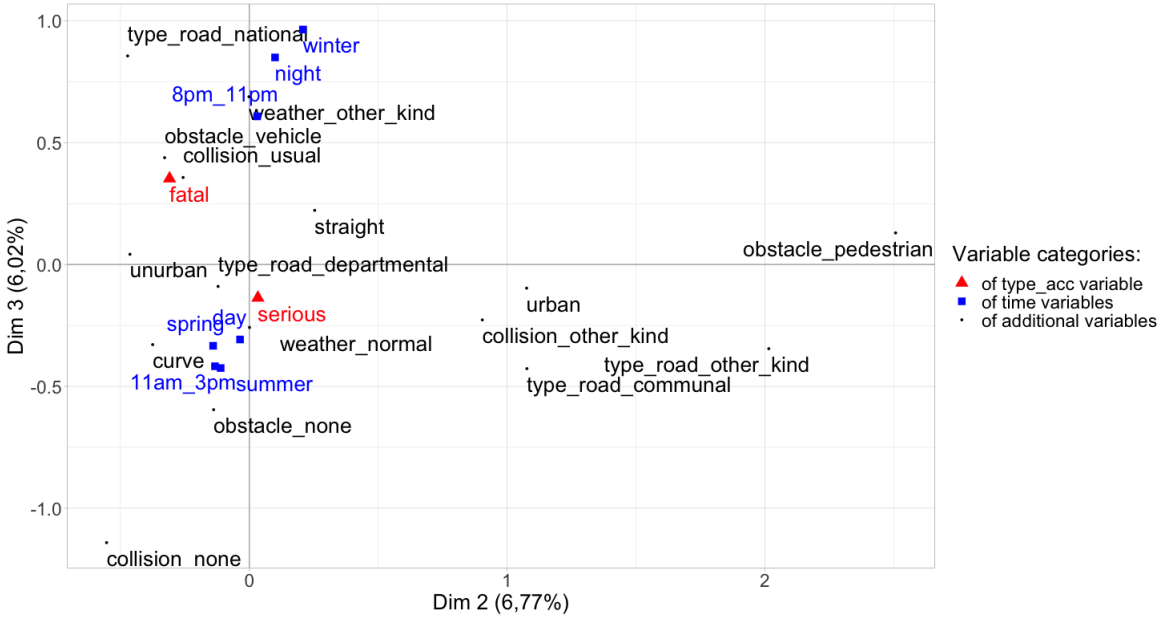

FIG 1.5: MCA temporal analysis: factorial plane made by axis 2 and axis 3.

1.3.2 MCA of road crash spatial variables

We performed a spatial analysis by considering the general features as well as the spatial ones as described in the Section (1.2.1). Temporal variables (*season*, *week*, *daytime* and *time*) are not considered in this analysis. Due to the several categories of the variable *canton*, the spatial analysis of Franche-Comté has been splitted into two analyses corresponding to Doubs and Jura departments. We will interpret only relationships from categories situated close one to another in the plot.

Doubs department

For the spatial analysis of Doubs department, the first four factorial axes explain 21,22% of the total inertia. Note that only the first 25 categories with the most important representation qualities were plotted.

Plot made by axis 1 and axis 2 given in Fig (1.7) explains 12,74% of the total inertia. Not all the variables are plotted, the variable *weather* is less well represented than the other categories and it does not figure on the plot. Categories which contribute the most for axis 1 are "none" (20,77%) and "vehicle" (14,06%) from *obstacle*, then "usual" from *collision* (12,88%); and for axis 2"pedestrian" (22,53%), "urban" (18,92%) and "other_kind" from *collision* (9,73%).

As mentioned before, associations can be highlighted by the proximity of categories on the factorial plot. Two groups of close categories with spatial connotations stand out: "type_road_other_kind", "type_road_communal", "Bethoncourt", "urban" and "Valentigney"; "Besançon", "collision_usual" and "obstacle_vehicle". The first group emphasizes that Bethoncourt and Valentigney accidents are more frequent in urban areas. The second one strongly insists that the canton of Besançon is more conducive to collisions with vehicles.

Additional plots made by combinations of axis 1, 2, 3 and 4 explain between 8,48% and 11,01% of the total inertia, it should be noted that compared to each other the associations do not differ. These plots indicates, in addition to what was said before, that Besançon accidents tend to be fatal when the weather is bad and more characterized by bumped pedestrians; Bethoncourt accidents are mostly associated with substance consumption and very strongly associated to "drug"; Baume-les-Dames accidents are more likely to be fatal; Besançon, Saint-Vit and Ornans accidents are mainly similar and more frequent on communal and national roads; most of Maîche accidents are not caused by collisions.

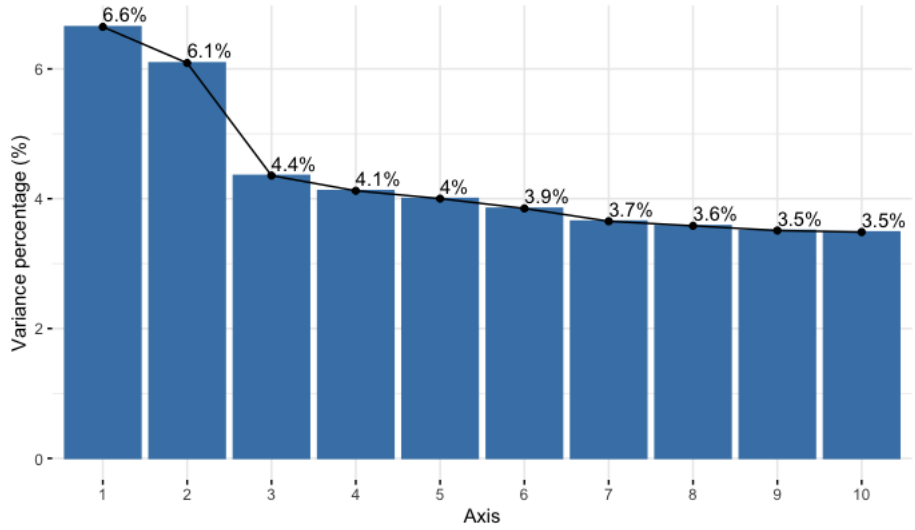

FIG 1.6: MCA spatial analysis, Doubs department: variance percentage explained by the first 10 axes.

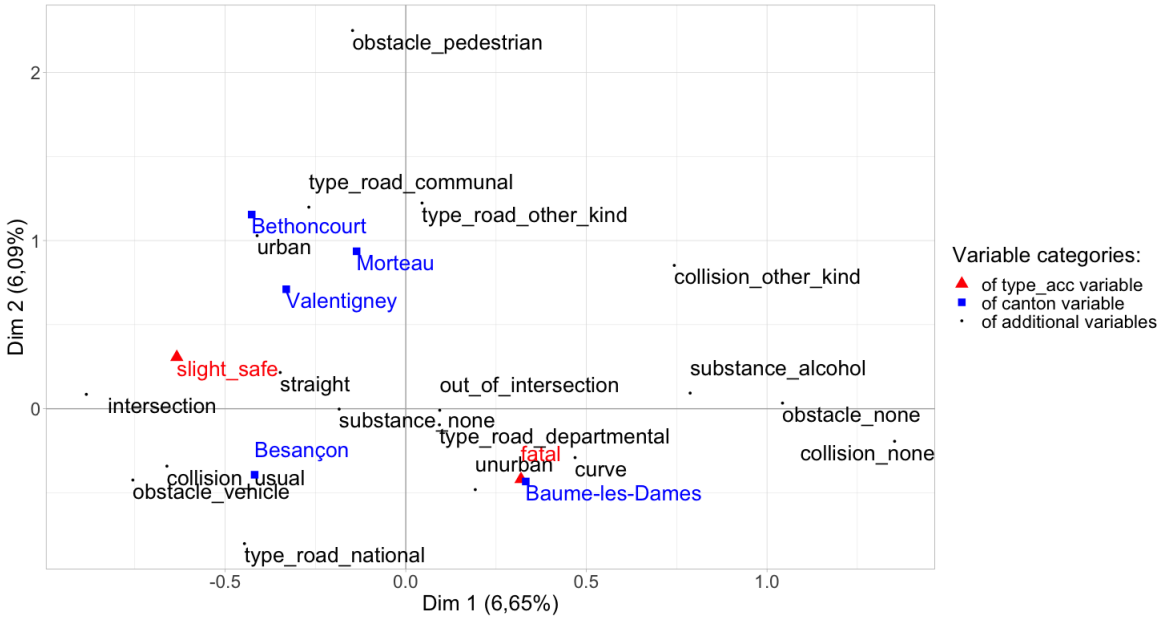

FIG 1.7: MCA spatial analysis, Doubs department: factorial plane made by axis 1 and axis 2.

Jura department

For the spatial analysis of Jura department, the first four factorial axes explain 21,88% of the total inertia. Note that only the first 25 categories with the most important representation qualities were plotted.

Plot made by axis 1 and axis 2 explains 13,08% of the total inertia and is given in FIG (1.9). Not all the variables are plotted, the variable *weather* is less well represented than the other categories and it does not figure on the plot. Categories which contribute the most for

axis 1 are "none" (18,34%) and "vehicle" (13,79%) from *obstacle*, then "usual" from *collision* (11,94%); and for axis 2 "pedestrian" (24,89%), "urban" (18,69%) and "other_kind" from *type_road* (11,23%).

We can distinguish three groups of close categories, with components of the *canton* variable that stand out in this factorial plane: "Saint-Lupicin", "fatal", "drug", "alcohol_drug", "alcohol" and "obstacle_none"; "Authume", "Saint-Laurent-en-Grandvaux", "unurban", "type_road_departmental" and "out_of_intersection"; "Champagnole", "urban" and "type_road_communal". The first group highlights that accidents in the canton of Saint-Lupicin are more likely to be fatal and associated to alcohol and drug consumption. The second one tells that accidents happening in Authume and Saint-Laurent-en-Grandvaux are more frequent in non urban areas, on departmental roads. Then, the third group of close categories shows that accidents in the canton of Champagnole are occuring more commonly in urban areas and on communal roads. Finally, the structure of this factorial plane tells that accidents happening in Authume, Saint-Laurent-en-Grandvaux and Saint-Lupicin are more likely to be fatal.

Additional plots made by combinations of axis 1, 2, 3 and 4 explain between 8,81% and 11,24% of the total inertia, it should be noted that compared to each other the associations do not differ. The cantons of Champagnole, Morez and Saint-Laurent-en-Grandvaux have been associated to each other in many factorial planes, it seems that accidents are more likely to happen in these cantons when the weather is qualified as "other_kind" (cloudy, rainy or snowy). This is opposed to accidents happening in Authume and Bletterans where accidents are more frequent when the weather is "normal". Many cantons have been related to alcohol or drug consumption: accidents occurring in the Dole canton are more commonly associated to drug, Moirans-en-Montagne and Saint-Claude cantons are instead matched to alcohol. Finally, the canton where accidents happen in higher proportion on highway is Dole, it is also the canton where accidents are more likely to be fatal, and finally, the canton where pedestrians are bumped in much higher amounts is Champagnole.

1.3.3 MCA of road crash temporal and spatial variables

Here was conducted a global MCA by considering both temporal variables (*time*, *season*, *daytime* and *week*) and spatial variables (*department*) as well as the most important general features such as *type_acc* and *substance*. The first three-factorial axes explain 25,85% of the total variance and only these axes were kept for further analysis. Two-dimensional geometrical representations of all 24 categories are given in Fig (1.10)-(1.12) and interpreted below.

The first two-dimensional map in Fig (1.10) gives the representation of categories on the plane made by axis 1 and axis 2 and it accounts for 19,13% of the total inertia. Categories with the greatest contribution to the axis 1 are "night" (25,75%), "midnight_6am" (13,83%) and "8pm_11pm" (11,88%) and respectively, "winter" (21,99%), "slight_safe" (21,10%) and "week_end" (9,33%) for axis 2.

Axis 1 shows the contrast between serious accidents occurring during the day (between

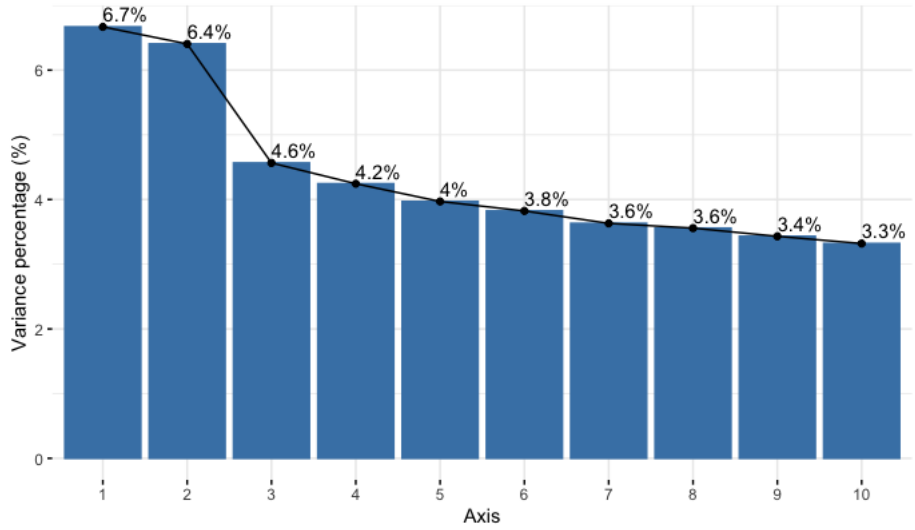

FIG 1.8: MCA spatial analysis, Jura department: variance percentage on the first 10 axes.

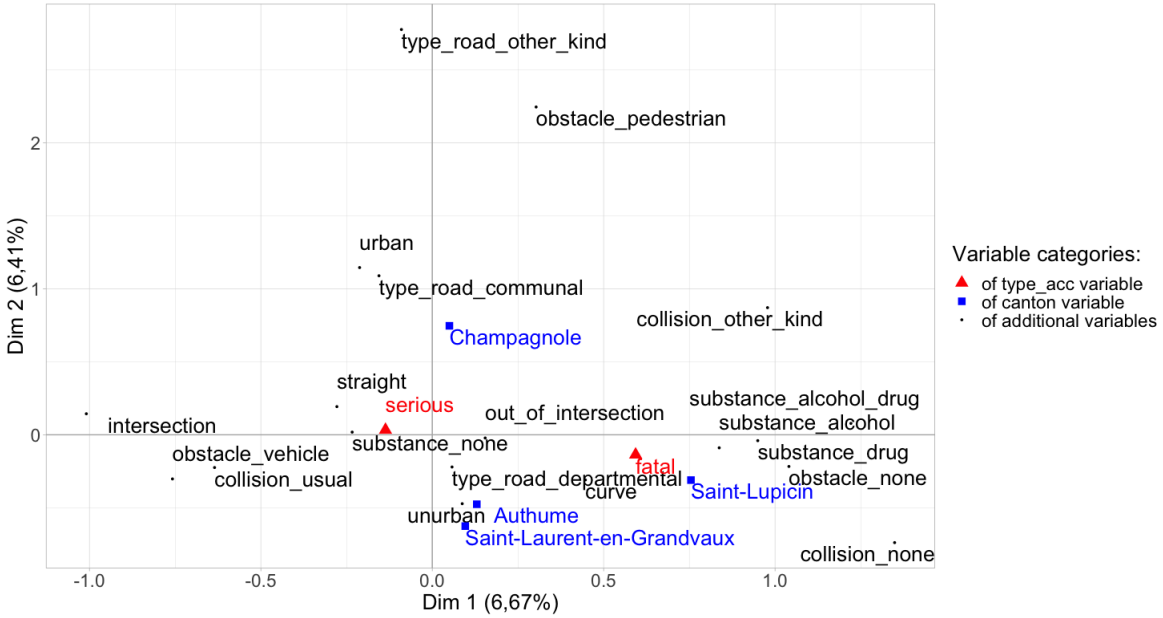

FIG 1.9: MCA spatial analysis, Jura departement: factorial plane made by axis 1 and axis 2.

7:00 a.m. and 7:59 p.m.) from fatal accidents happening during the night (between 8:00 p.m. and 6:00 a.m.). In the first situation, the accidents happen under good daylight conditions, during the spring and the summer time and mostly during the week-time (from Monday to Friday). In the second situation, accidents occur during the week-end and they are associated to alcohol/drug consumptions.

Axis 2 dissociates clearly slight accidents from serious and fatal ones. Accidents occurring in Territoire de Belfort are mostly slight accidents while accidents from Jura are in general serious or fatal. As for the first factorial axis, axis 2 shows the contrast week accidents from week-end accidents which are fatal most of the time. We can notice that accidents

occurring in the Doubs department are similar to those from the department of Haute Saône and relatively close to the origine represented by the average accident profile.

The first factorial plane reveals five groups of associated categories: "slight_safe", "Terr_Belfort" and "winter"; "weekday", "7am_10am" and "none"; "fatal", "week_end" and "drug"; "serious", "spring", "summer" and "Jura"; enfin "night", "8pm_11pm", "alcohol_drug", "midnight_6am" and "alcohol".

Notice also that from this plot, the group formed by categories "Territoire Belfort", "winter" and "slight_safe" is relatively far away from the origin of the plot. This is due to the fact that categories "Territoire Belfort" and "slight_safe" are of low relative frequencies, $6,20\%$ and respectively $11,47\%$, see also TAB 1.1.

Plot made by axis 1 and axis 3 given in FIG (1.11) explains $18,87\%$ of the total inertia. Categories with the greatest contribution to the axis 3 are "Terr_Belfort" $(14,96\%)$, "slight_safe" $(12,84\%)$ and "spring" $(10,87\%)$.

Axis 3 seems to contrast accidents occurring in spring/summer from those occurring in autumn/winter. In the first situation, accidents happen mostly during the day, on week-end and they may have minor consequences or, on the contrary, they may be very serious. In the second situation, the accidents occur during the night, on week-time and they may be serious. Again, the department of Jura is in opposite situation with the department Territoire de Belfort. We can also notice that the department Doubs seems to be the department where most of accidents are associated to drugs. Several groups of categories can be distinguished: "drug", "Doubs" and "summer"; "Haute_Saone" and "serious"; "none", "4pm_7pm" and "weekday"; "Jura", "autumn" and "winter".

Finally, the two-dimensional map of axis 2 and 3 given in FIG (1.12) explains $13,70\%$ of the total inertia. This plot highlights several associations that were analyzed in previous ones. For example, fairly similar modalities are "8pm_11pm" and "Doubs"; "spring", "11am_3pm", "drug", "week_end", "summer" and "fatal".

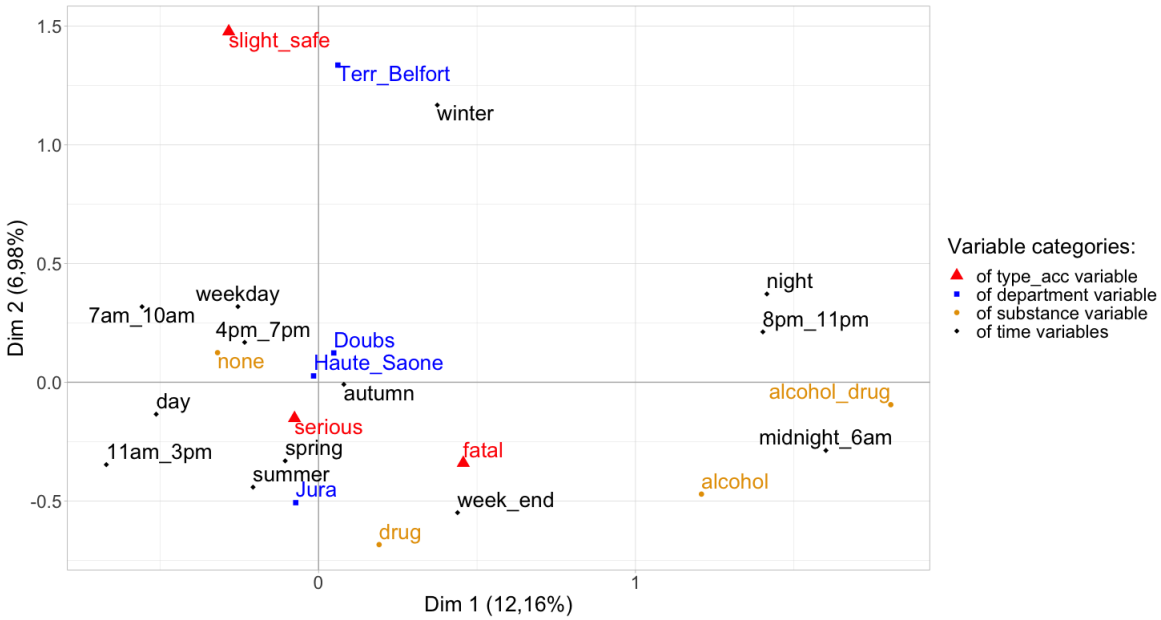

FIG 1.10: MCA global analysis: factorial plane made by axis 1 and axis 2.

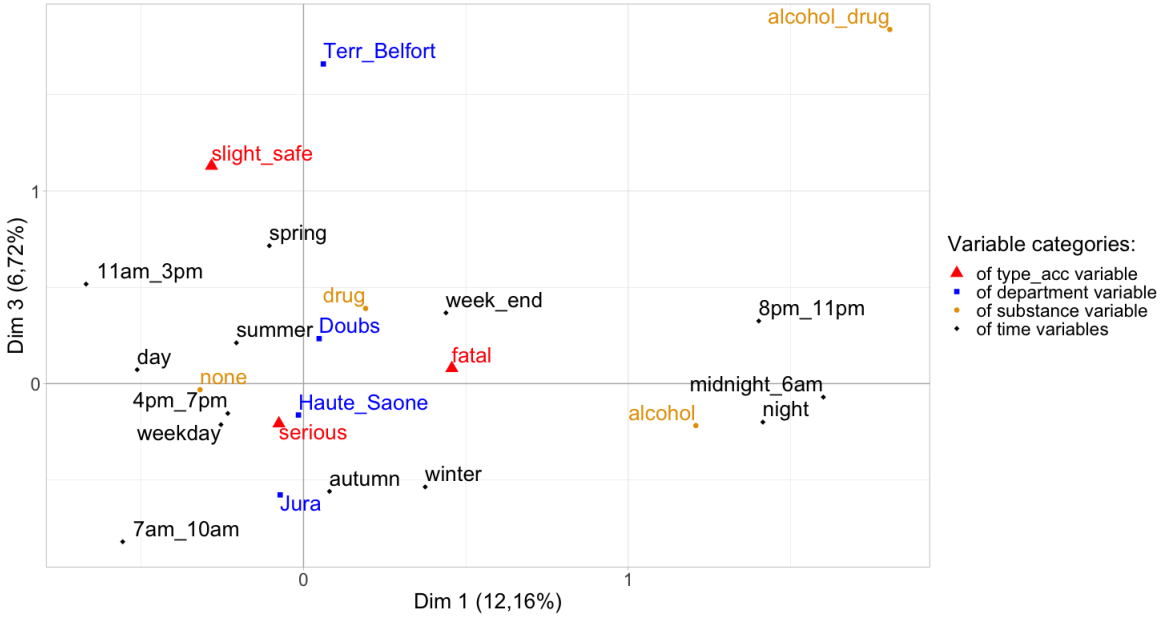

FIG 1.11: MCA global analysis: factorial plane made by axis 1 and axis 3.

1.3.4 Hierarchical log-linear modelling

The MCA performed in sections 1.3.1 and 1.3.2 reveals that there are associations between the gravity of the accidents (*type_acc*) and the drug/alcohol consumption (*substance*). Moreover, we could see during the spatial analysis that these associations are observed within each department. In order to describe more thoroughly the association patterns between these categorical variables and the variable *department*, several hierarchical log-linear models have been fitted on the related three-way contingency table (corresponding to the column "Observed values" of the table TAB 1.3).

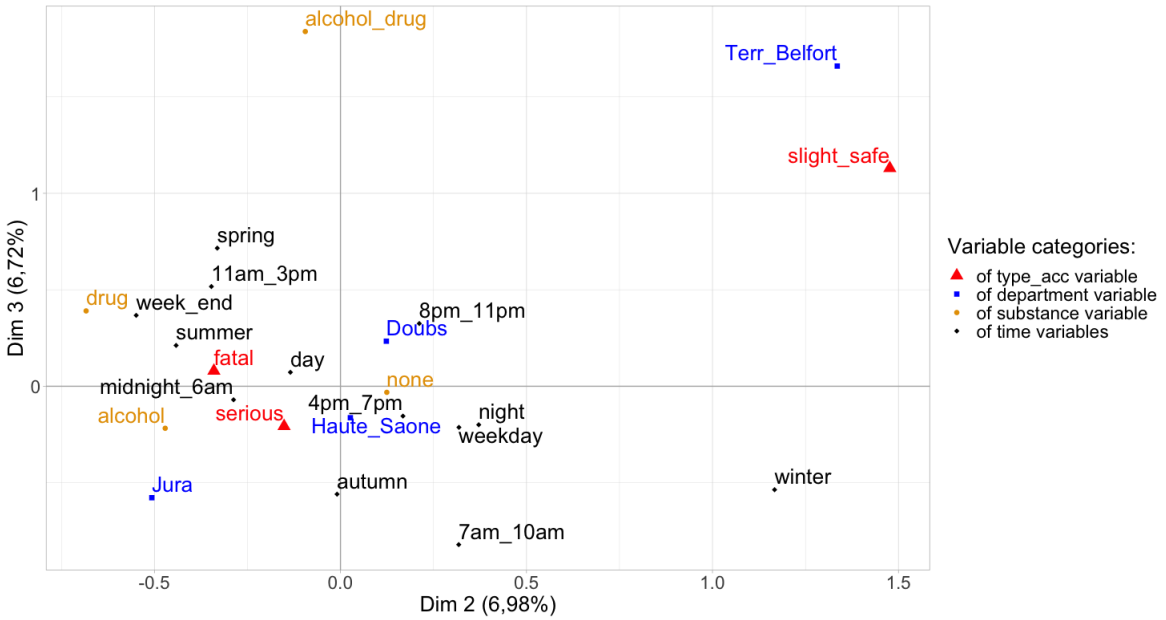

FIG 1.12: MCA global analysis: factorial plane made by axis 2 and axis 3.

TAB 1.4 gives the likelihood-ratio statistic G^2 of each hierarchical model and the 5%-level associated significance test (p-value). A model fits the data well if the null hypothesis of the goodness-of-fit test is accepted. The p-values in TAB 1.4 show that all the models fit the data poorly except (TS, TD, SD) which is close to the observed data (corresponding to the column "Fitted values" of the table TAB 1.3). This unsaturated hierarchical final model has been kept as the objective was to find the simplest model that fits the data adequately. It is written as

$$\log \mu_{kk'k''} = \lambda + \lambda_k^T + \lambda_{k'}^S + \lambda_{k''}^D + \lambda_{kk'}^{TS} + \lambda_{kk''}^{TD} + \lambda_{k'k''}^{SD},$$

where k is "slight_safe", "serious" or "fatal" for the categorical variable *type_acc* (T) ; k' is "none", "alcohol", "drug" or "alcohol_drug" for the variable *substance* (S) ; and k'' is "Doubs", "Haute_Saone", "Jura" or "Terr_Belfort" for the variable *department* (D).

This is the model with no three-factor interaction. The conditional association terms appear for each pair of variables, this means that no pair is conditionally independent. The odds ratios related to this model have been calculated and are given in TAB 1.5. Note that the baseline categories of *type_acc*, *substance* and *department* were respectively "slight_safe", "none" and "Doubs". For instance, the odds ratio relating the level "serious" of *type_acc* and "alcohol_drug" of *substance* at the level "Doubs" of *department* is calculated as

$$\frac{25,64 \times 221,37}{1149,58 \times 3,96} = 1,25.$$

Remind that the no three-factor interaction model means that the association between two variables is identical at each level of the third variable. Hence, in the same way, calculating this odds ratio with the fitted values regarding to the levels "Haute_Saone", "Jura" or "Terr_Belfort" of *department* would have given the same result.

In general marginal odds ratios may differ from conditional ones in a no three-factor

interaction model, however in this case marginal and conditional odds ratio are very close. This means that controlling or ignoring the third variable does not change significantly the association between the two variables. Only conditional odds ratio will be interpreted below as the interpretations of marginal ones are the same.

Regarding substances consumption, the odds for an accident to be serious or fatal increases when alcohol, drug or both are consumed. Indeed, the odds ratios for an accident to be serious are estimated to be respectively 1,64, 1,94 and 1,25 higher than slight when alcohol, drug and both are consumed compared with no consumption. Similarly, the odds ratios for an accident to be fatal are estimated to be respectively 2,76, 4,15 and 5,93 higher than slight when alcohol, drug and both are consumed. The highest risk for an accident to be fatal corresponds to drug and alcohol consumption, almost two times larger than alcohol consumption. Regarding the department where the accident happens, the odds to be serious or fatal decreases only for the department Territoire de Belfort. The odds ratio for an accident to be serious is estimated to be 0,48 times lower than slight when it occurs in this department compared with Doubs. Similarly, the odds for an accident to be fatal is estimated to be 0,39 times lower than slight when it occurs in Territoire de Belfort compared with Doubs. The highest risk for an accident to be fatal is when it occurs in Jura department compared with Doubs, almost four times larger than in Territoire de Belfort.

TAB 1.3: Three-way contingency table with *type_acc*, *substance* and *department* as categorical variables. Left side correspond to the observed values, right one is equal to the predicted frequencies from the log-linear model (TS, TD, SD).

		Observed values			Fitted values (TS, TD, SD)		
		type_acc			*type_acc*		
department	*substance*	slight_safe	serious	fatal	slight_safe	serious	fatal
Doubs	alcohol_drug	6	26	24	3,96	25,64	26,40
	drug	2	30	13	2,87	28,79	13,34
	alcohol	27	227	85	26,81	228,99	83,20
	none	220	1150	250	221,37	1149,58	249,06
Haute_Saone	alcohol_drug	2	15	20	2,48	17,60	16,92
	drug	4	21	14	2,32	25,60	11,08
	alcohol	11	165	35	15,58	145,92	49,50
	none	121	658	144	117,62	669,88	135,50
Jura	alcohol_drug	0	12	10	0,86	9,40	11,74
	drug	2	27	12	1,49	25,29	14,22
	alcohol	12	131	75	10,03	144,34	63,63
	none	89	802	201	90,62	792,97	208,42
Terr_Belfort	alcohol_drug	2	8	8	2,70	8,36	6,94
	drug	0	8	2	1,32	6,32	2,36
	alcohol	8	19	8	5,58	22,75	6,67
	none	62	157	25	62,40	154,58	27,03

TAB 1.4: Goodness-of-Fit Tests for log-linear models relating *type_acc* (T), *substance* (S) and *department* (D).

Model	G^2	p-value
(T, S, D)	246,39	0,00
(T, SD)	218,97	0,00
(S, TD)	179,65	0,00
(D, TS)	125,36	1,09e-12
(TS, TD)	58,61	3,99e-4
(TS, SD)	97,94	6,72e-11
(TD, SD)	152,23	0,00
(TS, TD, SD)	27,38	0,07
(TSD)	0,00	–

TAB 1.5: Odds ratio estimated from (TS, TD, SD) log-linear model. The table is divided into two parts, which are also splitted up into two parts: conditional odds ratios in top have been calculated respectively in left and right sides controlling levels of *department* and *substance* variable, and marginal odds ratios in bottom have been calculated respectively in left and right sides ignoring *department* and *substance* variable. Each odds ratio has been calculated as each level of *type_acc*, *substance* and *department* variable and was opposed respectively to "slight_safe", "none" and "Doubs".

Conditional odds ratios					
	serious	fatal		serious	fatal
alcohol_drug	1,25	5,93	Haute_Saone	1,10	1,02
drug	1,94	4,15	Jura	1,69	2,05
alcohol	1,64	2,76	Terr_Belfort	0,48	0,39
Marginal odds ratios					
	serious	fatal		serious	fatal
alcohol_drug	1,08	4,92	Haute_Saone	1,11	1,06
drug	1,91	4,07	Jura	1,68	1,98
alcohol	1,66	2,78	Terr_Belfort	0,47	0,41

1.3.5 Log-linear modelling using MCA

The goal here was to fit a log-linear model with more than three categorical variables. In order to do that, the global MCA analysis performed in the previous section has been used to choose several variables associated to the road crash severity and to fit a log-linear model with a limited number of parameters but well chosen. A similar analysis has been performed by Papagiotakos and Pitsavos (2004). The global MCA analysis revealed several groups of associated variables/categories such as "slight_safe", "Terr_Belfort" and "winter"; "night", "8pm_11pm", "alcohol_drug", "midnight_6am" and "alcohol", or even more "fatal",

"week_end" and "drug". These three groups suggest that it might exist interactions between the variables *type_acc*, *department* and *season* ; *daytime*, *time* and *substance* ; and finally, *type_acc*, *week* and *substance*.

Based on these considerations, the following log-linear model has been fitted:

$$\log \mu_{k_1 k_2 k_3 k_4 k_5 k_6 k_7} = \lambda + \lambda^{TD}_{k_1 k_3} + \lambda^{SD}_{k_2 k_3} + \lambda^{DaTiS}_{k_7 k_4 k_2} + \lambda^{TWS}_{k_1 k_6 k_2} + \lambda^{TDSe}_{k_1 k_3 k_5} + \lambda^{WTiS}_{k_6 k_4 k_2} + \lambda^{TiSD}_{k_4 k_2 k_3}, \quad (1.5)$$

where k_1 is "slight_safe", "serious" or "fatal" for the categorical variable *type_acc* (T); k_2 is "none", "alcohol", "drug" or "alcohol_drug" for the variable *substance* (S); k_3 is "Doubs", "Haute_Saone", "Jura" or "Terr_Belfort" for the variable *department* (D); k_4 is "7am_10am", "11am_3pm", "4pm_7pm", "8pm_11pm" or "midnight_6am" for the categorical variable *time* (Ti); k_5 is "spring", "summer", "autumn" or "winter" for the variable *season* (Se); k_6 is "weekday" or "week_end" for the variable *week* (W); and k_7 is "day" or "night" for the variable *daytime* (Da).

Several different log-linear models have been fitted and compared one to each other, this final model was the one with the smallest likelihood-ratio statistic G^2. This log-linear model contains specific second and third order interactions between the seven chosen variables based on the global MCA. As mentioned before, these interactions have been decided upon the groups of associated variables revealed in the global MCA. Considering all the second and third interaction terms in the log-linear model would cost in term of interpretation and feasibility.

TAB 1.6: Seven-way contingency table with *type_acc* (T), *department* (D), *substance* (S), *time* (Ti), *season* (Se), *week* (W) and *daytime* (Da) as categorical variables. Left side correspond to the observed values, right one is equal to the predicted frequencies from the log-linear model 1.5.

		Observed values			Fitted values		
		type_acc			*type_acc*		
D, S, Ti, Se, Da	*week*	slight_safe	serious	fatal	slight_safe	serious	fatal
Doubs							
none	weekday	10	27	4	8,33	32,01	7,02
7am_10am	week_end	2	8	0	2,06	10,96	1,95
spring							
day							
D, S, Ti, W, Da	*season*	slight_safe	serious	fatal	slight_safe	serious	fatal
Doubs	spring	8	36	8	5,90	31,34	5,57
none	summer	4	44	6	5,44	41,08	7,24
11am_3pm	autumn	5	30	2	3,68	30,19	6,15
week_end	winter	2	18	4	4,52	24,26	4,87
day							

TAB 1.6 gives two partial tables of the seven-way contingency table made with *type_acc*, *substance*, *department*, *time*, *season*, *week* et *daytime* as categorical variables. It is quite difficult, actually, to give the whole seven-way contingency table due to the high number of cells (3840). This table gives frequencies for categories "Doubs", "none", "7am_10am", "spring"

and "day" of respectively *department*, *substance*, *time*, *season* and *daytime* variables; and "Doubs", "none", "11am_3pm", "week_end" and "day" for respectively *department*, *substance*, *time*, *week* and *daytime* variables. The first partial table (top) details according to *week* and *type_acc* variables and the second partial table (bottom) details according to *season* and *type_acc* variables.

The odds ratios related to the fitted log-linear model from (1.5) have been calculated in the same way as in the previous section and are given in TAB 1.7. The baseline categories used of *type_acc*, *week* and *season* were respectively "slight_safe", "weekday" and "spring". Here, conditional odds ratios may differ according to which levels of the variables we have conditioned. However there are too many possible combinations of levels with our seven categorical variables used, hence only conditional odds ratios regarding to the fixed levels, as in TAB 1.7, were calculated. Notice a slight difference between the conditional odds ratios and the marginal ones. However, the interpretations are the same, except for the level "winter" of the variable *season* for which the odds ratios have a different conclusion depending on whether they are conditional or marginal.

Conditional and marginal odds ratios inform that the risk of an accident being serious or fatal increases if it occurs during the weekend. Indeed, the conditional and marginal odds ratios for an accident to be serious are estimated to be equal to 1,38 on week-ends and respectively 1,12 and 1,34 for an accident being fatal, on week-ends as well. Both kinds of odds ratios tell us that the risk for a crash to be serious or fatal than slightly increases if it occurs in summer or autumn in comparaison with spring. Conditional and marginal odds ratios for an accident to be serious are estimated to be respectively 1,42 and 1,27 if it occurs in summer, 1,54 and 1,37 if it is in autumn. They are estimated to be respectively equal to 1,41 and 1,03 for a fatal crash in summer, 1,77 and 1,23 in autumn. For those occurring during winter with respect to spring, interpretations differ regarding to conditional or marginal odds ratios. Indeed, the risk for an accident to be serious does not seem to depend on the fact that it occurs in winter with respect to spring, the conditional odds ratio being equal to 1,01. The marginal odds ratio is equal to 0,90 meaning that this risk decreases slightly if it occurs during winter. As for the fatal consequences of an accident occurring in winter, the risk would be slightly increased compared to spring according to the conditional odds ratio equal to 1,14; the marginal odds ratio equal to 0,91 informs us that this risk would be slightly reduced.

1.3.6 Ordinal regression modelling

An ordinal regression has been fitted on the response variable *type_acc* (ordered as slight_safe, serious and then fatal). The analysis was performed with all the explanatory variables except *canton* (due to too many categories).

Note that the categorical variable *type_acc* is considered as a response variable. It is relevant because it gives the severity of accidents and the aim of this study lies on understanding how the gendarmerie can avoid serious injuries.

The initial dataset has been split randomly into two sets: a training set (seventy-five

TAB 1.7: Odds ratio estimated from log-linear model 1.5. The table is divided into two parts, each part is also split up into two parts: conditional odds ratios in top have been calculated in left side controlling the levels of *department*, *substance*, *time*, *season* and *daytime* then right side controlling the levels *department*, *substance*, *time*, *week* and *daytime* variables, and marginal odds ratios in bottom have been calculated in left side ignoring *department*, *substance*, *time*, *season* and *daytime* then in right side ignoring *department*, *substance*, *time*, *week* and *daytime* variables. Each odds ratio has been calculated as each level of *type_acc*, *week* and *season* variable and was opposed respectively to "slight_safe", "weekday" and "spring".

Conditional odds ratios					
	serious	fatal		serious	fatal
			summer	1,42	1,41
week_end	1,38	1,12	autumn	1,54	1,77
			winter	1,01	1,14
Marginal odds ratios					
	serious	fatal		serious	fatal
			summer	1,27	1,03
week_end	1,38	1,34	autumn	1,37	1,23
			winter	0,90	0,91

percent of the initial one) and a test set (the remaining twenty-five percent). An ordinal regression model has been fitted on the training set, the results are given in TAB 1.8. Only five explanatory variables reveal to have an effect on the response variable: *time*, *substance*, *department*, *collision* and *area*. This model, with full parameters, gives a misclassification error of 28,84% on the test set.

Next, a variable selection has been performed by using AIC criterion (with backward selection). The final model is composed by the categorical variables *time*, *substance*, *department*, *collision* and *area*. This model gives a misclassification error of 29,00% on the test set, a score very close to the previous one. The odds ratio of these variables are given in FIG (1.13). Only odds ratios with confidence intervals not containing the value 1 (represented by the vertical dotted line) are interpreted.

Regarding the odds ratio, the highest risk for an accident to be serious or fatal is if substances have been consumed by one of the drivers involved. Indeed, the most two important odds ratios are alcohol_drug and drug which are equal to $\exp(1,16) = 3,19$ and $\exp(0,85) = 2,34$ respectively. The risk for an accident to be serious or fatal increases if the accident happens in non urban areas (odds ratio 2,10) or in the Jura department (odds ratio 1,42 with Doubs as reference). Accident involving uncommon collision also have a slightly increased risk (odds ratio 1,28). On the opposite, two categories have a protective effect and are associated with lower risk of serious or fatal accident. This is the case for Territoire de Belfort department (odds ratio 0,56 with Doubs as reference) and for the occurrence time between 4pm and 7pm (odds ratio 0,77).

Each department odds ratio has been represented on the map in FIG 1.14. Remind that the odds of Haute-Saône department were not significant and hence meaningless. The odds

TAB 1.8: Ordinal regression model results with *type_acc* as ordered response variable. The item * means that the p-value of the nullity coefficient test is less than 0,05. The category in parentheses correspond to the baseline category of the above categorical variable.

Attribute	Ordinal regression results				Attribute	Ordinal regression results			
	Coefficients	Estimate	p-value			Coefficients	Estimate	p-value	
substance	alcohol_drug	1,16	5,19e-8	*	*department*	Haute_Saone	-0,01	0,89	
(none)	drug	0,85	4,32e-5	*	(Doubs)	Jura	0,35	5,87e-5	*
	alcohol	0,46	1,14e-5	*		Terr_Belfort	-0,58	2,67e-4	*
season	spring	0,01	0,30		*obstacle*	vehicle	-0,02	0,85	
(winter)	summer	0,11	0,27		(none)	pedestrian	0,23	0,15	
	autumn	-0,05	0,67			other_kind	0,23	0,30	
week	week_end	-0,04	0,53		*shape_road*	curve	-0,01	0,90	
(weekday)					(straight)				
daytime	day	0,08	0,51		*collision*	usual	-0,02	0,91	
(night)					(none)	other_kind	0,25	0,04	*
time	11am_3pm	-0,08	0,50		*type_road*	communal	0,43	0,14	
(7am_10am)	4pm_7pm	-0,26	0,02	*	(highway)	departmental	0,39	0,15	
	8pm_11pm	-0,10	0,52			national	0,47	0,10	
	midnight_6am	0,18	0,27			other_kind	0,34	0,34	
weather	other_kind	0,00	0,97		*intersection*	intersection	-0,12	0,31	
(normal)					(out_of_intersection)				
area	unurban	0,74	3,97e-16	*					
(urban)									

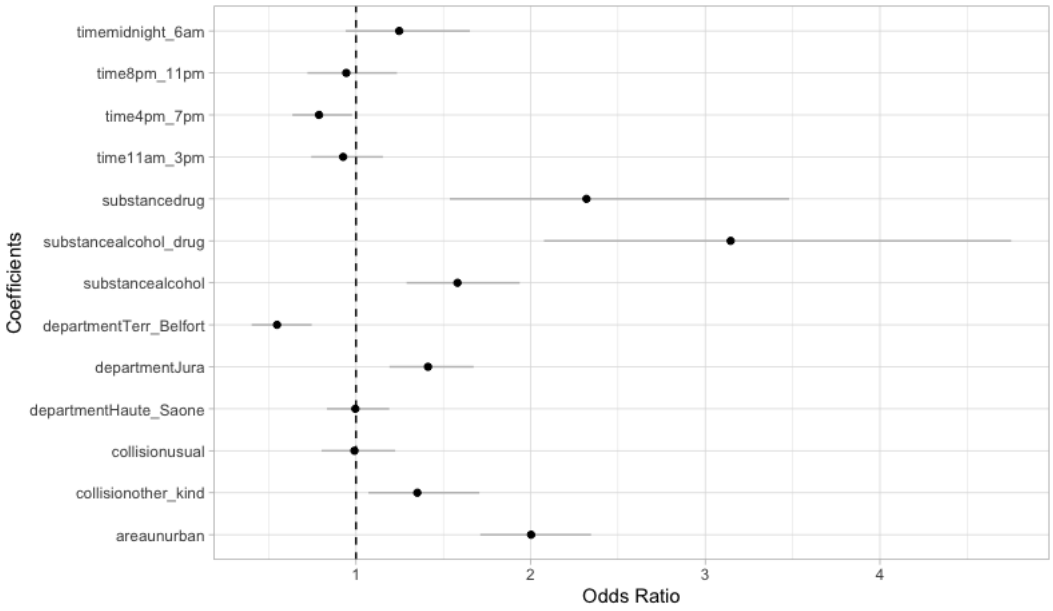

FIG 1.13: Odds ratios obtained by ordinal regression model. Grey lines represent the confidence intervals and black points the values of the odds ratios.

ratio of Doubs department is equal to 0 as it was the baseline category for the *department* variable. As a symbol, Jura and Territoire de Belfort departments have been represented in red and blue respectively due to their odds ratio: the riskiest and the less risky.

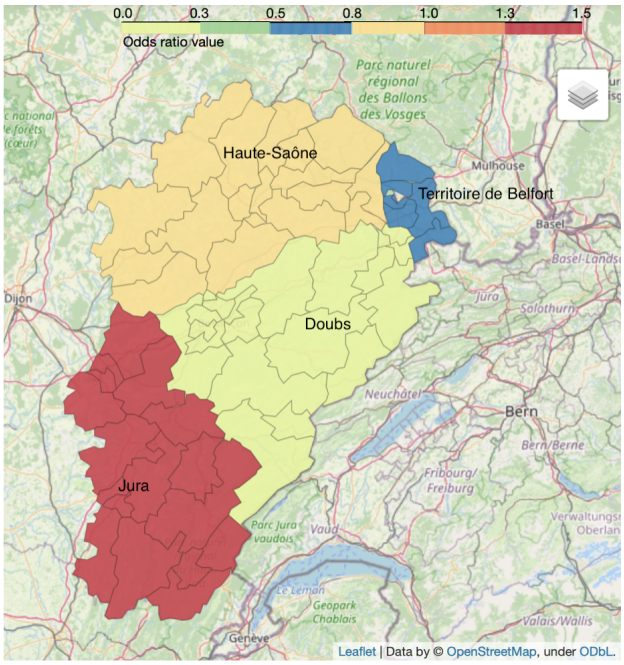

FIG 1.14: Franche-Comté map with department odds ratio. Each color corresponds to a department with its odds ratio value considering the Doubs department as the reference and each small division corresponds to a canton.

1.4 Summary of injury road accident analysis and discussions

A study of accidents with the purpose of mitigating the crash severity is critical for the well-being of a society and the safety concern posed by road crashes. The aim of this work was to understand factors which are the most influential in road accidents from the French region Franche-Comté. To respond to these issues, three statistical methods were used: Multiple Correspondence Analysis (MCA), log-linear model and ordinal logistic regression.

MCA, the only unsupervised or descriptive statistical method used in this study, allowed to assess relationships between the categorical variables and examine the associations between the different categories. Geometric representations of data in smaller dimension spaces were produced and proximities between several road crash related categories have been observed. This analysis allowed to establish a global vision of the data and to draw up temporal and spatial profiles of accidents occurring in the Franche-Comté region. Regarding the MCA temporal analysis, several associations have been highlighted. We remarked that accidents occurring during the week are different from those occurring during the weekend. Indeed, weekend accidents are more likely to happen during the night, be fatal and associated to alcohol/drug consumption. There was also a contrast between summer and winter accidents. The MCA spatial analysis revealed that several cantons of the Franche-Comté region are strongly related to alcohol/drug consumption (Bethoncourt, Saint-Lupicin or Dole) or to fatal accidents (Besançon, Baume-les-Dames, Authume, Saint-Laurent-en-Grandvaux, Saint-Lupicin and Dole).Bethoncourt, Besançon and Baume-les-Dames are situated in the Doubs department and others in the Jura department. Finally, the global MCA temporal and spatial analysis allows to analyse the departments in association to alcohol/drug con-

sumptions and crash severity in temporal ways. Many contrasts have been highlighted and groups of close categories have been analysed. It seems that Jura department is the department where most of accidents are serious or fatal, instead of Territoire de Belfort department which is associated to minor consequences.

The log-linear model was used next in order to evaluate dependencies between the road crash gravity, the alcohol/drug consumption and Franche-Comté departments. This tool models the multidimensional contingency table formed by these three categorical variables and describes associations and interactions among them. It allowed establishing patterns. The selected model concludes on no interaction between the categorical variables *type_acc*, *substance* and *department*. It corresponds to the model of homogeneous association, which means that each pair of variable were conditionally dependent. Odds ratios estimated from this model allowed to quantify the risks about alcohol/drug consumption and the department where the accident happened. We remarked that the highest risks for a serious or fatal accident to happen are if drug, alcohol or both are consumed. Conversely, the lowest risk for an accident to be serious or fatal is if it happens in the Territoire de Belfort department. The Jura department was, instead of the previous one, a location which increases this risk. Then, a second log-linear model was fitted, a non-hierarchical one. This model used results from the global MCA. The fairly similar groups of categories were translated into variable interactions which thus induced the fitted log-linear model. This model revealed many associations between variables. Odds ratios estimated from this model allowed to quantify the risk of an accident occurring during the week or the week-end, or occurring in spring, summer, autumn or winter. According to the conditional odds ratios calculated with respect to spring level, the highest risk for an accident to be serious or fatal is if it occurs in autumn.

The ordinal regression allowed the study to assess each effect of road crash related factors on the road crash gravity. Eight circumstances revealed to be influential on the accident gravity: the consumption of alcohol, drug or both; the period of the day between 4pm and 7pm; the roads situated outside urban areas; the roads situated in Jura or Territoire de Belfort departments; collisions qualified as "other_kind" (not usual as frontal or rear-end for example). Odds ratios estimated from this model allowed to quantify the risks due to each of these circumstances. Similarly to the log-linear analysis, the risk of an accident being fatal is highest if alcohol and drugs are consumed and lowest if the accident happens in the Territoire de Belfort department.

The results obtained with these three methods allow us to conclude that the most important factor to take into account for road crashes in Franche-Comté is the alcohol/drug consumption. As expected, this factor strongly influences the nature of accidents. Hence, based on results obtained with this statistical study, more efforts should be gathered by the National Gendarmerie of Besançon to prevent the alcohol/drug consumption especially in the cantons which were associated to this factor. For example, more alcohol/drug tests and driver awareness measures can be performed.

In order to be more precise for the spatial analysis, the future study would be focused on how GPS coordinate can be used to prevent accidents.

II Space and time analyses of road accidents

The previous part of this thesis is composed of the Chapter 1 that globally focused on the analysis of road crash injuries. Now, the purpose of this second part is to geographically anticipate the accident occurrences. Chapter 2 corresponds to the data preparation in order to be used in Chapter 3 for the statistical methods. The chapters 2 and 3 consider only the spatial component and Chapter 4 this time tries to undertake the temporal component of the data.

While the first part of the thesis is concerned with unsupervised and supervised classical statistical methods, the second part is now concerned with spatial statistics. As it was a field totally unknown to me, I decided to give in Chapter 2, 3 and 4 the implementations of the used statistical methods on the software `R` in an educational way.

The reader may pay attention now to the new notations: X denotes a *spatial point pattern*; Y a *Gaussian process*; Z a vector of covariates values.

Chapter 2

Spatial road crashes and related factors data handling

Spatial point pattern and areal interpolation methods

This work introduces the notion of spatial point patterns. First examples of possible manipulations on the software R of such objects are given. The spatial point pattern used and handled here corresponds to the road accident data that occurred between 2017 and 2019 in the CAGB (urban community of Besançon). Then, the research of relevant related factors to road accidents has been investigated and the information brought has been set properly on a global support. The support chosen is a regular 64×64 grid of cells 650×650 meters. The methods undertaken here to merge the several characteristics, given each at a different scale, into one support are interpolation methods. The purpose of the following work is to technically prepare all these spatial elements (road crash point pattern and related factors) for a statistical modelling in the next chapters and to give the possible existing R software tools that allow to do this pre-processing.

2.1 Introduction

The previous chapter focused on giving a multivariate statistical analysis of road crash data from the French region of Franche-Comté with special attention to road crash gravity. Multiple Correspondence Analysis, Log-linear models and ordinal logistic regression were performed in order to assess associations between the road crash injury and several important accident-related factors. The purposes of this analysis make it possible to raise awareness and insist more on the consequences of these accidents according to the behaviour of the drivers involved (such as alcohol/drug consumption for example). In addition to this behavioural preventive character, the objective now is the spatial analysis and prediction, that is, geographically anticipate the occurrence of these accidents.

Some spatial components were used in the previous analysis such as the canton or the

department where the accident occurred. The final goal now is to perform a spatial analysis at a finer scale. In order to fulfill this objective, the main interest is the exact locations of these accidents and this is what the analysis will focus on. This chapter deals with georeferenced road crash data recorded in several locations in the Doubs department of the French region of Franche-Comté. More particularly, we will present in the following sections the manipulations and implementations of specific statistical and graphical tools available in the software `R` (R Core Team, 2021) to handle this kind of spatial data for the fitting of a statistical model in the next chapter.

The following librairies will be used :

```
> library(FRK)
> library(ggplot2)
> library(leaflet)
> library(lgcp)
> library(maptools)
> library(rgdal)
> library(rgeos)
> library(sf)
> library(sp)
> library(spatstat)
> library(tidyverse)
```

`R` packages `FRK` (Zammit-Mangion and Sainsbury-Dale, 2022), `maptools` (Bivand et al., 2021b), `rgdal` (Bivand et al., 2021a), `rgeos` (Bivand et al., 2021c), `sf` (Pebesma et al., 2022), `sp` (Pebesma et al., 2021) and `spatstat` (Baddeley et al., 2021b) deal with spatial data and are used in order to handle this kind of data. The packages `ggplot2` (Wickham et al., 2021), and `leaflet` (Cheng et al., 2021) are used for plot. Finally `lgcp` (Taylor et al., 2021) is the package used to fit our statistical model and `tidyverse` (Wickham, 2021) for data wrangling basic operations.

The current chapter is structured as follows. Section 2.2 describes how to create a point pattern object, specific class in the software which handle geographical locations points, for our road crash data. Then Section 2.3 deals with environment resources that will be associated to the point pattern built in last section. More particularly, Section 2.3.1 is concerned in creating a grid support to welcome these resources and the interpolation of this information onto the latter grid is in Section 2.3.2.

2.2 Road accident spatial point pattern

Data used in this study concern road injury crashes, that are, accidents that occurred on a public road involving at least one vehicle and resulting in at least one victim requiring cares. These data are extracted from the French national analysis bulletin of road traffic injury accidents called BAAC file (*Bulletin d'Analyse des Accidents Corporels*) available as open data on the French platform www.data.gouv.fr. The BAAC data are filled in by the security forces present on the accident scene and next, data are treated, analyzed and put online by the national interdepartmental observatory of road safety (*Observatoire National*

Interministériel de la Sécurité Routière). Datasets from BAAC file include accident location information, as filled in on scene, as well as information regarding the characteristics of the accident and its location, the involved vehicles and their victims.

This study deals with road crashes that occurred between 2017 and 2019 in the CAGB (*Communauté d'Agglomération du Grand Besançon*), headquarters of the region Franche-Comté and urban community of Besançon (central municipality) composed of 68 cities.

```
> c_2019 <- read_delim("DATA/caracteristiques-2019.csv",
+                      delim = ";",
+                      escape_double = FALSE,
+                      trim_ws = TRUE)
> c_2018 <- read_csv("DATA/caracteristiques-2018.csv")
> c_2017 <- read_csv("DATA/caracteristiques-2017.csv")
```

Files extracted from the French open platform data.gouv are `.csv` extension files. Datasets `c_2017`, `c_2018` and `c_2019` are structured in order to extract only accidents that occurred in the CAGB and then merged into one dataset called `cagb` (command-lines not shown). This dataframe has 397 observations and two columns named x and y which represent respectively the longitude and the latitude of the accident location.

When dealing with spatial data, the first step is to pay attention to a very important aspect : the *coordinate reference system* (CRS) used. This system is a standard way to describe geographical data and is defined by a *projection*, a *datum* and a set of parameters. The projection corresponds to how the three dimensional angular system (defined by the longitude and the latitude) is transformed into a two dimensional planer system. Then, the datum is a model of the shape of the earth which estimates the angles of the angular system. Most commonly used CRSs have been assigned a unique identifier called *EPSG* code. The CRS used in the BAAC file is defined by the datum *WGS*84, also called *World Geodesic System 1984*, which is the global reference system. This CRS has EPSG:4326. Data from `cagb` are then visualized in Fig 2.1 using `leaflet` package as follows:

```
> leaflet(data = cagb) %>% addTiles() %>%
+    setView(lng = 6.01, lat = 47.24, zoom = 11) %>%
+    addCircleMarkers(~x, ~y,
+                     weight = 1,
+                     fillOpacity = 1,
+                     radius = 1,
+                     color = "black")
```

Several datasets will be loaded and used in the following, hence, it is important to transform them to a common CRS so they align with one another. For more convenience, the CRS chosen to be used is *RGF*93, also called *Lambert* 93, with EPSG:2154. This CRS has the advantage to deal with actual shapes, dimensions and distances.

Data in the form of a set of points $X = \{x_1, \ldots, x_n\}$ of $\mathbb{R}^2$, irregularly distributed over a study area, is called a spatial *point pattern* and refers to the locations as events. The reader may find comprehensive descriptions of spatial point pattern and applications in `R` respectively in Diggle (2013) and Baddeley et al. (2015). Road accident data extracted from the French open platform data.gouv are georeferenced points, hence, a spatial point pattern.

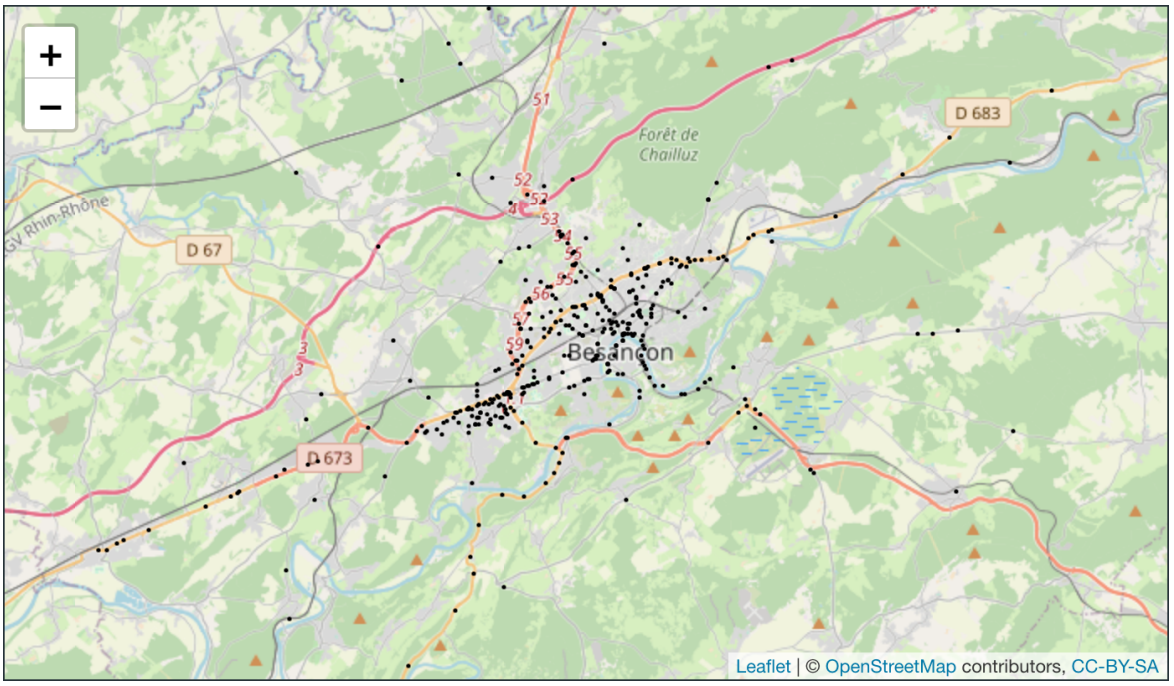

FIG 2.1: Spatial distribution of road crashes in the CAGB area.

In order to designate `cagb` as a point pattern in `R`, it needs to be converted into an object of type `ppp`. This class of the package `spatstat` represents a two-dimensional point pattern dataset. It specifies the locations of the points and the *window* (study area) in which the pattern was observed.

The first step then is to create an object of class `owin` which defines the observation window, that is, the study area of our spatial point pattern. The type of owin used here is "polygonal": a region whose boundary is a polygon or several polygons. The polygons of the Doubs department cities are loaded from the website france-geojson.gregoiredavid.fr in `.geojson` extension files. This website brings to the *France Geojson* project which contains the routes of the French geographical and administrative entities (regions, departments, cantons, districts and municipalities), data contain the postal code and the name of the entity. The object `cities` created below is a `SpatialPolygonsDataFrame` object, a class from the package `sp` for holding polygon geometries with attributes.

```
> u <- "https://france-geojson.gregoiredavid.fr/
        repo/departements/25-doubs/communes-25-doubs.geojson"
> downloader::download(url = u, destfile = "communes.GeoJSON")
> cities <- readOGR(dsn = "communes.GeoJSON")
> cities@proj4string
> cities <- spTransform(cities, CRSobj=CRS("+init=epsg:2154"))

CRS arguments: +proj=longlat +datum=WGS84 +no_defs
```

The CRS of `cities` is *WGS*84, obtained with its attribute `proj4string`. As mentioned before, it is more beneficial to work in *Lambert* 93. Hence the function `spTransform` from the package `sp` is used in order to transform to the new CRS. The cities polygons are plotted

in Fig 2.2 using command-lines from package ggplot2 below. For more convenience in using this package, the function SpatialPolygonsDataFrame_to_df from the package FRK is used in order to convert the SpatialPolygonsDataFrame cities to a dataframe. Indeed, this function creates columns X1, X2 and id which allow to plot polygons and values associated (as name in our case) in a simply way. The reader may find various plotting methods later. The object proj_plot is used in order to specify to ggplot2 that the CRS wished for plotting is *Lambert* 93. It will be used for the rest of the chapter.

```
> proj_plot <- '+proj=lcc +lat_1=49 +lat_2=44 +lat_0=46.5 +lon_0=3 +x_
   0=700000 +y_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m
   +no_defs'
> cities_df <- SpatialPolygonsDataFrame_to_df(cities)

> ggplot(cities_df, aes(x = X1, y = X2, group = id)) +
+      geom_polygon(colour='black',
+                   fill=ifelse(cities_df$name == "Besançon",
+                   "#1A8BAF", "white")) +
+      coord_sf(xlim = c(890000, 970000),
+               ylim = c(6660000, 6705000),
+               crs = st_crs(2154), datum = proj_plot) +
+      theme_bw() +
+      theme(panel.grid.major = element_line(colour =
+            "black", linetype = "dashed", size = 0.1),
+            panel.grid.minor = element_line(colour =
+            "black", linetype = "dashed", size = 0.1)) +
+      xlab("Longitude") +
+      ylab("Latitude")
```

Our observation window owin_cagb is plotted in Fig 2.3 as follows:

```
> cities <- gUnionCascaded(cities)
> owin_cagb <- as.owin(cities)

> owin_cagb_sf <- st_as_sf(owin_cagb)
> owin_cagb_sf <- st_transform(owin_cagb_sf, crs=2154)

> ggplot(owin_cagb_sf) +
+      geom_sf(color = 'black', fill = 'white') +
+      coord_sf(xlim = c(890000, 970000),
+               ylim = c(6660000, 6705000),
+               crs = st_crs(2154), datum = proj_plot) +
+      theme_bw() +
+      theme(panel.grid.major = element_line(colour =
+            "black", linetype = "dashed", size = 0.1),
+            panel.grid.minor = element_line(colour =
+            "black", linetype = "dashed", size = 0.1)) +
+      xlab("Longitude") +
+      ylab("Latitude")
```

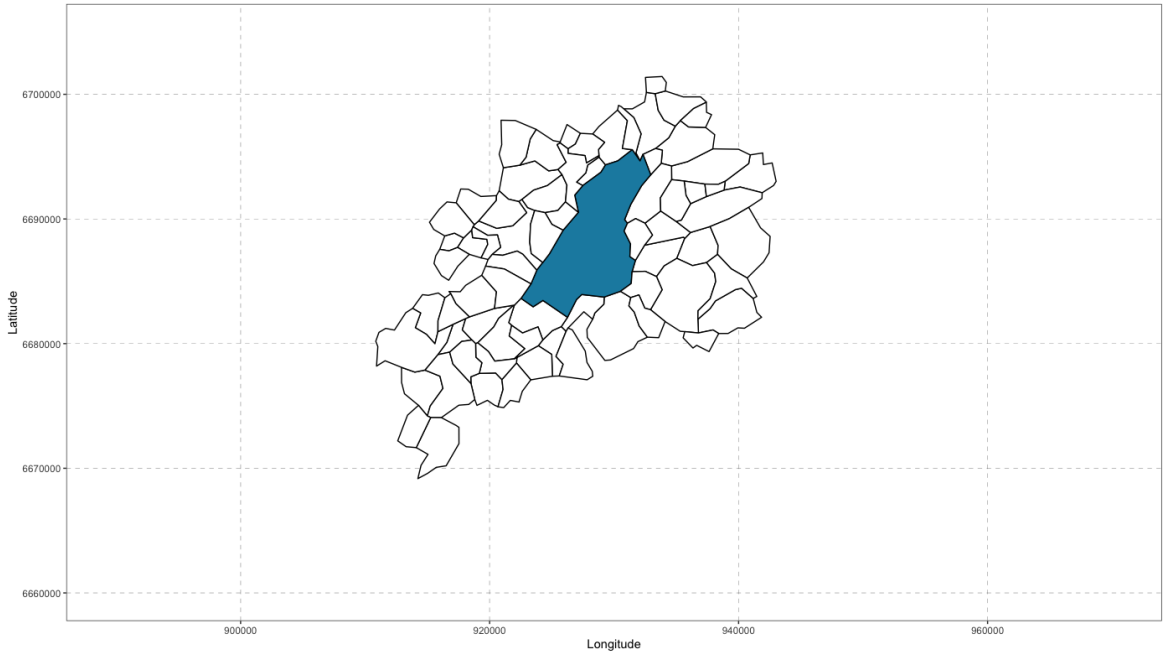

FIG 2.2: Polygons of CAGB cities. The blue one corresponds to the city of Besançon.

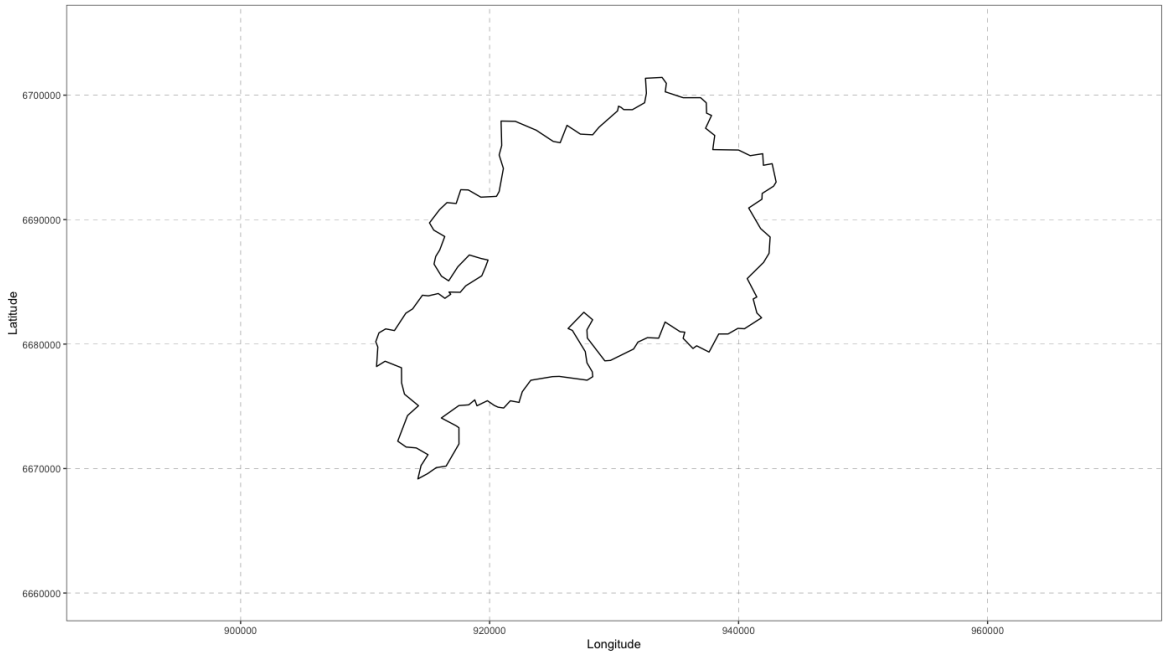

FIG 2.3: Polygons of the study area.

The cities polygons have been merged into one polygon using the function `gUnionCascaded` from the package `rgeos` and set as an object `owin` with package `maptools`. For more convenience using package `ggplot2`, an object `owin_cagb_sf` of class `sf` is created with `st_as_sf` from the package `sf`. This package is a support for simple features, a standardized way to encode spatial data made of geometries (points, lines, polygons, ...). The CRS is ensure with the `sf` function `st_transform` by specifying EPSG:2154, just as

similar as spTransform seen before.

As our owin object has been made, then an object of class ppp can be created with command-lines below. First, an object cagb_sf is created with st_as_sf. The coords argument takes the names of the numeric columns holding spatial coordinates and the CRS *WGS*84 is specified with the crs argument as it was the initial projection established when loading data from data.gouv. Finally, the CRS of cagb_sf is converted to *Lambert* 93 with st_transform as seen before.

```
> cagb_sf <- st_as_sf(cagb, coords = c("x", "y"), crs = 4326)
> cagb_sf <- st_transform(cagb_sf, crs=2154)
```

A ppp object called cagb_ppp is then created by using cagb_sf and owin_cagb with the function as.ppp from the package spatstat as follows:

```
> cagb_ppp <- as.ppp(st_coordinates(cagb_sf), owin_cagb)
> cagb_ppp <- as.ppp(cagb_ppp)
> area(owin_cagb)
> perimeter(owin_cagb)

[1] 513848149
[1] 150901
```

The points must lie inside the specified window but it might happen that some points lie outside, this may be due to input error. Hence these points called "rejects" can be excluded by directly reusing the function as.ppp. The point pattern cagb_ppp has 396 events in the study window which has an area of 513 848 149 square meters and a perimeter of 150 901 meters. The point pattern cagb_ppp can be visualized as follows and gives Fig 2.4:

```
> plot(cagb_ppp, pch = 20, cex = 0.25,
+       xlim = c(890000, 970000),
+       ylim = c(6660000, 6705000),
+       xlab = "Longitude", ylab = "Latitude",
+       ann = TRUE, axes = TRUE, main = "")
```

The point pattern can be also visualized with ggplot2 by using the sf objects cagb_sf and owin_cagb_sf as follows:

```
> ggplot() +
+      geom_sf(owin_cagb_sf, color = 'black', fill = 'white') +
+      geom_sf(cagb_sf, color = 'black', size = 0.5) +
+      coord_sf(xlim = c(890000, 970000),
+               ylim = c(6660000, 6705000),
+               crs = st_crs(2154), datum = proj_plot) +
+      theme_bw() +
+      theme(panel.grid.major = element_line(colour =
+            "black", linetype = "dashed", size = 0.1),
+            panel.grid.minor = element_line(colour =
+            "black", linetype = "dashed", size = 0.1)) +
+      xlab("Longitude")+
+      ylab("Latitude")
```

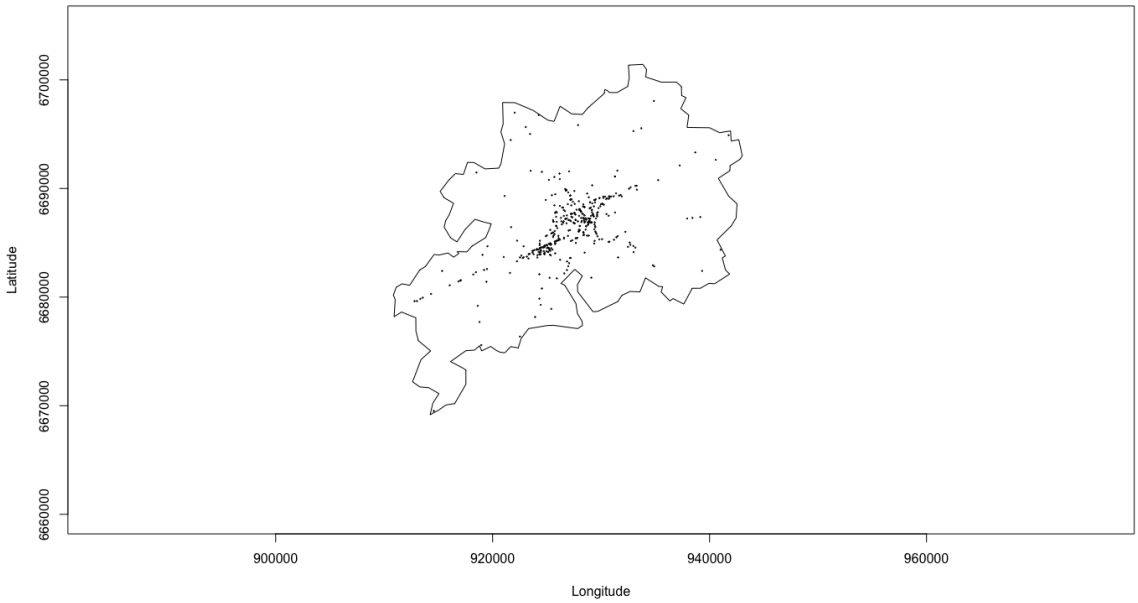

FIG 2.4: Road accident spatial point pattern in the CAGB. Using `plot` function.

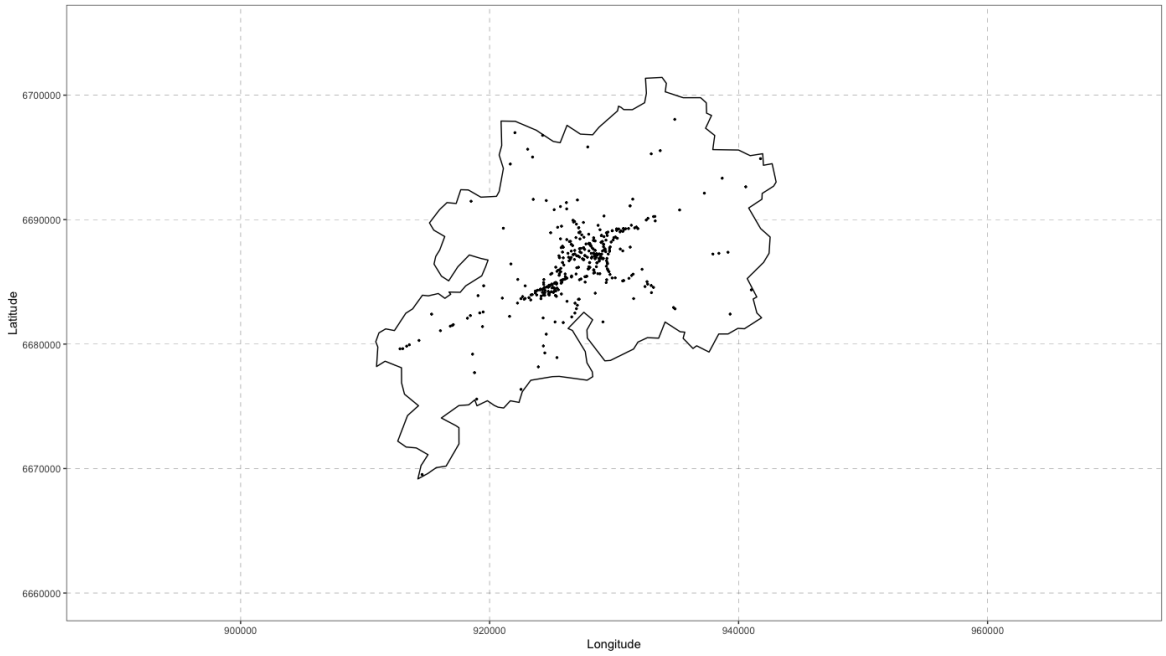

FIG 2.5: Road accident spatial point pattern in the CAGB. Using `ggplot2` functions.

Command-lines above give FIG 2.5.

It might be interesting to divide the window into subregions $B_1, \ldots, B_m$, called *quadrats*, and count the numbers of points falling in each quadrat. In practice, *quadrat counting* is a simple way to check for some statistic properties of a point pattern. Quadrat counting is performed in `spatstat` by the function `quadracount` and the `plot` method can be used to display quadrats as follows:

```
> par(mfrow = c(1, 2))
> plot(quadratcount(cagb_ppp, nx = 3, ny = 3), main="")
> plot(quadratcount(cagb_ppp, nx = 9, ny = 9), main="")
```

FIG 2.6 displays two ways of quadrat counting on our ppp object cagb_ppp. The left and respectively the right panel show the point pattern in 3×3 and 9×9 cells, as it was specified with nx and ny arguments.

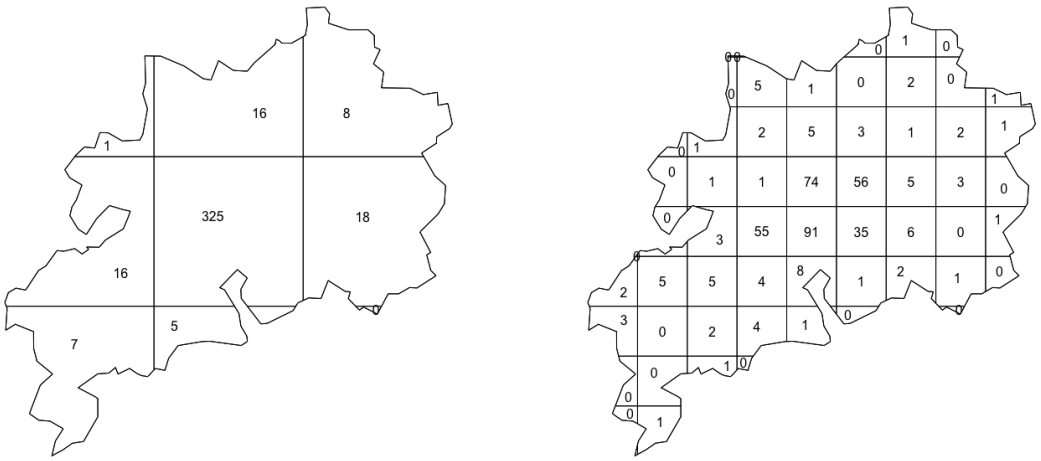

FIG 2.6: Plot of quadrats of road accident spatial point pattern. Left to right : 3×3 quadrat sand 9×9 quadrats.

Finally, the ppp object cagb_ppp can be exported using the function saveRDS.

```
> saveRDS(cagb_ppp, "DATA/cagb_ppp.rds")
```

2.3 Handling auxiliary information and covariates

The current section focuses on the creating of a database of the environment characteristics of road crashes locations, such as sociodemographic data, global and road infrastructure data, so that the impact of these environment resources on road accidents can be assessed. The information will be used in the next chapter in order to fit spatial statistical models such as *log-Gaussian Cox processes* (LGCPs). In practice, fitting a LGCP requires to perform computations on a regular grid. To be as rigorous as possible, covariate data collected will be directly associate with this grid.

2.3.1 Computational support grid

The computational grid is created by using the getpolyol function from the lgcp package as follows:

```
> polyolay <- getpolyol(data = cagb_ppp, cellwidth = 650, ext = 2)
> grid_cagb <- polyolay$fftpoly
> area(grid_cagb)

[1] 1730560000
```

The arguments `cellwidth` and `ext` are chosen through preliminary statistical methods relating to the model itself, we will therefore not give any further details here. As the CRS of `cagb_ppp` is *Lambert* 93, the grid used then the same CRS. Finally, the grid can be extracted from the attribute `fftpoly`. The object `grid_cagb` is of class `SpatialPolygons` from package `sp`, similar to `SpatialPolygonsDataFrame` seen before.

This regular grid 64×64 covers an area of 1 730 560 000 square meters, this means, the 4 096 cells are of 650m $\times$650m. This grid can be visualized as follows:

```
> grid_cagb_sf <- st_as_sf(grid_cagb)

> grid_cagb_sf %>%
+      dplyr::select(geometry) %>%
+      ggplot() +
+          geom_sf( color = 'black', fill = 'white') +
+          coord_sf(xlim = c(880000, 975000),
+                   ylim = c(6660000, 6710000),
+                   crs = st_crs(2154), datum = proj_plot) +
+          theme_bw() +
+          theme(panel.grid.major = element_line(colour =
+              "black", linetype = "dashed", size = 0.1),
+              panel.grid.minor = element_line(colour =
+              "black", linetype = "dashed", size = 0.1)) +
+          xlab("Longitude")+
+          ylab("Latitude")
```

As seen before, for more convenience using `ggplot2`, an `sf` object `grid_cagb_sf` has been created. Command-lines above gives Fig 2.7.

Finally the object `polyolay`, which contains the grid, is exported using the function `saveRDS` as seen before as follows:

```
> saveRDS(polyolay, "DATA/polyolay.rds")
```

2.3.2 Areal interpolation

Environment resource data were extracted from various sources such as the French statistical institute INSEE (*Institut National de la Statistique et des Etudes Economiques*), OpenStreetMap (data by © OpenStreetMap contributors under the Open Database Licence) or the French open platform data.gouv seen before. Sociodemographic and global infrastructure data (such as school locations, shop locations or gas station locations for example) have been extracted from the website of INSEE www.insee.fr. Road infrastructure data (such as traffic light locations, give way or stop locations for example) have been extracted from www.openstreetmap.fr. Finally, the network structure of roads has been extracted from

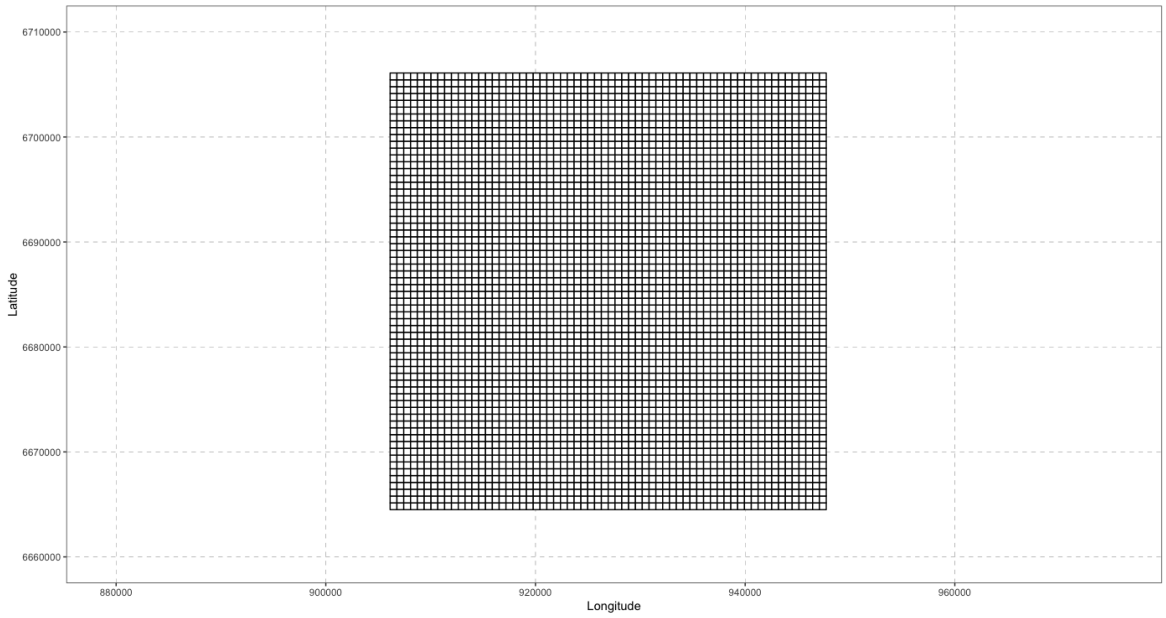

FIG 2.7: Computational support grid.

www.data.gouv.fr. Data collected were reported between 2015 and 2020, this information has been considered unchanged over the years in order to associate them to road crashes that occurred between 2017 and 2019.

The data formats are incompatible as they have been collected at different geographical scales or units. For instance, sociodemographic data are provided as polygons, road network as lines and infrastructure locations as points. However, these heterogeneous data need to be simultaneously used. Hence it is necessary to merge them into one spatial support : this problem is called the *areal interpolation* problem. The process of interpolation consists in transforming data from source zones to target zones (which in our case are the cells of `grid_cagb`). Let $t_1, \ldots, t_T$ and $s_1, \ldots, s_S$ be the sets of target zones and source zones respectively. We denote by Z_{t_i} the value of the target variable Z (variable of interest needed to be interpolated) on the target zone t_i and respectively by Z_{s_j} on the source zone s_j, $i = 1, \ldots, T$ and $j = 1, \ldots, S$. Then, for the intersection zone I_{t_i,s_j} between t_i and s_j, the value of Z is noted as Z_{t_i,s_j}. The variables to be interpolated that will be considered in the following are extensive variables (Do et al., 2021). Only one interpolation method will be used and presented briefly below, for more details on interpolation methods see for example Do et al. (2021).

Sociodemographic data

French sociodemographic file extracted from INSEE is a `.shp` file, that is, a shapefile. Shapefiles are file formats for systems containing all the information linked to object geometries (points, lines or polygons). Data are loaded below using the function `st_read`, from the package `sf`, which helps to read simple features from `.shp` files. It is a heavy file to

download and to load (around 3 minutes and 42 seconds for 1.04Go).

```
> T1 <- Sys.time()
> france <- st_read("DATA/Filosofi2015_carreaux_200m_metropole.shp",
+                   quiet=TRUE)
> france <- st_transform(france, crs=2154)
> T2 <- Sys.time()
> difftime(T2, T1)

 Time difference of 3.692795 mins
```

The data are structured (command-lines not shown) into an object `grid_insee_sf` that extracts only geometries falling in the CAGB and wished covariates that give per geometry:

- `Ind` : the number of individuals
- `Ind_18_24` : number of individuals between 18 and 24 years old
- `Ind_65_79` : number of individuals between 65 and 79 years old
- `Ind_80p` : number of individuals more than 80 years old

Sociodemographic data provided by INSEE are given at a very fine scale. Indeed, they are associated to a grid of cells 200×200 meters, which is finer than the target zones, polygon geometries of 650×650 meters. An illustration of this phenomenon can be visualized as follows:

```
> sub_grid <- grid_cagb_sf[c(1358:1360,1294:1296),]

> st_crs(grid_insee_sf) <- st_crs(sub_grid)
> inters_insee <- st_filter(grid_insee_sf, sub_grid)
> inters_insee <- inters_insee %>% select(geometry)
> inters_point <- st_centroid(inters_insee)

> ggplot() +
+   geom_sf(data = sub_grid, fill = 'white') +
+   geom_sf(data = inters_insee, color = 'red', fill = NA) +
+   geom_sf(data = inters_point, color = 'red') +
+   coord_sf(xlim=c(913500, 917500), ylim=c(6677000, 6679200),
+            crs = st_crs(2154), datum = proj_plot) +
+   theme_bw() +
+   theme(panel.grid.major = element_line(colour = "black",
+                                         linetype="dashed",
+                                         size=0.1),
+         panel.grid.minor = element_line(colour = "black",
+                                         linetype="dashed",
+                                         size=0.1)) +
+   xlab("Longitude") +
+   ylab("Latitude")
```

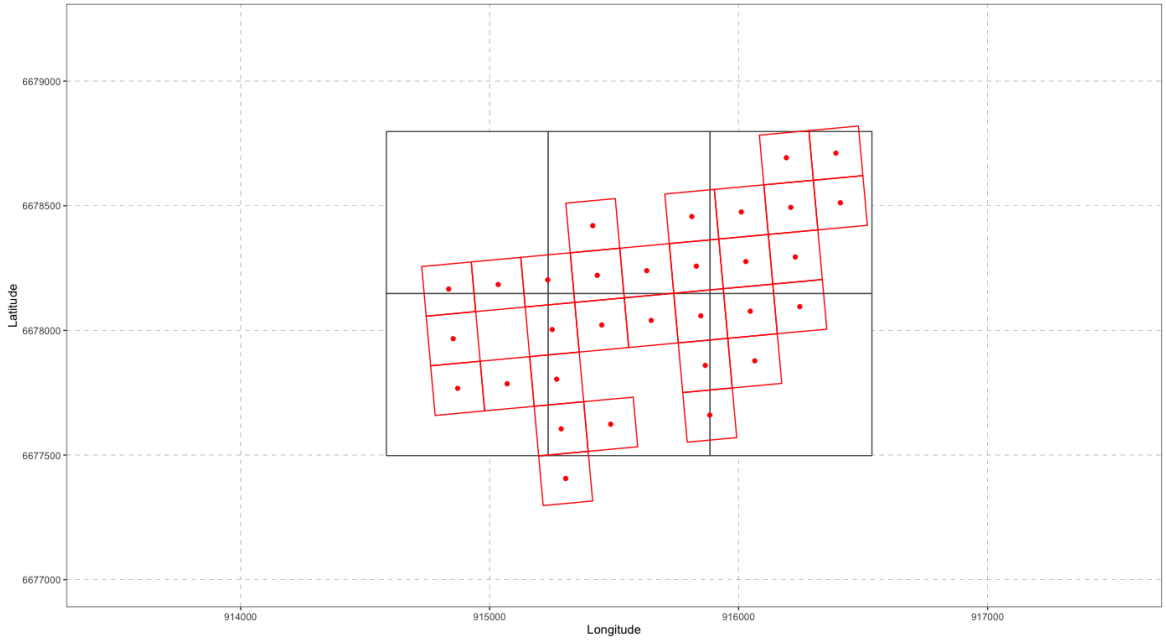

FIG 2.8: Plot of INSEE data cells with centroids (in red) overlaying computational grid cells (in black).

The FIG 2.8 represents the source zones (and centroids associated) in red that overlay a subset of target zones (computational grid cells numbered 1358 to 1360 and 1294 to 1296) in black. Indeed, the overlap of `grid_insee_sf` on `grid_cagb_sf` has been computed using the function `st_filter`, from the package `sf`, which joins according to geometries. Classical functions of this package require that both `sf` objects have exactly the same CRS, which can be done with `st_crs`. Then the centroids of `grid_insee_sf` overlapping cells are obtained with `st_centroid` from `sf` package.

INSEE data cells (source zones) need to be transferred into the form of the computational grid `grid_cagb_sf` cells (target zones). As mentioned before, sources are much smaller than targets, which theoretically make the interpolation better. Indeed, sources are more likely to be included in targets. The method that will be introduced below is the *point-in-polygon* (PIP) method (Do et al., 2021). As its name implies, sources are points and targets are polygons. As INSEE data cells are polygons, it has to be represented by points, hence the use of centroid as described before. Finally, the value to be interpolated on a target zone is the aggregation of all source points located in it, that is, the sum of the values of all these source points. The PIP method is realized as follows:

```
> grid_insee_point <- st_centroid(grid_insee_sf)
> inters <- st_intersects(grid_cagb_sf, grid_insee_point, sparse = F)
> inters_yes <- apply(inters, 1, function (x) any(x != F))
> targets <- c("Ind", "Ind_18_24", "Ind_65_79", "Ind_80p")
> for(z in targets){
+   grid_cagb_sf[, z] <- rep(0, nrow(grid_cagb_sf))
+   for (i in which(inters_yes)) {
+   grid_cagb_sf[i, z] <- sum(st_drop_geometry(grid_insee_point)[
```

```
    inters[i,], z])
+   }
+ }
```

There are four target variables to be interpolated : `Ind`, `Ind_18_24`, `Ind_65_79` and `Ind_80p`. Two sets t_i, $i = 1, \ldots, 4\ 096$ and s_j, $j = 1, \ldots, 3\ 386$ of target zones and source zones respectively. First, all centroid sources are computed. The function `st_intersects` is used in order to identify if both arguments geometries share any space and intersection zones I_{t_i,d_j} are obtained and stocked in `inters_yes`. Then for each target variables, the aggregation of all source points located in it is computed using the function `sum`. Remark that, as said before, `sf` objects encode geometries of spatial data. Datasets of this class always contain geometries in a column called *geometry*. This column does not have to be mentioned properly in `R` operations in order to be considered, it is a sort of "attached" column. The function `st_drop_geometry` consists, as its name implies, to drop out geometries from the `sf` dataset. Otherwise without using this function, the function `sum` applied above would have summed up geometries too, even if there is only one target variable selected in the command-line.

```
> sum(grid_insee_sf$Ind)
> sum(grid_cagb_sf$Ind)

[1] 178232
[1] 178232
```

In our case, the PIP method is sufficient. Indeed for instance, the total number of individuals on the whole grid `grid_insee_sf` is 178 232, same as `grid_cagb_sf`, which means that there is no information lost. Note that INSEE data cells relate to information collected in 2015, and according to an INSEE report published in 2018 concerning the population census in 2015, there are 197 754 inhabitants in the CAGB (INSEE, 2018). The difference between 197 754 and 178 232 is due to confidentiality problems. In conclusion, our sociodemographic data interpolated on `grid_cagb_sf` is quite close to the reality. Two variables have been created from available variables : *prop18* and *prop65* which give respectively the proportion of individuals between 18 and 24 years old and the proportion of individuals more than 65 years old.

Global and road infrastructures

Global infrastructure data have been extracted from INSEE and transformed into the following seven variables :

- *health*: locations of healthcare institutions (such as hospitals, doctor offices or drug-stores for example)
- *school*: locations of schools (until the end of highschool)
- *college*: locations of schools (after highschool)
- *shop*: locations of shops (such as food shops, shops or restaurants for example)

- *station*: locations of stations (such as train stations or taxi services for example)
- *gasoline*: locations of gas stations
- *leisure*: locations of establishments for leisure (such as cinemas or tennis courts for example)

File extracted is a `.csv` extension file. Data were loaded, structured, split in order to keep only resources that fall in the CAGB and transformed into a `sf` object where geometries are points.

Road infrastructure data have been extracted from OpenStreetMap and transformed into the variables *intersection* and *radars* which give respectively the locations of intersections (traffic lights, stop and give way) and the locations of speed cameras. Files extracted are `.gpkg` which is a file format for geospatial data. Data loaded as `.gpkg` files are of class `sf` with points as geometries.

All these variables have to be interpolated on `grid_cagb_sf`. As sources are points, the method to be applied is the PIP method described before. Another way to implement this method consists in using functions of package `sp` as follows:

```
> radars_sf <- st_read("DATA/radardevitesse.gpkg")
> radars_sf <- radars_sf %>% dplyr::select(geom)
> radars_sf <- st_transform(radars_sf, crs=2154)
> radars_sf <- st_intersection(owin_cagb_sf, radars_sf)
> radars_sf$radars <- rep(0, nrow(radars_sf))

> radars_sp <- as(radars_sf, 'Spatial')
> radars_sp <- spTransform(radars_sp, CRS("+init=epsg:2154"))
> radars_sp <- aggregate(x = radars_sp["radars"],
+                        by = grid_cagb,
+                        FUN = length)
> radars_sf <- st_as_sf(radars_sp)
> radars_sf[is.na(radars_sf$radars),]$radars <- 0

> grid_cagb_sf <- st_join(grid_cagb_sf, radars_sf,
+                         join = st_nearest_feature)
```

First, the function `st_intersection` is used in order to keep only speed cameras locations located in the observation window. Then, a `SpatialPolygonsDataFrame` `radars_sp` is created using the function `as(, 'Spatial')`. The spatial aggregation is performed by applying the function `aggregate` from package `sp` to the spatial object `radars_sp`. The argument `by = grid_cagb` specifies that the aggregation has to be done according to it and `FUN = length` specifies that the aggregation consists in the sum of the values. Then, the object `radars_sp` is converted once more into a `sf` object in order to be merged with `grid_cagb_sf`. To do so, the function `st_join` is used with argument `join = st_nearest_feature` which, as its name implies, join according to nearest geometries. As `radars_sf` and `grid_cagb_sf` have the same geometries, this command-line acts as a classical `merge` operation.

All target variables mentioned above have been interpolated to grid_cagb_sf similarly as the variable *radars*.

Road network

The structure of road network has been extracted from the French open platform data.gouv as a `.shp` file and loaded as a `sf` object with line type geometries.

```
> road_sf <- st_read("DATA/l_voiries_cagb_lg_r27.shp")
> road_sf <- st_transform(road, crs=2154)
```

The objective here is to be able to calculate the length of municipal roads and the length of national roads (departmental, national and highways) per cells of the grid grid_cagb_sf. To do so, one has to simply compute the length of the roads in the intersection zones, just as follows:

```
> road_sf <- road_sf[, c("statut")] %>%
+      filter(statut %in% c("COMMUNALE",
+                           "COMMUNAUTAIRE",
+                           "VOIE PRIVEE"))
> road_sf <- st_transform(road_sf, crs = st_crs(grid_cagb_sf))
> inters <- st_intersection(grid_cagb_sf, road_sf)
> inters$len <- st_length(inters)
> grid_cagb_sf$Id <- 1:nrow(grid_cagb_sf)
> join <- st_join(grid_cagb_sf, inters)
> interpol <- group_by(join, Id) %>%
+      summarize(length = sum(len))

> interpol$length <- as.numeric(interpol$length)
> interpol[is.na(interpol$length),]$length <- 0
> interpol <- st_drop_geometry(interpol)

> grid_cagb_sf <- merge(data_sf, out, by = "Id")
> grid_cagb_sf <- grid_cagb_sf %>%
+      select(-Id)
> grid_cagb_sf <- grid_cagb_sf %>%
+      rename(municipal_length = length)
```

The variable *statut* of road_sf gives the type of the road : COMMUNALE, COMMUNAUTAIRE and VOIE PRIVEE corresponds to municipal roads ; ETAT and DEPARTEMENT corresponds to national roads. The length of each road segments is computed with st_length from sf package. Then, the total road length is obtained by summing up according to cells, thanks to the variable Id created above. This variable helps later in the merge operation, in order to associate the length of the roads to grid_cagb_sf. This is another way to merge two `sf` objects, instead of st_nearest_feature seen before. Command-lines above created the variable *municipal_length* for municipal roads, the same method can be applied for the length of national roads.

2.3.3 Final support of covariate values

All interpolation methods made before bring to a spatial object `grid_cagb_sf` which has, per cell, values of 13 variables which are *prop18*, *prop65*, *health*, *school*, *college*, *shop*, *station*, *gasoline*, *leisure*, *intersection*, *radars*, *municipal_length* and *national_length*. One last step is to make the intersection between `grid_cagb_sf` and `owin_cagb_sf` in order to keep only cells that fall in the CAGB :

```
> st_crs(owin_cagb_sf) <- st_crs(grid_cagb_sf)
> grid_cagb_sf <- st_intersection(grid_cagb_sf, owin_cagb_sf)
> area(grid_cagb_sf)

[1] 513848149
```

The grid `grid_cagb_sf` covers now an area of 513 848 149 square meters and is composed of 1 362 cells. Each variable can be plotted, here are command-line examples for Fig 2.9 which represents range of values of the variable *shop* per `grid_cagb_sf` cells :

```
> grid_cagb_sf %>%
+      dplyr::select(shop) %>%
+      ggplot() +
+          geom_sf(aes(fill = shop))+
+          scale_fill_gradient(low = "#F5FAFD", high = "#AA2A10",
+          breaks = c(0,50), limits = c(0,50)) +
+          coord_sf(xlim = c(905000, 950000),
+                   ylim = (6665000, 6705000),
+                   crs = st_crs(2154), datum = proj_plot) +
+          theme_bw() +
+          theme(panel.grid.major = element_line(colour = "black",
+                                                linetype="dashed",
+                                                size=0.1),
+                panel.grid.minor = element_line(colour = "black",
+                                                  linetype="dashed",
+                                                  size=0.1)) +
+          xlab("Longitude") +
+          ylab("Latitude") +
+          labs(fill = paste("Shops located", "\n", "per cells"))
```

Values of *shop* variable with a small range from 0 to 50 have been plotted, instead of real values that goes to 353, it is only for the sake of clarity. Hence, grey cells are cells where values are more than 50. The reader may find the plots of every remaining variables in Appendix A.

Finally, the covariate data are exported with the function `st_write` from package `sf` as follows:

```
> st_write(obj = grid_cagb_sf, "DATA/grid_cagb_sf.shp", delete_layer =
    TRUE)
```

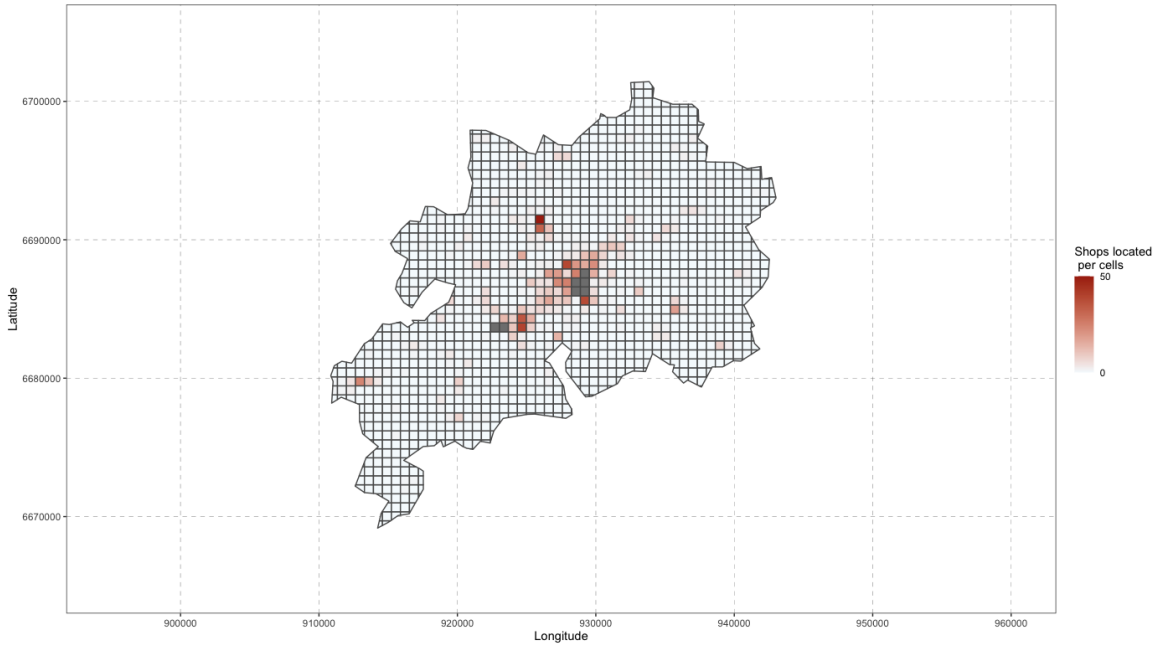

FIG 2.9: Plot of *shop* variable values, between 0 and 50, per cells of `grid_cagb_sf`.

Chapter 3

Spatial modelling road accidents in the urban community of Besançon using log-Gaussian Cox processes [1]

Homogeneous and inhomogeneous Poisson point processes, Poisson models aggregation and log-Gaussian Cox processes

In order to prevent and/or forecast road accidents, the statistical modelling of spatial dependence and potential risk factors is a major asset. The focus in the following is on the georeferenced location of accidents. We crossed these events with covariates characterizing the study geographical area such as sociodemographic and infrastructure measures. After a variable selection (poisson model, poisson models aggregation and random forest), the occurrence of accidents was modelled by using a spatial log-Gaussian Cox process. The results of this analysis enable us to identify principal risk factors of road accidents and critical areas. The data used are road accidents that occurred between 2017 and 2019 in the CAGB (urban community of Besançon).

3.1 Introduction

Chapter 1 focused on giving a multivariate statistical analysis of road crash data from the French region of Franche-Comté with special attention to road crash gravity. This analysis has a behavioural preventive character. In order to fulfill the main goal of spatial analysis of road crashes, the subsequent analyses should be focused on the spatial preventive character. Chapter 2 focused on the spatial point pattern of road accidents of the CAGB (*Communauté d'Agglomération du Grand Besançon*) to be fitted and associated covariates. More particularly, it detailed the spatial tools used on the software `R`. The main goal now,

[1] This chapter leads to an article writing in progress for publication.

with the available covariates, is to set boundaries for CAGB zones to be avoided and identify the accident risk factors. Our road crash data are considered as a point pattern and assumed to be a realization of an underlying stochastic process called *point process*. The crucial element when studying point processes is the expected number of events per unit area, known as the intensity of the point process. The context of road accidents directly implies the consideration of the intensity of the process as a random process. The point process proposed to model our road crash data in this chapter is the *log-Gaussian Cox process* (Moller et al., 1998) (LGCP) for which the log-intensity is a Gaussian process.

The current chapter focuses on the fitting of LGCP and similarly as Chapter 2, the objective is to delve deeper into the understanding and practical implementation of the spatial LGCP using `R`.

The following librairies will be used :

```
> library(fields)
> library(ggplot2)
> library(leaflet)
> library(lgcp)
> library(maptools)
> library(miscFuncs)
> library(raster)
> library(sf)
> library(sp)
> library(spatstat)
> library(tidyverse)
```

`R` packages `fields` (Nychka et al., 2021), `leaflet` (Cheng et al., 2021), `maptools` (Bivand et al., 2021b), `raster` (Hijmans et al., 2022), `sf` (Pebesma et al., 2022), `sp` (Pebesma et al., 2021) and `spatstat` (Baddeley et al., 2021b) are tools for spatial data. The packages `lgcp` (Taylor et al., 2021) and `miscFuncs` (Taylor, 2021) are packages used to fit the statistical model LGCP. Finally `ggplot2` (Wickham et al., 2021) and `tidyverse` (Wickham, 2021) are used for plot and for data wrangling basic operations.

The current chapter is structured as follows. Section 3.2 introduces the notion of point processes and gives the main properties of homogeneous and inhomogeneous Poisson point processes. It also gives two possible nonparametric estimation methods of the intensity of a point process. Section 3.3 first gives the statistical definition of LGCPs and presents various possible inference methods for the intensity process: *minimum contrast method*, *maximum likelihood* and *Bayesian inference*. As the Bayesian inference is the common choice in practice and is adopted in this chapter, related methods such as computation aspects and predictions are detailed. Finally, this section gives also performance assessment tools and risk measures. We suggest in Section 3.4 a new variable selection method based on an aggregation of Poisson regressions and the permutation variable importance criterion. Finally, Section 3.5 corresponds to the fits of the LGCPs using the covariates thus selected previously. The best LGCP model is used to create maps indicating the riskiest areas of the CAGB and to identify the risk factors. A special attention is accorded to the implementation on the software `R` of the theoretical tools defined in the last sections. A short conclusion

is given in Section 3.6 and supplementary materials (such as Bayesian computation tools implementations) are given in Section 3.7.

3.2 Point process basics

Section 2.2 in Chapter 2 introduced the notion of spatial point pattern. This refers to observed spatial locations of events in some set $W \subset \mathbb{R}^2$ (Baddeley et al., 2015; Cressie and Wikle, 2011). More particularly, consider a point pattern is a set of points $X = \{x_1, \ldots, x_n\}$ where $x_i \in W \subset \mathbb{R}^2$ has a known location for all $i = 1, \ldots, n$. The number n of the points in the pattern is not fixed in advance and may be any nonnegative number including zero. For a bounded region $B \subset W$, $X \cap B$ denotes the subset of X consisting of points falling in the region B and $n(X \cap B)$ denotes the number of points of X that fall in B. The analysis of a spatial point pattern lies on many characteristics. Indeed, we can ask ourself: Are the points uniformly located over a region or instead clustered ? Does the density of points depend on an explanatory variable ? and so on... Hence, the analysis does not lie on the points themselves but about the way the points were generated (Baddeley et al., 2015, Chapter 5). Indeed, a spatial point pattern X is seen as a realization of an underlying process $\mathbf{X}$ denoted as *point process*.

Point processes are widely used in several fields. For instance epicentres of earthquakes, gorilla nesting sites, positions of trees in a forest or locations of cases of cancer are point patterns that can be modelled by point process models. Many other examples have been developed in the literature: Mohler (2014) used data consisting of violent crimes occurring in Chicago in order to predict homicide; Olsbo et al. (2013) wanted to more precisely describe the spatial structure of epidermal nerve fibers (the outmost part of the skin); Murotani et al. (2019) analysed air voids which can be present in the concrete and may affects its properties; then in the aim of estimating biodiversity of forests, Tovo and Favretti (2018) focused on the tendency of plants to form clusters of individuals for the Barro Colorado Island and Pasoh rainforests. Several different points of view, statistically speaking, were considered in these articles.

Let $\mathbf{X}$ be a point process, namely a stochastic mechanism whose outcomes is a point pattern $X = \{x_1, \ldots, x_n\}$ of any size n, where n is a nonnegative integer. As remarked by Cressie and Wikle (2011), *one fundamental property of a spatial point process is the expected number of events in a given region* B, namely $\mathbb{E}(n(\mathbf{X} \cap B))$. This can be evaluated by means of the *intensity function* $\lambda(u)$ that is a measure of the potential for an event to appear at any location $u \in B \subset W$. Let denote by B_u a small region located at u of volume $|B_u|$. Mathematically speaking, the intensity function $\lambda(\cdot)$ is defined by:

$$\lambda(u) = \lim_{|B_u| \to 0} \frac{\mathbb{E}(n(\mathbf{X} \cap B_u))}{|B_u|}, \quad u \in W \tag{3.1}$$

provided that this limit exists. Then,

$$\mathbb{E}(n(\mathbf{X} \cap B)) = \int_B \lambda(u) du, \quad B \subset W.$$

One of the main goals in statistical modelling and analysis of spatial point pattern is the study of the intensity function $\lambda(\cdot)$ as it is a powerful tool to characterize the spatial behaviour of a point process. We start by considering first the simplest way to model a spatial point pattern, the *homogenous Poisson process*, for which the intensity function is constant everywhere and proceed next with the *inhomogeneous Poisson* of varying intensity function. More generally, the *Cox process* are modifications of the inhomogeneous Poisson process to incorporate random influences in the intensity function, namely the intensity function may depend now on unobservable external factors as well as observable covariates. Finally, we will consider the *log-Gaussian Cox process* (LGCP) which is a Cox process with log-intensity modelled by a Gaussian process. Our final goal is to model the spatial point pattern of CAGB road accident data by LGCP with log-intensity depending also on observable covariates reflecting socio-demographic as well as road characteristics of the CAGB region.

3.2.1 Homogeneous Poisson process

An idealized standard process is the *homogeneous* Poisson point process, also called *complete spatial randomness* (CSR), characterized by two properties:

HPP1 homogeneity : $\mathbb{E}[n(\mathbf{X} \cap B)] = \lambda|B|$, $\lambda > 0$, which means that the expected number of events falling in a region $B \subset W$ is proportional to its area;

HPP2 independence : $n(\mathbf{X} \cap B_1), n(\mathbf{X} \cap B_2), \ldots, n(\mathbf{X} \cap B_m)$ are m independent random variables, whenever $B_1, B_2, \ldots, B_m$ are disjoint regions of W.

Property **HPP1** is equivalent to the fact that the intensity function $\lambda(\cdot)$ defined in Eq (3.1) is constant and equal to $\lambda > 0$. The parameter λ represents the average number of random points per unit area, which in our case of road crashes can be seen as an occurrence rate, and is known as the *intensity* of the point process. The two properties **HPP1** and **HPP2** together imply that the number $n(\mathbf{X} \cap B)$ of points falling in a region B has a Poisson distribution with mean $\lambda|B|$, $\lambda > 0$. Furthermore, the homogeneous Poisson point process is stationary and isotropic which respectively mean that the distribution of $\mathbf{X}$ is invariant under translation and rotation. Homogeneous Poisson point processes are extensively presented in Diggle (2013, Chapter 4) or Baddeley et al. (2015, Chapter 5).

A way to simulate homogeneous Poisson point process realizations is as follows:

```
> plot(rpoispp(30, nsim = 3), main = "", pch = 20, cex = 1)
```

Fig 3.1 gives three possible realizations of a homogeneous Poisson point process with an intensity value λ of 30. The simulations have been made with `rpoispp` from package `spatstat`. This function takes the intensity of the Poisson process as first argument, hence in the homogenous case, a constant is required. The default window, in which the point pattern is simulated, is the unit square. The function returned a list of `ppp` objects as `nsim = 3` specified that three realizations were required.

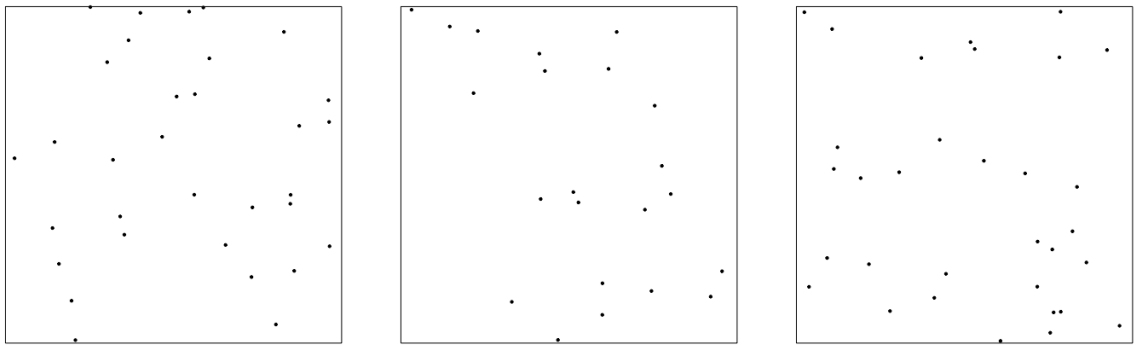

FIG 3.1: Three simulated realizations of the Poisson point process with intensity 30 in the unit square.

3.2.2 Inhomogeneous Poisson process

The Poisson process of completely random patterns presented in the previous section is mostly used as the standard reference in comparison with other patterns. Indeed, the intensity $\lambda > 0$ of this process is constant. However, the most important model for many practical purposes is the *inhomogeneous* Poisson point process. This class of models is a modification of homogeneous Poisson point processes in which the constant intensity is replaced now by a spatially varying intensity function $\lambda(u)$ depending on the spatial location u, $u \in W \subset \mathbb{R}^2$. The inhomogeneous Poisson point process $\mathbf{X}$ with intensity function $\lambda(\cdot)$ is defined by the following two properties:

IPP1 intensity : for all bounded region $B \subset W$, $\mathbb{E}[n(\mathbf{X} \cap B)] = \int_B \lambda(u)\mathrm{d}u$, supposed to be finite;

IPP2 independence: $n(\mathbf{X} \cap B_1), n(\mathbf{X} \cap B_2), \ldots, n(\mathbf{X} \cap B_m)$ are m independent random variables, whenever $B_1, B_2, \ldots, B_m$ are disjoint regions of W.

Property **IPP1** implies that the point process $\mathbf{X}$ has intensity function $\lambda(\cdot)$, has defined in EQ (3.1). The two properties together imply the Poisson distribution: for all bounded region $B \subset W$, $n(\mathbf{X} \cap B)$ has a Poisson distribution with mean $\int_B \lambda(u)\mathrm{d}u$. For more details on inhomogeneous Poisson point process, see for example Diggle (2013, Chapter 6), Baddeley et al. (2015, Chapter 9) and the references therein. There is no restrictions on the function $\lambda(u)$ to be used, as long as the function is non-negative and locally integrable. For example we can simulate an inhomogeneous Poisson process with intensity function $\lambda_1(u) = 200(x+y)^2$, $u = (x, y) \in \mathbb{R}^2$, as follows:

```
> lambda_1 <- function(x, y) {200 * (x+y)^2}
> lambda_1_im <- as.im(lambda, W = square(1))
> plot(lambda_1_im, main = "")
> plot(rpoispp(lambda_1, win = square(1)), main = "",
+      pch = 20, cex = 1, add = TRUE)
```

Command-lines above give Fig 3.2, that is, the plot of a realization of an inhomogeneous Poisson process overlaying the intensity function λ_1. To do so, the first step is to create the intensity function. Then in order to obtain a realization of the point process with λ_1 as intensity function, the same function as in Section 3.2.1 `rpoispp` is used but this time with the function λ_1, not a constant. In order to overlay the plots of the theoretical intensity and the simulated point pattern, the function is converted to a pixel image using the function `as.im` from the package `spatstat`. The object `lambda_1_im` of class `im` is plotted and the point pattern is simply associated to the later plot by specifying `add = TRUE`. Note that as before, the study window chosen is the unit square (`square(1)`). The Fig 3.2 shows a high density of points in the top right corner of the window.

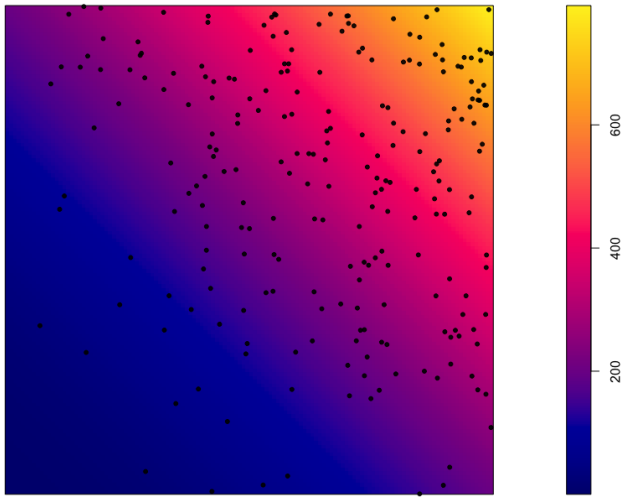

Fig 3.2: Plot of intensity function λ_1 and realization of the associated inhomogeneous Poisson point process in the unit square.

The intensity function of a point process is assumed to be the major part of the generation of the data and the most important is to know whether the intensity is homogeneous or not. Inhomogeneity of the intensity reflects the spatial distribution of points which may be more or less abundant in different regions of space. For most real life problems, it is more realistic to assume that the underlying point process is inhomogeneous, that means, driven by a non constant intensity function. In our case, Fig 2.1 from Section 2.2 in Chapter 2 showed a high density of points in the middle of the window. In order to inspect how the intensity function in our case would be, the same plot as Fig 3.2 is obtained as follows:

```
> cagb_ppp <- readRDS("DATA/cagb_ppp.rds")

> plot(density(cagb_ppp), main = "")
> plot(cagb_ppp, main = "", pch = 20, cex = 1, add = TRUE)
```

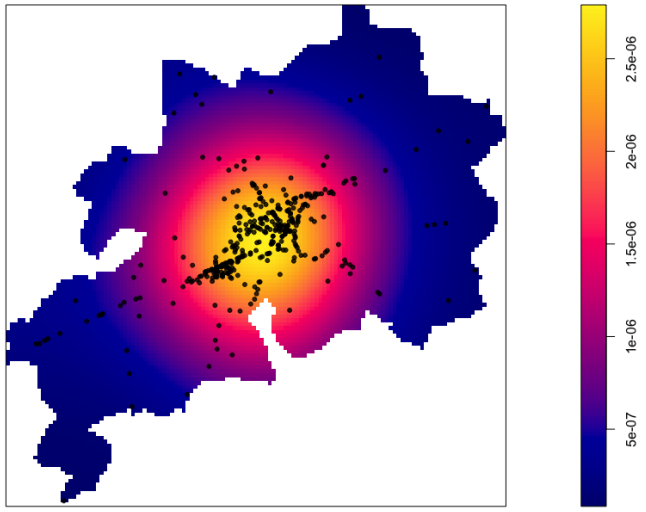

FIG 3.3: Plot of road accident spatial point pattern in the CAGB overlapping an estimation of its intensity function.

The point pattern `cagb_ppp` created in Section 2.2 is loaded with `readRDS`. A possible estimate of the intensity of our point process here is produced with the function `density` from `spatstat` which computes an intensity function from a point pattern in argument (more details will be explained later about this function in the next Section). As expected, the abundance of points in the middle of FIG 3.3 is reflected in intensity values. Indeed, the warmer the colors of the intensity, the more points there are.

3.2.3 Estimation of the intensity function of a point process

The previous section shows how the investigation of the intensity of a point process is an important step in point pattern analysis. Indeed, the intensity function is the expected density of point per unit area and represents hence a reference descriptive characteristic for a point process. It is interpreted as the incidence rate of the events recorded in the point pattern. This rate of occurrence is the most important property of a point process to be analysed when the goal of analysis is the prevention of the events. The main task in order to fulfill our goal of geographical anticipation of road crashes of the CAGB is to map the spatial variation in intensity and investigate, if it is the case, whether the intensity depends on covariates. As said before, the intensity of the point process that generated our point pattern of road crashes will be considered as inhomogeneous. Some preliminary estimation investigations of the intensity will be presented briefly below.

If the intensity is suspected to be inhomogeneous, it can be estimated by nonparametric methods such as *quadrat counting*, mentioned in Section 2.2 of Chapter 2, or *kernel estimation*. More details and other existing estimation methods are given for example in Baddeley et al. (2015, Chapter 6) and Diggle (2013, Chapter 5).

Quadrat counting

Quadrat counting is a statistical technique to estimate the intensity function and that allows to check if regions of equal area, from the observation window, have almost the same number of points (as they would have if the point process were homogeneous). Indeed subregions (quadrats) B_j, $j = 1, \ldots, m$, of the observation window each have $n_j = n(X \cap B_j)$ number of points. All are estimates of $\mathbb{E}[n(\mathbf{X} \cap B_j)]$ and are equal on average in the case where the intensity is homogeneous. A simple estimate of the intensity function, as average intensity in each quadrat, can be obtained by dividing each quadrat counts by the area of the associated quadrats. The quadrat method is very similar to the histogram method for density estimation.

The quadrat counts made in Fig 2.6 from Section 2.2 in Chapter 2 have been created once again below with `quadratcount` from `spatstat` package. Then, estimates of the intensity function can be obtained with the function `intensity`, from `spatstat`, which takes an object `quadratcount` as argument. Fig 3.4 and Fig 3.5 are plotted as follows:

```
> q3_counts <- quadratcount(cagb_ppp, nx = 3, ny = 3)
> q9_counts <- quadratcount(cagb_ppp, nx = 9, ny = 9)

> par(mfrow=c(1, 2))
> plot(q3_counts, main = "")
> plot(intensity(q3_counts, image = TRUE), main = "")
> plot(q9_counts, main = "")
> plot(intensity(q9_counts, image = TRUE), main = "")
```

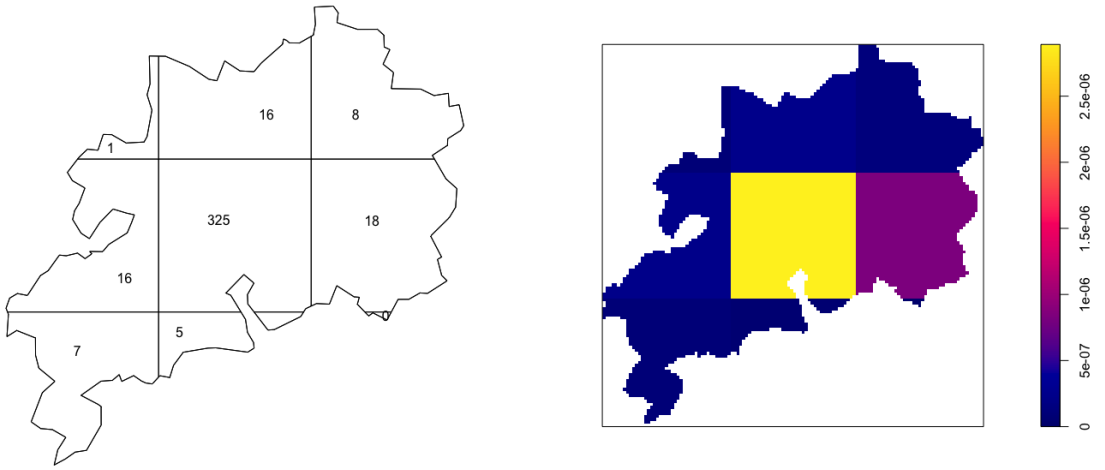

Fig 3.4: Quadrat counting for CAGB road accidents data. Quadrat counts 3×3 in left and intensity estimates (point per square meters) in right.

Intensity estimates in Fig 3.4 and Fig 3.5 suggest that the intensity may be quite elevated in the middle of the plots, as before with Fig 3.3.

One way to assess whether the intensity function is homogeneous or inhomogeneous is using the quadrat counting test of homogeneity. The null hypothesis of this test is that the intensity is homogeneous and the alternative one is that the process is not a homogeneous

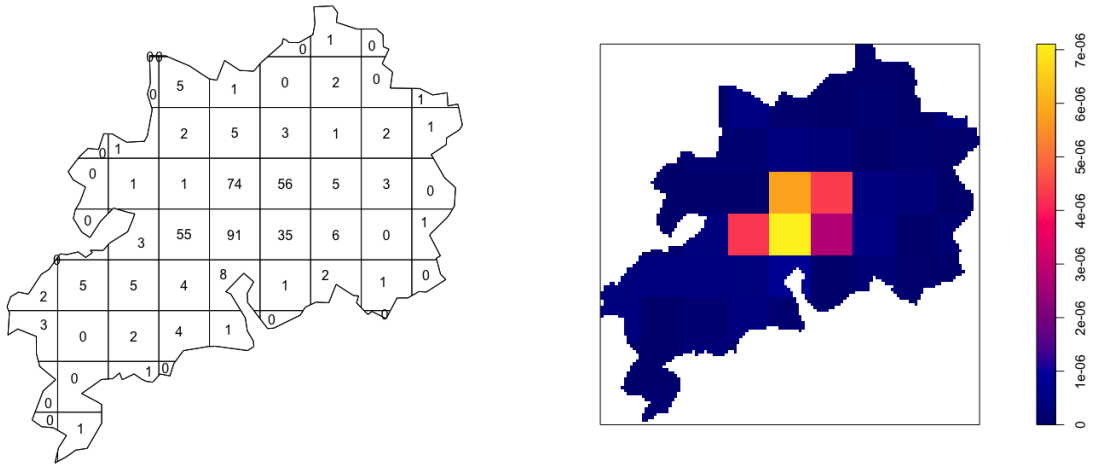

FIG 3.5: Quadrat counting for CAGB road accidents data. Quadrat counts 9×9 in left and intensity estimates (point per square meters) in right.

Poisson process. However, the power of the test depends on the size of quadrats and is only optimal when the quadrats are neither very large nor very small. This means that the decision of the test depends on the choice of quadrat size and it is then questionable. In order to show the weakness of this test, consider CAGB data and the estimation of the intensity function by quadrats of size 2×1 first and then, of size 9×9:

```
> quadrat.test(cagb_ppp, nx = 2, ny = 1)

Chi-squared test of CSR using quadrat counts

data:  cagb_ppp
X2 = 0.82724, df = 1, p-value = 0.7261
alternative hypothesis: two.sided

Quadrats: 2 tiles (irregular windows)
```

The quadrat counting test of homogeneity is performed with `quadrat.test` from `spatstat` package. Numbers of quadrats in the x and y directions have to be specified in the command-line. A grid 2×1 of quadrats for our point pattern has been created, the associated test results in not rejecting the null hypothesis of a homogenous Poisson process for the CAGB road accidents data. This result is not expected since the point pattern clearly exhibits a strong spatial inhomogeneity with more point in the middle of the window. Here the result is unsatisfactory because the quadrat size considered for the test is too big. However, shifting number of quadrats as `nx = 9` and `ny = 9`, for example, gives `p-value < 2.2e-16` which means that the null hypothesis is rejected and hence, we conclude to an inhomogeneous intensity function. Hence, quadrat counting can be used for standing assumptions on the intensity but should not be adopted for a final diagnostic.

Kernel estimation

Another way to estimate the intensity function nonparametrically is the kernel estimation. We use here the exact terms of Baddeley et al. (2015, Section 6.5.1, Page 168) who illustrates this method metaphorically and allows to visualize it in an interesting way:

"Our favorite analogy is to imagine placing one square of chocolate on each data point. Using a hair dryer we apply heat to the chocolate so that it melts slightly. The result is an undulating surface of chocolate; the height of the surface represents the estimated intensity function of the point process. The total mass of chocolate is unchanged.

There are many kernel estimators but only the standard one will be briefly presented here. For any spatial location u in the observation window $W \subset \mathbf{R}^2$, the *uncorrected* kernel estimator of $\lambda(u)$ is defined as

$$\tilde{\lambda}(u) = \sum_{i=1}^{n} \boldsymbol{\kappa}_\sigma(u - x_i) \tag{3.2}$$

where $u = (x, y) \in W$ and the kernel function $\boldsymbol{\kappa}_\sigma(\cdot)$ is the melted square of chocolate placed at data point location x_i. The kernel $\boldsymbol{\kappa}_\sigma$ may be a probability density function symmetric about the origin and the most used choice is the Gaussian distribution : $\boldsymbol{\kappa}_\sigma(u) = (2\pi\sigma^2)^{-1} \exp\{-||u||^2/2\sigma^2\}, u = (x, y) \in W$. The standard deviation $\sigma > 0$ of the Gaussian distribution is specified as the *smoothing bandwidth* of the kernel. This kernel estimator is named uncorrected as it does not take into account edge effects. Other kernel estimators with a correction are given in Baddeley et al. (2015, Chapter 6). For more details on the kernel method, the reader may refer to Silverman (1986).

An estimation of the intensity function of `cagb_ppp` with the uncorrected estimator can be obtained as follows:

```
> k_density_50 <- density(cagb_ppp, sigma = 50, edge = FALSE)
> plot(k_density_50, clipwin = owin_bes,
+      main = "", xlim = c(925000,932944), ylim = c(6685000, 6690000))
> contour(k_density_50, clipwin = owin_bes,
+         xlim = c(925000,932944), ylim = c(6685000, 6690000),
+         add= TRUE, drawlabels = F)
> persp(k_density_50, main="", zlab = "Intensity estimated")
```

As seen in Section 3.2.2, the function `density` is used. This function actually computes kernel density estimates. The argument `sigma` specifies the smoothing bandwidth σ as mentioned above. The estimate can be plotted with `plot`. Then, contour lines can be added to the previous plot using `contour`. On the other hand, in order to visualize this estimate as a perspective view, the function `persp` is used. For the sake of clarity, a focus is made using a smaller `owin` object : `owin_bes` by using the optional argument `clipwin` of the function `contour`. The object `owin_bes` has been created in the same way as `owin_cagb` seen in Section 2.2 in Chapter 2 and relates to the polygon boundaries of the city of Besançon, headquarter of the CAGB. The plot of the uncorrected kernel estimation of the intensity function with smoothing bandwidth $\sigma = 50$ is plotted in the left panel of Fig 3.6. The

perspective view is given in the left panel of Fig 3.7.

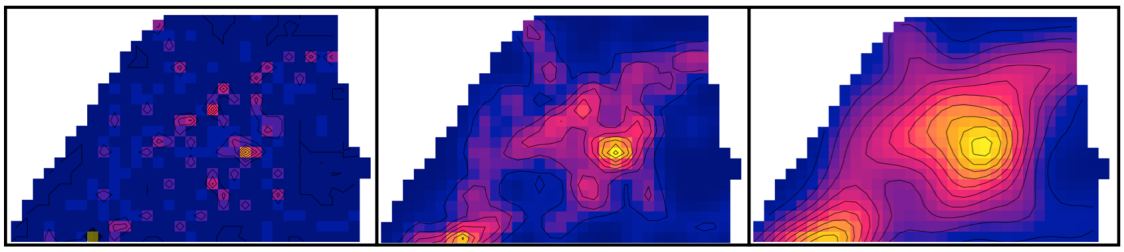

Fig 3.6: Kernel estimates of intensity for CAGB road crashes using different smoothing bandwidths. Left to right: smoothing bandwidth σ equal to 50, 250 and 500. Same colour code interpretation as Fig 3.3 (plot legend not displayed as it has been resized).

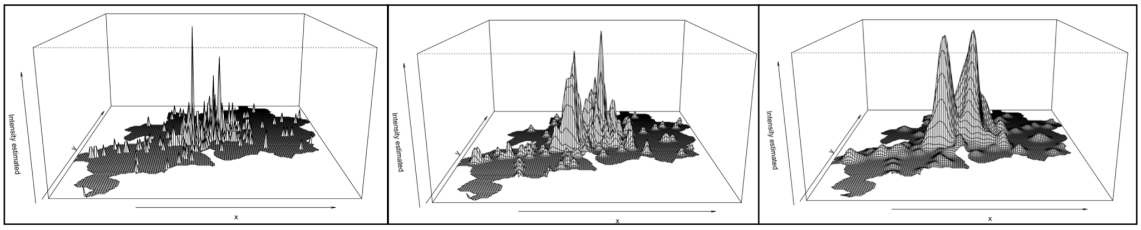

Fig 3.7: Perspective views of intensity kernel estimates of CAGB road crashes using different smoothing bandwidths. Left to right: smoothing bandwidth equal to 50, 250 and 500.

Varying the smoothing bandwidth of the uncorrected kernel estimator produces the remaining panels of both Fig 3.6 and Fig 3.7. The larger the bandwidth is, the smoother the estimated intensity is. Indeed, the smallest value $\sigma = 50$ produces an irregular intensity surface, while larger values appear to slightly smooth the intensity.

As for quadrat counting, where the size of quadrats is chosen by the user, kernel estimation depends heavily on the smoothing bandwidth σ and less on the choice of kernel. However, several methods are available for selecting the bandwidth such as algorithms minimising some criterion for example. The likelihood cross-validation method (Loader, 1999) is computed with `bw.ppl` as follows:

```
> smoothing_band <- bw.ppl(cagb_ppp)
> smoothing_band
> opt_density <- density(cagb_ppp, sigma = smoothing_band, edge =
   FALSE)
> plot(opt_density, main="")
> persp(opt_density, main="", zlab = "Intensity estimated")

 sigma
 787.5311
```

Smoothing bandwidth estimate value obtained by the chosen algorithm is $\hat{\sigma} = 787.5311$ (much larger than the ones tried before) and Fig 3.8 displays the kernel estimate of the

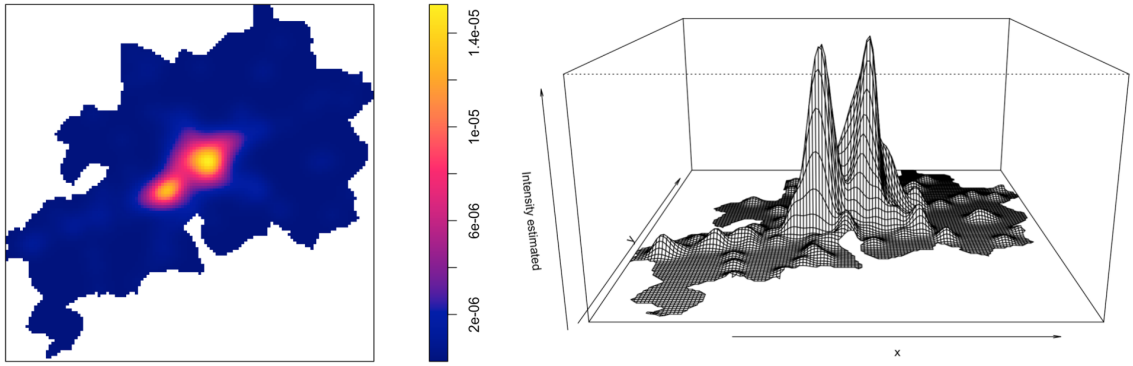

FIG 3.8: Plot of a kernel estimate of intensity of CAGB road crashes with smoothing bandwidth selected by likelihood cross-validation method. Left to right : classical plot and perspective plot.

intensity function of the point process that generated our CAGB road crashes point pattern computed for this value $\hat{\sigma}$. All these plots, and those produced with the quadrat counting method, suggest that the intensity may be elevated in the middle of the observation window, which actually almost corresponds to the south of Besançon city.

3.3 Log-Gaussian Cox Processes (LGCP)

The previous section introduced the notion of point processes and in particular, the intensity function of a point process. In the case of the inhomogeneous Poisson point process, the intensity function is varying spatially. Section 3.2.3 introduced two nonparametric methods used in order to estimate this intensity function. Other nonparametric methods exist and, moreover, methods which this time inspect how the intensity of points depends on the values of a covariate. Now, in this section, the statistical modelling of the intensity function of the point process that generated our CAGB road crashes point pattern will be investigated. More particularly, a log-Gaussian Cox process (LGCP) will be proposed to model the road crashes point pattern, where the intensity depends on covariates and also on a spatial Gaussian field capturing the spatial random effect.

Related works

The LGCP can be used to model various situations but its practical implementation faces important computational challenges. A few methods have been proposed for running and fitting this statistical model. For example Bayisa et al. (2020) used LGCPs in order to improve the northern Sweden ambulance system in both space and time. It seems that most of the emergency calls are located in populated areas, but the main concerns are for inhabitants of large rural areas who may need fast access to prehospital care. The fitted model has three components: a spatial component which was estimated by a quartic kernel whose bandwidth has been selected using K-means clustering; a temporal component which was estimated by a Poisson regression model; and a (unobservable) spatio-temporal Gaus-

sian process whose properties were estimated using minimum contrast estimation. Another field where LGCPs have proven useful is wildfire prevention. Opitz et al. (2020) used a spatio-temporal model with the aim of giving preventive measures and operational forecasts in the French Mediterranean basin. They faced a significant computational challenge due to the high dimension of the data. Indeed, a large amount of information from covariates such as for example vegetation type, weather or urbanization was used. Bayesian inference was made using the *integrated nested Laplace approximation* (INLA). Ramírez and Valencia (2021) trained LGCPs in the field of road crashes in order to understand and analyze the occurrence of accidents, from a spatial and temporal point of view, in Bogota the capital city of Colombia. They included several spatial and temporal covariates associated to accidents such as road characteristics, demographic conditions or even weather temporal factors. Bayesian estimation was made using *Markov chain Monte Carlo* (MCMC) method with the *Metropolis-adjusted Langevin algorithm* (MALA), a version of the commonly used *Metropolis-Hastings algorithm*. Diggle et al. (2013) worked on LGCP in order to illustrate and explain the use of the model. They developed different versions of the model with increasing complexity, each adapted to a specific example: an introductory spatial model whose events were hickory trees; a multivariate spatial model with four different types of events corresponding to four different genotypes of bovine tuberculosis; a spatial model with covariate information available for lung cancers observed on a geographical region of interest partitioned into a set of subregions; and finally a spatio-temporal model for gastro-intestinal disease. MCMC was also used for the Bayesian inference. Many other possible applications may be given, however few fast computational methods are available for the moment. We start by presenting briefly the statistical definition of a LGCP and the Bayesian approach used for parameter inference and continue next with the LGCP modelling of our road crash data and the associated computational algorithm.

3.3.1 LGCP statistical definition

As remarked in Section 3.2.2, the road crashes points seem to be clustered. This suggests to consider the intensity as inhomogeneous. More particularly, our data will be modelled by a *Cox* process. A Cox process is essentially a Poisson process for which the intensity function is random. It is a natural way to model a inhomogeneous point processes as it assumes that the unknown spatially varying intensity function $\Lambda(u)$ is random.

Formally, a Cox process is defined by the following two assumptions (Diggle et al., 2013):

CP1 there exists a non-negative stochastic process $\Lambda(\cdot) = \{\Lambda(u) : u \in W\}$, called *driving* intensity process;

CP2 Conditionally on the realization $\Lambda(\cdot) = \lambda(\cdot)$, the point process $\mathbf{X}$ is an inhomogeneous Poisson process with intensity function $\lambda(\cdot)$.

More explicitly, conditionally on $\Lambda(\cdot) = \lambda(\cdot)$, the count variable $n(\mathbf{X} \cap B_j)$ is Poisson distributed with mean $\int_B \lambda(u)(\mathrm{d}u)$ for all bounded $B \subset W$ and $n(\mathbf{X} \cap B_1), \ldots, n(\mathbf{X} \cap B_m)$ are (conditionally) independent whenever $B_1, \ldots, B_m$ are disjoint. We can see Λ and $\mathbf{X}$ as

two different levels of randomness, so that the Cox process is often called a *doubly stochastic point process*.

The random variation in the intensity function can be seen as a spatial random effect and is often attributable to an unobserved covariate. The spatial variation in the intensity function can make the point pattern look clustered, that is, with more abundant points in some areas. For instance in our case of road crashes, the curvature of a road can be a potential external factor. Indeed, a road more curved in some location can cause more accidents instead of another location where the road is completely straight. The situation can also be seen in a different way as: a road more curved will make the drivers slow instead of a road completely straight where drivers would accelerate and lead to more accidents as the speed is a risk factor. The observation remains the same, the curvature of the road affects the driving and hence, the number of accidents.

There are many possible ways to build a Cox process, the most natural approach to circumvent the nonnegativity assumption of the intensity function is to deal with the logarithm of the driving intensity, hence the concept of the log-Gaussian Cox process. A LGCP is a Cox process whose log-intensity is modelled by a Gaussian process (Moller et al., 1998), say $Y = \{Y(u) : u \in \mathbb{R}^2\}$ (i.e. the joint distribution of any finite vector $(Y(u_1), \ldots, Y(u_n))$ is Gaussian). More precisely,

$$\log(\lambda(u)) = Y(u), \quad u \in W. \tag{3.3}$$

The Gaussian random field Y is usually assumed to be the restriction on W of a stationary and isotropic Gaussian random field defined on $\mathbb{R}^2$ (i.e. invariant under translations and rotations), so that it is completely specified by its mean $\mu = \mathbb{E}[Y(u)]$, its variance $\sigma^2 = \text{Var}[Y(u)]$ and its covariance function

$$C(h) := \text{Cov}(Y(u), Y(u+h)), \quad u, h \in \mathbb{R}^2.$$

As the covariance function depends only on the distance between points and not on their directions, it can be modeled by

$$\text{C}(h) = \sigma^2 \mathbf{r}(||h||/\phi), \quad h \in \mathbb{R}^2,$$

where $\phi > 0$ is a scale parameter and $||\cdot||$ is a suitable norm on $\mathbb{R}^2$ (for instance the Euclidean norm), $u \in \mathbb{R}^2$. Moller et al. (1998) suggest several parametric forms for the function $\mathbf{r}$ that ensure that the covariance function $C(\cdot)$ is well-defined (i.e. semi-definite positive function). A common choice for $\mathbf{r}$, that we will also adopt for the next of this chapter, is the exponential function which leads to the following covariance function:

$$\text{C}(h) = \sigma^2 \exp\left(-\frac{||h||}{\phi}\right), \; h \in \mathbb{R}^2.$$

The covariance function of the Gaussian process Y can be specified using function `CovFunction` from package `lgcp` as follows:

```
> cf <- CovFunction(exponentialCovFct)
```

For the remaining of our work, we will also set $\mu = -\sigma^2/2$ which is a convenient re-parameterisation since it enables to have $\mathbb{E}[\Lambda(u)] = \mathbb{E}[\exp(Y(u))] = 1$ (Diggle et al., 2013; Taylor et al., 2015), meaning that we have no prior information on the mean random spatial effect. Note that due to properties of the log-Gaussian distribution, the intensity function $\lambda(\cdot)$ and by consequence the process $\mathbf{X}$ are totally specified by the parameters σ and ϕ of the Gaussian process Y, so efficient estimation of these parameters is crucial. The parameters of the Gaussian process Y are often transformed onto log-scale, $\eta = \{\log(\sigma), \log(\phi)\}$, as it is more appropriated for the computation algorithm that will be used further (Taylor et al., 2015).

In applications, one is often interested in the effect of spatial covariates. One way to introduce them in the model is to add a linear term in the log-intensity that depends on the covariates This results in the model

$$\log(\lambda(u)) = Z(u)^\top \beta + Y(u), \quad u \in W,$$

where $Z(u)$ is a vector of covariate values observed at the position u and β is a vector of unknown parameters accounting for the effect of the different covariates. The notation $^\top$ stands for the vector or matrice transpose. The parameters of the model now are $\eta = \{\log(\sigma), \log(\phi)\}$, the parameters of the Gaussian process Y, and β, the vector of covariate effects. For parameter estimation, three approaches can be considered: *minimum contrast estimation*, *maximum likelihood* and *Bayesian estimation* (Diggle et al., 2013) which we describe in Sections 3.3.2–3.3.4 below.

Finally, in practice, to fit the model in a tractable way, the common approach is to make computation on a very fine regular grid. The study region of space is divided into a $C \times C$ grid of equally spaced cells and the log-intensity is assumed to be constant over each grid cell. Hence, the log-intensity within a given cell, for example the ith cell, is constant and specified by its value at its centroid c_i, $i = 1, \ldots, C^2$. The number of events, fitted with a LGCP, is then treated as cell counts on this grid :

$$n(\mathbf{X} \cap c_i) \sim \text{Poisson}(\lambda(c_i))$$

with

$$\log(\lambda(c_i)) = Z(c_i)^\top \beta + Y(c_i), \tag{3.4}$$

where $n(\mathbf{X} \cap c_i)$ denotes the number of events in the ith cell of the grid with centroid c_i, for $i = 1, \ldots, C^2$. Usually, the grid is fine enough so that the cell-counts are very small (ideally 0 or 1) but it should allow a good compromise between accuracy of approximation and computational complexity. The computational grid is also useful for the Bayesian inference parameters as described in Section 3.3.4.

3.3.2 Minimum contrast method

We consider first the LGCP model with the intensity process modelled by Eq (3.3). The parameters to be estimated are σ and ϕ. The *minimum contrast method* is one way to estimate these unknown parameters.

The minimum contrast method, also known as *least-squares* approach, consists in choosing the parameters $\hat{\sigma}$ and $\hat{\phi}$ which minimize the squared discrepancy between the empirical and the theoretical second-order moments. Let $C(\cdot)$ be the covariance function of the Gaussian process Y as presented in Section 3.3.1. Moller et al. (1998) chose estimate values $\hat{\sigma}^2$ and $\hat{\phi}$ of respectively σ^2 and ϕ that minimize the discrepancy between a nonparametric estimate of the covariance function $\hat{C}(\cdot)$ and the theoretical one $C(\cdot)$ as follows

$$\int_{\epsilon}^{u_0} \left[\hat{C}(u)^{\alpha} - C(u)^{\alpha}\right]^2 \mathrm{d}u$$

where $0 \leq \epsilon < u_0$ and $\alpha > 0$, $u \in \mathbb{R}^2$. Moller et al. (1998) gave possible example choices for the parameters ϵ, u_0 and α. Usually, ϵ is the minimum distance between two points in the point pattern and u_0 and α are user-specified constants chosen so that the above integral is well defined. However, it seems difficult to give appropriate values for these parameters. Hence, minimum contrast estimation is generally considered as useful for providing preliminary estimates of the parameters (Diggle et al., 2013; Taylor et al., 2015). The minimum contrast method can be computed as follows:

```
> minimum.contrast(data = cagb_ppp,
+                  model = "exponential",
+                  method = "g",
+                  intens = density(cagb_ppp),
+                  transform = log)

[Univariate spatial minimum contrast]
Nonparametric heterogenous PCF estimation...done.
Starting values are (321.61, 2.99); optimising exponential correlation
    function...

done.
   $estimates
            scale variance
   [1,] 617.4396 1.850986

   $discrepancy
       Squared discrepancy
   [1,]            2150.972
```

The function `minimum.contrast` from `lgcp` package takes as argument the point pattern `cagb_ppp`. Then the assumed theoretical form of the spatial correlation function is specified in the argument `model`. Various definitions of the minimum contrast method are given in Brix and Diggle (2001), Diggle et al. (2013) and Taylor et al. (2015). In the package

`lgcp`, the parametric functions that can be used are the pair correlation function (PCF) g and Ripley's K function (Ripley, 1977). The method of spatial minimum contrast here is defined with `method = "g"` and is associated with `transform = log` which simply plays for contrast criterion. Finally, an estimation of the intensity of `cagb_ppp` is specified using the function `density` as seen in Section 3.2.3. The parameter estimates are then $\hat{\phi} = 617.440$ and $\hat{\sigma}^2 = 1.851$.

These values will be useful as preliminary estimates in the computation algorithm further. In particular, they might be helpful for the choice of the grid size. Indeed, the scale parameter ϕ of the covariance function $C(\cdot)$ of Y actually determines the spatial correlation in the process. Hence the approximate spatial scale of 617 meters tells that a grid of close dimension cells might be necessary to capture the dependence structure in the Gaussian process Y. The choice of such an appropriate grid size can be seen as follows:

```
> chooseCellwidth(cagb_ppp, cwinit = 650)
```

The function `chooseCellwidth` from the package `lgcp`, as its name implies, helps in choosing the cell width in order to set up the computational grid. The desired cell width is specified in the argument `cwinit`. In our case, a cellwidth of 650 meters is chosen as it might be appropriate for an efficient computational grid size. The above command-lines produce the plot of the observation window of the `ppp` object given and computational grid associated.

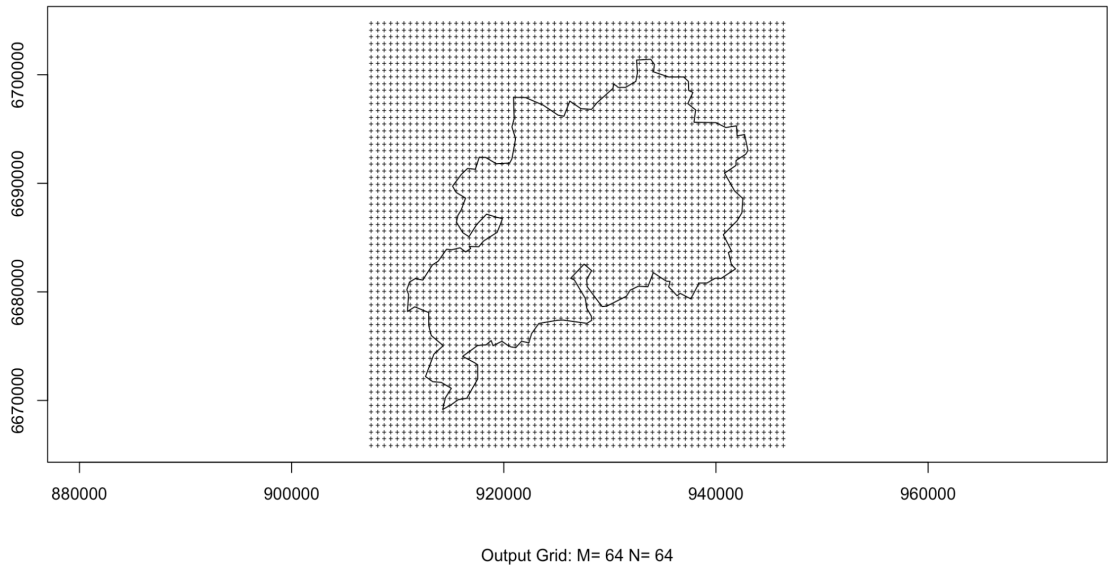

FIG 3.9: Output of `chooseCellwidth(cagb_ppp, cwinit = 650)`.

The FIG 3.9 displays output grid of size 64×64. One the cell width is chosen, it remains to create the grid. The object `polyolay`, created in Section 2.3.1 from Chapter 2 is loaded as follows:

```
> polyolay <- readRDS("DATA/polyolay.rds")
```

Remind that the grid have been created with a cell width of 650 meters.

3.3.3 Maximum Likelihood Estimation

Maximum likelihood is one of the most used method for estimating unknown parameters. However, as we will briefly describe below, in the case of Cox and LGCP models, this approach can not be used.

Consider first a general Cox point process $\mathbf{X}$. For a realization of the intensity process, $\Lambda(u) = \lambda(u)$, we have an inhomogeneous Poisson point process of intensity $\lambda(\cdot)$ governed by some parameter θ. In this case, the likelihood for θ (computed with respect to the reference homogeneous Poisson process of intensity $\lambda = 1$) is given by (Baddeley et al., 2015, Chapter 9)

$$\ell(\theta\,; X) = \lambda(x_1)\lambda(x_2)\dots\lambda(x_n)\exp\left(\int_W (1-\lambda(u))\mathrm{d}u\right), \tag{3.5}$$

where $X = \{x_1, \dots, x_n\}$ is the observed point pattern within the observation window W. The first term of the likelihood given in Eq. 3.5 is the probability of observing data in points $x_1, \dots, x_n$ and $\exp\left(\int_W(-\lambda(u))\mathrm{d}u\right)$ is the probability of not observing any other points in the window W. The constant factor $\exp\int_W du = \exp(|W|)$ is due to rescaling with respect to the Poisson process of $\lambda = 1$ and it is usually omitted from the likelihood expression.

Then, we get that the probability density of the Cox process $\mathbf{X}$ is given by (Baddeley et al., 2015, Chapter 12)

$$\mathcal{L}(\theta\,; X) = \mathbb{E}\left[\prod_{i=1}^{n}\lambda(x_i)\exp\left(-\int_W \lambda(u)\mathrm{d}u\right)\right].$$

The high dimensionality of the integration makes this likelihood analytically intractable (Moller et al., 1998; Diggle et al., 2013; Baddeley et al., 2015) even in the case of computation on a fine regular grid. The commonly used alternative is then the Bayesian inference.

3.3.4 Bayesian inference

Our goal is to use the data $X = \{x_1, \dots, x_n\}$ in order to make inferences about the latent process Y and the parameters $\theta = (\eta, \beta)$ which parametrize the covariance function of Y and the covariate effects (see Eq (3.4)). In the Bayesian paradigm, Y as well as θ are considered as random variables and the statistical problem may be summarized by a *hierarchical model* as described in Cressie and Wikle (2011, Chapter 2):

$$\begin{aligned} \text{parameter level} &: \text{distribution of } \theta; \\ \text{latent process level} &: \text{distribution of } Y|\theta; \\ \text{observation level} &: \text{distribution of } \mathbf{X}|\theta, Y. \end{aligned}$$

Here the process Y is unobserved and often referred to as a *latent process* and only the point pattern X is observed. In the LGCP case, the hierarchical model writes (Cressie and Wikle,

2011, Chapter 4):

$$\begin{aligned}
\text{parameter level} &: \theta = (\beta, \eta) \text{ with } \beta \text{ the covariate effects and} \\
&\quad \eta = (\log(\sigma), \log(\phi)) \text{ parametrizing the covariance functions of } Y; \\
\text{latent process level} &: \text{Given } \theta,\ Y(\cdot) \text{ is a Gaussian process and} \\
&\quad \Lambda(u) = \exp\left(Z(u)^\top \beta + Y(u)\right),\ u \in W; \\
\text{observation level} &: \text{Given } \Lambda(\cdot) = \lambda(\cdot),\ \mathbf{X} \text{ is an inhomogeneous Poisson process} \\
&\quad \text{with intensity function } \lambda(\cdot).
\end{aligned}$$

To make inferential statements, the Bayesian paradigm assigns a prior distribution $\pi(\theta)$ to the model parameters $\theta = (\beta, \eta)$ and focuses on the posterior distribution $\pi(\theta, Y|X)$, the conditional distribution of θ and Y given the observations X, given by Bayes' Theorem:

$$\begin{aligned}
\pi(\theta, Y|X) &\propto \pi(X|\theta, Y)\pi(\theta, Y) \\
&\propto \pi(X|\theta, Y)\pi(Y|\theta)\pi(\theta).
\end{aligned}$$

The latent field is unobserved and plays, during the estimation procedure, almost the same role as an unknown parameter. For this reason, and for the convenience of notation, we will note $\Theta = (\theta, Y)$ the unobserved part of the model to be estimated (both parameters and latent process). We then have the following model:

$$\begin{aligned}
\text{unobserved level} &: \Theta \text{ with distribution } \pi(\Theta) \\
\text{observation level} &: \mathbf{X} \text{ with distribution } \pi(X|\Theta),
\end{aligned}$$

where

$$\pi(\Theta) = \pi(Y|\theta)\pi(\theta)$$

and

$$\pi(\Theta|X) \propto \pi(X|\Theta)\pi(\Theta) = \pi(X|\theta, Y)\pi(Y|\theta)\pi(\theta).$$

Computational aspects

From a computational point of view, the Bayesian inference for LGCP presents two main difficulties: the choice of the prior distribution and the computation of the posterior distribution. As $\theta = (\beta, \eta)$, we can chose a prior of the product $\pi(\theta) = \pi(\beta)\pi(\eta)$. The package `lgcp` allows to choose a multivariate Gaussian prior for β and a multivariate Gaussian prior for $\eta = (\log \sigma, \log \phi)$:

$$\beta \sim \mathcal{N}(\mu_\beta, \Sigma_\beta) \qquad \text{and} \qquad \eta \sim \mathcal{N}(\mu_\eta, \Sigma_\eta).$$

Following Diggle et al. (2013) or Taylor et al. (2015), we have chosen

$$\beta \sim \mathcal{N}(0, 10^6), \quad \log \sigma \sim \mathcal{N}(\log(1), 0.15) \quad \text{and} \quad \log \phi \sim \mathcal{N}(\log(2000), 0.15).$$

The prior for β corresponds to a very flat prior that conveys almost no information (zero mean and very high variance). Note that $\mu_\phi = 2000$ has been chosen by preliminary fits on our data, this results from the trace plot given in FIG B.6 in APPENDIX B. These priors are defined in `lgcp` as follows:

```
> priors <- lgcpPrior(
+  etaprior = PriorSpec(LogGaussianPrior(mean = log(c(1, 2000)),
+                                        variance = diag(0.15, 2))),
+  betaprior = PriorSpec(GaussianPrior(mean = rep(0, 8),
+                                      variance = diag(10^6, 8))))
```

The above command-lines are an example of how priors can be defined with the function `lgcpPrior` for a model with seven covariates. Indeed, in `betaprior` the mean and the variance are of dimension eight (which correspond to intercept and the seven covariates). The object `priors` will be used in a LGCP fit further.

Actually, with LGCP, it is not possible to have an analytically tractable expression for the posterior $\pi(\theta, Y|X)$. Hence, we use Markov chain Monte Carlo (MCMC) algorithm in order to simulate the unknown posterior distribution. Another way to approximate this unknown distribution is the integrated nested Laplace approximation (INLA). For more details on INLA and a comparative evaluation of the performance between MCMC and INLA see Taylor and Diggle (2013).

MCMC methods generate samples from a Markov chain whose stationary distribution is the target of interest, in our case $\pi(\theta, Y|X)$. The commonly used algorithm is the Metropolis-Hastings algorithm. The reader may find more details in Robert (1996); Robert and Casella (2004); Robert (2007); Brooks et al. (2011).

We note $\Theta = (\theta, Y)$ the unobserved part of the model (model parameters and latent process) and $\{(\Theta^{(N)}\}_{N\geq 1}$ the Markov Chain generated by the Metropolis-Hasting algorithm in order to estimate the posterior distribution $\pi(\Theta|X)$. The Markov-Chain is generated following the global principle given in Algorithm 1 and relies on proposal and acceptance/rejection. The proposal distribution is defined by a transition kernel $q(\Theta^*|\Theta)$.

Typically, the Markov chain $\{\Theta^{(N)}\}_{N\geq 1}$ produced by Algorithm 1 is irreducible, ergodic and converges to the target distribution $\pi(\Theta|X)$. For necessary conditions on q to ensure these properties, see Robert and Casella (2004) for example. Beyond this basic properties, the choice of q is critical to achieve quick convergence and good mixing properties of the Markov chain.

The design of q used in our Metropolis-Hastings type MCMC method is a mix of random walk and Langevin proposal kernels (Taylor et al., 2013). This method is known as the Metropolis-adjusted Langevin algorithm (MALA) and has been recommended by several authors working on inferential methods for spatial and spatio-temporal LGCP, for more details see Moller et al. (1998) and the references therein. We will only give basic elements, the reader may find further comprehensive details about MALA in Moller et al. (1998), Brix and Diggle (2001), Taylor et al. (2013) or Taylor et al. (2015). As suggested by the latter authors, it is better to work with a transformation of the Gaussian process Y, namely Γ for

Algorithm 1 Metropolis-Hastings

(I) Initialization $\Theta^{(0)}$ of the chain at time 0;

(II) Update from $\Theta^{(N)}$ to $\Theta^{(N+1)}$ as follows:

(a) Generate a proposal Θ^* with distribution $q(\Theta^*|\Theta^{(N)})$

(b) Compute the acceptance ratio

$$p = \min\left\{1, \frac{\pi(\Theta^*|X)\; q(\Theta^{(N)}|\Theta^*)}{\pi(\Theta^{(N)}|X)\; q(\Theta^*|\Theta^{(N)})}\right\}.$$

(c) Set

$$\Theta^{(N+1)} = \begin{cases} \Theta^* & \text{with probability } p, \\ \Theta^{(N)} & \text{with probability } 1-p. \end{cases}$$

instance, implying a matrix that diagonalizes the covariance function. This transformation appears to reduce the computational complexity and improve the mixing of the Markov chain. With $\Theta^* = \{\beta^*, \eta^*, Y^*\}$ and $\Theta^{(N)} = \{\beta^{(N)}, \eta^{(N)}, Y^{(N)}\}$, the proposal is defined as

$$q(\Theta^*|\Theta^{(N)}) = \mathcal{N}\Big[\Theta^*;\; \Theta^{(N)} + \frac{h^2}{2}\Sigma\nabla\log\{\pi(\Theta^{(N)}|X)\},\; h^2\Sigma\Big]$$

where $\mathcal{N}(a; b, c)$ denotes a multivariate Gaussian density with mean b and variance c evaluated at a and $h > 0$ is a scaling parameter. The parameter h is adjusted on some preliminary runs of the algorithm in order to achieve an average acceptance rate of 0.574 for the MCMC algorithm (Moller et al., 1998; Diggle et al., 2013). Finally, the component Σ is composed of Σ_Γ, Σ_β and Σ_η where all are approximations of the negative inverse of the Fisher information associated matrices (Diggle et al., 2013; Taylor et al., 2013, 2015). The construction of Σ_Γ, Σ_β and Σ_η are based on initial guesses at Γ, β and η, hence approximate values $\hat{\phi}$ and $\hat{\sigma}^2$ obtained by minimum contrast method used before will be useful. The initial values for the run of the MCMC algorithm can be declared as follows:

```
> INITS <- lgcpInits(etainit = log(c(sqrt(1.85), 617)), betainit =
    NULL)
```

The function `lgcpInits` from package `lgcp` helps the MCMC algorithm to calibrate the proposal density using the provisional estimates given if specified. The argument `etainit`, as its name implies, is used in order to declare initial values for η which must be always presented in the required form $\{\log(\sigma), \log(\phi)\}$. Then, the argument `betainit` is for parameter β. We followed Taylor et al. (2015) by specifying nothing. If no initial value is declared then β will be estimated from an overdispersed Poisson fit to the cell counts, ignoring spatial correlation. Finally, note that Y (or Γ) is not specified as the user can not do it, actually, a sensible value is chosen by the MCMC function.

Bayesian Prediction

In a Bayesian predictive framework, the distribution of a "new" observation X^* (in our case a new point pattern) is given by the *posterior predictive distribution* defined by

$$\pi(X^*|X) = \int \pi(X^*|\Theta)\pi(\Theta|X)\mathrm{d}\Theta, \tag{3.6}$$

where $X = \{x_1, \ldots, x_n\}$ denotes the data, $\pi(X^*|\Theta)$ is the data distribution and $\pi(\Theta|X)$ the posterior distribution. Note that if Θ were known, the predictive distribution for X^* would simply be $\pi(X^*|\Theta)$. As Θ is unknown, the predictive distribution for X^* is a mixture of the data distributions $\pi(X^*|\Theta)$ with respect to the posterior distribution $\pi(\Theta|X)$.

As the LGCP is computed on a grid, the goal here is to predict the number of events per cell of the computational grid. Considered an observed point pattern $X = \{x_1, \ldots, x_n\}$ within an observation window W, then the prediction of the number of events per cell i is given by the optimal predictor n_i^*, the posterior predictive expectation of $n(\mathbf{X}^* \cap c_i)$ given the point-process data X :

$$n_i^* = \mathbb{E}[n(\mathbf{X}^* \cap c_i)|X], \tag{3.7}$$

where $\mathbb{E}[\cdot|X]$ is computed with respect to the posterior predictive distribution given in Eq (3.6) and $\mathbf{X}^*$ denotes an independent replicate of the data with the same values Θ that produced the observed point pattern X (Cressie and Wikle, 2011, Chapter 2); a realization of $\mathbf{X}^*$ is X^*. Considering double conditioning with respect to Θ and X, we get

$$\begin{aligned} n_i^* &= \mathbb{E}[n(\mathbf{X}^* \cap c_i)|X] \\ &= \mathbb{E}\Big[\mathbb{E}\big[n(\mathbf{X}^* \cap c_i)|X, \Theta\big]\big|X\Big] \\ &= \mathbb{E}_{\pi(\Theta|X)}\big[\lambda(c_i|\Theta)\big], \end{aligned} \tag{3.8}$$

as $\mathbf{X}^*$ is independent of point-process data X and $n(\mathbf{X}^* \cap c_i)|\Theta \sim \text{Poisson}(\lambda(c_i|\Theta))$ with

$$\lambda(c_i|\Theta) = \exp\big(Z(c_i)^\top \beta + Y(c_i)\big). \tag{3.9}$$

In Eq (3.8), the notation $\mathbb{E}_{\pi(\Theta|X)}$ means expectation with respect to the measure $\pi(\Theta|X)$.

From a practical point of view, the prediction n_i^* is computed using Monte-Carlo simulation methods to approximate the expectation by the average of simulated-based quantities and the MCMC algorithm to generates samples from a Markov chain whose stationary distribution is the target distribution $\pi(\theta|X)$ (Gamerman and Lopes, 2006). In the Monte Carlo Markov Chain (MCMC) method, the expectation in Eq (3.8) may then be approximated by

$$n_i^* \simeq \frac{1}{N}\sum_{l=1}^{N} \lambda(c_i|\Theta^{(l)}), \quad i = 1, \ldots, C^2, \tag{3.10}$$

where $\Theta^{(l)}$ denotes the lth sample replicate of the Markov chain defined in Algorithm 1.

Indeed, the Markov chain $(\Theta^{(N)})_{N\geq 1}$ being ergodic, the *Ergodic theorem* implies

$$\frac{1}{N}\sum_{l=1}^{N}\lambda(c_i|\Theta^{l)}) \xrightarrow{a.s.} \mathbb{E}_{\pi(\Theta|X)}[\lambda(c_i|\Theta)], \quad \text{as } N\to+\infty. \tag{3.11}$$

For the sake of clarity, we give all software implementations in Section 3.5 which is devoted to LGCP modelling of the CAGB data.

3.3.5 Model performance assessment and risk measures

The goal in assessments for model fitting, which actually corresponds to posterior predictive diagnostics, is to determine whether the observed data are representative of the type of data expected under the assumed model. Our main goal is to predict as accurate as possible the number of accident per cell, so we suggest below several methods in order to evaluate the performances of competitor models.

Weighted Mean Squared Error and R-squared value

The first metric is based on the classical mean square error (MSE):

$$\text{MSE} = \frac{1}{C^2}\sum_{i=1}^{C^2}(n_i^* - n_i)^2,$$

where $n_i = n(X\cap c_i)$ denotes the number of events for the ith cell of centroid c_i of the $C\times C$ computational grid and n_i^* the predicted number of events for the ith cell. With the MSE-metric, equal weights $1/C^2$ of sum equal to one are given to all squared differences between the observed and the predicted values. In our situation, it may be advisable to weight differently the squared discrepancies between the observed and the predicted values. We suggest then the second metric which is a weighted MSE (wMSE) defined as follows

$$\text{wMSE} = \sum_{i=1}^{C^2} w_i(n_i^* - n_i)^2, \tag{3.12}$$

where

$$\omega_i = \frac{\tilde{\omega}_i}{\sum_{j=1}^{C^2}\tilde{\omega}_j}, \tag{3.13}$$

with

$$\tilde{\omega}_i = \frac{1}{\text{Var}(n(\mathbf{X}^*\cap c_i)|X)}, \quad i=1,\ldots,C^2, \tag{3.14}$$

and $\mathrm{Var}(n(\mathbf{X}^*\cap c_i)|X)$ is computed with respect to the posterior predictive distribution given in Eq (3.6). Considering again double conditioning with respect to Θ and X, we get

$$\begin{aligned}
&\mathrm{Var}\big[n(\mathbf{X}^*\cap c_i)\big|X\big]\\
&=\mathbb{E}\Big[\mathrm{Var}\big[n(\mathbf{X}^*\cap c_i)\big|X,\Theta\big]\Big|X\Big]+\mathrm{Var}\Big[\mathbb{E}\big[n(\mathbf{X}^*\cap c_i)\big|X,\Theta\big]\Big|X\Big]\\
&=\mathbb{E}_{\pi(\Theta|X)}\big[\lambda(c_i|\Theta)\big]+\mathrm{Var}_{\pi(\Theta|X)}\big[\lambda(c_i|\Theta)\big],
\end{aligned}\tag{3.15}$$

as $\mathbf{X}^*$ is independent of the point-process data X and $n(\mathbf{X}^*\cap c_i)|\Theta\sim\mathrm{Poisson}(\lambda(c_i|\Theta))$. Recall that $\lambda(c_i|\Theta)$ is defined in Eq (3.9).

From a practical point of view, the variance given in Eq (3.15) may be evaluated by MCMC approximation. In the previous section, devoted to Bayesian prediction, the expectation term from the right-side of Eq (3.15) is approximated by

$$\frac{1}{N}\sum_{l=1}^{N}\lambda(c_i|\Theta^{(l)})\underset{N\to+\infty}{\longrightarrow}\mathbb{E}_{\pi(\Theta|X)}\big[\lambda(c_i|\Theta)\big].$$

The variance term in the right-side of Eq 3.15 can be approximated in a similar way by

$$\frac{1}{N}\sum_{l=1}^{N}\left(\lambda(c_i|\Theta^{(l)})-\frac{1}{N}\sum_{k=1}^{N}\lambda(c_i|\Theta^{(k)})\right)^2\underset{N\to+\infty}{\longrightarrow}\mathrm{Var}_{\pi(\Theta|X)}\big[\lambda(c_i|\Theta\big].$$

So, in practice, we use the approximation

$$\mathrm{Var}(n(\mathbf{X}^*\cap c_i)|X)\simeq\frac{1}{N}\sum_{l=1}^{N}\lambda(c_i|\Theta^{(l)})+\frac{1}{N}\sum_{l=1}^{N}\left(\lambda(c_i|\Theta^{(l)})-\frac{1}{N}\sum_{k=1}^{N}\lambda(c_i|\Theta^{(k)})\right)^2.\tag{3.16}$$

In order to compare the performances of different models, we have also considered the R-squared value defined by

$$R^2=1-\frac{\sum_{i=1}^{C^2}(n_i^*-n_i)^2}{\sum_{i=1}^{C^2}(\bar{n}-n_i)^2},\tag{3.17}$$

where $\bar{n}=\frac{1}{C^2}\sum_{i=1}^{C^2}n_i$ is the mean number of events per cell of the computational grid.

Confusion matrices

An appropriate LGCP model would predict correctly the event occurrences when they were initially observed. In order to asses this model performance, we need to compute the

probability that the expected number of events per cell is equal to 0:

$$\begin{aligned}
\mathbb{P}\big(n(\mathbf{X}^* \cap c_i) = 0 \big| X\big) &= \mathbb{E}\big[\mathbb{I}_{\{n(\mathbf{X}^* \cap c_i)=0\}} | X\big] && (3.18)\\
&= \mathbb{E}\Big[\mathbb{E}\big[\mathbb{I}_{\{n(\mathbf{X}^* \cap c_i)=0\}} | X, \Theta\big]\Big| X\Big] && (3.19)\\
&= \mathbb{E}\Big[\mathbb{P}\big(n(\mathbf{X}^* \cap c_i) = 0 \big| X, \Theta\big)\Big| X\Big]\\
&= \mathbb{E}_{\pi(\Theta|X)}\big[\exp(-\lambda(c_i|\Theta))\big]\\
&\simeq \frac{1}{N}\sum_{l=1}^{N} \exp(-\lambda(c_i|\Theta^{(l)}))\big]. && (3.20)
\end{aligned}$$

These predicted probabilities can help us to classify the observations into two classes: if the predicted probability is "high" (namely higher than a threshold to be defined), then we classify the corresponding observation as belonging to the class **0** = "no events occurred", otherwise it belongs to the class **1** = "at least one event occurred". These predicted classes can then be compared to the observed values which have been also classified in these two **0**/**1** classes by applying the same rule. *Confusion matrices* are crossed tables of the observed and predicted classes as follows:

		Predicted class	
		0	**1**
Observed class	**0**	True negative (TN)	False positive (FP)
	1	False negative (FN)	True positive (TP)

Several metrics are based on the quantities contained in the confusion matrix (TN, FN, FP and TP) and are commonly used in machine learning in order to assess the model performance. The reader may find more details on the confusion matrix implementations and these metrics in Section 3.5.2.

Exceedance probabilities and relative risk

In order to visualize the spatial variation of the random intensity function

$$\Lambda(c_i) = \exp(Z(c_i)^\top \beta + Y(c_i)), \quad i = 1, \ldots, C^2, \tag{3.21}$$

we use, as in Taylor et al. (2013), the *exceedance probabilities* defined by

$$\begin{aligned}
\mathbb{P}\big(\Lambda(c_i) > t \big| X\big) &= \mathbb{E}\Big[\mathbb{P}\big(\Lambda(c_i) > t \big| X, \Theta\big)\Big| X\Big]\\
&= \mathbb{E}_{\pi(\Theta|X)}\big[\mathbb{I}_{\{\lambda(c_i|\Theta)>t\}}\big],
\end{aligned} \tag{3.22}$$

where $\mathbb{I}$ denotes the indicator function. Once again, the MCMC approximation is used and yields

$$\mathbb{P}\big(\Lambda(c_i) > t \big| X\big) \simeq \frac{1}{N}\sum_{l=1}^{N} \mathbb{I}_{\{\lambda(c_i|\Theta^{(l)})>t\}}.$$

The random intensity function takes into account both the spatial random effect through the latent random field Y and the covariate effects through the parameter β. One is often interested in studying the two effects separately and focus first on the spatial random effect. In the sequel, the Markov Chain $(\Theta^{(N)})_{N\geq 1}$ is noted

$$\Theta^{(N)} = (\beta^{(N)}, \eta^{(N)}, Y^{(N)}), \quad N \geq 1.$$

If one wants to study specifically the spatial random effects, one can focus on the *spatial relative risk*

$$\begin{aligned} \mathbb{P}\big(\exp(Y(c_i)) > t \big| X\big) &= \mathbb{E}\Big[\mathbb{P}\big(\exp(Y(c_i)) > t \big| X, \Theta\big)\Big| X\Big] \\ &= \mathbb{E}_{\pi(\Theta|X)}\left[\mathbb{I}_{\{\exp(Y(c_i))>t\}}\right] \\ &\simeq \frac{1}{N}\sum_{l=1}^{N} \mathbb{I}_{\{\exp(Y^{(l)}(c_i))>t\}} \end{aligned} \quad . \tag{3.23}$$

This quantity is interpreted as the residual spatial effect after integration of the spatial variability due to the spatial covariates.

Parameter a posteriori distribution and associate effects

Finally, it is interesting to visualize the a posteriori distribution of the parameters $\theta = (\beta, \eta)$.

The parameter β corresponds to the covariates effect. More precisely, the jth component β_j corresponds to the effect of the jth covariate Z_j and the exponential $\exp(\beta_j)$ is interepreted as a relative risk. Hence, $\beta_j < 0$ means that higher values of Z_j imply reduced risk of road crashes, while $\beta_j > 0$ corresponds to increased risk. The value $\beta_j = 0$ is interpreted as an absence of effect of the covariate Z_j. In the MCMC method, the a posteriori distribution of β_j is assessed through the empirical distribution $\{\beta_j^{(N)}\}_{N\geq 1}$ along the Markov chain. Typically, the histogram of the MCMC sample $\{\beta_j^{(l)} : 1 \leq l \leq N\}$ allows to visualize the a posteriori distribution of β_j. One can compare the a posteriori distribution to the a priori distribution and infer how much the observations are informative for the statistical inference. In particular, recall that the a priori distribution for β_j has been chosen as a very flat prior. A credible interval with level 95% for β_j is obtained by taking the empirical quantiles of order 2.5% and 97.5% of the MCMC sample. Of particular interest is to determine whether 0 lies inside or outside the credible interval for β_j, so as to determine the statistical significance of the effect of the covariate Z_j.

Similarly, the posterior distribution for $\eta = (\log \sigma, \log \phi)$ is visualized via an histogram of the MCMC sample $\{\eta^{(l)} : 1 \leq l \leq N\}$. This yields a Bayesian estimation of the parameters of the latent process Y.

3.4 Log-Gaussian Cox Processes pre-processing: variable selection

Our main goal is the analysis of georeferenced road crashes in CAGB and more exactly, to prevent and/or forecast road accidents. Taking into account the spatial characteristics of these accidents therefore seems essential. Hence, the statistical modelling of spatial dependence and potential risk factors is a major asset. More specifically regarding risk factors, the information will be incorporated through covariates in order to build a model according to a "explanatory-predictive tradeoff". Hence, it has to be decided which covariates will play a part in the analysis.

We will use a LGCP to model the occurence of the road crashes in CAGB and we will include spatially explanatory covariates in the varying intensity function $\lambda(\cdot)$ as explained in Section 3.3. However, algorithms used in order to run a LGCP are computationally challenging and time consuming. For example, Ramírez and Valencia (2021) fitted a "saturated" model (i.e. a model with all the available covariates) for which the total computation time was around two days by using a 2.2 GHz Intel core i7-4702MQ processor with 16Gb of RAM and they reduced the computation time to one day by using a model based only on the significant covariates chosen manually. We suggest in this work *automatic variable selection methods* and *variable selection criterion* in order to choose the most important covariates, in a small number, to be used next in a LGCP model allowing in this way an important reduction of the computation time. In order to do that, we need to do first several preliminary data processing operations such as normalization and interpolation of covariates on the chosen grid. Next, variable selection methods based on Poisson aggregation and variable importance criterion will be used in order to choose the most important covariates to be plugged-in the LGCP model. We consider all these preliminary treatments as *LGCP pre-processing*.

We consider the covariates as structured and associated to the computational grid of the LGCP as described in Chapter 2. The data are loaded as follows:

```
> cagb_covar_sf <- st_read('DATA/grid_cagb_sf.shp', quiet = TRUE)
> summary(cagb_covar_sf)[c(1, 6), ]

   prop18             prop65             health            school
Min.   :0.000000   Min.   :0.0000   Min.   : 0.0000   Min.   :0.00000
Max.   :0.200000   Max.   :0.7000   Max.   :80.0000   Max.   :7.00000

   college             shop             station             gasoline
Min.   :0.00000   Min.   :  0.0000 Min.   :0.00000   Min.   :0.00000
Max.   :5.0000    Max.   :353.0000 Max.   :3.00000   Max.   :2.00000

   leisure           intersection      radars      municipal_length
Min.   : 0.0000   Min.   : 0.0000  Min.   :0.000000  Min.   :    0.0
Max.   :14.0000   Max.   :23.0000  Max.   :1.000000  Max.   :13811.7
```

```
   national_length      geometry
Min.   :   0.0     POLYGON      :1362
Max.   :9761.8
```

For more convenience, covariate data are named `cagb_covar_sf` instead of `grid_cagb_sf` as it is named in Chapter 2. Note that the covariates are on different scales. In order to make each covariate equally important, a min-max normalization is applied as follows:

```
> minmax_norm <- function(x) return((x - min(x)) / (max(x)-min(x)))

> cagb_covar_sf <- cagb_covar_sf %>%
+              mutate(radars = fct_norm(radars),
+                     health = fct_norm(health),
+                     school = fct_norm(school),
+                     college = fct_norm(college),
+                     leisure = fct_norm(leisure),
+                     gasoline = fct_norm(gasoline),
+                     station = fct_norm(station),
+                     intersection = fct_norm(intersection),
+                     shop = fct_norm(shop),
+                     municipal_length = fct_norm(municipal_length),
+                     national_length = fct_norm(national_length))
```

The total number of cells included in the observation window is 1 362. Each cell will represent an observation. The covariates are $\{Z_1, Z_2, \ldots, Z_{13}\} = \{prop18, prop65, health, school, college, shop, station, gasoline, leisure, intersection, radars, municipal\_length, national\_length\}$. Then, the response variable to be explained will be the cell event counts of the computational grid, that is, the number of accidents per cell. A column *accidents* will be created and merged to the covariates in order to have a global data-frame. Remind that in Section 2.2, the `sf` object `cagb_sf` has been structured and corresponds to our road crash point pattern. The PIP interpolation method, as seen in Section 2.3.2, is used in order to aggregate road crash points `cagb_sf` located in polygon cells of `cagb_covar_sf` as follows:

```
> st_crs(cagb_sf) <- st_crs(cagb_covar_sf)
> cagb_sf$accidents <- rep(0, nrow(cagb_sf))
> cagb_covar_sp <- as(cagb_covar_sf, 'Spatial')
> cagb_sp <- as(cagb_sf, 'Spatial')

> cagb_sp <- aggregate(x = cagb_sp["accidents"],
+                      by = cagb_covar_sp,
+                      FUN = length)
> cagb_sf <- st_as_sf(cagb_sp)
> cagb_sf[is.na(cagb_sf$accidents), ]$accidents <- 0

> st_crs(cagb_sf) <- st_crs(cagb_covar_sf)
> selection_sf <- st_join(cagb_covar_sf, cagb_sf,
+                         join = st_covers)
> selection <- st_drop_geometry(selection_sf)
```

Similar command-lines as in Section 2.3.2 have been used. In order to use the function `aggregate`, `SpatialPolygonsDataFrame` objects `cagb_sp` and `cagb_covar_sp` have been created with `as(, 'Spatial')` as seen in the previous chapter. Then, the `sf` object `selection_sf` is created with `st_join` from package `sf` but this time with argument `join = st_covers`, another way to join geometries that gives the same results as methods employed in Section 2.3.2. Finally, the function `st_drop_geometry` is used in order to remove geometries from `selection_sf` and to create the `data.frame` object `selection`.

3.4.1 Poisson aggregation

We suggest in the following a variable selection method based on Poisson aggregation and variable importance criterion. Indeed, the Poisson distribution is the simplest distribution for count data. The average number of road crash occurrences per cell of the computational grid will be modelled in function of the available covariates. Actually, Poisson regression is quite similar to the LGCP as presented in Section 3.3 except that the Poisson regression completely ignores the spatial correlation.

Mathematically speaking, let N_i be the random variable denoting the number of accidents per i-th cell and n_i the observed value of N_i. Then, N_i follows a Poisson distribution of mean λ_i which we model by the following Poisson regression model based on covariate data $Z_1, \ldots, Z_{13}$ described in the previous section:

$$\begin{aligned} \log(\lambda_i) &= f(\mathbf{z}_i), \quad i = 1, \ldots, C^2 \\ f(\mathbf{z}) &= b_0 + \mathbf{z}^\top \mathbf{b} \end{aligned}$$

for $\mathbf{z}_i = (Z_{ij})_{j=1}^{13}$ and $\mathbf{b} = (b_j)_{j=1}^{13}, b_j \in \mathbb{R}$, $j = 1, \ldots, 13$. The predicted number of accidents per cell, $\hat{N}_i$, is given by:

$$\hat{N}_i = \hat{f}(\mathbf{z}_i) = \exp(\hat{b}_0 + \mathbf{z}_i^\top \hat{\mathbf{b}})$$

where $\hat{b}_j, j = 1, \ldots, 13$ are computed from sample data iteratively by using the re-weighted least squares criterion (Agresti, 2002, Chapter 4). In practice, count data exhibit sometimes larger variability than the mean, this phenomenon is known as overdispersion and is represented by a dispersion parameter Φ. Literally, this is defined as $\mathrm{Var}(N_i) = \Phi\mathbb{E}[N_i]$ with $\Phi > 1$. A way to overcome this situation is to use the *quasi-Poisson* model that introduces the dispersion parameter Φ into the model as suggested in Agresti (2002, Chapter 4).

In order to choose the most important variables, we suggest to proceed with a Poisson model aggregation instead of fitting a classical single Poisson regression model. More exactly, the Poisson model aggregation is an ensemble method that combines several Poisson regression models in order to get a better fit than a single Poisson regression model would do; it works in a similar way as the well-known *random forest algorithms* (Breiman, 2001) except that we consider a Poisson regression model now instead of a regression tree.

Globally speaking, the statistical learning method proposed is as follows. In this approach, we generate M different bootstrapped training data sets by randomly selecting with replacement n units among the initial n ones. Let B_m be these bootstrapped data sets,

$m = 1, \ldots, M$. We select also randomly and without replacement a set of p_0 covariates among the initial 13 covariates; let $A_m \subset \{1, \ldots, 13\}$ be the set of the labels of the chosen covariates. Let denote by $\mathbf{z}_i^* = (Z_{ij})_{j \in A_m}$ the p_0-dimensional vector of measures of the selected covariates on the i-th unit, for $i \in B_m$. We fit next a Poisson regression model on the data set $(n_i, \mathbf{z}_i^*), i \in B_m$ and determine $\hat{b}_0$, $\hat{\mathbf{b}} = (\hat{b}_j)_{j \in A_m}$. The predicted number of road crashes $\hat{N}^{B_m}$ at some point $\mathbf{z}$ is given by:

$$\begin{aligned} \hat{N}^{B_m}(\mathbf{z}) &= \hat{f}^{B_m}(\mathbf{z}), \\ &= \exp(\hat{b}_0 + \mathbf{z}^\top \hat{\mathbf{b}}). \end{aligned} \tag{3.24}$$

Finally, the m predicted values for N_i at $\mathbf{z}$ are averaged to obtain the final predicted value for N_i at $\mathbf{z}$:

$$\hat{N}(\mathbf{z}) = \frac{1}{M} \sum_{m=1}^{M} \hat{N}^{B_m}(\mathbf{z}). \tag{3.25}$$

Here our main purpose is the variable selection and we need a variable importance criterion. As suggested in Breiman (2001), we used the variable permutation criterion as *variable importance measure* (VIM). This process consists in, once the model is trained on the data set $(n_i, \mathbf{z}_i^*), i \in B_m$ to determine $\hat{f}^{B_m}(\cdot)$, shuffling the jth covariate values, $j = 1, \ldots, 13$ for the units not selected in the bootstrapped sample B_m. Then, predictions errors are evaluated on the initial sample and on the permuted sample. The permutation variable importance is defined to be the increase in the model error when this single covariate has been randomly permuted. This procedure breaks the relationship between the covariate and the response variable, thus the increase in the model error is indicative of how much the model depends on the variable.

More formally, the VIM is computed as follows. Dealing with a bootstrapped sample means that there are remaining observations not used: these observations form the *out-of-bag* (OOB) sample, named OOB^{B_m}. We fit a Poisson model on the data set $(n_i, \mathbf{z}_i^*), i \in B_m$ to determine $\hat{f}^{B_m}(\cdot)$; we compute the predictions $\hat{N}^{B_m}(\mathbf{z}_i) = \hat{f}^{B_m}(\mathbf{z}_i)$ for $i \in OOB^{B_m}$ and evaluate the prediction error $err_{OOB^{B_m}}$ defined as the average of the squared discrepancies between the predicted and the observed values:

$$err_{OOB^{B_m}} = \frac{1}{|OOB^{B_m}|} \sum_{i \in OOB^{B_m}} (\hat{N}^{B_m}(\mathbf{z}_i) - n_i)^2,$$

where $\mathbf{z}_i = (Z_{ij})_{j=1}^{13}$ for $i \in OOB^{B_m}$. We permute randomly the values of the jth covariate, $j \in \{1, \ldots, 13\}$, on the OOB^{B_m} data set and we compute the error in a similar way:

$$err_{OOB^{B_m,j}} = \frac{1}{|OOB^{B_m}|} \sum_{i \in OOB^{B_m}} (\hat{N}^{B_m}(\mathbf{z}_i^j) - n_i)^2,$$

where $\mathbf{z}_i^j = (Z_{i\ell}^{(j)})_{\ell=1}^{13}$ for which the values of the covariate Z_j have been permuted on the OOB^{B_m} set.

The VIM of the jth variable named $VIM^{B_m,j}$ corresponds simply to the difference be-

tween these two errors. The VIM is evaluated for each covariate on each training set B_m, $m = 1, \ldots, M$. Finally, after running the M iterations of the process, all VIMs are averaged in order to have one VIM value for each covariate named VIM^j as follows

$$VIM^j = \frac{1}{M} \sum_{m=1}^{M} VIM^{B_m,j}. \tag{3.26}$$

The higher the VIM value is, the more important the corresponding variable is. The corresponding variable selection algorithm is described in Algorithm 2. In order to evaluate the accuracy of this variable selection method, the whole dataset `selection` is splitted into a training set s_{train} = `selection_train` and a test set s_{test} = `selection_test`.

Algorithm 2 Poisson models aggregation

(I) For the mth bootstrapped training sample B_m selected from `selection_train`, $m = 1, \ldots, M$, do:

1. Select randomly without replacement $p_0 \leq 13$ covariates among the original covariates $\{Z_1, \ldots, Z_{13}\}$; let A_m be the set of the selected variables;
2. Fit a Poisson regression on the sample data $(n_i, \mathbf{z}_i^*), i \in B_m$ where n_i is the observed number of accidents on the mth sample and $\mathbf{z}_i^*$ is the vector of recorded values of the p_0 selected covariates for $i \in B_m$; determine $\hat{f}^{B_m}(\cdot)$;
3. Evaluate the out-of-bag error $err_{OOB^{B_m}}$ on the OOB^{B_m} observations

$$err_{OOB^{B_m}} = \frac{1}{|OOB^{B_m}|} \sum_{i \in OOB^{B_m}} (\hat{N}^{B_m}(\mathbf{z}_i) - n_i)^2$$

   where for $i \in OOB^{B_m}$, $\mathbf{z}_i$ is the vector of recorded values of the 13 covariates and $\hat{N}(\mathbf{z}_i) = \hat{f}^{B_m}(\mathbf{z}_i)$ are the prediction values.

4. For $j = 1, \ldots, 13$ do:
   (a) Shuffle randomly the values of the j-th variable on the OOB sample, let $\mathbf{z}_i^j = (Z_{i\ell}^{(j)})_{\ell=1}^{13}$ be the new vector of observations, for $i \in OOB^{B_m}$.
   (b) Evaluate the out-of-bag error $err_{OOB^{B_m,j}}$:

$$err_{OOB^{B_m,j}} = \frac{1}{|OOB^{B_m}|} \sum_{i \in OOB^{B_m}} (\hat{N}^{B_m}(\mathbf{z}_i^j) - n_i)^2,$$

   (c) Evaluate the variable selection criterion $VIM^{B_m,j}$

$$VIM^{B_m,j} = err_{OOB^{B_m,j}} - err_{OOB^{B_m}}$$

5. Compute the prediction of the number of accidents on the two whole subsets `selection_train` and `selection_test`, respectively denoted as $\hat{N}_{s_{train}}^{B_m}$ and $\hat{N}_{s_{test}}^{B_m}$

(II) Average all the predictions for the subsets `selection_train` and `selection_test`

$$\hat{N}_s = \frac{1}{M} \sum_{m=1}^{M} \hat{N}_s^{B_m}, \qquad s \text{ in } \{s_{train}, s_{test}\}$$

As mentioned before and described in Algorithm 2, the aim of the aggregation process is to average all the predictions computed in the M iterations. The mean of predictions $\hat{N}_{s_{train}}$ and $\hat{N}_{s_{test}}$, where $\hat{N}_s = \frac{1}{M} \sum_{m=1}^{M} \hat{N}_s^{B_m}$ for $s \in \{s_{train}, s_{test}\}$, are used then in order to evaluate model accuracy metrics. First, the Mean Squared Errors (MSE) of our process on `selection_train` and `selection_test` are defined as

$$MSE_s = \frac{1}{n_s} \sum_{i=1}^{n_s} (\hat{N}_{s,i} - N_{s,i})^2, \qquad s \text{ in } \{s_{train}, s_{test}\}. \tag{3.27}$$

Then, the R-squared values on `selection_train` and `selection_test` are defined as

$$R_s^2 = 1 - \frac{\sum_{i=1}^{n_s}(\hat{N}_{s,i} - N_{s,i})^2}{\sum_{i=1}^{n_s}(\bar{N}_s - N_{s,i})^2}, \qquad s \text{ in } \{s_{train}, s_{test}\} \tag{3.28}$$

where $\bar{N}_s$ is the mean value on the sample s.

The Poisson models aggregation is not already computed in any `R` package to the best of our knowledge. Hence, a function `rf_poisson` has been created in order to compute the whole process described in Algorithm 2. This function takes as arguments the training and the test sets, the response variable name, the covariate names, the number of covariates to be randomly selected and the number of models to be fitted. These arguments are respectively specified with `train`, `test`, `N`, `varZ`, `mtry` and `M` attributes. Note that in the following, for the sake of clarity, we tried to split the function according to each step of Algorithm 2, as the function is heavy to read. First, storing matrices are declared :

```
> rf_poisson <- function(train, test, N, varZ, mtry, M){

+   #Storing results
+   Nhat_train_Bm <- matrix(NA, nrow(train), M)
+   Nhat_test_Bm <- matrix(NA, nrow(test), M)
+   VIM <- matrix(NA, nrow = length(varZ), ncol = 1)
+   rownames(VIM) <- varZ
```

Objects `Nhat_train_Bm`, `Nhat_test_Bm` and `VIM` will contain respectively values of $\hat{N}_{s_{train}}^{B_m}$, $\hat{N}_{s_{test}}^{B_m}$ and VIM^j, $m = 1, \ldots, M$, $j = 1, \ldots, 13$.

The next steps then correspond to the aggregation. Before fitting a Poisson regression, a bootstrap sample `Bm`, a random selection of `mtry` covariates `Zstar` and the formula of the Poisson model `FORM` are setted as follows:

```
+   for(m in (1:M)){ # Start M Poisson models

+   #Bootstrap sample Bm
+   b <- sample(nrow(train), nrow(train), replace = TRUE)
+   Bm <- train[b, ]

+   #Covariate random sampling
+   v <- sample(length(varZ), mtry)
+   Zstar <- varZ[v]

+   #Writing formula
+   FORM <- as.formula(paste(N, "~ ",
+                       paste(Zstar, collapse = "+")))
```

Secondly, the mth Poisson model with `N` as the response variable and `Zstar` as covariates is computed as follows:

```
+   #Model fitting
+   fhat_Bm <- glm(FORM, data = Bm, family = "quasipoisson")
```

The function `glm` with argument `family = "quasipoisson"` is used in order to fit the model, named `fhat_Bm`. Note that this choice enables to focus on the overdispersion in Poisson models.

Then, the mth OOB error is evaluated as follows:

```
+   #OOB error with no permutation
+   oob_Bm <- train[-b, ]
+   err_oob_Bm <- predict(fhat_Bm,
+                         newdata = oob_Bm,
+                         type = "response")
+   err_oob_Bm <- mean((err_oob_Bm - oob_Bm[, N])^2)
```

The OOB observations OOB^{B_m} are the remaining observations not used in the sample B_m, hence we simple remove the observations taken for `Bm` in `train`. Then, the error $err_{OOB^{B_m}}$ is computed using the predictions and the observations, named `err_oob_Bm`.

Next, the variable importance criterion has to be setted. Hence, a `for` loop is also used in order to iteratively permute the jth covariate in the OOB observations as follows:

```
+   #OOB error with permutation
+   for(j in varZ){
+      ind_j <- which(colnames(oob_Bm) == j)
+      oob_Bm_j <- gdata::resample(oob_bm[, ind_j],
+                                  size = nrow(oob_bm))
+      oob_Bm_j <- oob_Bm %>%
+              dplyr::select(-j) %>%
+              cbind(oob_Bm_j)
+      colnames(oob_Bm_j) <- c(colnames(oob_Bm[, -ind_j]), j)
+      err_oob_Bm_j <- predict(fhat_Bm,
+                              newdata = oob_Bm_j,
+                              type = "response")
+      err_oob_Bm_j <- mean((err_oob_Bm_j - oob_Bm_j[, N])^2)

+      #VIM
+      VIM[j, ] <- sum(VIM[j, ], err_oob_Bm_j - err_oob_bm,
+                      na.rm = TRUE)
+   }
```

The jth covariate values of observations `oob_Bm` are permuted with the function `resample` from the package `gdata`. Then, the jth covariate non permuted is removed and replaced by the new permuted values in `oob_Bm`. The sample containing the vector of recorded values for which the jth covariate values have been permuted is called `oob_Bm_j`. Then, similarly as previous command-lines, the mth OOB permuted error $err_{OOB^{B_{m,j}}}$, named `err_oob_Bm_j`, is obtained. Finally, the VIM is evaluated by summing up the difference between `err_oob_Bm_j` and `err_oob_Bm` to the previous differences computed at $m-i$th iterations, $i = 1, \ldots, m-1$.

The final step for the mth iteration is to compute the predictions of the response variable `N` on the train and test sets, `train` and `test`, as follows:

```
+  #Forecasting
+ Nhat_train_Bm[, m] <- predict(fhat_Bm,
```

```
+                                  newdata = train,
+                                  type = "response")
+ Nhat_test_Bm[, m] <- predict(fhat_Bm,
+                               newdata = test,
+                               type = "response")

+  } #M fitted models
```

The `for` loop process for the aggregation is now done, `M` Poisson regressions have been fitted and associated predictions for `train` and `test` sets have been computed at each step. The final stage then is to average all the predictions and the VIM values in order to evaluate MSE and R-squared values. This is done as follows:

```
+  #Mean predictions and VIM
+  Nhat_train <- rowMeans(Nhat_train_Bm[, 1:M], na.rm = TRUE)
+  Nhat_test <- rowMeans(Nhat_test_Bm[, 1:M], na.rm = TRUE)
+  VIM <- VIM/M

+  #Aggregation assessment
+  R2_train <- 1 - Metrics::rse(train[, N], Xhat_train)
+  R2_test <- 1 - Metrics::rse(test[, N], Xhat_test)
+  MSE_train <- Metrics::mse(train[, N], Xhat_train)
+  MSE_test <- Metrics::mse(test[, N], Xhat_test)
```

The objects `Nhat_train` and `Nhat_test` correspond respectively to $\hat{N}_{s_{train}}$ and $\hat{N}_{s_{test}}$. R-squared values on `train` and `test` are evaluated using the function `rse` from the package `Metrics`. The MSE values are computed similarly using the function `mse`.

The function `rf_poisson` is fully computed, it remains to return the results as follows:

```
+  #Function results
+  rf <- list(R2_train, R2_test, MSE_train, MSE_test, VIM)
+  return(rf)
+
+ }
```

3.4.2 Implementations of Poisson aggregation and variable selection on CAGB data

Train/Test subsample sets

The first step is to split up the dataset `selection` into two subsets `selection_train` and `selection_test`. Remind that there are 396 accidents located in the 1 362 cells that overlay the CAGB window. This means that the split up of `selection` might create two subsets unequally shared according to the number of road crashes per cell. Moreover, this phenomenon known as *imbalanced class* could degrade the performance of the learning method further. Hence, it is a major asset to rectify this possible case. The method proposed is as follows. A temporary column `accidents_class` is created and is defined as the class **1** if one or more accident are located in the cell, class **0** otherwise. Then, the subsets `selection_train` and

selection_test are formed by following a kind of balance class. This process is computed as follows:

```
> selection <- selection %>%
+      mutate(accidents_class = case_when(accidents != 0 ~ 1))
> selection[is.na(selection$accidents_class), ]$accidents_class <- 0
> selection$accidents_class <- as.factor(selection$accidents_class)
> prop.table(table(selection$accidents_class))

        0         1
0.8883994 0.1116006
```

The column accidents_class is created with the function mutate from the package dplyr. As expected, we are in a case of imbalanced class. Indeed, the class 0 is represented around 88.84% and class 1 as 11.16%. Classical methods for balanced class usually form the subsets by following the same balance of the initial dataset. However in the case of imbalanced class, the purpose of main solution methods is to bring back to a case of balanced class if possible (Menardi and Torelli, 2018). Several methods are available for solving this problem. The method used here is Synthetic Minority Oversampling Technique (SMOTE) which oversamples the minority class. The reader may find more details about overcoming imbalance class methods, especially the SMOTE method, in Chawla et al. (2002). The proposed solution to our imbalance class problem is computed as follows:

```
> set.seed(123)
> index <- createDataPartition(selection$accidents_class, p = 0.75,
   list = FALSE)
> selection_train <- selection[index, ]
> selection_test <- selection[-index, ]

> selection_train <- DMwR::SMOTE(accidents_class~.,
+                                data.frame(selection_train),
+                                k = 2,
+                                perc.over = 300,
+                                perc.under = 200)
+ prop.table(table(selection_train$accidents_class))

> selection_train$accidents <- round(selection_train$accidents)

  0   1
0.6 0.4
```

First, the function createDataPartition from the package caret is used in an attempt to balance the class distributions within the splits as the initial dataset selection. Then, the SMOTE method is employed with the function SMOTE of package DMwR. This generates a new dataset, following the specified parameters (we will not go into details here), that addresses the class imbalance problem. Note that this method is setted only for selection_train as models would be trained on it. The subset selection_test will be only used for model assessments. Finally, accidents_class is removed:

```
> selection_train <- selection_train %>%
+          dplyr::select(-accidents_class)
> selection_test <- selection_test %>%
+          dplyr::select(-accidents_class)
```

Poisson aggregagtion on CAGB data

As the subsets `selection_train` and `selection_test` are now created properly, the aggregation can be fitted. First, storing objects and arguments of the function `rf_poisson` are initialized as follows:

```
> N <- "accidents"
> varZ <- c("radars", "health", "school", "college",
+           "leisure", "gasoline", "station", "municipal_length",
+           "national_length", "prop18", "prop65",
+           "intersection", "shop")
> mtry <- c(4, 6, 8, 10, 13)
> M <- 1000
> res <- list()
```

The parameter `mtry`, which corresponds to the number of covariates randomly selected to be fitted, is tuned with five possible values: 4, 6, 8, 10 and 13. Note that 13 corresponds to the total number of available covariates. The parameter `M`, which corresponds to the number of iterations, is setted to 1 000. Finally, a `list` object `res` is declared in order to contain the results of the function `rf_poisson`.

Then, the run of the aggregation is as follows:

```
> T1 <- Sys.time()

> i <- 1
> for(p0 in mtry){
+  res[[i]] <- rf_poisson(selection_train,
+                         selection_test,
+                         varA,
+                         varZ,
+                         mtry = p0,
+                         M))
+  i <- i+1
+ }

> T2 <- Sys.time()
> difftime(T2, T1)

 Time difference of 3.084634 mins
```

A `for` loop is used in order to iteratively run the aggregation with the *p0*th element of the vector `mtry` as number of covariates to be fitted in Poisson models. The computation time of these 5 000 models is 3 minutes and 4.8 seconds.

Results and interpretations

The results of this aggregation are compared with several fits (the corresponding `R` command-lines are not given). Indeed, linear regressions, Poisson regressions, penalized models (Lasso), Boosting and Random Forest models have been also fitted. We will only present the results of a saturated Poisson regression, a Poisson regression with covariates selected with Akaike Information Criterion (AIC) and a Random Forest model. Note that the Random Forest has been fitted using also the permutation criterion and the hyperparameters, such as for example the number of covariates to be randomly selected at each iteration, have been tuned by cross validation. The results are shown in Tab 3.1.

Tab 3.1: Mean squared error (MSE) and R-squared (R^2) values of various statistical models fitted on the road crash data.

		`selection_train`		`selection_test`	
Model fitted	**Number of covariates introduced**	**MSE**	**R-squared**	**MSE**	**R-squared**
Poisson regression	13	1.14	0.70	0.42	0.69
Poisson regression with AIC	7	1.19	0.68	0.45	0.67
Poisson regressions aggregation	4	1.39	0.63	0.58	0.57
	6	1.16	0.69	0.46	0.66
	8	1.09	0.71	0.42	0.69
	10	1.07	0.72	0.41	0.70
	13	1.18	0.69	0.42	0.69
Random Forest	8	0.28	0.92	0.51	0.62

Tab 3.1 gives the performance assessment of a saturated Poisson regression (which means that all the covariates have been included), a Poisson regression with the seven covariates selected with AIC criterion, the Poisson aggregations with different numbers of covariates to be randomly selected and fitted into the models (four, six, eight, ten and thirteen) and finally, a Random Forest model. The reader may find in Appendix B the Tab B.1 which is very similar to Tab 3.1, however the statistical methods used have been fitted on different samples of the dataset `selection`. First, the same results as Tab 3.1 are given. Then, the methods are compared according to the training samples which are simply a training set where the distribution of the column `accidents_class` of `selection` have been preserved (which means around 89% for the class **0** and 11% for the class **1**), referred as *Distribution preserved* in Tab B.1, and finally the whole dataset `selection`, which means that no training and test sets have been created, referred as *No split rule* in Tab B.1.

The best model, according to MSE and R-squared values on `selection_train`, is the Random Forest with a MSE and a R-squared respectively equal to 0.28 and 0.92. However, on `selection_test`, this model is ranked almost the last with a MSE and a R-squared respectively equal to 0.51 and 0.62. This specific case seems similar as overfitting cases. However, the Random Forest model has been fitted with cross validation in order to find the

best tuning parameters (we will not go into details here) that could improve values of MSE and R-squared on selection_test and the best model choice is this one (with for example eight covariates to be randomly selected at each iteration).

The saturated Poisson regression and the saturated aggregation (both with all available covariates) are only presented in order to be compared with the parsimonious models. It seems that the two best models, in terms of the explanatory-predictive tradeoff, are Poisson regressions aggregation with eight and ten covariates. Indeed, the MSE values on selection_train are respectively 1.09 and 1.07 whereas the saturated aggregation MSE value is 1.18. R-squared values are respectively 0.71 and 0.72 instead of 0.69 for the saturated model. However on selection_test they are quite equivalents. Indeed, MSE values of aggregations with 8 and 10 covariates are respectively 0.42 and 0.41 where the saturated aggregation MSE is equal to 0.42. Then, R-squared values are respectively 0.69 and 0.70 where the saturated one is 0.69.

Globally, the R-squared values of all these fits do not exceed 0.72 and 0.70 on respectively selection_train and selection_test. The methods employed did not give possibilities to improve the predictions. This means that eventually there is: the need of more informative covariates; another statistical model framework; no solutions to better anticipate the number of road crashes per cells. Some tentative answers will be given further with the LGCP fit.

Variable importance visualization

Back to the importance of the covariates, it could be interesting to produce variable importance plot similar to the classical plots usually given in practice. To do so, the VIM values returned by the function rf_poisson will be used. We proposed here to plot the ratio of the VIM values compared to the maximum one. Here is how to compute such a plot:

```
> vim_4 <- as.data.frame(res[[1]][[5]])
> vim_4 <- cbind(rownames(vim_4), vim_4)
> colnames(vim_4) <- c('covariate', 'vim')
> vim_4 <- vim_4 %>%
+  mutate(imp = vim*100/max(vim))

> ggplot(data = vim_4, aes(x = reorder(covariate, +imp), y = imp)) +
+  geom_bar(stat = "identity", fill = "steelblue") +
+  coord_flip() +
+  labs(x = "Covariate", y = "Importance")
```

The VIM values of the Poisson models aggregation where four covariates were randomly selected at each iteration have been chosen as an example for the command-lines, named vim_4. First, the VIM values are handled in order to create a dataframe with the name of the covariates, the VIM associated to them and the importance value as columns, named respectively covariate, vim and imp. The importance value, as mentioned before, is simply the ratio of the VIM compared to the maximum one. Then functions of the package ggplot2 are used in order to produce a barplot that plots the decreasing importance of each covariate. Command-lines above produced Fig 3.10. The four covariates that are the most important

are *municipal_length*, *shop*, *station* and *national_length*.

As the reader can see, it is possible sometimes to have negative values for permutation importances. In these cases, the predictions on the shuffled data happened to be more accurate than the real data. This happens when the covariate seems to be really not informative but random chance caused the predictions on shuffled data to be more accurate (Genuer and Poggi, 2019, Chapter 4).

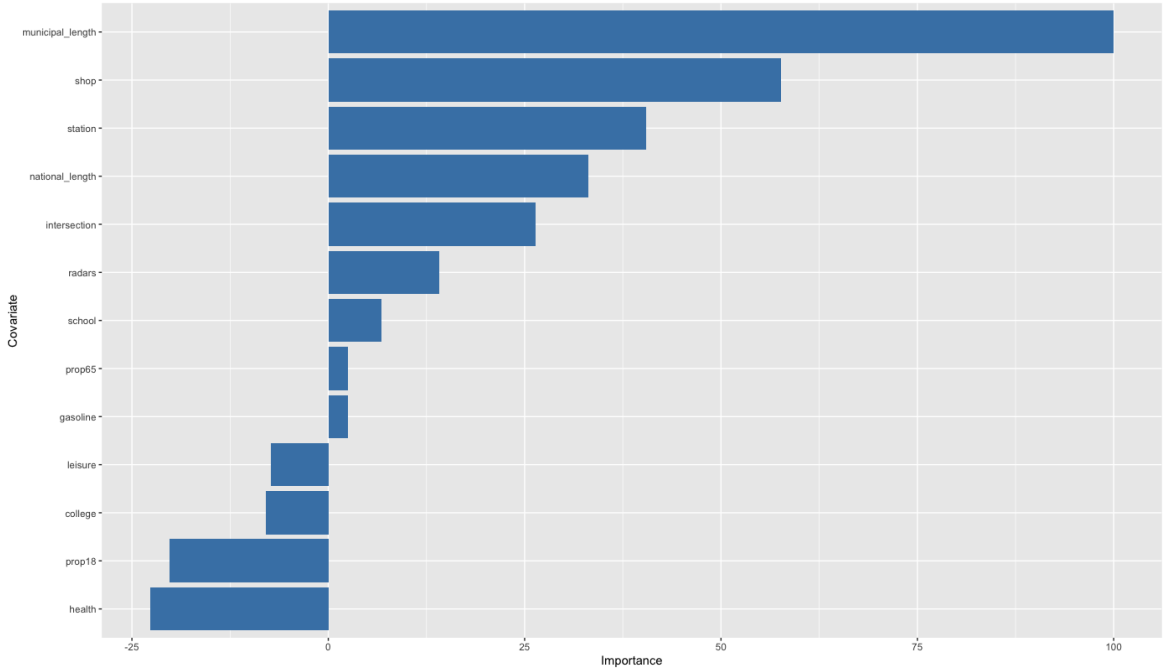

FIG 3.10: Variable importance plot for Poisson models aggregation with four covariates randomly included at each iteration.

The variable importance plots for all the remaining Poisson models aggregation and the Random Forest model have been produced using the same command-lines as previously. The results are given in APPENDIX B in FIG B.1, FIG B.2, FIG B.3, FIG B.4 and FIG B.5.

Recall that the definition of LGCP is written as

$$n(\mathbf{X} \cap c_i) \sim \text{Poisson}(\lambda(c_i))$$
$$\log(\lambda(c_i)) = Z(c_i)^\top \beta + Y(c_i).$$

Six subsets of covariates have been chosen and hence, give the following LGCP model formulas :

LGCP1 : $\mathbf{X} \sim$ *municipal_length* + *national_length* + *intersection* + *radars* + *prop65* + *prop18* + *station*

LGCP2 : $\mathbf{X} \sim$ *municipal_length* + *shop* + *station* + *national_length*

LGCP3 : $\mathbf{X} \sim$ *municipal_length* + *national_length* + *intersection* + *station* + *prop65* + *leisure*

LGCP4 : $\mathbf{X} \sim$ *municipal_length* + *national_length* + *intersection* + *shop* + *prop65* + *station* + *prop18* + *gasoline*

LGCP5 : $\mathbf{X} \sim$ *municipal_length* + *national_length* + *prop65* + *intersection* + *prop18* + *shop* + *school* + *gasoline* + *radars* + *station*

LGCP6 : $\mathbf{X} \sim$ *municipal_length* + *intersection* + *school* + *shop* + *health* + *national_length* + *station* + *prop18*

The covariates of model **LGCP1** correspond simply to the covariates selected with AIC on the Poisson regression. Then for each remaining model, the covariates have been chosen as being the most important covariates in the model relating respectively to the number introduced. Indeed for example, as four covariates were set to be randomly selected and fitted in the Poisson regressions of the aggregation then the most four important covariates have been chosen for **LGCP2**. The same process was adopted for **LGCP3**, **LGCP4**, **LGCP5** and **LGCP6**.

3.5 Log-Gaussian Cox Processes : fit, diverse assessments and result interpretations

3.5.1 Fit

In the previous section, six subsets of covariates have been selected by variable selection methods. The next step then is to fit LGCP models with these choices. The model **LGCP1** will be taken as example for the implementation in `R`. First, the model is defined as follows:

```
> FORM <- X ~ municipal_length + national_length + intersection +
   radars + prop65 + prop18 + station
```

Then the next step is to interpolate the covariates of this model onto the computational grid `polyolay`. Remind that in Section 2.3, an object `SpatialPolygons` has been extracted from `polyolay` and has been used in order to interpolate all the available covariates on. The purpose of interpolating the covariate values directly on a grid that is the same as the computational grid `polyolay` was to perform the interpolation methods ourself. Indeed, we were able to know exactly what was happening with our data and compare later the covariate values support that will be used in the fit of the LGCP models. Hence, the process of interpolation that will be used in the following will only give the same values as `cagb_covar_sf`, produced in Section 2.3.3. The classical steps for interpolating the covariates of a model on the computational grid as recommended in Taylor et al. (2015) using the package `polyolay` are the following:

```
> cagb_covar_sp <- as(cagb_covar_sf, 'Spatial')
> cagb_covar_sp@data <- guessinterp(cagb_covar_sp@data)

> cagb_covar_sp@data <- assigninterp(df = cagb_covar_sp@data,
+                           vars = colnames(cagb_covar_sp@data),
+                           value = "ArealWeightedSum")

radars interpolation via ArealWeightedMean
health interpolation via ArealWeightedMean
```

```
school interpolation via ArealWeightedMean
college interpolation via ArealWeightedMean
leisure interpolation via ArealWeightedMean
gasoline interpolation via ArealWeightedMean
station interpolation via ArealWeightedMean
intersection interpolation via ArealWeightedMean
shop interpolation via ArealWeightedMean
municipal_length interpolation via ArealWeightedMean
national_length interpolation via ArealWeightedMean
prop18 interpolation via ArealWeightedMean
prop65 interpolation via ArealWeightedMean
```

First, the function `guessinterp` from package `lgcp` assigns by default the interpolation method as the area-weighted mean for any numeric column. As its name implies, this function is used in order to guess provisional interpolation methods for the columns of the data frame. If the user wants to replace the assigned interpolation process, this can be done by using the function `assigninterp` and specifying the desired method in the `value` argument. There are three possible types of interpolation methods employed in the package `lgcp`, the reader may find more details in Taylor et al. (2015, Section A). The method used here is `ArealWeightedSum`. However, as our covariates have already been interpolated on the computational grid previously, any of the three interpolation methods would lead to the same result. Finally, the process of interpolation is as follows:

```
> Zmat <- getZmat(formula = FORM, data = cagb_ppp,
+                 regionalcovariates = cagb_covar_sp
+                 cellwidth = 650,
+                 ext = 2,
+                 overl = polyolay)

Using 'cellwidth' and 'ext' from overl
aggregating regional covariate information ...
loading polygon overlay ...
interpolating ...
Time Taken:  6.912689
```

The function `getZmat` constructs a design matrix for routine used in `lgcp`. Note that the arguments `cellwidth` and `ext` have been mentioned in Section 2.3.1. The component `cellwidth` has been explained above in Section 3.3.2. Then, the argument `ext` specifies the amount by which the computational grid will be extended, we will not go into details here, the reader may find comprehensive motivations in Davies and Bryant (2013).

The covariate data can be visualized by using the command-line `plot(Zmat)` which produces a sequence of plots, almost similar to the ones produced in Section 2.3.3 and in the Appendix A.

The final step from the LGCP routine is to run the MCMC algorithm. In order to do that, we need the spatial point pattern (contained in `cagb_ppp`), the covariates interpolated onto the computational grid and contained in the object `Zmat`, the model formula specified with `FORM`, the covariance function (contained in the object `cf`), the initial values of β and η

specified in INITS and the priors specified in priors. Note that in Section 3.3.4, the priors have been specified for a model of 7 covariates, just as **LGCP1**. The fit is computed as follows:

```
> nsimul <- 1500000
> bnin <- 250000
> rt <- 1250

> BASEDR <- "RESULTS/res_LGCP1/"

> lgcp1 <- lgcpPredictSpatialPlusPars(
+      formula = FORM,
+      sd = cagb_ppp,
+      Zmat = Zmat,
+      model.priors = priors,
+      model.inits = INITS,
+      spatial.covmodel = cf,
+      mcmc.control = mcmcpars(mala.length = nsimul,
+                              burnin = bnin,
+                              retain = rt,
+                              adaptivescheme =
+                                  andrieuthomsh(inith = 1,
+                                  alpha = 0.5, C = 1,
+                                  targetacceptance = 0.574)),
+      output.control = setoutput(gridfunction =
+                        dump2dir(dirname = BASEDR, forceSave = TRUE)),
+      cellwidth = CELLWIDTH, ext = EXT)

> save(list = ls(), file = file.path(BASEDR, "lgcp1.RData"))
```

The above command-lines run the MALA chain for 1 500 000 iterations, with an initial burn-in of 250 000 iterations, followed by 1 250 000 iterations of which every 1 250th sample is saved. This leads to a sample $\{(\Theta^{(N)}\}_{N\geq 1} = \{(\beta^{(N)}, \eta^{(N)}, Y^{(N)})\}_{N\geq 1}$ with $N = 1000$. The argument adaptivescheme is specific to the MALA proposal, the reader is refered to Taylor et al. (2013) for an explanation of these options. The results of lgcp1 are saved in an object by specifying the argument output.control and with the function save used in the last command-line.

The models **LGCP2**, **LGCP3**, **LGCP4**, **LGCP5** and **LGCP6** have been fitted by using similar command-lines, respectively named lgcp2, lgcp3, lgcp4, lgcp5 and lgcp6. However, note that the objects FORM, Zmat and priors are specific to the models themselves.

3.5.2 Fits assessment

The models **LGCP2**, **LGCP3**, **LGCP4**, **LGCP5** and **LGCP6** have been fitted in the previous section. The purpose now is to compare these models in order to chose the best one. The criteria that will be used further in order to choose the best model are the wMSE and the R-squared values, then the confusion matrices as introduced in Section 3.3.5. The

best model will then be used to map the riskiest zones of the CAGB and to identify the riskiest factors as introduced in Section 3.3.5.

Weighted mean squared errors and R-squared values

The goal now is to compute the weighted mean square error, wMSE, defined in Eq (3.13) for the six models. Recall that the wMSE is defined as follows

$$\text{wMSE} = \sum_{i=1}^{C^2} \frac{\tilde{\omega}_i}{\sum_{j=1}^{C^2} \tilde{\omega}_j} (n_i^* - n_i)^2,$$

where

$$\tilde{\omega}_i = \frac{1}{\text{Var}(n(\mathbf{X}^* \cap c_i)|X)}, \quad i = 1, \ldots, C^2 \tag{3.29}$$

and $\text{Var}(n(\mathbf{X}^* \cap c_i)|X)$ will be approximated by Monte-Carlo and MCMC simulations as given in Eq (3.16). In order to do that, we need to compute first n_i^*, the expected number of events per cell of the grid, defined in Section 3.3.4.

An approximation of n_i^*, for the model **LGCP1** can be computed as follows:

```
> lgcp1_pred <- lgcp:::expectation.
   lgcpPredictSpatialOnlyPlusParameters(lgcp1, numCases)[[1]]

|============================================================| 100%
```

The function `expectation` from the package `lgcp` computes joint expectations under the model with all parameters. The function `numCases`, in argument of `expectation`, gives the expected number of events in each cell of the computational grid. The object `lgcp1_pred` is a 64×64 matrix. The observed and the predicted values can be compared graphically. In order to do that, the initial road crash values have to be interpolated on the computational grid in order to obtain a matrix 64×64 just as `lgcp1_pred`. Indeed, remind that `selection_sf` corresponds only to the cells including in the window `owin_cagb_sf`. Hence, a `sf` object `obs_sf` has been created with the `SpatialPolygons` object extracted from `polyolay` on which the road crash point pattern `cagb_sf` has been interpolated similarly as previous methods used (command-lines not shown as the process was similar to many other processes in the previous and current chapter). Then, as `lgcp1_pred` is simply a matrix, the spatial marker that could have informed us of the different geometric structures of these cells is lost. Hence, one way to assign back exactly the same spatial structures as `obs_sf` for example is to create a `RasterLayer` object. This is done as follows:

```
> obs_sp <- as(obs_sf, 'Spatial')
> r <- raster(ncol = 64, nrow = 64)
> crs(r) <- crs(obs_sp)
> extent(obs_sp)
> extent(r) <- extent(obs_sp)
> obs_ras <- rasterize(obs_sp, r, field = "accidents")

> lgcp1_pred_ras <- raster(lgcp1_pred)
```

```
> extent(lgcp1_pred_ras) <- extent(obs_ras)

> plot(obs_ras, main = "")
> plot(lgcp1_pred_ras, main = "")

class       : Extent
xmin        : 906135.5
xmax        : 947735.5
ymin        : 6664498
ymax        : 6706098
```

First, a `RasterLayer` object `obs_ras` is created from `obs_sf`. Indeed, the `sf` object `obs_sf` is our reference spatial structure from which we want to assign the same geometries to `lgcp1_pred`. To do so, first a `SpatialPolygonsDataFrame` object `obs_sp` is created as it is necessary later. Then, a 64×64 `RasterLayer` object `r` is created using the function `raster` from the package `raster`. The spatial extent of this `RasterLayer` object is specified by using the extent of `obs_sp`. An `extent` object is simply giving the spatial boundaries of the object in argument. For example, the lowest latitude value of `obs_sp` is 6 664 498. Then, the function `rasterize` from the package `raster` allows to transfer the values containing in the column `accidents` of the object `obs_sp` to the raster cells `r`. Finally, the `RasterLayer` object `lgcp1_pred_ras` is created with the function `raster` and its extent is specified using the extent of `obs_ras`. The two `RasterLayer` objects can be then plotted using the function `plot`. Note that it was not necessary to set `obs_ras` in order to create `lgcp1_pred_ras`. Indeed to specify its extent, we could have used the following command-line : `extent(polyolay$fftpoly)`, as it is the same. The creation of `obs_ras` is used as manipulation examples of `raster` package functions but more particularly for the sake of comparison between the observed and fitted values.

Fig 3.11 shows the observed number of road crashes per cell of the computational grid. Fig 3.12 represents the predicted values from `lgcp1`, that are, the expected number of road crashes per cell of the computational grid. The grey background of the plots corresponds to the cells where no accidents happened. Graphically speaking, these plots are similar. The same structure as Fig 3.11 can be recognized in Fig 3.12. Moreover, note that the five green cells in the middle of Fig 3.11, that represent globally cells where between eight and fourteen accidents happened, seem to have been predicted properly as it can be seen in Fig 3.12.

Now, that the expected number of events per cell of the grid n_i^* has been approximated, we can approximate the variance formula as given in Eq (3.16). For the sake of clarity, we make a copy of `lgcp1_pred` named `EX`. The function that computes the approximation of the variance given in Eq (3.16) is as follows :

```
> VARX <- function(Y, beta, eta, Z, otherargs){
+  ca <- diff(otherargs$mcens[1:2]) * diff(otherargs$ncens[1:2])
+  X <- ca * otherargs$poisson.offset[1:otherargs$M, 1:otherargs$N]
+   * exp(matrix(Z % * % t(beta), otherargs$M * otherargs$ext,
+  otherargs$N * otherargs$ext)[1:otherargs$M, 1:otherargs$N] + Y)
```

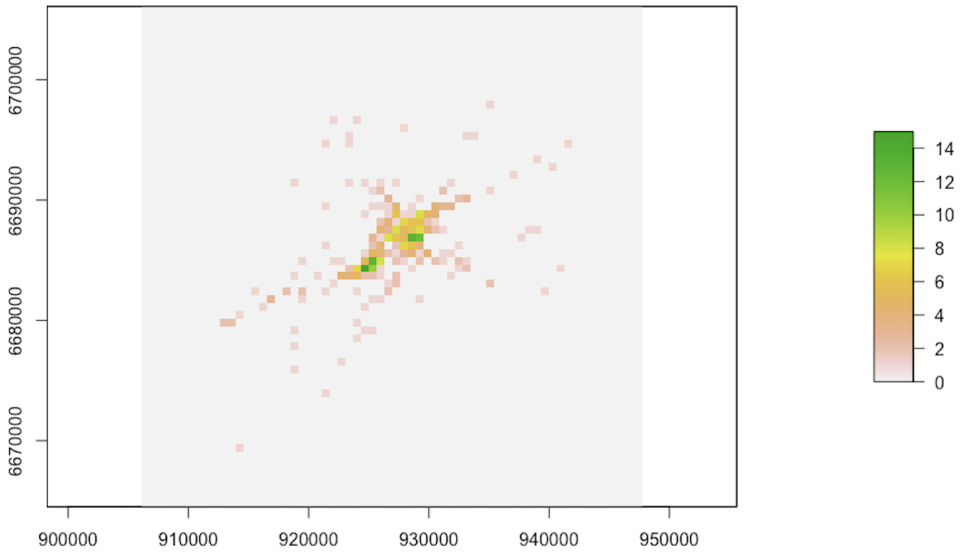

FIG 3.11: Plots of road crash observed values on a 64 × 64 grid.

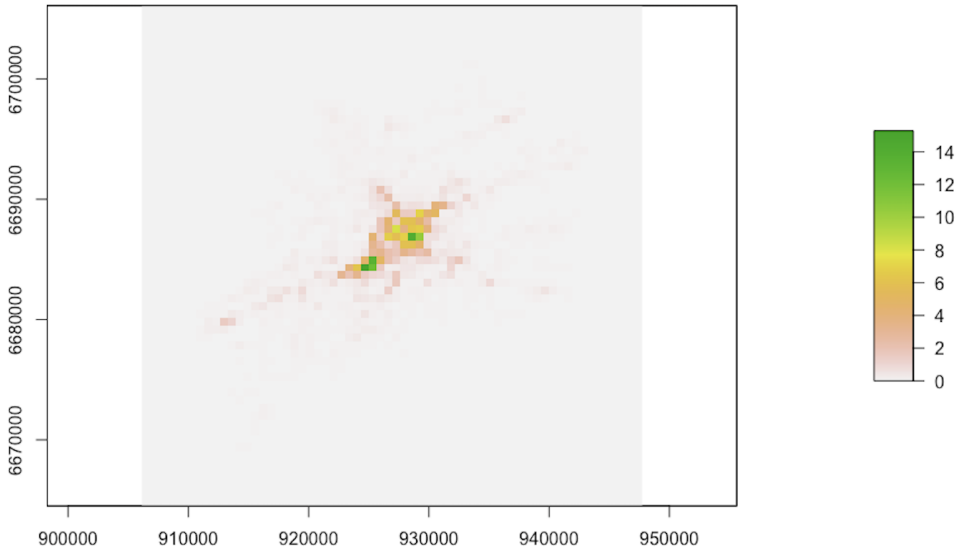

FIG 3.12: Plots of road crash fitted values from `lgcp1` on a 64 × 64 grid

```
+  return(EX + (X-EX)^2)}

> lgcp1_VARX <- lgcp:::expectation.
   lgcpPredictSpatialOnlyPlusParameters(lgcp1, VARX)[[1]]

|======================================================================| 100%
```

Note that this function is not so obvious. We will not go into details here. The most important command-line is the option `return` in which the EQ (3.16) is clearly written. Then,

the function `expectation` used the function `VARX` to compute the variance from `lgcp1`. Note also that `EX` should have been incorporated as an argument in the function `VARX` but we did not proceed with this direction as it was difficult to implement it. Finally, the wMSE for the model **LGCP1** is computed as follows:

```
> w1_tilde <- 1/lgcp1_VARX
> w1 <- w1_tilde/(sum(w1_tilde, na.rm = TRUE))
> lgcp1_wMSE <- sum(w1*(as.matrix(obs_ras) -
+                         as.matrix(lgcp1_pred_ras))^2,
+                 na.rm = TRUE)
> lgcp1_wMSE

[1] 0.01241512
```

The vector of weights $\tilde{\omega}$ is computed as defined in Eq (3.14). The second command-line allows to set `NA` values outside the window area. This implies that only the cells inside the window `owin_cagb` are considered meaning that the wMSE is written as

$$\text{wMSE} = \sum_{i=1}^{C_W} \frac{\tilde{\omega}_i}{\sum_{j=1}^{C_W} \tilde{\omega}_j} (n_i^* - n_i)^2,$$

where C_W is the number of cells included in the study window W, equal to 1 362. The value of the wMSE computed from `lgcp1` is 1.24e-02. Finally, the vector of weights $\omega_i = \tilde{\omega}_i / \sum_{j=1}^{C_W} \tilde{\omega}_j, i = 1, \ldots, C_W$ is computed as defined in Eq (3.13) and can be also visualized with the following command-lines:

```
> w1_ras <- raster(w1)
> extent(w1_ras) <- extent(owin_cagb)
> plot(w1_ras, main = "")
```

The Fig 3.13 shows that the most important weights are essentially close to the boundaries of the study window. However, note that the weights are in a close range.

The same process for computing `lgcp1_wMSE` has been adopted for the remaining models **LGCP2**, **LGCP3**, **LGCP4**, **LGCP5** and **LGCP6**. During this process, note that `RasterLayer` objects `lgcp2_pred_ras`, `lgcp3_pred_ras`, `lgcp4_pred_ras`, `lgcp5_pred_ras` and `lgcp6_pred_ras` have been created just as `lgcp1_pred_ras`. The results are shown in Tab 3.2. The best model to be chosen would be `lgcp2` as it has the lowest wMSE value, which is 1.12e-02. However, the models seem to be quite equivalents regarding to their wMSE values.

To assess the performance of each model, the second proposed metric is the R-squared value as defined in Eq (3.17) and recalled below:

$$R^2 = 1 - \frac{\sum_{i=1}^{C^2} (n_i^* - n_i)^2}{\sum_{i=1}^{C^2} (\bar{n} - n_i)^2}.$$

The R-squared values of each model are given in Tab 3.2.

The R-squared for `lgcp1` is computed as follows:

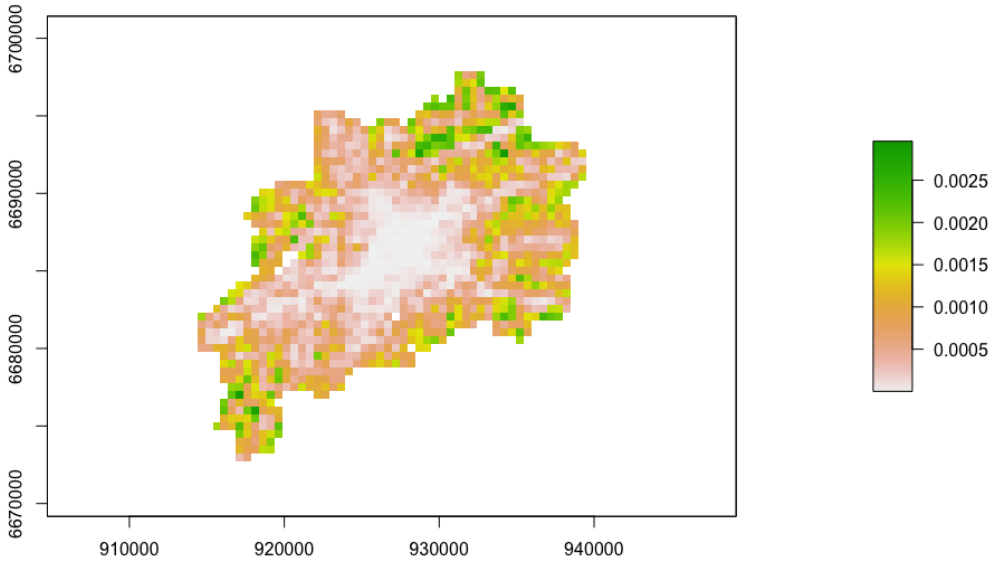

FIG 3.13: Plot of weights used in the wMSE for model `lgcp1`.

TAB 3.2: Metric results of the LGCP models fitted.

	Model					
Metric	**LGCP1**	**LGCP2**	**LGCP3**	**LGCP4**	**LGCP5**	**LGCP6**
wMSE	1.24e-02	1.12e-02	1.16e-02	1.21e-02	1.20e-02	1.18e-02
R-squared (%)	91.07	91.08	91.13	90.99	91.39	91.42

```
> lgcp1_pred_sp <- rasterToPolygons(lgcp1_pred_ras)
> lgcp1_pred_sf <- st_as_sf(lgcp1_pred_sp)
> st_crs(lgcp1_pred_sf) <- st_crs(obs_sf)
> lgcp1_pred_sf <- st_intersection(lgcp1_pred_sf, owin_cagb)
> lgcp1_pred_sf <- st_join(lgcp1_pred_sf, obs_sf, join = st_covers)
> colnames(lgcp1_pred_sf) <- c("pred", "obs", "geometry")

> lgcp1_R2 <- 1-Metrics::rse(lgcp1_pred_sf$obs, lgcp1_pred_sf$pred)
> lgcp1_R2

[1] 0.9107213
```

First, from the `RasterLayer` object `lgcp1_pred_ras`, a `SpatialPolygonsDataFrame` named `lgcp1_pred_sp` has been created in order to be next converted into a `sf` object `lgcp1_pred_sf`. To do so, the explicit function `rasterToPolygons` from the package `raster` has been used. The goal of this conversion was to use the function `st_intersection` from package `sf` in order to consider only the cells inside the window `owin_cagb`, which definitely

means that the R-squared value is written as

$$R^2 = 1 - \frac{\sum_{i=1}^{C_W}(n_i^* - n_i)^2}{\sum_{i=1}^{C_W}(\bar{n} - n_i)^2},$$

where C_W is the number of cells included in the study window W, equal to 1 362. Then, the observed values containing in `obs_sf` are merged to the fitted values of `lgcp1_pred_sf` with the function `st_join`. The column names are modified for the sake of clarity. Finally, the R-squared metric is computed with the function `rse` of the package `Metrics`. The R-squared value of the final model chosen `lgcp1` is 91.07e-02 which literally enables us to say that 91.07% of the predicted values are correlated to the observed values.

The same process used for computing `lgcp1_R2` has been also adopted for the remaining models **LGCP2**, **LGCP3**, **LGCP4**, **LGCP5** and **LGCP6** and the results are given in Tab 3.2. The best model to chose, regarding to the R-squared values, would be **LGCP6**. However, note that the values are in a close range.

Take a look back at the variable selection results given in Section 3.4.2. The maximum R-squared value on the test set was equal to 0.70. This is correct but in practice, the purpose is to have higher R-squared values. This suggested that, possibly, the covariate information was not good enough to explain and/or predict well the road crashes. Hence, this expressed the need to have maybe more informative covariates or maybe, the remaining of explanation percentage is relating to a noise. The Gaussian process in the proposed method LGCP, as it is a latent field, acts actually as a noise, that means, the hazard that can not be explained by the covariates. If we compare the aggregation of Poisson models and the LGCP, the LGCP seems to be more appropriate as the R-squared values are higher.

To conclude, the model that has been chosen and considered as the best one is **LGCP3**. Indeed **LGCP3** has almost the lowest wMSE, really close to **LGCP2**, but has a higher R-squared value. The model **LGCP3** has also shown the best results in the next section.

Confusion matrices

The goal now is to compute the probability that the expected number of accidents per cell is equal to 0 in order to assess if the potential best model `lgcp3` predicts correctly the accident occurrences if they were initially observed, as described in Section 3.3.5. Unfortunately it was difficult to compute the predictive distribution given in Eq (3.20) with the functions of the package `lgcp`. An alternative that can give similar results and possible to be computed is to generate a random Poisson variable with mean the intensity estimate from `lgcp3`. This can be computed as follows:

```
> pois_proba0 <- function(Y, beta, eta, Z, otherargs){
+  ca <- diff(otherargs$mcens[1:2]) * diff(otherargs$ncens[1:2])
+  lambda <- ca *
+  otherargs$poisson.offset[1:otherargs$M, 1:otherargs$N]
+  *exp(matrix(Z % * % t(beta), otherargs$M * otherargs$ext,
+  otherargs$N * otherargs$ext)[1:otherargs$M, 1:otherargs$N] + Y)
```

```
+   x <- dpois(0, lambda)
+   return(x)
+ }

> lgcp3_proba0 <- lgcp:::expectation.
    lgcpPredictSpatialOnlyPlusParameters(lgcp3, pois_proba0)[[1]]

|============================================================| 100%
```

As seen before with the function `VARX`, the creation of such a function as `pois_proba0` is not so obvious. The most important command-line is `x <- dpois(0, lambda)` which returns the value of the Poisson probability density function. Especially, it will return the probability that the number of events is equal to 0. Note that the Poisson distribution has mean `lambda`, the intensity estimated from `lgcp3`. The object `lgcp3_proba0` is a 64×64 matrix containing the probability that no accident happened in the cells of the computational grid (from a random variable simulated with `lgcp2` parameter estimates). Note that other possible simulations could have been generated instead of the ones resulting from the function `dpois`. For example, it would have been interesting to simulate an inhomogeneous Poisson point process on the study window `owin_cagb` with our intensity estimate coming from `lgcp3` as parameter. Instead of probabilities, we would have had counts.

These predicted probabilities are used now to classify the observations and to determine the confusion matrix as defined in Section 3.3.5 and recalled below:

		Predicted class	
		0	**1**
Observed class	**0**	True negative (TN)	False positive (FP)
	1	False negative (FN)	True positive (TP)

The observed values and the probabilities in `lgcp3_proba0` will be converted into two classes: **0** for "no accident happened" and **1** for "at least one accident happened". The transformation of observed values into this binary class is simply realized by converting the count values different from 0 to the class **1**. However, for classifying the predicted probability values obtained with `lgcp3_proba0` into two classes, a decision rule needs to be defined. An observation belongs to the class **0** if its predicted probability is higher than a threshold (to be set in practice), otherwise it belongs to the class **1**. True negative and true positive values are respectively the count of predicted classes **0** and classes **1** when it was observed. Then, the false negative and false positive values are respectively the count of predicted classes **0** and classes **1** when it was not the case. In practice, the choice of an ideal threshold may be based on subjective criteria (for example, one can take the threshold equal to 0.5) or by optimization criteria based on quantities computed from the confusion matrix such as the *Accuracy* or the *Sensitivity* as defined below.

Several metrics can be derived from the confusion matrix, for example :

- *Accuracy*: $(TP+TN)/(TP+FP+FN+TN)$, which is the ratio between the number of correct predictions and the total number of predictions. This metric is useful when false negatives and false positives have similar costs.

- *Sensitivity* (also called *recall*) : $TP/(TP+FN)$, which is the ratio between the number of true positives and total number of observed positives. This metric is useful when identifying the positives is crucial and hence, the occurrence of false negatives is not wished.

- *Specificity*: $TN/(TN+FP)$, which is the ratio between the number of true negatives and the total number of observed negatives. This metric is useful when the goal is to cover all true negatives and hence, the occurrence of false positives is not wished.

- *Precision*: $TP/(TP+FP)$, which is the ratio between the number of true positives and the total number of predicted positives. This metric is useful when the occurrence of false positives is not wished.

- *F1-Score*: $2(\text{Sensitivity} \cdot \text{Precision})/(\text{Sensitivity} + \text{Precision})$, which is the harmonic mean of the precision and sensitivity. This metric is useful in a case of imbalanced class distribution, when the cost of false positives and false negatives differs.

A way to assess the performance for the classification problems at various threshold settings is the ROC curve (*Receiver Operating Characteristics*) from which the metric *Area Under The Curve* (AUC) is derived. The ROC curve plots the specificity values on the x-axis and the sensitivity values on the y-axis with respect to different threshold values. As to improve the sensitivity you have to decrease the specificity, or vice versa, an ideal model would be the one which allows to improve one of this metric without decreasing the other. The AUC value allows to rate this tradeoff between sensitivity and specificity (James et al., 2013). Hence, it is also interesting to compute this metric.

We propose to compute the confusion matrix metrics for several threshold values and the AUC result for the model `lgcp3`. A way to implement it this is as follows:

```
> #Storing results
> threshold <- seq(0.01, 0.99, by=0.01)
> sensi <- rep(NA, length(threshold))
> speci <- rep(NA, length(threshold))
> acc <- rep(NA, length(threshold))
> prec <- rep(NA, length(threshold))
> f1_score <- rep(NA, length(threshold))

> k <- 1
> for(t in threshold){

+  #Data wrangling step
+  tmp <- raster(lgcp3_proba0)
+  extent(tmp) <- extent(obs_ras)
+  tmp <- rasterToPolygons(tmp)
+  tmp <- st_as_sf(tmp)
+  st_crs(tmp) <- st_crs(obs_sf)
+  tmp <- st_intersection(tmp, owin_cagb)
+  tmp <- st_join(tmp, obs_sf, join = st_covers)
```

```
+   colnames(tmp) <- c("proba", "obs", "geometry")

+   #Observed values transformation to binary class
+   tmp[tmp$obs >= 1, ]$obs <- 1
+   tmp$obs <- as.factor(tmp$obs)

+   #Fitted values transformation to binary class
+   tmp$pred_bin <- tmp$proba
+   tmp[tmp$pred_bin > t, ]$pred_bin <- 0
+   tmp[tmp$pred_bin != 0, ]$pred_bin <- 1
+   tmp$pred_bin <- as.factor(tmp$pred_bin)

+   #Confusion matrix setting
+   tab <- table(tmp$pred_bin, tmp$obs)

+   #Confusion matrix metrics computation
+   sensi[k] <- tab[2, 2]/(tab[1, 2] + tab[2, 2])
+   speci[k] <- tab[1, 1]/(tab[1, 1] + tab[2, 1])
+   acc[k] <- (tab[1, 1] + tab[2, 2])/ sum(tab)
+   prec[k] <- tab[2, 2]/(tab[2, 2]+tab[2, 1])
+   f1_score[k] <- 2*(prec[k]*sensi[k])/(prec[k]+sensi[k])

+   k <- k+1
+ }

+ #AUC computation
> pROC::roc(response = tmp$obs, predictor = tmp$proba)$auc

Setting levels: control = 0, case = 1
Setting direction: controls > cases
Area under the curve: 0.9503
```

First, a vector `threshold` is set containing various threshold values to be used. Then, storing objects `sensi`, `speci`, `acc`, `prec` and `f1_score` are declared for respectively each confusion matrix metrics sensitivity, specificity, accuracy, precision and F1-score. A `for` loop enable to iteratively run the creation of a `sf` object named `tmp` (as temporary) that contains the observed values of `obs_sf` and the probabilities that no accident happened of `lgcp3_proba0`, the transformation of these probabilities to the binary class regarding to the tth element of `threshold`, and finally the computation of the corresponding confusion matrix and its associated metrics. Finally, the AUC is computed using the function `roc` of the package `pROC`. Note that the command-lines used above contain redundant steps. Indeed, it would have been more optimal not to compute the class of the observed values at each iteration of the `for` loop for example.

The first metric AUC value is 95.03e-02 which tells that how much the model `lgcp3` enables to distinguish between classes. The higher the AUC value is, the better model is for predicting the classes **0** and **1**. The final model `lgcp3` is the best model regarding to the

AUC values of every model (Tab 3.3). However, note that these values are in a close range.

Now, the final values of sensitivity, specificity, accuracy, precision and F1-score depend on the research goal. As seen before, the metrics of the confusion matrix have different interpretations. In our case, the most important one is the sensitivity. Indeed, identify truly where accidents happened is crucial, we do not want to miss a cell where initially accidents happened. There would rather be some extra false positives, which means cells where accidents are supposed to happen but finally not, over saving some false negatives. The results of the metrics regarding to different levels of sensitivity are computed as follows:

```
> for(s in c(0.80, 0.85, 0.90)){
+  ind <- which(sensi >= s)[1]

+  print(paste("Threshold:", threshold[ind]))
+  print(paste("Sensitivity:", sensi[ind]))
+  print(paste("Specificity:", speci[ind]))
+  print(paste("Accuracy:", acc[ind]))
+  print(paste("Precision:", prec[ind]))
+  print(paste("F1-Score:", f1_score[ind]))
+  cat("\n")
+ }

[1] "Threshold: 0.85"
[1] "Sensitivity: 0.809210526315789"
[1] "Specificity: 0.902479338842975"
[1] "Accuracy: 0.892070484581498"
[1] "Precision: 0.510373443983402"
[1] "F1-Score: 0.625954198473282"

[1] "Threshold: 0.87"
[1] "Sensitivity: 0.855263157894737"
[1] "Specificity: 0.8752066115702487"
[1] "Accuracy: 0.872980910425844"
[1] "Precision: 0.462633451957295"
[1] "F1-Score: 0.600461893764434"

[1] "Threshold: 0.9"
[1] "Sensitivity: 0.927631578947368"
[1] "Specificity: 0.821487603305785"
[1] "Accuracy: 0.8333333333333333"
[1] "Precision: 0.394957983193277"
[1] "F1-Score: 0.554027504911591"
```

Three levels for sensitivity values are set: 80%, 85% and 90%. Then, a `for` loop iteratively identify the threshold value that allows to obtain a minimum sensitivity value equal to the sth element of `c(0.80, 0.85, 0.90)` using the function `which`, and then computes the metrics associated. The threshold value starting from which the probabilities values imply the class to be 0 in order to obtain a sensitivity value of 90% is 0.90. Then the sensitivity, specificity,

TAB 3.3: Confusion matrix results.

	Model					
Metric	**LGCP1**	**LGCP2**	**LGCP3**	**LGCP4**	**LGCP5**	**LGCP6**
Sensibility	91.45	90.13	92.76	92.11	90.13	92.76
Precision	33.49	38.70	39.50	34.06	37.23	34.90
Specificity	77.19	82.07	82.15	77.60	80.91	78.26
Accuracy	78.78	82.97	83.33	76.87	81.92	79.88
F1-Score	49.03	54.15	55.40	49.73	52.69	50.72
AUC	94.51	95.17	95.03	94.68	94.67	95.06

accuracy, precision and F1-score values associated to this threshold are respectively 92.76%, 82.15%, 83.33%, 39.50% and 55.40%.

The precision value is 39.50% which seems to be very low. However, remind that we are in a case of imbalanced class, as seen in the beginning of Section 3.4.2. Hence, this result was expected. Indeed, the sensitivity was the metric to improve which implies to promote false positives. Here, the precision value equal to about 40% means that, globally, when a true positive is predicted, one and a half false positives are predicted. But note that in this case, false positives are tolerable. On the other hand, note that the low F1-score value is expected also. Indeed, the F1-score is not so high if one of the measures precision or sensitivity is improved at the expense of the other.

The same process for computing `lgcp3_proba0` and hence the metrics sensitivity, specificity, accuracy, precision, F1-score and AUC has been also adopted for the remaining models **LGCP1**, **LGCP2**, **LGCP4**, **LGCP5** and **LGCP6**. This enable us to compare the metric values among all models. The results are given in TAB 3.3. Note that the sensitivity, specificity, accuracy, precision and F1-score values are associated to a threshold that allowed to obtain a minimum sensitivity value of 0.90, as seen previously with the model `lgcp3`. The best model with respect simultaneously to the sensitivity, specificity, accuracy, precision and F1-score values is **LGCP3**. However, note that the values are in a close range.

These results are questionable. Indeed, the thresholds for a probability to be converted to the class **0** in order to have, for example, a minimum sensitivity value of 90%, are very high. Is the choice of these threshold values tolerable? If the threshold is set at $t = 0.70$, which is an equidistant value between the standard threshold $t = 0.50$ and our threshold $t = 0.90$, the value of sensitivity, precision, specificity, accuracy and F1-score, for model `lgcp3`, would be respectively 65.13%, 72.79%, 96.94%, 93.39% and 68.75%.

3.5.3 Results interpretation

Remind that the goals of the whole analysis was to be able, by manipulating road crash data as spatial point pattern, to be confident on saying which geographical zone of the study area is critical and which environment factor is the most dangerous. The statistical model chosen, LGCP, helps us to fulfill these goals. Indeed, a map of predicted probabilities can be produced from the model results in order to easily identify the risky spatial areas. Then, parameter estimates of the covariates in a Poisson model framework also enable to conclude

on the risky character of the environment factor, as mentioned in Section 3.3.5.

Riskiest zones of the CAGB

Firstly, we propose to map the probability that the number of events exceeds a given threshold t as defined in Eq (3.22) that we recall:

$$\mathbb{P}\big(\Lambda(c_i) > t \big| X\big) = \mathbb{E}_{\pi(\Theta|X)}\big[\mathbb{I}_{\{\lambda(c_i|\Theta)>t\}}\big]$$

For example, it could be interesting to map the probability that at least one accident happens per cell of the computational grid. This is computed as follows:

```
> exceed <- function(Y, beta, eta, Z, otherargs){
+  ca <- diff(otherargs$mcens[1:2]) * diff(otherargs$ncens[1:2])
+  lambda <- ca *
+  otherargs$poisson.offset[1:otherargs$M, 1:otherargs$N]
+  * exp(matrix(Z % * % t(beta), otherargs$M * otherargs$ext,
+  otherargs$N * otherargs$ext)[1:otherargs$M, 1:otherargs$N] + Y)

+  d <- dim(lambda)
+  return(matrix(as.numeric(lambda >= 1), d[1], d[2]))
+ }

> lgcp3_ex1 <- lgcp:::expectation.lgcpPredictSpatialOnlyPlusParameters
   (lgcp3, exceed)[[1]]

|============================================================| 100%
```

As seen before, the creation of such functions requiring joint expectations is not so obvious. The most important command-line of this function is the option `return` which is clearly explicit. The function `expectation` will use the function `exceed` to compute the probability that the number of road crashes exceed one. Note that the threshold t (in this case $t = 1$) should have been incorporated into the function `exceed` as an argument but was not as it was heavy to do with the package `lgcp`.

The object `lgcp3_ex1` is a 64×64 matrix containing the probabilities, per cell, that at least one accident occurred. However, the spatial component that could inform about the geographical location of each observation in the matrix has been lost, as usually with the run of `expectation`. Hence, as seen many times before, the matrix `lgcp3_ex1` will be converted to a `RasterLayer` object in order to get back the spatial component. The reference spatial object that will be used is, again, `obs_sf`. The whole process is computed as follows:

```
> lgcp3_ex1_ras <- raster(lgcp3_ex1)
> extent(lgcp3_ex1_ras) <- extent(obs_sf)
> lgcp3_ex1_sp <- rasterToPolygons(lgcp3_ex1_ras)
> lgcp3_ex1_sf <- st_as_sf(lgcp3_ex1_sp)
> st_crs(lgcp3_ex1_sf) <- st_crs(owin_cagb)
> lgcp3_ex1_sf <- st_intersection(lgcp3_ex1_sf, owin_cagb)
```

The matrix lgcp3_ex1 is used to create the RasterLayer object lgcp3_ex1_ras where the extent is coming from obs_sf. Then, the RasterLayer object is converted to a sf object named lgcp3_ex1_sf, by transitioning to a SpatialPolygonsDataFrame as it is necessary. The goal of this conversion is to be able to easily use the functions of package sf. Finally, only the cells inside the study window owin_cagb are kept with the use of st_intersection.

The probabilities contained in lgcp3_ex1_sf are plotted as follows:

```
> lgcp3_ex1_sf %>%
+  ggplot() +
+      geom_sf(aes(fill = layer))+
+      scale_fill_gradient(low = "#F5FAFD", high = "#AA2A10",
+      breaks = c(0, 0.50, 1), limits = c(0, 1)) +
+      coord_sf(xlim = c(905000, 950000),
+               ylim = (6665000, 6705000),
+               crs = st_crs(2154), datum = proj_plot) +
+      theme_bw() +
+      theme(panel.grid.major = element_line(colour = "black",
+                                            linetype="dashed",
+                                            size=0.1),
+            panel.grid.minor = element_line(colour = "black",
+                                              linetype="dashed",
+                                              size=0.1)) +
+      xlab("Longitude") +
+      ylab("Latitude") +
+      labs(fill = paste("Probability that at least", "\n",
+                        "one accident occurred"))
```

As lgcp3_ex1_sf is coming from a RasterLayer object, the numeric column attached, containing the probabilities, is named layer. Then, remind that the object proj_plot has been set in Section 2.2 in order to specify to the package ggplot2 that the coordinate reference system (CRS) wished for plotting is *Lambert* 93. Command-lines above generated Fig 3.14.

Fig 3.14 shows that the zones where at least one accident could occur with a high probability are in the centre of the study window.

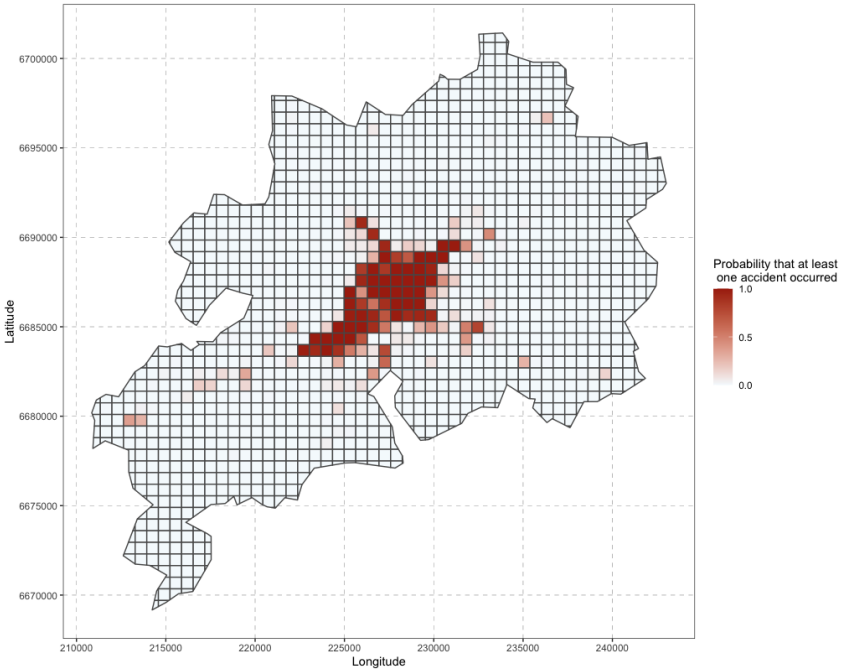

FIG 3.14: Plot of the posterior probability that the number of road crashes exceeds 1.

It is of interest to compute the exceedance probabilities and to plot the risk map for higher values of t. Consider for example $t = 5$ and $t = 10$ which correspond to map more riskier cells which are produced in FIG 3.15 and FIG 3.16 by using the same process as the one used for computing FIG 3.14. Indeed, as we can see, less red cells are displayed in FIG 3.15 and we remark very few cells of the highest risk in FIG 3.16.

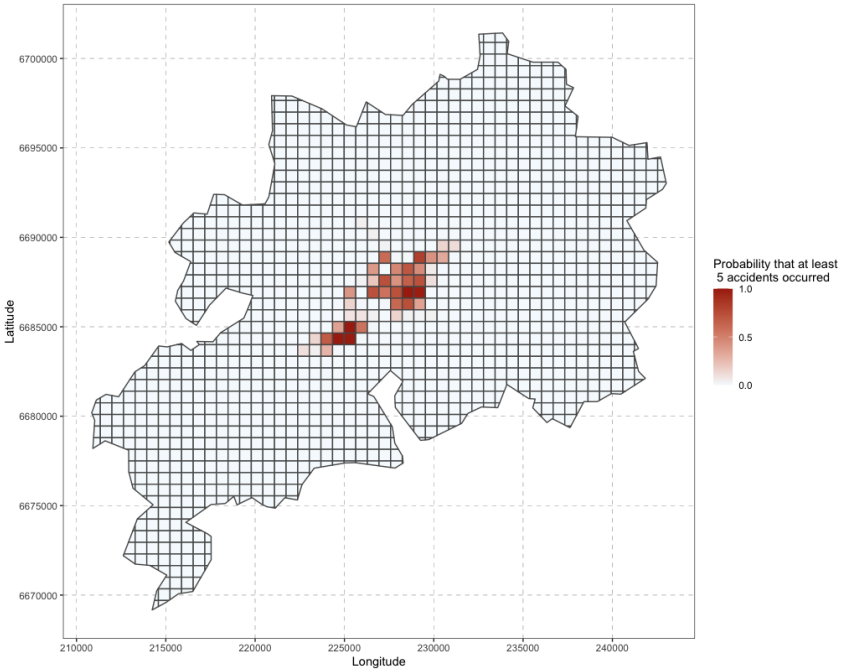

FIG 3.15: Plot of the posterior probability that the number of road crashes exceeds 5.

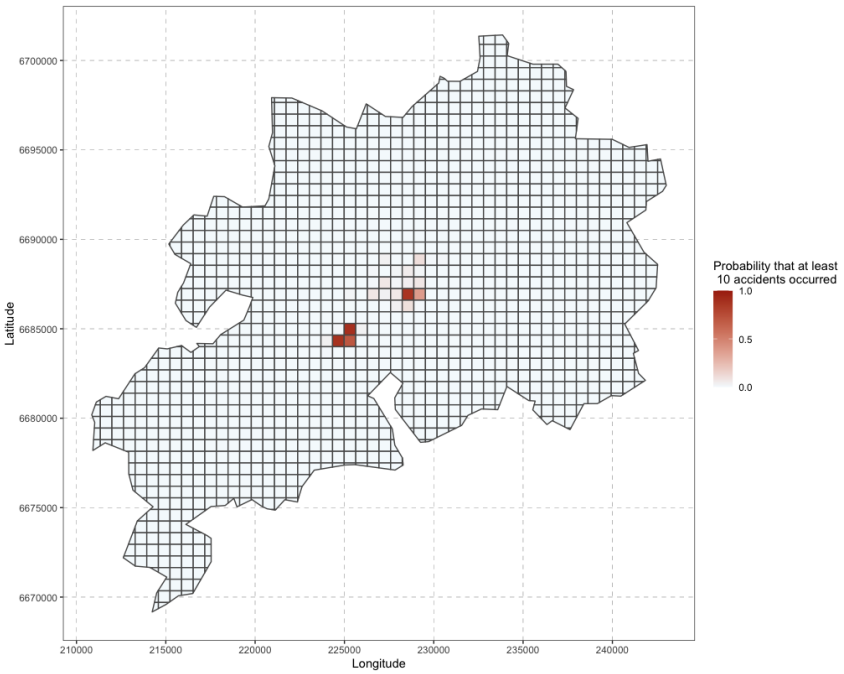

FIG 3.16: Plot of the posterior probability that the number of road crashes exceeds 10.

Finally, we propose to plot the boundaries of the riskiest cell on a map in order to visualize explicitly where the zone is in the urban community of Besançon. The code below generates the result in FIG 3.17 :

```
> lgcp3_ex10_sf_84 <- st_transform(lgcp3_ex10_sf, crs = 4326)
> max(lgcp3_ex10_sf_84$layer)

> lgcp3_ex10_sf_84 %>%
+       filter(layer == max(layer)) %>%
+       leaflet() %>%
+       addProviderTiles("Esri.WorldImagery") %>%
+       addPolygons(color = "red")

[1] 0.901
```

The `sf` object `lgcp3_ex10_sf` results from the same command-lines that produced `lgcp3_ex1_sf`, as mentioned previously, in order to compute FIG 3.16. Then, a `sf` object is created from `lgcp3_ex10_sf` by converting the CRS to *WGS* 84, named `lgcp3_ex10_sf_84`. This step is necessary for using functions from the package `leaflet`. Finally the riskiest cell, which means the cell with the highest probability that at least ten accidents happen, is plotted using the function `leaflet`. More particularly, the function `addPolygons` is used to, as its name implies, add graphics elements and layers to the map. The function `addProviderTiles` simply adds a tile layer from a known map provider.

The riskiest cell plotted in FIG 3.17 has a probability of 0.90 that more than ten accidents happen. This zone corresponds to an important junction that links a big national road ($N57$) and a departmental one ($D673$), that are roads known to have a high traffic density in the

study window.

FIG 3.17: Plot of the riskiest cell in the CAGB regarding to its probability that at least 10 accidents occur.

Finally, the latent field Y can be seen as a noise, which means the hazard relating to the road crashes which can not be explained only with the covariate information. Hence, it can be also of interest to plot the probability that the relative risk exceeds a given threshold t defined in EQ (3.23) that we recall:

$$\mathbb{P}\big(\exp(Y(c_i)) > t \big| X\big) = \mathbb{E}_{\pi(\Theta|X)}\big[\mathbb{I}_{\{\exp(Y(c_i))>t\}}\big]$$

The plots of probabilities that the relative risk exceeds $t = 2$ or $t = 5$ are computed as follows:

```
> ep <- exceedProbs(c(2, 5))
> ex <- lgcp:::expectation.lgcpPredict(lgcp3, ep)
> par(mfrow = c(2, 1))
> plotExceed(ex[[1]], 'ep', lgcp2, asp = 1, ylab = "", xlab = "")

|======================================================================| 100%
```

First, the function `exceedProbs` from the package `lgcp` computes the approximation of the exceedance probabilities for thresholds 2 and 5. Then, as already seen before, the function `expectation` is used. Finally, to plot these exceedance probabilities, the function `plotExceed` from the package `lgcp` is used. The command-lines above produced FIG 3.18.

Riskiest environment factors

In order to identify which factor increases the most the probability that an accident occurs, we look at the estimator of the β parameter as it can be expressed also as a relative

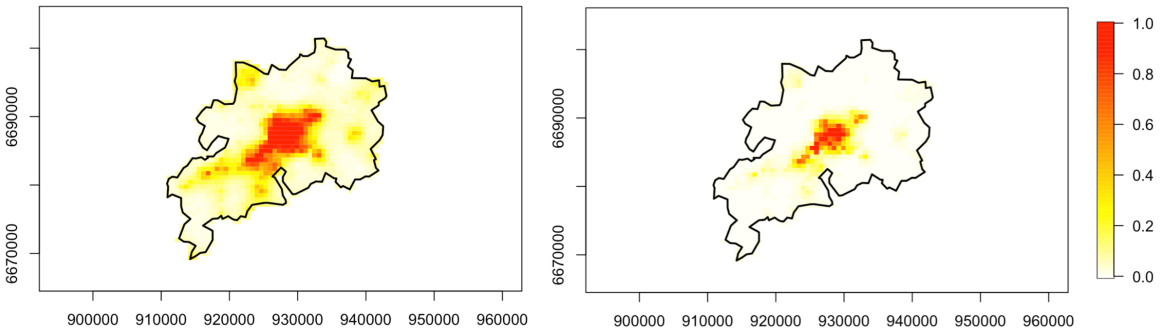

FIG 3.18: Plot that the relative risk exceeds a given threshold. Left to right : threshold equal to 2 and 5.

risk, as described in Section 3.3.5. These coefficients are obtained as follows:

```
> parsummary(lgcp3)
```

The function `parsummary` from the package `lgcp` produces a summary table for the model parameters β and η.

The results for β are given in TAB 3.4. As the covariates have been normalized, as mentioned in Section 3.4, each one can be fairly compared. Hence, it is easy to say that the riskiest factor is *national_length*, with median 87.80. Then, in second, third and fourth position there are *municipal_length*, *intersection* and *station* with respectively median 26.65, 2.99 and 2.12. The factors *prop65* and *leisure* were not found to be significant (95% CRIs contained both the value 1).

TAB 3.4: Parameter estimates of **LGCP3** model.

Parameter	**Median**	**Lower 95% CRI**	**Upper 95% CRI**
$\exp(\beta_{Intercept})$	4.78e-08	2.54e-08	8.34e-08
$\exp(\beta_{municipal\_length})$	26.65	8.10	101.71
$\exp(\beta_{national\_length})$	87.80	33.96	264.83
$\exp(\beta_{intersection})$	2.99	1.50	7.14
$\exp(\beta_{station})$	2.12	1.09	4.31
$\exp(\beta_{prop65})$	3.53	0.61	17.49
$\exp(\beta_{leisure})$	2.18	0.87	5.57

In other words, a cell zone where major roads are nationals is risky as it increases the probability that an accident happens, higher than if the major roads were municipals. Then a cell zone where the intersections are in a high number increases also the probability of accident occurrence. Finally, if a cell zone contains one or more stations such as train or taxi stations, the probability is also increased. Note that these results do not conclude that the national or municipal roads, the intersections or the number of stations are the factor that cause accidents. For example, a zone composed of one or more stations is usually a busy area. Hence, the traffic density could possibly be the factor that influences the accident occurrence.

3.6 Summary of the spatial analysis of the urban community of Besançon road crashes and discussions

Summary results The LGCP model is a tool for modelling a spatial point pattern assumed to be generated by an inhomogeneous Poisson process from which the intensity is a stochastic process assumed to vary spatially. The analysis goals of this chapter were to identify the critical areas of the CAGB while highlighting the risk factors.

A significant number of factors were available for the analysis. However, the LGCP model represents a challenge in terms of calculation and computation time. In order to choose which factors would play a role in the analysis, we proposed to proceed with a variable selection before fitting the LGCP based on the Poisson distribution, which is as close as possible to the LGCP statistical modelling. The variable selection methods, mainly inspired by Random Forest, established six variable combination to be fitted in the LGCP. Hence, six LGCP models have been fitted and have been then assessed in order to choose the best model from both explanatory and predictive points of view. The best model consists of the covariates *national_length*, *municipal_length*, *intersection*, *station*, *prop65* and *leisure*. This model enables to conclude that 91.13% of the predicted values are correlated to the observed values. In addition, we are able to predict in 92.76% of cases the occurrence of accidents.

Finally, to meet the objectives of the analysis mentioned above, we proposed to produce maps plotting the probability that one, five and ten accidents occur. This allows us to conclude on the riskiest area of the CAGB, corresponding to an area where the $N57$ and $D673$ roads intersect and represents an area known to be very busy. Then, we are able to say that the most risky factors in the hierarchical order are the length of national roads, the length of municipal roads, the number of intersection and the number of stations (train or taxis) per unit area.

Discussions Work realized in this chapter may be extended in several directions. It would have been interesting to compare the LGCP fits to other spatial statistical modellings. One possible extension with the same framework as the one set for the LGCP fits, is to model the number of road crashes per cell by with a Poisson distribution in a semiparametric model, inspired from Generalized Additive Models, for which the road crash spatial components longitude and latitude could be fitted in smooth functions (Wood, 2017, Chapter 7). Some similar examples will be given in Chapter 4.

On the other hand, it would have been interesting also to compare the LGCP prediction performances to other statistical models. The methods undertaken for the variable selection are actually models that can be compared to the LGCP as the statistical framework was the same: the number of road crashes per cell of the grid; covariates values per cell of the grid. Actually, the results have shown, according to the R-squared values, that the LGCP fits are the best. However, there exist many machine learning methods such as boosting models, support vector machines, neural networks, ... (Bao et al., 2019; Tang et al., 2020; Dong et al., 2015; Li et al., 2020). One major difficulty to use such methods on our road crash data is the lack of information recorded at a very fine scale (for example the traffic

density data per day or even per hour of the day).

Focusing on the performances of the model LGCP, we proposed to compute a weighted Mean Squared Error (wMSE), a R-squared value and the confusion matrix in the framework where the number of accidents has been classed into **0** for "no accident happened" and **1** for "at least one accident happened". These metrics enable us to decide which fit among the LGCP fits was the best according to respectively its wMSE, its R-squared value and its performance of predicting well that accidents happened when it was actually the case. On top of these metrics, it would have been interesting to simulate a realization from an inhomogeneous Poisson point process with intensity estimated from our LGCP fits. Then, this point pattern just generated can be compared to the observed road crash point pattern. To do so, a comparison metric has to be set. I would have enjoyed to focus on this latter proposition of comparison if the time would have allowed it. My first thoughts go to the nearest neighbour distance.

Then, the LGCP models fitted in this chapter are only spatial models. Another direction for further research would be to consider spatio-temporal LGCP as suggested in Taylor et al. (2015). Unfortunately, due to the thesis time constraints and the fact that a spatio-temporal LGCP is very time costing, this proposition has not been deeply studied. Moreover, in order to drive this kind of spatio-temporal analysis, the main elements were missing: the temporal covariates. Chapter 4 handles traffic density data. However, these data are available only for a very short subset of the CAGB study area: the city of Besançon. We suggest in Chapter 4 to predict the traffic density data by using kriging. In order to adapt the LGCP spatial analysis made here into a spatio-temporal one, it is actually possible to only focus on Besançon city. But we assumed that the use of predicted values based on kriging was not rigorous for this kind of statistical modelling. It could be interesting to collect road traffic density data that are displayed on the apps *Google Maps* or *Waze* for example.

Finally, it could have been interesting to combine the goals of Chapter 1 and the goals of this chapter, that means, fit a spatial analysis with a special attention on accident injuries. Several methods can be investigated in order to achieve it. For example, our road crashes point pattern can be modelled by marked point processes that allow to treat labelled point pattern (Baddeley et al., 2015, Section 1.1.3). For example, in our case the road accidents would be labelled as "slight", "serious" and "fatal" as in Chapter 1. On the other hand, we can keep the grid for which we have the accidents located in the cells and the covariates values, and we can set a cell *score* that is the sum of the injury scores. More precisely if slight, serious and fatal injuries are scored respectively as 1, 2 and 3, a cell where three accidents of each category have occurred would have a final score that is equal to 6. Bao et al. (2019) have used a similar method as the one described just before.

3.7 Supplementary material

We give in this section the supplementary diagnostics needed for validation of the MCMC algorithm and prior distributions.

3.7.1 Markov Chain Monte Carlo diagnostic checks

After running the MCMC algorithm, monitoring methods have to be employed. This include to check that the Markov chain is mixing well and converging. We will follow the steps suggested in Taylor et al. (2015) based on widely used graphical methods for MCMC convergence diagnosis.

An empirical approach to convergence control is the trace plot, which is a time series plot that shows the realizations of the Markov chain at each iteration. This graphical method is used to visualize the Markov chain in order to detect deviant or non-stationary behaviors.

The trace plots for the parameters β and η are computed as follows:

```
> traceplots(lgcp3)
```

To produce the trace plots for the parameters β and η, the function `traceplots` from the package `lgcp` is used. The trace plot of parameter σ is shown in Fig 3.19. The plot looks satisfactory as it shows random scatter around a mean value between 0.9 and 1.2 (approximatively). The reader may find all the trace plots in Appendix B.

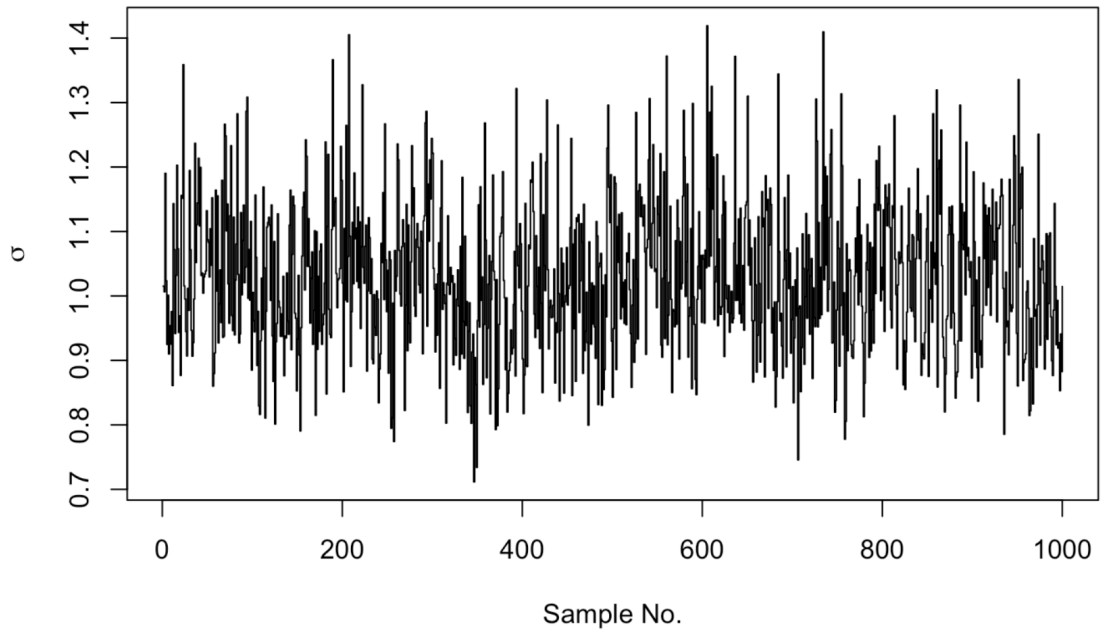

Fig 3.19: Trace plot for the parameter σ from `lgcp3`.

The autocorrelation plots can also be used in order to check the mixing of the chain. These are plots that graphically summarize the strength of a relationship between an observation in a time series with observations at prior time steps, called lags. For fast-mixing Markov chains, lag-k autocorrelation values drop down to (practically) zero as quickly as k increases. On the other hand, high lag-k autocorrelation values for larger k indicate the presence of a high degree of correlation and slow mixing of the Markov chain. The autocorrelation check of the latent Gaussian process and the autocorrelation plots for the parameters β and η are computed as follows:

```
> lagch <- c(1, 5, 15)
> Sacf <- autocorr(lgcp3, lagch, inWindow = NULL)
> for(i in 1:3){
    +  image.plot(xvals(lgcp3), yvals(lgcp3),
    +              Sacf[, , i], zlim = c(-1, 1),
    +              axes = FALSE, xlab = "", ylab = "", asp = 1,
    +              sub = paste("Lag:", lagch[i]))
    +  plot(cagb_ppp$window, add = TRUE)
    +  scalebar(5000, label = "5 km")
    + }

>  parautocorr(lgcp3)

|============================================================| 100%
```

First, different lags of the chain are chosen: 1, 5 and 15 for example. The function `autocorr` from package `lgcp` computes cell-wise for selected autocorrelations of the Gaussian process of `lgcp3`. Then, the results are plotted with the function `image.plot` on which the study window `owin_cagb` is added with the `plot` call and the autocorrelation plots are obtained with the function `parautocorr` from the package `lgcp`. The results are given in Fig 3.20.

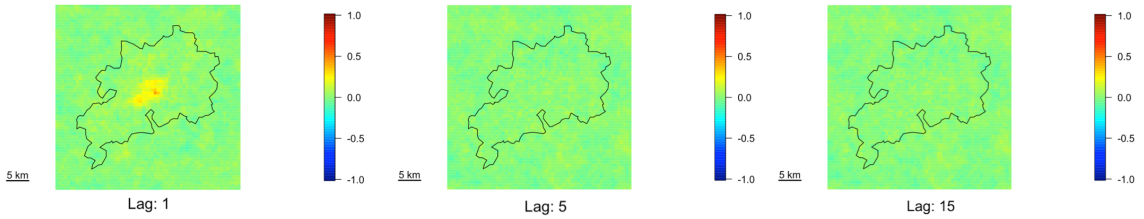

Fig 3.20: Autocorrelation plot of the gaussian process from `lgcp3`. Left to right: lag 1, 5 and 15.

The three plots in Fig 3.20 show that there is little autocorrelation, especially the left panel. However, this is very slight. Then, the autocorrelation plot for the parameter σ is shown in Fig 3.21. The reader may find the remaining autocorrelation plots for parameters ϕ and β in Fig B.7 in Appendix B. Note that there are little autocorrelation in these plots but this is very slight. These plots are still satisfying. To completely avoid this case, it is suggested to run again the algorithm for a longer period of time.

The diagnostic checks performed allow to check that the Markov Chain was mixing well and has converged. As these monitoring steps have been established satisfactory, the best chosen model `lgcp3` can be used to make inferences.

3.7.2 Prior and posterior distributions

Finally, it also of interest to make visual assessment of prior and posterior distributions, in order to graphically visualize how the posterior distribution acts relating to the prior

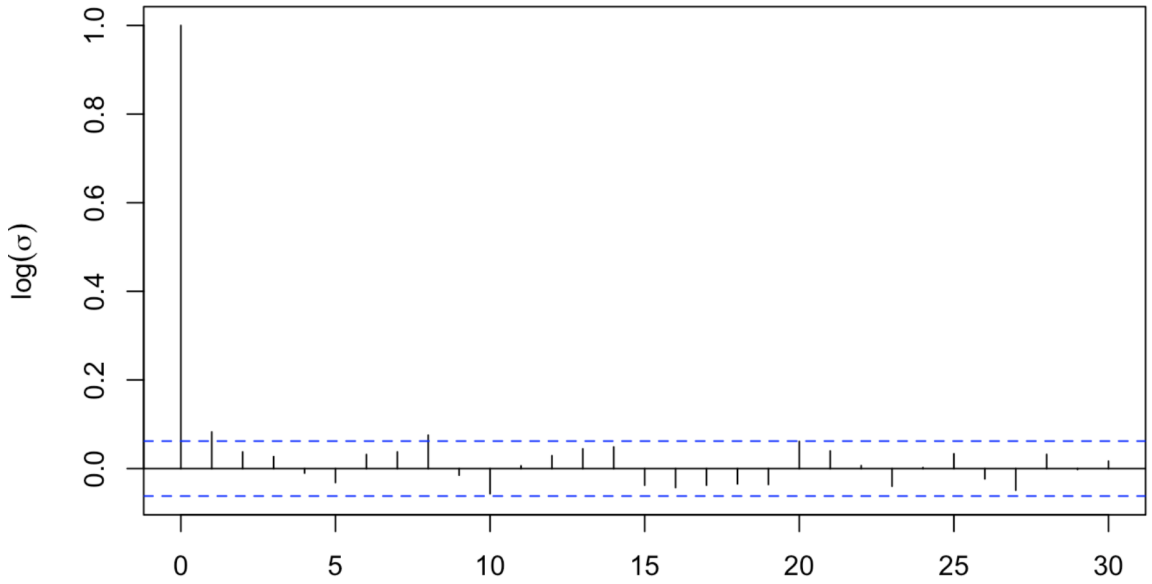

FIG 3.21: Autocorrelation plot for the parameter σ from `lgcp3`.

distribution. The corresponding plots are computed as follows:

```
> par(mfrow = c(3, 3))
> priorpost(lgcp3)
```

The function `priorpost` from the package `lgcp` plots the prior distributions, as a red line, and the posterior distributions, as a histogram, of the model parameters η and β. The command-lines above produce the FIG 3.22.

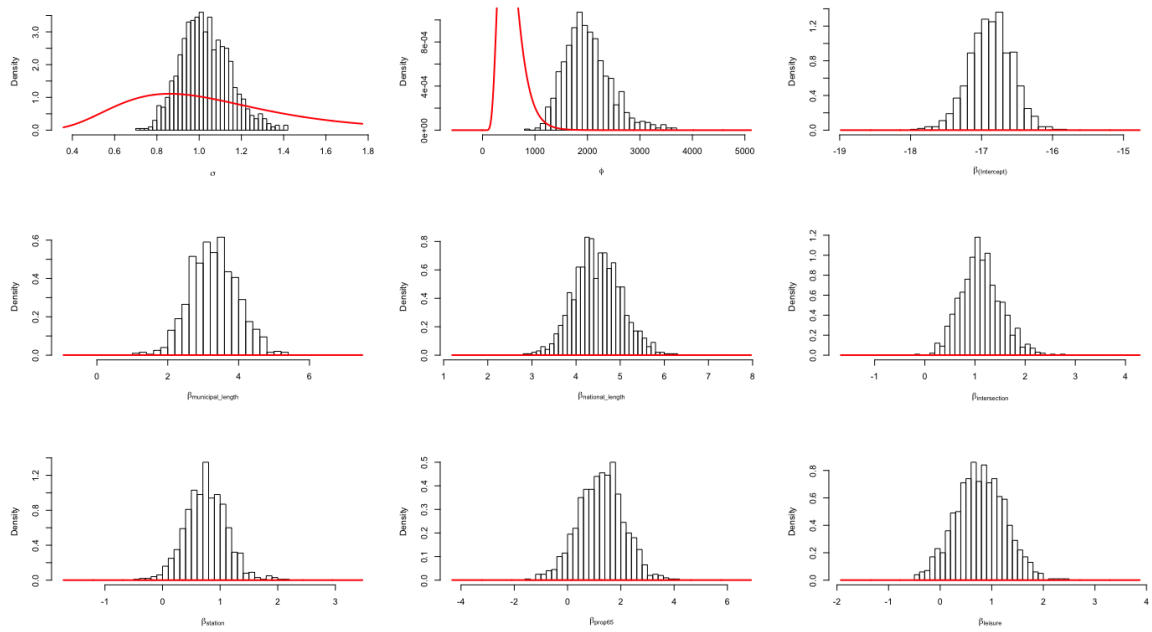

FIG 3.22: Plot of the prior and posterior distributions of each parameter.

While the covariate effects β seem to be well identified by the data, the priors for the

parameters η are not so close to the data. This tended to be not so well identified, the reader can find similar results in Taylor et al. (2015) or Ramírez and Valencia (2021) for example.

Chapter 4

Road accident space and time preliminary analyses of the city of Besançon [1]

Linear network, kriging and Generalized Additive Models

In order to prevent and/or forecast road accidents, the statistical modelling of spatio-temporal dependence and potential risk factors is a major asset. The focus in the following is on the road accidents located in space and time. A short descriptive statistical study on the road network where the accidents happened has been developed. Then, the road traffic density has been used in a geostatistical framework in order to simulate a road crash point pattern that lies on the road network. This simulated point pattern has been merged to the road crash point pattern observed in order to be analysed into space and time in an epidemiological adapted framework known as case-control study. The spatio-temporal statistical modelling method used here is the Generalized Additive Model. This work aims at giving first exploratory space and time analyses, based on the road accident data that occurred between 2017 and 2019 in the city of Besançon.

4.1 Introduction

Chapter 3 focused on giving a spatial analysis of road crash data from the CAGB, urban community of Besançon. The method of log-Gaussian Cox processes has been used in order to fulfill the main goals that were : highlight the most critical zones of the CAGB and identify the riskiest accident factors. The final goal now is to perform spatio-temporal analyses. Remind that in Chapter 1, we explored some temporal components that were associated to the road accident data from the region of Franche-Comté. The results of the Multiple Correspondence Analysis and the ordinal logistic regression models show that the time components and the road crash injury were dependent. However, these temporal

[1] This chapter corresponds to a work in progress with Benjamin Taylor started when visiting University of Lancaster in April and May 2022.

components were handled as categorical variables. Hence, the focus of this chapter is to analyse, this time, the road accident point pattern into a space and time framework with suitable spatio-temporal statistic tools. The analysis proposed here is inspired by common techniques used in the analysis of spatially-referenced case-control data in epidemiology. Our road crash data represent the cases in the study and "no crash data" represent the group of controls. Unfortunately, the group of controls is missing in our study so we propose to proceed with a simulation of the controls. Once the group of controls has been generated, we are interested in predicting the probability for a point location to be a case by considering semiparametric models based on environment characteristics into space and time. More precisely, the final goal is to identify the critical zones of the city of Besançon according to the time.

As the goal now is focused on the spatial and temporal components of our road crashes point pattern, we begin by exploring the temporal characteristics of the data. Then, a huge pre-processing work for the fitting of semiparametric models is presented in the following sections. The motivation of this pre-processing work is the simulation of the group of controls. The group of controls has to be simulated on the road network. Hence, the first part of the pre-processing tries to introduce the notion of linear networks and how to handle point patterns on a linear network. Then, the group of controls can be simulated according to a given quantity. The second part of the pre-processing work is focused on the manipulations of the traffic density data on which the simulation of the controls will be based. As our traffic density data are sparse data, the method of kriging is used in order to extrapolate these data beyond the observed locations. Finally a semiparametric space and time analysis is fitted, inspired from Generalized Additive Models, with the aim of estimating the probability of being a case, into space and time.

Similarly as Chapter 2 and Chapter 3, this chapter details the manipulations and implementations of specific statistical tools for spatial and temporal modelling from the software `R` (R Core Team, 2021) used for this exploratory analysis.

The following librairies will be used :

```
> library(areal)
> library(gstat)
> library(ggplot2)
> library(leaflet)
> library(lubridate)
> library(maptools)
> library(mgcv)
> library(raster)
> library(sf)
> library(sp)
> library(spatstat)
> library(tidyverse)
```

`R` packages `areal` (Prener et al., 2022), `leaflet` (Cheng et al., 2021), `maptools` (Bivand et al., 2021b), `raster` (Hijmans et al., 2022), `sf` (Pebesma et al., 2022), `sp` (Pebesma et al., 2021) and `spatstat` (Baddeley et al., 2021b) are tools for spatial data. The packages `gstat`

(Pebesma and Graeler, 2022) and `mgcv` (Wood, 2022) are packages used respectively to do kriging and to fit GAMs. Finally `ggplot2` (Wickham et al., 2021), `lubridate` (Spinu et al., 2021) and `tidyverse` (Wickham, 2021) are used respectively for plot, data wrangling basic operations and handling dates.

The current chapter is structured as follows. Section 4.2 gives descriptive analyses of the spatio-temporal data of the road crashes point pattern of the city of Besançon. Section 4.3 is focused on the generation of the group of controls, based on short road network analyses and traffic density data predicted by kriging. Then Section 4.4 described the semiparametric statistical modelling and its results.

4.2 Space and time descriptive analyses of Besançon road crashes

The purpose of this analysis is to describe the space and time components of the road crash point pattern of the city of Besançon. Chapters 2 and Chapter 3 have only focused until now on the spatial characteristics of these accidents. Before modelling the road crash point pattern in a space and time framework, this subsection will focus on handling such spatio-temporal data in the software `R` and describing globally the point pattern in space and time.

First, a new dataframe is loaded that corresponds to the road crashes point pattern of Besançon with the date of the accident occurrence associated :

```
> bes <- read_csv("DATA/bes.csv")
> head(bes)
> class(bes$date)

date                          latitude    longitude
-----------------------------------------------------
3 janvier 2017 19h46:00     47.24508      6.02350
7 janvier 2017 04h12:00     47.24267      6.02574
15 janvier 2017 10h13:00    47.24433      6.00460
27 janvier 2017 12h20:00    47.22535      5.97899
12 février 2017 03h34:00    47.26256      6.04514
15 février 2017 17h10:00    47.22113      5.96667

[1] "character"
```

The dataframe is composed of three columns: `date`, `latitude` and `longitude`. The first column corresponds to the date of the accident: the day, the month, the year and the time with hours, minutes and seconds. The coordinate reference system (CRS) used for the coordinates is $WGS84$. The first step is to specify to the software that the column `date` has to be handled in a date-time class. The class used in this study is `POSIXct`, which stores both date and time (contrary to the class `date` that is only a date class). A similar class as `POSIXct` is `POSIXlt`, the difference is that the class `POSIXlt` stores the hours, minutes, seconds, day, month and year separately. The transformation of `date` is computed as follows:

```
> date_format <- "%d %B %Y %Hh%M:%S"
> bes <- bes %>%
+  mutate(date_single = as.POSIXct(date, format = date_format))
```

The format on which the date will be handled is specified in `date_format`. Then, the class is defined with the function `as.POSIXct`.

The advantages of using the class `POSIXct` is that it is easier to handle the temporal characteristics of the variable. Indeed, many functions of the software, in the base or with packages such as `lubridate` for example, help to directly deal with the year, the month, the day and the time. For instance, the number of accidents per month in a year can be visualized as follows:

```
> m_levels <- c("January", "February", "March", "April",
+               "May", "June", "July", "August",
+               "September","October", "November", "December")
> tmp <- tmp %>%
+  mutate(year = year(date),
+         month = factor(months(date), levels = m_levels))

> tmp %>%
+  ggplot(aes(month)) +
+  geom_bar(width = 1) +
+  facet_wrap(~year, nrow = 1) +
+  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

The year and the month of each observation in the dataframe are obtained respectively with the functions `year` and `month`. Command-lines above produced the Fig 4.1. The number of road crashes per month vary from year to year, we can make the hypothesis that there is probably no seasonality. What could probably be interesting is to investigate why in April, May and June of 2018, the number of road crashes is much higher than in 2017 and 2019.

We propose now to plot the number of road crashes per trimester in a year. This is computed as follows:

```
> q_levels <- c("Q1", "Q2", "Q3", "Q4")
> tmp <- tmp %>%
+  mutate(quarter = factor(quarters(date), levels = q_levels))

> tmp %>%
+  ggplot(aes(quarter)) +
+  geom_bar(width = 1) +
+  facet_wrap(~year, nrow = 1) +
+  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

The trimester of each observation is obtained with the function `quarter`. Command-lines above produced the Fig 4.2. Graphically, it is easier to see that the number of road crashes is higher in 2018 than in 2017 and 2019.

The data can also be plotted in space and time. First, the creation of the `sf` object from `bes` is computed as follows:

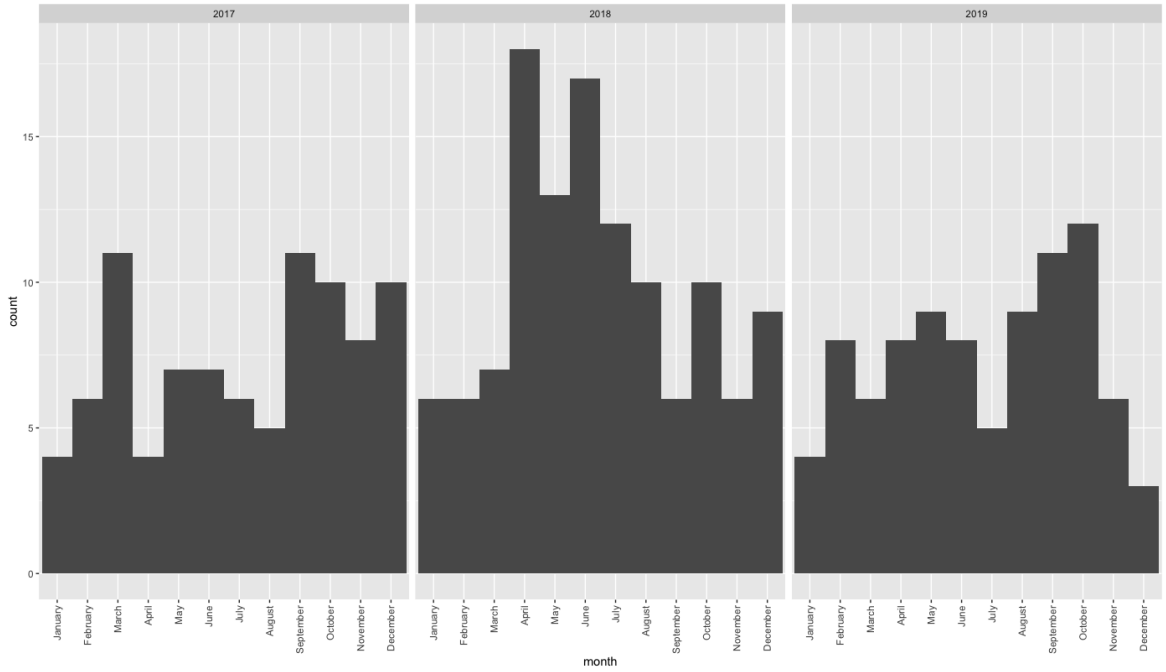

Fig 4.1: Number of road crashes per month in a year.

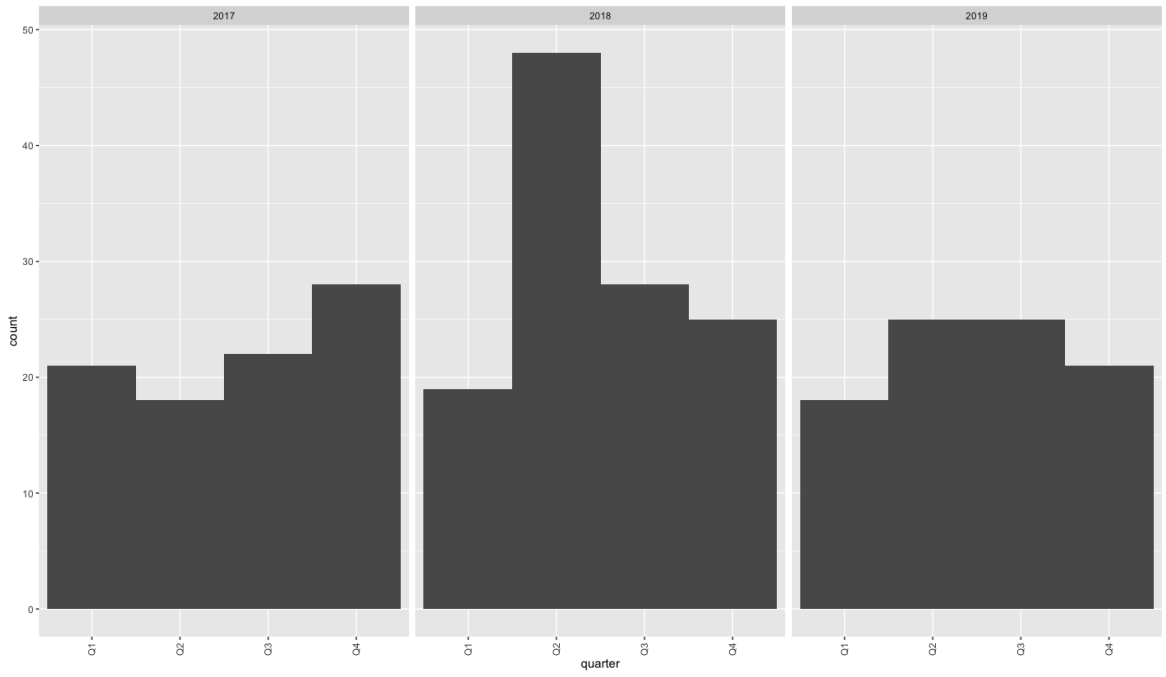

Fig 4.2: Number of road crashes per trimester in a year.

```
> bes_sf <- st_as_sf(bes, coords = c("longitude", "latitude"), crs = 4
   326)
> bes_sf <- st_transform(bes_sf, crs = 2154)
> class(bes_sf$date)

[1] "POSIXct" "POSIXt"
```

The CRS is modified in order to use *Lambert* 93 as usual. On the other hand, as you can see, the conversion to a `sf` object did not modify the class of `date`.

Then, the road crashes point pattern of the city of Besançon can be visualized as follows:

```
> owin_bes_sf <- st_as_sf(owin_bes)
> st_crs(owin_bes_sf) <- st_crs(bes_sf)

> ggplot() +
+  geom_sf(data = owin_bes_sf, color = "black", fill = "white") +
+  geom_sf(data = bes_sf, color = "black", size = 0.5) +
+  coord_sf(xlim = c(915000, 945000),
+           ylim = c(6680000, 6700000),
+           crs = st_crs(2154), datum = proj_plot) +
+  theme_bw() +
+  theme(panel.grid.major = element_line(colour = "black",
+        linetype = "dashed", size = 0.1),
+        panel.grid.minor = element_line(colour = "black",
+        linetype = "dashed", size = 0.1)) +
+  xlab("Longitude") +
+  ylab("Latitude")
```

An object `owin_bes_sf` is created from `owin_bes` with the function `st_as_sf`. The `owin` object `owin_bes` corresponds to the polygon boundary of the city of Besançon used respectively in Chapter 3. Command-lines above produced the Fig 4.3.

Fig 4.3: Road accident spatial point pattern in the city of Besançon.

We propose now to plot the road crash point pattern per day in a year. In Chapter 1, a variable *week* was used in order to compare the injury accidents that happened during the week to the ones happening during the weekend. Our results have concluded on the fact that accidents happening during the weekend were riskier than the ones happening during

the week. Hence, it is interesting to look at the road crashes point pattern in space and time with the time component decomposed into week and weekend. Indeed for example, one can ask if road crashes can be more abundant in some locations according to if they happened during the week or the weekend. This can be visualized as follows:

```
> tmp <- bes_sf %>%
+  mutate(day = factor(weekdays(date), levels = d_levels,
+         week = ifelse(day %in% c("Saturday", "Sunday"),
+                "weekend", "week"),
+         year = year(date))

> ggplot() +
+  geom_sf(data = owin_bes_sf) +
+  geom_sf(data = tmp, size = 0) +
+  facet_grid(week~year) +
+  theme_bw() +
+  theme(panel.grid.major = element_line(colour ="black",
+          linetype = "dashed", size = 0.1),
+     panel.grid.minor = element_line(colour = "black",
+          linetype = "dashed", size = 0.1),
+     axis.text.x = element_blank(),
+     axis.text.y = element_blank())
```

The day of week of each observation is obtained with the function `weekdays`. Command-lines above produced Fig 4.4. It seems that accidents that happen during the weekend have no geographical preferences compared to the ones happening during the week but it should be properly analysed.

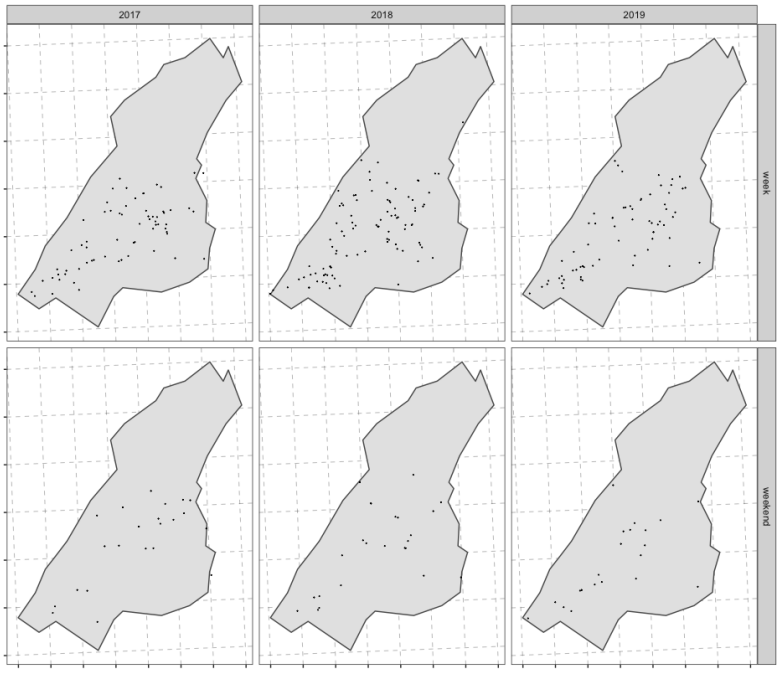

Fig 4.4: Road crash point pattern per time of the week in a year.

Similarly as the previous plot, the road crash point pattern can be visualized according to the trimester of the year. This is computed as follows:

```
> tmp <- tmp %>%
+  mutate(quarter = factor(quarters(date), levels = q_levels))

> ggplot() +
+  geom_sf(data = owin_bes_sf) +
+  geom_sf(data = tmp, size = 0) +
+  facet_grid(year~quarter) +
+  theme_bw() +
+  theme(panel.grid.major = element_line(colour ="black",
+          linetype = "dashed", size = 0.1),
+     panel.grid.minor = element_line(colour = "black",
+          linetype = "dashed", size = 0.1),
+     axis.text.x = element_blank(),
+     axis.text.y = element_blank())
```

Command-lines above produced the Fig 4.5. Graphically, it is hard to make assumptions on a possible space and time trend.

Fig 4.5: Road crash point pattern per trimester in a year.

This section gives descriptive analyses of the space and time road crashes point pattern of Besançon and enables to make assumptions on the potential temporal component that could influence these accident occurrences. Before fitting the space and time analysis in order to confirm a potential temporal effect, some elements are missing. The next section corresponds to the solution that we proposed to theses issues and is defined as *pre-processing*.

4.3 Spatio-temporal analysis pre-processing : simulation of the controls

In order to build a spatio-temporal analysis, we propose to adapt the case-control studies used in a classical epidemiological framework to our road accident study. This case-control study is designed to determine if an exposure is associated with an outcome. This can be described as follows : first, identify the cases (group known to have the outcome) and the controls (group known to be free of the outcome) ; then, look back in time to learn which subjects in each group had the exposure. In our case of road crashes, the recorded accident locations are the cases, the controls would be simulated locations in the study window and the potential exposure would be an environment characteristic such as road infrastructures for example.

The motivation of the following section is the simulation of the group of controls, as it is missing in the case-control study framework that we wish to adapt. The controls will be a point pattern in the study window. The question is then : how should we simulate the group of controls? Firstly, the controls in the framework of case-control studies can be potentially considered as the cases. Indeed, the goal of the modelling is to estimate the probability of being a case. Hence, the group of controls has to be simulated on the road network. Secondly, the points will be simulated according to the traffic density data. However, the available traffic data are sparse. Hence, we propose to extrapolate the data beyond the observed locations using kriging.

The following section introduces the notion of linear networks and what point patterns on linear networks. Then, a brief introduction to kriging is given, followed by the fit of this method on the traffic density data. Finally, in the end of this section, we generate several point patterns on the road network according to the traffic density data and hence form the group of controls.

4.3.1 Point processes on linear networks

This section aims at giving solutions on how to handle the road network by using the software `R` and especially, the road crashes point pattern of the city of Besançon on the road network. The road network is categorized as a linear network in the field of spatial statistics. A network can be railways, rivers or the work way of a worm. Baddeley et al. (2021a) gave many example of spatial point patterns on networks such as reported crimes in a neighbourhood, positions of spider webs on a urban brick wall or locations of spines of a neuron. The first strategy undertaken in order to analyse point patterns on a linear network is to adapt, if it is possible, the statistical tools already existing for point patterns in two dimensional space for the new setting of a linear network. An example of kernel estimation is given in the following that enables to make hypotheses on the riskiest road segments of the city of Besançon.

Linear network basics

Following Baddeley et al. (2015), a linear network, denoted by L, is defined as

$$L = \bigcup_{i=1}^{N} l_i, \tag{4.1}$$

where l_i are line segments and $N < \infty$ is the number of segments. Line segments l_i are set to $[u_i, v_i] = \{\omega : \omega = tu_i + (1-t)v_i, 0 \leq t \leq 1\}$ where u_i, $v_i \in \mathbb{R}^2$ are the *vertices* of l_i.

Line segments meet themselves only at their endpoints. In the case where two road cross each other, each road is splitted into segments that end at the meeting-point which is a *vertex* of the network. The number of segments which exit from each vertex is referred to as the *degree* of that vertex.

The most common distance measure used between any two points u and v in the network L is the *shortest-path distance*. A *path* between two points u and v is defined to be a sequence $x_0, x_1, \ldots, x_p$ of points in L with $x_0 = u$, $x_p = v$ and $[x_i, x_{i+1}] \subset L$. The shortest path distance is simply the minimum of the lengths of all paths from u to v, where the length of a path is the sum of the Euclidean distances between each points of the sequence $x_0, x_1, \ldots, x_p$.

We take the `sf` object `road_sf` which is the road network of the CAGB used in Chapter 2. First, we keep only the road network that is inside the city of Besançon as follows:

```
> st_crs(road_sf) <- st_crs(owin_bes_sf)
> road_sf <- st_intersection(road_sf, owin_bes_sf)
```

The road network can be visualized with the package `leaflet` as follows:

```
> road_sf_84 <- st_transform(road_sf, crs = 4326)

> leaflet() %>%
+  addTiles() %>%
+  setView(lng = 6.025490, lat = 47.236168, zoom = 15) %>%
+  addPolylines(data = road_sf_84,
+  color = "black",
+  opacity = 0.75,
+  weight = 1)
```

Remind that the creation of the `sf` object `road_sf_84`, by converting the CRS to *WGS* 84, was required for using functions from the package `leaflet`. Command-lines above, by switching values of attribute `zoom`, produced Fig 4.6.

Our purpose now is to describe how networks are handled in the software `R`. As for point patterns that have their own classes such as `ppp`, linear networks can also be manipulated in a specific class named `linnet`. An object of this class represents a linear network of straight line segments and contains information about each segment, vertex and connectivity of the network. The creation of such an object with our road network is as follows:

```
> road_sp <- as(road_sf, 'Spatial')
> road_linnet <- as.linnet.SpatialLines(road_sp)
```

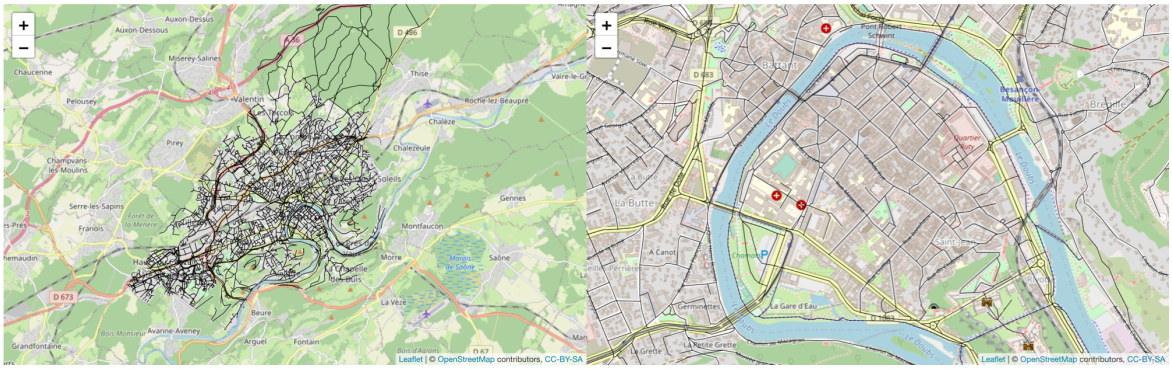

FIG 4.6: Road network of the city of Besançon. Left to right : anoverview and a focus on Besançon city centre.

```
> road_linnet

Linear network with 34665 vertices and 36161 lines
Enclosing window: rectangle = [922571.8, 932940.5] x
                              [6682230, 6695093] units
```

First an object `SpatialLines`, collection of polygonal lines, is created from `road_sf` in order to use the function `as.linnet.SpatialLines` from the package `maptools` that, as its name implies, converts such an object into an object of class `linnet`. Note that different methods exist for creating `linnet` objects such as for example the function `linnet`, that creates a linear network from vertices given by a `ppp` object. The object `road_linnet` is composed of 34 665 vertices and 36 161 lines.

Point patterns on linear networks

A point pattern on a linear network L is a finite collection of points $X = \{x_1, \ldots, x_n\}$ of $\mathbb{R}^2$ where each x_i represents a location on L. As mentioned in Chapter 3 Section 3.2.1, the point process reference models are usually assumed to be stationary and isotropic. However, these assumptions in a point process framework on a linear network are difficult to stand. Indeed, a linear network is not a homogeneous space as there is no guarantee that points will still live on the underlying network after applying some transformations and/or rotations (Baddeley et al., 2015).

To specify in the software `R` that a given point pattern lies on an network, the class `lpp` stands. An object of this class contains information such as the linear network, the spatial coordinates of the points and other kind of coordinates that specify which line segment of the network contains the point and the position of the point along this segment (between 0 and 1 where 0 corresponds to the first endpoint and 1 the second one).

Before creating an object of class `lpp`, it is interesting to visualize the point pattern and the network simultaneously.

```
> bes_sf_84 <- st_transform(bes_sf, crs = 4326)

> leaflet() %>%
```

```
+   addTiles() %>%
+   setView(lng = 6.025490, lat = 47.236168, zoom = 15) %>%
+   addPolylines(data = road_sf_84,
+   color = "black",
+   opacity = 0.75,
+   weight = 1) %>%
+   addCircleMarkers(data = bes_sf_84,
+   weight = 1,
+   radius = 3,
+   fillOpacity = 1,
+   color = "blue")
```

FIG 4.7: Road network and road accident point pattern of the city of Besançon. Note that the points do not lie on the network.

The FIG 4.7 shows a focus on the city centre of Besançon. The points plotted do not lie on the network. This situation is avoided with the creation of a `lpp` object as the coordinates of the point pattern are computed by projecting the locations onto the lines of the network. This is computed as follows:

```
> bes_ppp <- as.ppp(st_coordinates(bes_sf), owin_bes)
> bes_lpp <- lpp(bes_ppp, road_linnet)

> tmp <- as.data.frame(cbind(bes_lpp$data$x, bes_lpp$data$y))
> tmp <- st_as_sf(tmp, coords = c("V1", "V2"), crs = 2154)
> tmp <- st_transform(tmp, crs=4326)

> leaflet() %>%
+   addTiles() %>%
```

```
+   setView(lng = 6.025490, lat = 47.236168, zoom = 17) %>%
+   addPolylines(data = road_sf_84,
+   color = "black",
+   opacity = 0.75,
+   weight = 1) %>%
+   addCircleMarkers(data = tmp,
+   weight = 1,
+   radius = 3,
+   fillOpacity = 1,
+   color = "blue")
```

The final step was to create a `ppp` object from `bes_sf` and to associate it to the `linnet` object `road_linnet` with the function `lpp` from the package `spatstat`. Then, in order to visualize it with the package `leaflet` similarly as in Fig 4.7, the point pattern from the `lpp` object `bes_lpp` has been extracted and named `tmp`, like *temporary*, as its use is only to show that the point pattern lie well on the network now. Command-lines above produced Fig 4.8.

Fig 4.8: Road network and road accident point pattern of the city of Besançon. Use of the function `lpp`. Note that the points now lie on the network.

Summary information of the object `bes_lpp` is computed as follows:

```
> summary(bes_lpp)

Point pattern on linear network
296 points
Linear network with 34665 vertices and 36161 lines
Total length 655830.7 units
Average intensity 0.000451336 points per unit length
```

```
Unmarked
Enclosing window: rectangle = [922571.8, 932940.5] x
[6682230, 6695093] units
```

The call of `summary` gives information about the point pattern on the linear network such as the number of events, that are 296, or the average density points per unit length. The first property that should be investigated from this point pattern on the network is the intensity of the underlying point process.

Intensity of a point process on a linear network

Let $X = \{x_1, \ldots, x_n\}$ be a point pattern on a linear network L. A point process $\mathbf{X}$ on a linear network L has *homogenous* intensity λ if

$$\mathbb{E}[n(\mathbf{X} \cap B)] = \lambda \ell(B) \tag{4.2}$$

where $\ell(B)$ denotes the total length of subset $B \subseteq L$ and $n(X \cap B)$ denotes the number of points of X that fall in B. Note that λ represents now the expected number of points per unit length of network. The homogeneous Poisson process on a linear network is defined similarly as the one in two dimensions defined in Section 3.2 where the property **HPP1** is simply replaced with Eq 4.2.

Alternatively, a point process $\mathbf{X}$ on a linear network L has *inhomogeneous* intensity function $\lambda(u)$ defined at all locations u on L if

$$\mathbb{E}[n(\mathbf{X} \cap B)] = \int_L \lambda(u) \mathrm{d}(u), \tag{4.3}$$

where $\mathrm{d}u$ denotes integration with respect to arc length and $B \subseteq L$ a subset. Here $\lambda(u)$ represents the expected number of points per unit length of network, in the vicinity of location u. Note also that the inhomogeneous Poisson process on a linear network is defined similarly as the one in two dimensions defined in Section 3.2.2 where the property **IPP1** is simply replaced with Eq 4.3.

In practice, the purpose of point pattern analysis is the estimation of the spatially varying density of events. Intensity functions of inhomogeneous point processes can be estimated non-parametrically with kernel smoothing. Whereas these methods were simple for spatial point pattern data in two dimensions, as briefly introduced in Section 3.2.3, kernel estimation on a linear network is mathematically and computationally difficult (Baddeley et al., 2021a).

Various kernel smoothing techniques have been proposed, Baddeley et al. (2015) suggested that the method of choice is the *equal-split continuous* (Okabe et al., 2009). However, as mentioned above, algorithms of kernel estimation are computationally challenging and timely consuming. Hence, we follow Rakshit et al. (2019) and estimate the intensity with the method of convolution kernel estimation, which can be computed rapidly using the so-called Fast Fourier Transform computation algorithm. The convolution kernel estimator of

the inhomogeneous intensity $\lambda(\cdot)$ is defined as

$$\tilde{\lambda}(u) = \frac{1}{c(u)} \sum_{i=1}^{n} \boldsymbol{\kappa}(u - x_i), \qquad u \in L \subset \mathbb{R}^2, \tag{4.4}$$

where $\boldsymbol{\kappa}(\cdot)$ is the smoothing kernel and $c(u) = \int_L \boldsymbol{\kappa}(v-u)\mathrm{d}v$ is the convolution of the kernel $\boldsymbol{\kappa}$ with the arc-length measure on the network. More particularly, the kernel $\boldsymbol{\kappa}$ is a function of two spatial locations u and v, $u, v \in \mathbb{R}^2$, in contrast to spatial point pattern in two dimensions where for example the kernel estimator given in Eq 3.2 in Section 3.2.3 was a function of the distance between u and v.

An estimation of the intensity function of `bes_lpp` with the convolution estimator can be obtained as follows:

```
> smoothing_band <- bw.scott(bes_lpp)
> opt_density <- densityQuick.lpp(bes_lpp, sigma = smoothing_band)
> plot(opt_density, main = ")
```

It is difficult to choose suitable bandwidth values when dealing with linear networks. Few methods are available to set these values. We used Scott's rule (Rakshit et al., 2019) to determine the smoothing bandwidth by computing the function `bw.scott` from the package `spatstat`. Then, the estimation is computed with the function `densityQuick` from `spatstat`, it returns an object of class `linim` which represents a pixel image on the linear network.

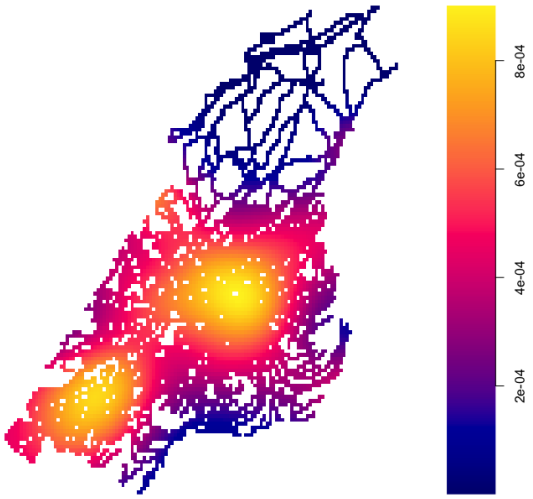

Fig 4.9: Kernel estimate of intensity for the road crashes of the city of Besançon on road network.

Command-lines above produced the Fig 4.9. The expected number of road crashes per unit length of road network is higher in the centre and in the bottom left of the city of

Besançon. We propose now to compute the density estimation again in order to better visualize the situation in Besançon city centre. The creation of a `SpatialPolygons` object, focused on the city centre of Besançon, is as follows:

```
> x <- c(928000,  928000, 929500, 929500, 928000)
> y <- c(6687000, 6685500, 6685500, 6687000, 6687000)
> centre_sp <- cbind(x, y)
> centre_sp <- Polygon(centre_sp)
> centre_sp <- Polygons(list(centre_sp), 1)
> centre_sp <- SpatialPolygons(list(centre_sp))
> centre_sp@proj4string <- CRS("+init=epsg:2154")

> centre_sp <- spTransform(centre_sp, CRSobj=CRS("+init=epsg:4326"))
> centre_sf <- st_as_sf(centre_sp)
> centre_sf %>%
+  leaflet() %>%
+  addProviderTiles("Esri.WorldImagery") %>%
+  addPolygons(color = "red")
```

First, the coordinates of the polygon vertices are set in `x` and `y` in order to form a `SpatialPolygons` object `centre_sp` with the functions `Polygon` and `Polygons` from `spatstat`. The coordinates are given in *Lambert*93, hence the CRS 2154 is specified in the attribute `proj4string` of `centre_sp`. In order to visualize with `leaflet`, the CRS is transformed to *WGS*84 and a `sf` object from `centre_sp` is created. Command-lines above produced Fig 4.10. The polygon `centre_sp` sets the limits of the city centre of Besançon, called *la Boucle* by the inhabitants, which means "the loop" due to its shape.

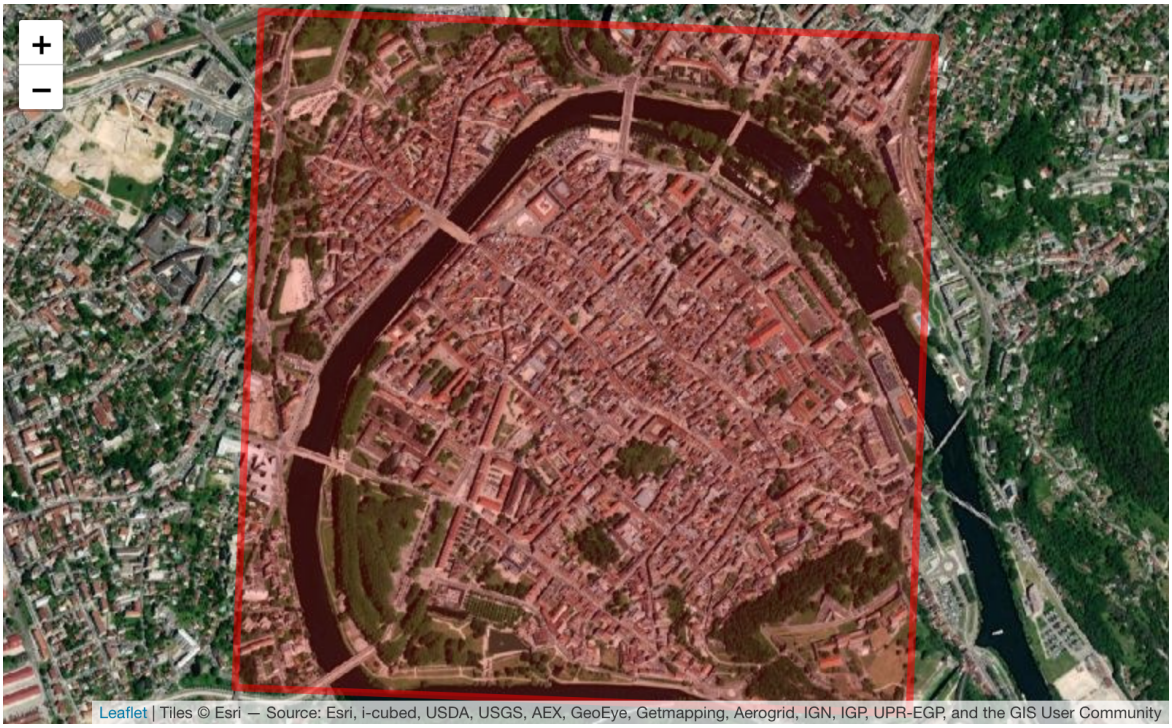

Fig 4.10: Polygon boundaries focused on Besançon centre.

Now that a new smaller study window has been chosen, the next step is to intersect the point pattern of Besançon road crashes and the road network both with the new window, and then follow each step of the kernel estimation similarly as above. This is done as follows:

```
# Intersections
> centre_sf <- st_transform(centre_sf, crs = 2154)
> r_centre_sf <- st_intersection(road_sf, centre_sf)
> b_centre_sf <- st_intersection(bes_sf, centre_sf)

# Creation of linnet, ppp and lpp objects
> centre_sp <- spTransform(centre_sp, CRSobj=CRS("+init=epsg:2154"))
> r_centre_sp <- as(r_centre, 'Spatial')
> r_centre_linnet <- as.linnet.SpatialLines(r_centre_sp)
> centre_owin <- as.owin(centre_sp)
> b_centre_ppp <- as.ppp(st_coordinates(b_centre_sf), centre_owin)
> b_centre_lpp <- lpp(b_centre_ppp, r_centre_linnet)

# Kernel estimation
> density_centre <- densityQuick.lpp(b_centre_lpp,
+                          sigma = bw.scott(b_centre_lpp))
> plot(density_centre, main = "")
```

Command-lines above produced the Fig 4.11.

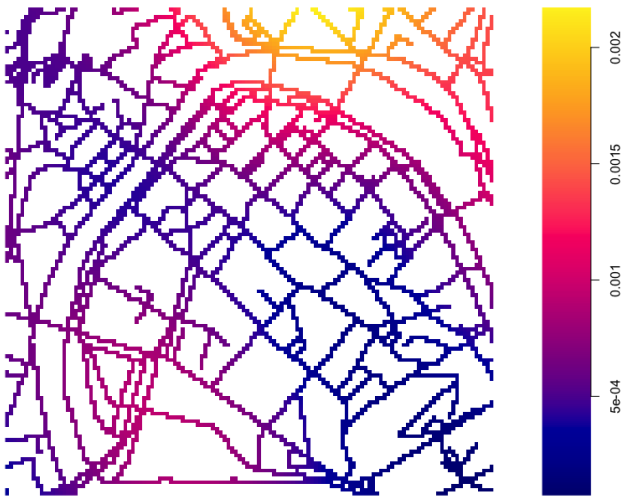

Fig 4.11: Kernel estimate of intensity for the road crashes of Besançon city centre.

It is of interest to overlay this intensity estimation on the map of Besançon city centre. To do so, the `linim` object `density_centre` has to be transformed into a `sf` object. This is done as follows:

```
> d_centre_ras <- raster(as.matrix(density_centre))
```

```
> extent(d_centre_ras) <- extent(centre_sp)
> d_centre_sp <- rasterToPolygons(d_centre_ras)
> d_centre_sp@proj4string <- CRS("+init=epsg:2154")
> d_centre_sp <- spTransform(d_centre_sp,
+                            CRSobj=CRS("+init=epsg:4326"))
>  d_centre_sf <- st_as_sf(d_centre_sp)

> pal <- colorNumeric(palette = "Blues", domain = d_centre_sf$layer)
> d_centre_sf %>%
+  leaflet() %>%
+  addProviderTiles("Esri.WorldImagery") %>%
+  addPolygons(fillOpacity = 0.5, weight = 0.5, color = ~pal(layer))
```

The steps from a `linim` object to a `sf` proposed here is the transitions to `RasterLayer` objects, as seen in Section 3.5.2, and `SpatialPolygonsDataFrame`. The `sf` object `d_centre_sf` is composed of polygons that were pixels of `density_centre`. Finally, we set the colors to the color palette `Blues` for the sake of clarity on the map. Command-lines above produced Fig 4.12. The expected number of road crashes per unit length of road network is higher in the top right, above the river *Doubs*, where stand two bridges called *Pont de la République* and *Pont Robert Schwint* that are busy places. Note that since February the 14^{th} of 2022, this bridge is closed to all motorised vehicles (Eme-Ziri, 2022).

Fig 4.12: Kernel estimate of intensity for the road crashes of Besançon city centre. With package `leaflet`.

4.3.2 Kriging

The purpose of this chapter is to perform space and time analyses of the road crashes point pattern of the city of Besançon. The goal of such a study is to obtain relevant temporal information that is related to road crashes. An important related factor in this case is the traffic density. For example Tang et al. (2020) used the average traffic volume per lane as a covariate in their supervised method named *TrAdaBoost.R2*, Ma et al. (2015) also used the traffic volume as covariate in order to fit a long short term memory neural network for road accident analysis and Park et al. (2018) used the traffic congestion in a gradient boosted decision trees model to analyse road crashes data. During this thesis, we tried to get such information. Thanks to the *Direction Départementale des Territoires* (DDT) of Doubs department and the *Grand Besançon Metropole* (CAGB), under confidentiality agreements, traffic density data has been made available. The mobility and transportation ministries set meters that record the number of vehicles crossing at fixed locations in Besançon during one week of the year. Data recorded during this week are considered to be constant over the year and are then used in order to get information about the annual average daily traffic for example.

However, the traffic density is available only at some locations in the city of Besançon and represent a sparse sample data. Hence, it would be difficult to use this information into a spatio-temporal analysis. However, this information still remains important in the road accident framework. The space and time framework analysis proposed in this chapter is inspired from case-control studies in order to identify the critical zones of Besançon into space and time. The statistical modelling that will be fitted further will enable to estimate the probability of controls being a case. This last state suppose that a control can potentially be considered as a case. Hence, the traffic density will be used for the simulation of the group of controls as it is a relevant information for road accidents.

First, the traffic density data are loaded as follows:

```
> traffic <- read_delim("DATA/traffic.csv",
+   delim = ";", escape_double = FALSE, trim_ws = TRUE)
> head(traffic)

latitude   longitude   year    tmja
-----------------------------------
47.22063   5.97827     2017    12509
47.26064   6.04771     2017    13276
47.24841   6.02992     2017     9506
47.25085   6.03414     2017    12696
47.25648   6.02068     2017    24217
47.26336   6.04335     2017    20236
```

The column containing the traffic density is *tmja* which stands for *traffic journalier moyen annuel* which means the annual average daily traffic. The traffic density data have been collected during 2017 and 2019 in the city of Besançon.

As traffic density data are sparse, the idea is to extrapolate beyond the observed locations

and to predict the traffic density on a given zone (that will be set further). Such method is part of the field of geostatistics. Geostatistics consist in making maps of the quantities of interest, such as the traffic density values, for a larger area than the finite numbers of locations where it has been recorded. That means, predictions in places where we do not have available data. The geostatistical method of prediction is known as *kriging*. It is assumed that the sparse sample data are realizations of an underlying stochastic process. Let $t(u_i)$ denotes the traffic values at observation points u_i, $i = 1, \ldots, n$. The process $T = \{T(u) : u \in D \subset \mathbb{R}^2\}$ that generated these values can be decomposed as

$$T(u_i) = m(u_i) + Y(u_i) + \epsilon(u_i), \qquad i = 1, \ldots, n, \tag{4.5}$$

where Y is a zero mean Gaussian process and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. The component $m(u_i)$ can be specified by a regression term as

$$\sum_{j=1}^{p} z_j(u_i)\beta_j,$$

where $z_j(u_i)$ is the measure of the jth covariate at u_i and β_j the associated effect. This case in geostatistics is referred as *universal kriging*. The goal is to kriging our traffic density data. A related information is the air pollution (Dekoninck and Severijnen, 2022). In the case of our geostatistical model includes this covariate, we would need to know the values of air pollution at each of the prediction locations in order to be able to predict the traffic density. The air quality index, for the city of Besançon, is provided by the organization *ATMO* Bourgogne-Franche-Comté. However, such data are recorded at only two fixed locations. Hence, this relevant information related to the traffic density can not be used. On the other hand, exploratory correlation analyses have been made in order to find if traffic density data are correlated (in some sense) to another continuous information such as the population for example, for which we do have the data at all the locations. However these analyses did not give significant results.

As no covariates are available, the component $m(\cdot)$ can be specified simply as m, which is a special case of universal kriging where the p-vector of covariates is $z(\cdot) = 1$ with $p = 1$ and $\beta = m \in \mathbb{R}$ unknown. This specification is known as *ordinary kriging*.

The Gaussian process Y in Eq 4.5 is usually assumed to be a second order stationary Gaussian process, which means that $\forall u, h \in D \subset \mathbb{R}^2$:

$$\begin{aligned} \mathbb{E}[Y(u)] &= 0 \\ \mathrm{Cov}(Y(u), Y(u+h)) &= C_Y(h) \\ \mathrm{Cov}(Y(u), Y(u+0)) &= C_Y(0) = \mathrm{Var}(Y(u)) = \sigma_Y^2 \geq 0, \end{aligned} \tag{4.6}$$

and intrinsic which means that $\forall u, h \in D$:

$$\begin{aligned} \mathbb{E}[Y(u+h) - Y(u)] &= 0 \\ \mathrm{Var}(Y(u+h) - Y(u)) &= 2\gamma_Y(h). \end{aligned} \tag{4.7}$$

The function $\gamma_Y(\cdot)$ is called the semivariogram. Developing the second state in Eq 4.7 gives $\gamma_Y(h) = C_Y(0) - C_Y(h)$. The semivariogram $\gamma_Y(\cdot)$ is unknown is estimated based on data $t(u_1), \ldots, t(u_n)$. The processes $T(\cdot)$ and $Y(\cdot)$ are related through the relation given in Eq 4.5. The following expressions are straightforward (Cressie, 1993, Chapter 3), $h \in \mathbb{R}^2$:

$$\gamma_T(h) = \gamma_Y(h) + \sigma_\epsilon^2 \mathbb{I}_{\{h \neq 0\}}$$
$$C_T(h) = C_Y(h) + \sigma_\epsilon^2 \mathbb{I}_{\{h=0\}}.$$

where $\gamma_T(h)$ and $C_T(h)$ denote respectively the semivariogram and the covariance function of the process T. An appropriate estimation of $\gamma_Y(\cdot)$ is the empirical semivariogram (Cressie, 1993, Chapter 3) given by

$$\gamma_T^*(h) = \frac{1}{2n(h)} \sum_{i=1}^{n(h)} [t(u_i + h) - t(u_i)]^2,$$

where $t(u_i + h)$ and $t(u_i)$ are observed values of T at locations u_i and $u_i + h$ and $n(h)$ is the number of paired comparisons at lag h.

A wide range of theoretical models for semivariogram and corresponding covariance functions have been proposed (Cressie, 1993, Chapter 2). The model that will be fitted further is the Gaussian semivariogram model given by

$$\gamma_T(h) = \begin{cases} \sigma_\epsilon^2 + \sigma_Y^2[1 - \exp(-\frac{h}{\phi})^2] & h > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

The shape of semivariogram models, such as the one given above, is specified basically by three parameters σ_ϵ^2, $\sigma_\epsilon^2 + \sigma_Y^2$ and ϕ respectively called the *nugget*, the *sill*, and the *range*. The value obtained by subtracting the nugget from the sill is called the *partial sill*.

The semivariogram is the cornerstone of kriging. Indeed, we predict T at any new points u_0 by

$$\hat{T}(u_0) = \sum_{i=1}^{n} \omega_i t(u_i),$$

where the weights ω_i are made to sum to 1, $\sum_{i=1}^n \omega_i = 1$, and depend on the semivariogram γ_T. For more details see (Cressie, 1993, Chapter 3).

Remind that the framework proposed here is to adapt the case-control study to our situation. As the group of controls is missing, we proposed to generate it based on the road network and the traffic density data. The latter data, as they are sparse, will be predicted using kriging. The aim of the case-control study that will be fitted further in Section 4.4 is to perform a space and time analysis. For each timed case, one (or more) control has to be set on the same date(in order to eliminate confounders at the design stage). Hence, we propose to do kriging for traffic density values per year. The steps are as follows : plot the empirical semivariogram ; fit the Gaussian semivariogram model ; kriging.

The year 2017 will be taken as example for `R` command-lines in the following. The first step is to transform traffic data into a `SpatialPolygonsDataFrame` as follows:

```
> traffic_2017 <- traffic %>%
+  filter(year == 2017) %>%
+  dplyr::select(-year)
> traffic_2017_sp <- traffic_2017
> coordinates(traffic_2017_sp) = ~ longitude + latitude
> proj4string(traffic_2017_sp) <- CRS("+init=epsg:4326")
> traffic_2017_sp <- spTransform(traffic_2017_sp,
+                                CRS("+init=epsg:2154"))
```

Then the empirical semivariogram is obtained as follows:

```
> traffic_2017_sp@data <- traffic_2017_sp@data %>%
+  mutate(tmja = sqrt(tmja))
> spplot(traffic_2017_sp, "tmja")

> vg_2017 <- variogram(tmja~1, data = traffic_2017_sp)
> plot(vg_2010, pch = 4, cex = 0.5)
```

First, the data are transformed in order to estimate easier the semivariogram further. The squared root function `sqrt` is the function chosen to be applied as it as shown good results during our preliminary tests of the method. The locations of the data, as well as the specific attribute `tmja` are plotted using the function `spplot` from `sp`. The range of squared root values are between 23 and 202 as shown in Fig 4.13. Then, the empirical semivariogram is obtained with the function `variogram` from the package `gstat`. The result is plotted in Fig 4.14.

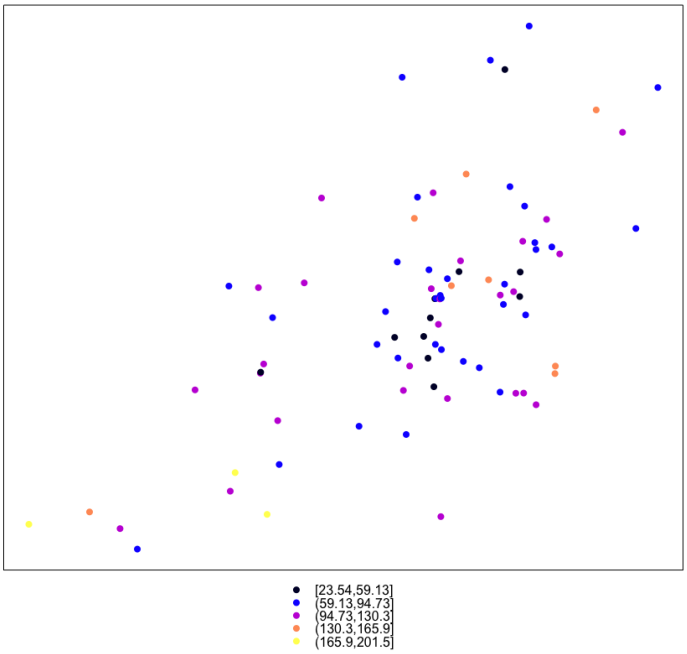

Fig 4.13: Squared root of traffic density values of 2017.

The next step is now to fit a theoretical semivariogram model. A wide range of models have been proposed. In practice, several models are fitted and visualized using the empirical

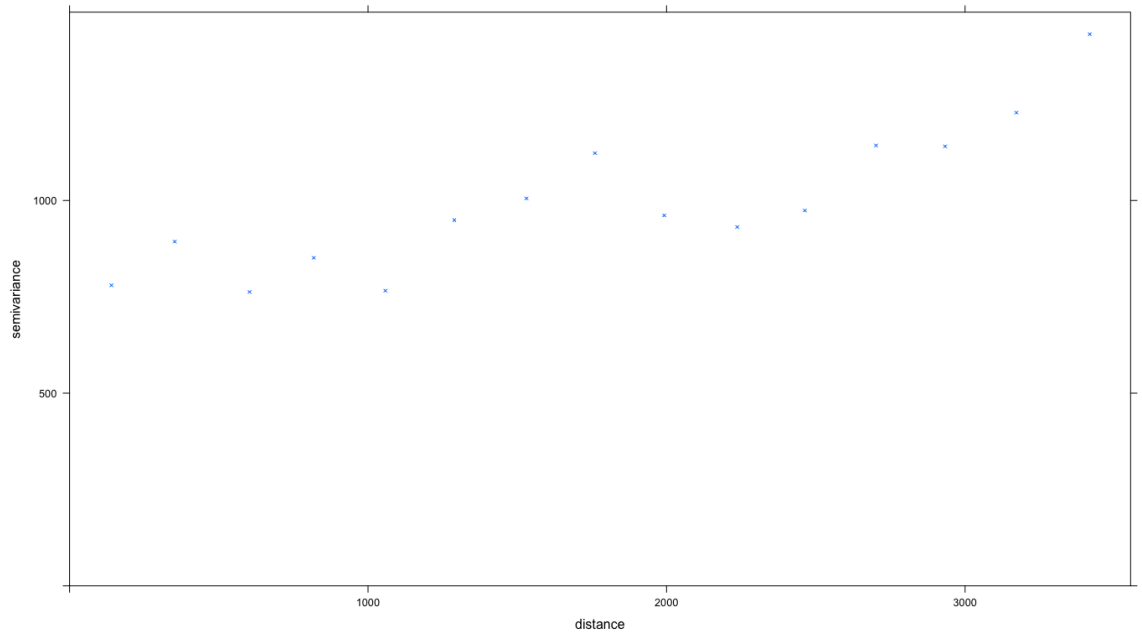

FIG 4.14: Empirical semivariogram of traffic density values of 2017.

semivariogram. The one that seems to fit suitably our data is the Gaussian semivariogram model given in EQ 4.8. This can be computed as follows:

```
> vgfit_2017 <- fit.variogram(vg_2017, model = vgm(model = "Gau",
+               psill = 1750,
+               range = 1000,
+               nugget = 750))
> plot(vg_2017, vgfit_2017, pch = 4, cex = 0.5)
```

The function `fit.variogram` from `gstat` is used with specified attributes `psill`, `range` and `nugget` that respectively correspond to the partial sill, the range and the nugget parameters of semivariogram models. The result is given in FIG 4.15. The reader may find also the fit of a Gaussian semivariogram model for traffic density data of 2019 in APPENDIX C.

The final step now is to do the kriging. Before fitting this method, an output grid has to be set onto which the traffic density values will be predicted. The construction of a regular grid covering the observation window `owin_bes_sf` is as follows:

```
> extent(owin_bes_sf)
> x <- seq(921034, 934434, by = 100)
> y <- seq(6682133, 6695533, by = 100)
> pred_grid <- expand.grid(x, y)
> pred_grid <- SpatialPixels(SpatialPoints(pred_grid))
> proj4string(pred_grid) <- CRS("+init=epsg:2154")

class       : Extent
xmin        : 922525.6
xmax        : 932943.1
ymin        : 6682110
```

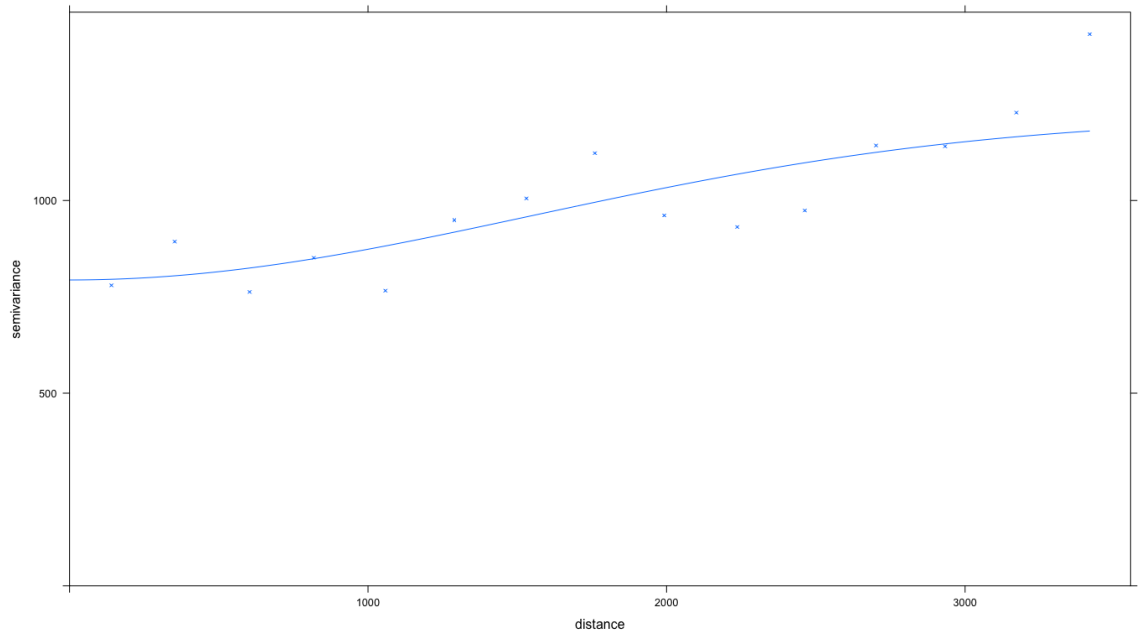

FIG 4.15: Fit of Gaussian semivariogram model (solid line) and empirical semivariogram (crosses) of traffic density data of 2017.

```
ymax        : 6695556
```

Given the extent of the window `owin_bes_sf` and the wish to predict the traffic density per cell of 100×100 meters, the extent of the grid are set as from 921 034 to 934 434 and from 6 682 133 to 6 695 533, respectively longitude and latitude values.

Then, the grid is created using the functions `expand.grid`, `SpatialPoints` and `SpatialPixels` that respectively creates a data frame from all combinations of the vectors `x` and `y`, creates objects of class `SpatialPoints` and defined a spatial grid from the points as a `SpatialPixels` object which has a `GridTopology` object in its list of attributes. The class `SpatialPixels` is a class for defining a pixels, forming a possibly incomplete rectangular grid of arbitrary dimension. On the other hand, the class `GridTopology` is a class for defining a rectangular grid of arbitrary dimension. Both are classes of the package `sp`. The overlay of the window `owin_bes_sf` on the grid `pred_grid` is computed as follows:

```
> pred_grid_sf <- st_as_sf(pred_grid)
> st_crs(pred_grid_sf) <- st_crs(owin_bes_sf)

> ggplot() +
+  geom_sf(data = predictgrid_sf, color ='black', size = 0.05) +
+  geom_sf(data = owin_bes_sf, color = 'red', fill = NA) +
+  coord_sf(xlim = c(915000, 940000),
+           ylim = c(6680000, 6697000),
+           crs = st_crs(2154), datum = proj_plot) +
+  theme_bw() +
+  theme(panel.grid.major = element_line(colour = "black", linetype =
   "dashed", size = 0.1),
```

```
+   panel.grid.minor = element_line(colour = "black", linetype = "
    dashed", size = 0.1)) +
+   xlab("Longitude") +
+   ylab("Latitude")
```

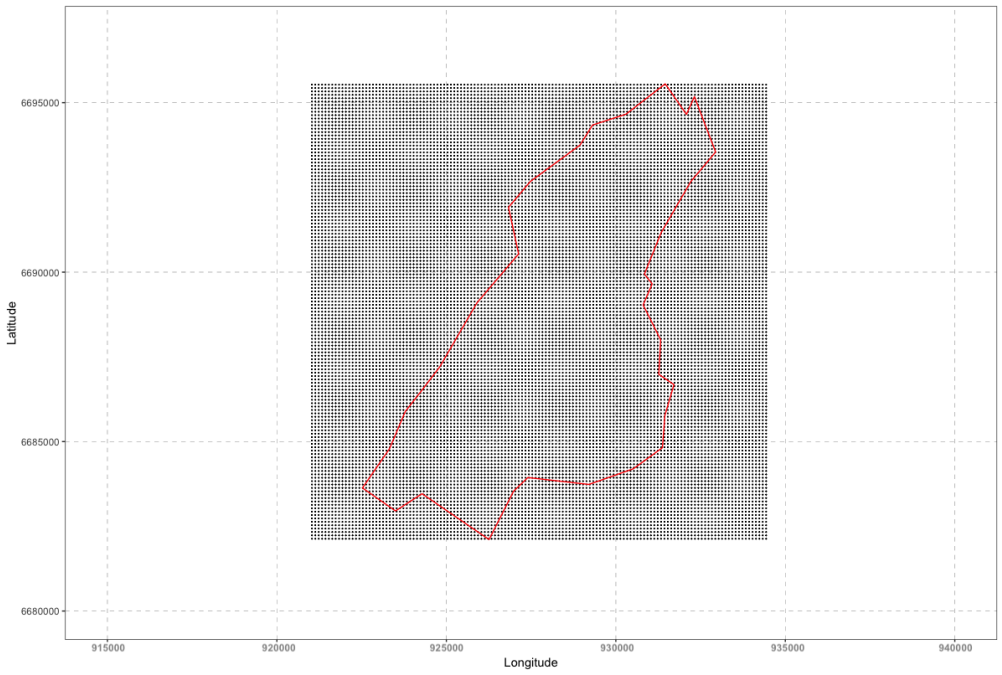

FIG 4.16: Overlay of the study window of the city of Besançon on the kriging grid.

Command-lines above produced FIG 4.16. As the grid `pred_grid` onto which the traffic density values will be predicted has been created, we are now in position to perform the kriging. This is done as follows:

```
> k_2017 <- krige(tmja~1,
+                 traffic_2017_sp,
+                 pred_grid,
+                 model = vgfit_2017)
> k_2017@data$var1.pred <- k_2017@data$var1.pred^2
> spplot(k_2017, "var1.pred")

[using ordinary kriging]
```

The kriging is performed with the function `krige` from `gstat`. The object `k_2017` is of class `SpatialPixelsDataFrame` and contains the predicted values in the column *var1.pred* that can be plotted using `spplot`. Command-lines above produced FIG 4.17. Note that as the values have been transformed earlier, they have to be transformed onto squared scale in order to go back to the original values.

FIG 4.17 shows higher traffic density values in the bottom left that corresponds to the bottom left of the study window. On the other hand, the lowest values of predicted density traffic data are located in the middle and the top of the study window. The object `k_2019` that corresponds to the predicted values for the year 2019 has been obtained with the same

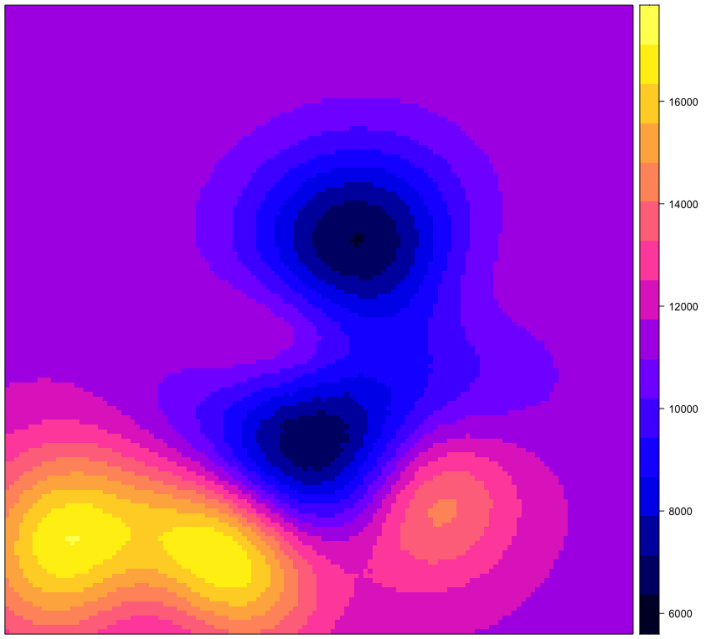

FIG 4.17: Predicted values of traffic density of 2017 for cells of `pred_grid`.

command-lines used for the year 2017. The plot of the predicted values for the year 2019 is given in FIG 4.18. As for 2017, FIG 4.18 shows higher values in the bottom left.

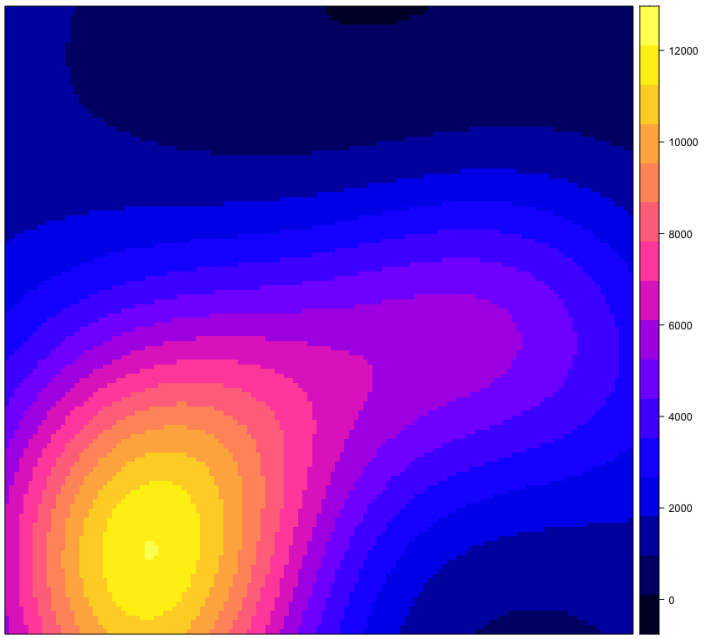

FIG 4.18: Predicted values of traffic density of 2019 for cells of `pred_grid`.

Traffic density data are considered to be almost unchanged from year to year. Hence, we propose to create an object `k_2018` that corresponds to the traffic density values over the grid `pred_grid` by averaging `k_2017` and `k_2019`. Traffic density values for the year 2018 are

plotted in Fig 4.19.

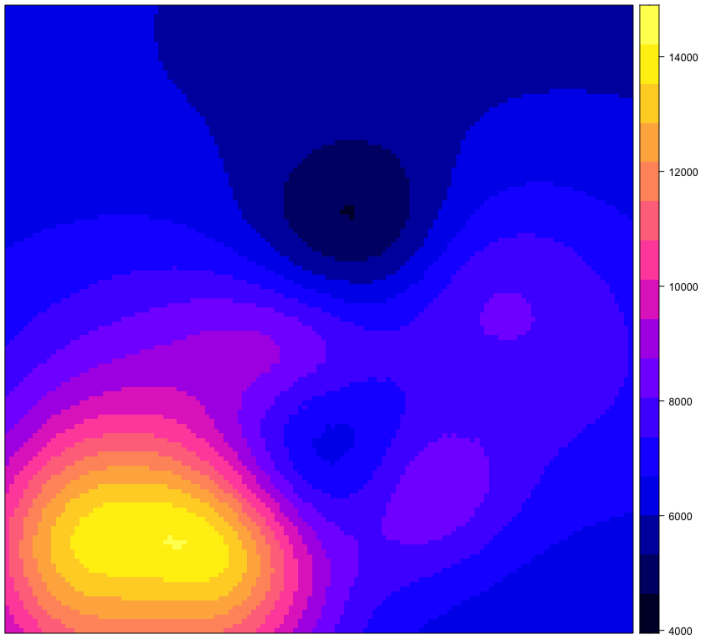

Fig 4.19: Traffic density values of 2018 for cells of `pred_grid`.

4.3.3 Simulation of the group of controls

The Section 4.3.1 introduced the notion of linear networks and gave also first preliminary spatial non-parametric analyses of the intensity of the point process that generated the road crash point pattern on the road network in the city of Besançon. The purpose was to show various manipulations on the software `R` that were available to handle the road network. Next, the Section 4.3.2 introduced the notion of kriging and gave the results of this method that allowed to extrapolate traffic density data over a grid that overlays the study window. We propose now to combine these analyses by simulating point patterns on the road network according to traffic density data.

The traffic density values have been extrapolated for each cell of the grid `pred_grid` for each year 2017, 2018 and 2019. The group of controls can be then simulated. To do so, we suggest to generate independent realizations of a Poisson point process on a linear network according to a given intensity. More particularly, a point pattern is produced on the road network based on traffic density data. This is computed as follows:

```
> k_2017_im <- k_2017
> k_2017_im@data <- k_2017_im@data %>%
+  dplyr::select(var1.pred)
> k_2017_im <- as(k_2017_im, 'SpatialGridDataFrame')
> k_2017_im <- as(k_2017_im, 'im')

> road_psp <- as.psp(road_sp)
```

```
> controls_2017<- rpoisppOnLines(t_2017_im, road_psp)
> controls_2017

Planar point pattern: 1237 points
window: rectangle = [922571.8, 932940.5] x [6682230, 6695093] units
```

First, a pixel image k_2017_im of class im is created from the SpatialPixelsDataFrame object k_2017. Then, an object road_psp of class psp is created from road_sp with as.psp from spatstat. An object of this class represents a line segment pattern in the two-dimensional plane. The creation of these objects was necessary in the following. Finally, a point pattern on the linear network according to traffic density value is simulated using the function rpoisppOnLines from spatstat. This function computes realizations of Poisson point processes from which the intensity is specified in the first argument. The point pattern controls_2017 on the network road_psp produced is a realization of a Poisson process with intensity k_2017_im and is composed of 1 237 points. Note that the range of number of points is related to the intensity given.

Many implementation methods are available in order to simulate a point pattern on a network according to preliminary information. For example, the function rpoislpp from spatstat produces also a realization of a Poisson process with a specified intensity on a given linear network. Actually, this function is very similar to the previous one except that the linear network can be specified as an object of class linnet. On the other hand, we can also compute the intersection between the pixel image k_2017_im and the road network road_sf and simply generate a point pattern with the function rpoispp seen in Section 3.2.1. The intersection of the pixel image and the network can be computed, in another way than using st_intersection from sf, as follows:

```
> extract_2017 <- t_2017_im[road_linnet, drop = FALSE]
> plot(extract_2017, main = "")
```

The subset between the pixel image t_2017_im and the linear network road_linnet is plotted in Fig 4.20. The function [,] from spatstat is named extract.im. The argument drop = FALSE specifies simply that the values outside the subset (here road_linnet) are assigned to NA values.

Point patterns on the road network for the years 2018 and 2019 have been created using the same command-lines for the creation of controls_2017, named respectively controls_2018 and controls_2019. They are composed respectively of 727 and 928 points. The three point patterns produced are plotted in Fig 4.21. Points seem to be more abundant in the bottom left than on the top of the study window, as expected.

Ratio of cases and controls We focus on the ratio of matching case and controls. Indeed, how many controls do we have to include in the analysis ? According to Setia (2016), the most optimum case-control ratio is one for one, denoted as 1:1. However in many situations, the number of cases can be poor. Hence, the number of controls can be increased in order to increase the statistical power of the analysis (as the number of cases is limited). If data are available at no extra cost, the number of controls for each case is not limited. However

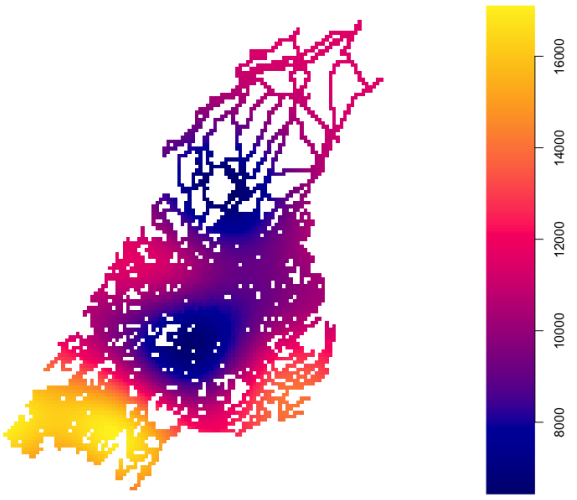

FIG 4.20: Plot of the intersection between the pixel image of traffic density values of 2017 and the road network.

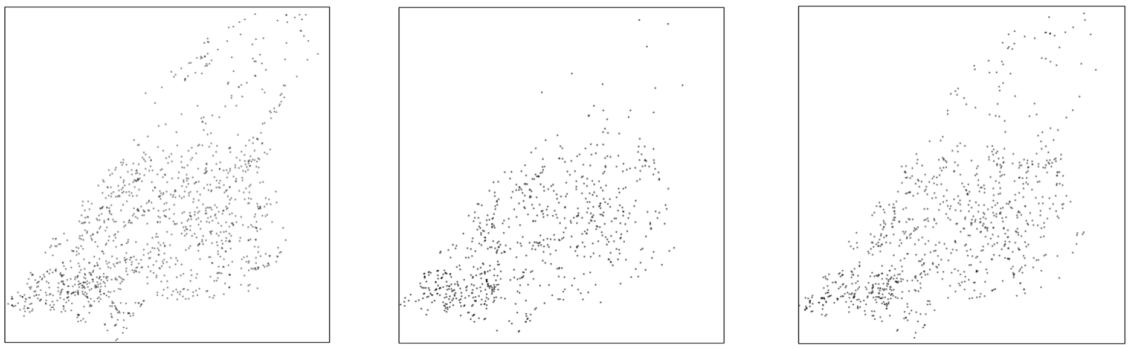

FIG 4.21: Point patterns simulated on the road network of Besançon according to traffic density values. Left to right: year 2017, 2018 and 2019.

in the cases where it is expensive to collect data, the optimal ratio is four controls per case, 1:4. Our situation is a specific case of figure as the controls are simulated. Hence, the only cost here is the computation time. We will not conclude on saying that the total number of cases, that is the total number of road crashes (296), is low or high. Hence we propose to set two different datasets where the matching ratio is 1:1 and 1:4.

The dataset design is as follows : each case would have one (or four) control that is sharing the same date, with values of the same covariates and a column *flags* that is the label of the observation (**1** for a case and **0** for a control). The first step is to assign the same dates as the cases to the controls. The creation of the dataset where the matching ratio is 1:1 is taken as an example for the software command-lines. For the year 2017, the process of matching the dates and the controls are computed as follows:

```
> dates_2017 <- bes_sf %>%
+  filter(year(date) == 2017) %>%
+  dplyr::select(date) %>%
+  st_drop_geometry()
> nrow(dates_2017)

> controls_2017_sf <- st_as_sf(controls_2017) %>%
+  filter(label == "point") %>%
+  dplyr::select(geom)

> s <- sample(nrow(controls_2017_sf), nrow(dates_2017))
> c_1_2017_sf <- controls_2017_sf[s, ]
> c_1_2017_sf <- bind_cols(dates_2017, c_1_2017_sf)
> c_1_2017_sf <- st_as_sf(c_1_2017_sf)
> colnames(c_1_2017_sf) <- c("date", "geom")

[1] 87
```

First, the dates of the year 2017 from the cases are extracted and stored in `dates_2017`. Then, for the sake of simplicity, the `ppp` object `controls_2017` is transformed into a `sf` object `controls_2017_sf`. This transformation implies the creation of a column `label` that describes the type of row, that is if it is a point or the window. Hence, we select only the points and then keep only the column that contains the geometries. Then, a number of points `c_1_2017_sf` of the object `controls_2017_sf` is randomly selected, equal to the number of dates of the year 2017 which is 87. Finally, the match is simply computed by merging the two objects `c_1_2017_sf` and `dates` with `bind_cols` from the package `dplyr`. Note that the use of this function was necessary as it is an efficient tool for binding many data frames (of any type) into one, contrary to the function `cbind` that makes a dispatch between `dataframe` and `sf` objects. Note also that the call of the function `st_as_sf` is required after the use of `bind_cols`.

The same command-lines have been computed for the years 2018 and 2019. The objects `c_1_2018_sf` and `c_1_2019_sf`, then created, are composed respectively of 118 and 91 controls. The three objects `c_1_2017_sf`, `c_1_2018_sf` and `c_1_2019_sf` are simply merged into one object `c_1_sf`.

The same whole process has been adopted for the creation of the object `c_4_sf`, that is similar to `c_1_sf`, where the matching ratio is 1:4.

4.4 Generalized Additive Model: fits and results

The space and time analysis framework proposed is to adapt our situation as a case-control study. To do so, the previous sections introduced the notion of a linear network that has been used later in order to simulate a group of controls on the road network and kriging that consisted in extrapolating the traffic density values on a grid that overlaps the study window of the city of Besançon that helped to simulate the group of controls according to the traffic density. As all the necessary elements for a spatio-temporal modelling are available,

we are now in position to drive this space and time case-control analysis. The final goal is to predict the probability for a point location on the road network to be a case based on environment characteristics. More particularly, the idea is to map the riskiest zones of Besançon into space and time. We suggest to introduce the environment characteristics as linear predictors and spatio-temporal components in a nonlinear way into the semiparametric statistical method, known as Generalized Additive Models, initially introduced by Hastie and Tibshirani (1986). This class of model is common used in the field of spatial statistics in order to fit space and/or time analyses. For instance, Wood (2017, Chapter 7) gives many data and fit examples such as spatio-temporal analyses of fish eggs, Feng (2022) fitted a spatio-temporal GAM on COVID-19 data and Wang and Brown (2012) also used a space and time GAM to model criminal incidents.

Let $F \in \{0,1\}$ be the binary response variable that labels the point locations as **1** = "case" and **0** = "control". In our case, the model takes the following form for $F \sim binomial(1, \mu_i)$

$$\text{logit}(\mu_i) = Z_i^\top \beta + f(x_i, y_i, t_i), \qquad i = 1, \ldots, n, \tag{4.9}$$

where $\text{logit}(a) = \ln(a/(1-a))$, $t \in]0,1[$, is the logistic link function, Z_i is the vector of environment covariate values for the ith observation (more details in the next section) with associated effect β and f is a nonparametric smooth function of x_i, y_i and t_i that are respectively the longitude, latitude and temporal coordinate of the ith observation. The function f is approximated into a multivariate basis function such as tensor product spline basis (Wood, 2017, Chapter 5). Model parameters are estimated using a penalised maximum likelihood approach.

4.4.1 Fits and results

As previously, the manipulations of the dataset of matching ratio 1:1 is taken as an example for the software command-lines that will be used for fitting GAMs. First, the controls are merged to the cases as follows :

```
> c_1_sf <- c_1_sf %>%
+  mutate(flags = 0)
> bes_sf <- bes_sf %>%
+  mutate(flags = 1)
> bes_c_1_sf <- rbind(bes_sf, c_1_sf)
```

Note that a label has been created in order to specify which observation is a control or a case before merging the objects. The column `flags` labels the control observations as **0** and the cases observations as **1**.

Then, we propose to incorporate information into the analysis through covariates.

Environment characteristics Remind that the goal of the analysis is to perform a spatio-temporal analysis as a case-control study. In order to determine if one exposure (or more) are associated to the outcome which is the accident occurrence, additional information through covariates will be incorporated. The covariates used are the same as the ones used in Chapter

2 and Chapter 3, that are *prop18, prop65, health, school, college, shop, station, gasoline, leisure, intersection, radars, municipal_length* and *national_length*. Data collected for the following work were reported between 2017 and 2020. This information has been considered unchanged over the years in order to associate them to road crashes that occurred between 2017 and 2019.

This information will be associated this time to a point, instead of a polygon as seen in Chapter 2 and Chapter 3. The covariate values are disposed on a grid, named `grid_bes_sf`, that is the same as the one used for kriging in Section 4.3.2. Note that the choice of this grid of cells 100×100 meters was motivated by having the information at a finer scale than in the previous chapters 2 and 3 (given at cells of 650×650 meters). However, this finer scale implies the use of different interpolation methods as the ones used in Chapter 2. The reader may find the interpolation details used to create the object `grid_bes_sf` in Section 4.6.

The covariate values of the grid `grid_bes_sf` can be associate to the controls and the cases as follows:

```
> st_crs(bes_c_1_sf) <- st_crs(grid_bes_sf)
> bes_c_1_sf <- st_intersection(bes_c_1_sf, grid_bes_sf)
```

The association of the grid with the covariates values and the object `bes_c_1_sf` has been computed using simply the function `st_intersection` as the intersection between polygons and points are points. It is a easier way to process for the merger wished.

The final step before fitting the GAMs is the one as follows:

```
> bes_c_1 <- cbind(bes_c_1_sf, st_coordinates(bes_c_1_sf)) %>%
+  st_drop_geometry()
```

The need of separate columns that contain the longitude and the latitude of the points implied the use of the function `st_coordinates`. The longitude and the latitude are now part of `bes_c_1` in columns respectively automatically named `X` and `Y`. Then, the geometries have been dropped out in order to have simply a `dataframe` object. The total number of observation in this dataset is equal to 592.

The same whole process has been adopted for the creation of the dataframe `bes_c_4`, that is similar to `bes_c_1`, where the matching ratio is 1:4. The total number of observation in this dataset is equal to 1 480.

Generalized additive model fits We now discuss potential models. The date will not be given in its simple form. In order to introduce temporal components in the following GAMs, the creation of temporal covariates is needed. We proposed to incorporate the time given at different scales such as trimester, month and day of week. The command-lines which give the creation of the corresponding columns in the datasets, named respectively `quarter`, `month` and `day`, are not shown.

The dataset `bes_c_1` and the temporal covariate `quarter` will be taken as examples for the command-lines in this section. First, the simplest model of the form given in Eq (4.9) is as follows

$$\text{logit}(\mu_i) = f(\texttt{X}_i, \texttt{Y}_i, \texttt{quarter}_i), \qquad i = 1, \ldots, 592,$$

where f is a smoothing function. In practice, a satisfactory approach to approximate the smooth functions depending on covariates measured on different scales is to use tensor product smooth bases. The reader may find more details on the latter notion in Wood (2017, Chapter 5). In our case the longitude and the latitude are given in meters and the time in terms of trimester, month or day. Hence, this situation expresses the need to use tensors. Then, the use of cubic spline is a common choice. The model above is implemented as follows:

```
> gam_quarter <- gam(flags ~ -1 + te(X, Y, quarter, bs = 'cr', k = 4),
+                    data = bes_c_1,
+                    family = binomial(link = logit))
> summary(gam_quarter)

Family: binomial
Link function: logit

Formula:
flags ~ -1 + te(x, y, quarter, bs = "cr", k = 4)

Approximate significance of smooth terms:
                  edf Ref.df Chi.sq  p-value
te(x,y,quarter) 23.65  26.66  63.18 8.13e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.116   Deviance explained = 11.8%
UBRE = 0.30282  Scale est. = 1         n = 592
```

The GAM is fitted using `gam` from the package `mgcv`. Then, the smooth term used is `te()` with attributes `bs = 'cr'` and `k = 4` which corresponds respectively to the tensor product, cubic spline and the number of knots (hyperparameter for splines).
The spatio-temporal term is significant with a p-value equal to 8.13e-05.

We propose now to fit a similar model as above but with including the covariate *national_length*:

$$\text{logit}(\mu_i) = \texttt{national\_length}_i\beta + f(\texttt{X}_i, \texttt{Y}_i, \texttt{quarter}_i), \qquad i = 1, \ldots, 592,$$

where β is the associated effect of the covariate *national_length*. This covariate was considered as an important risk factor in Chapter 3. This model is computed as follows:

```
> gam_quarter_nat <- gam(flags ~ national_length +
+                           te(X, Y, quarter, bs = 'cr', k = 4),
+                           data = bes_c_1,
+                           family = binomial(link = logit))
> summary(gam_quarter_nat)

Family: binomial
Link function: logit
```

```
Formula:
flags ~ te(X, Y, quarter, bs = "cr", k = 4)

Parametric coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          -0.2521     0.1051  -2.399 0.016461 *
national_length       1.9022     0.5424   3.507 0.000453 ***

Approximate significance of smooth terms:
edf Ref.df Chi.sq  p-value
te(x,y,quarter) 24.03  27.32  68.79 2.14e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.134   Deviance explained = 13.7%
UBRE = 0.28451  Scale est. = 1         n = 592
```

The covariate *national_length* and the spatio-temporal term are both significant. The exponential of the estimate of *national_length* is equal to 6.70 which means that the increase of length of national roads increases the probability of being a case. In other global words, the environment covariate *national_length* is a risky factor.

Several models have been fitted, with each temporal component and each environment covariate, both on the datasets `bes_c_1` and `bes_c_4`. The results are respectively summarized in TAB C.1 and TAB C.2 in APPENDIX C. Globally for models on data `bes_c_1`, the significant covariates are *shop*, *intersection* and *national_lenght*. The spatio-temporal components that are always significant are the ones with the temporal covariates `quarter` and `day`. Then for models on data `bes_c_4`, the significant covariates are *shop*, *gasoline*, *intersection* and *national_lenght*. All the spatio-temporal components, with the temporal covariates `quarter`, `month` and `day`, are almost always significant. The model forms proposed in this chapter are set in order to make assumptions, more research is needed to implement more rigorous models implying covariate selection, spline hyperparameter tuning and is planned to be treated later.

Riskiest zones of Besançon into space and time For now, it is interesting to map the estimated probability at each location of being a case, according to the temporal component. With the fitted model `gam_quarter_nat`, this can be computed as follows :

```
> plot(owin_bes_sf)
> vis.gam(gam_quarter_nat, view = c("X", "Y"),
+          plot.type = "contour", type = "response",
+          cond = list(quarter = 1), color = "terrain",
+          too.far = 0.08, add = TRUE)
> plot(owin_bes_sf, add = TRUE)
```

The function `vis.gam` from the package `mgcv` enables to produce perspective or contour plot views of `gam` objects. The attribute `view` has to contain the names of the two main effect

terms to be displayed on the x and y dimensions of the plot. Then, the attribute `view` is a list of the values to use for the other predictor terms. Note that the column `quarter` has been labelled differently as seen previously in Section 4.2 and is of type `double` with values 1, 2, 3 and 4. Here the value equal to 1 of the covariate `quarter` has been taken as an example. Command-lines above produced Fig 4.22.

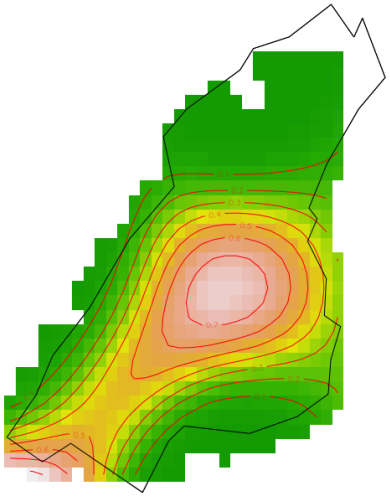

Fig 4.22: Plot of estimated probabilities of being a case from the fit `gam_quarter_nat` at level `1` of `quarter`. Green to red: probability values from 0 to 1.

The plots for the remaining values 2, 3 and 4 of the covariate `quarter` have been produced using similar command-lines as above. The corresponding plots are given in Fig 4.23. The comparison of the estimated probabilities according to each level of `quarter` is slightly varying. The probability of being a case is higher in the middle of the study window as well as in the bottom left, for each trimester. The middle of the study window corresponds to a circle of centre approximatively equal to the train station *Gare Viotte* of the city of Besançon. Then the bottom left of the study window corresponds globally to the departmental road *D*673. Remind that in Chapter 3, the riskiest cell of the urban community of Besançon (CAGB) was a cross between roads *N*57 and *D*673. The zone of this cell is clearly included in the bottom left of the study window here.

4.5 Summary of the space and times analyses of the Besançon road crashes and discussions

Summary results In order to conduct a space and time analysis of the Besançon road crashes, the chosen statistical method was inspired from case-control studies. As only the group of cases (occurred accidents) was available, the group of controls had to be simulated. To do so, two big steps have been set: the manipulation of the road network; the traffic

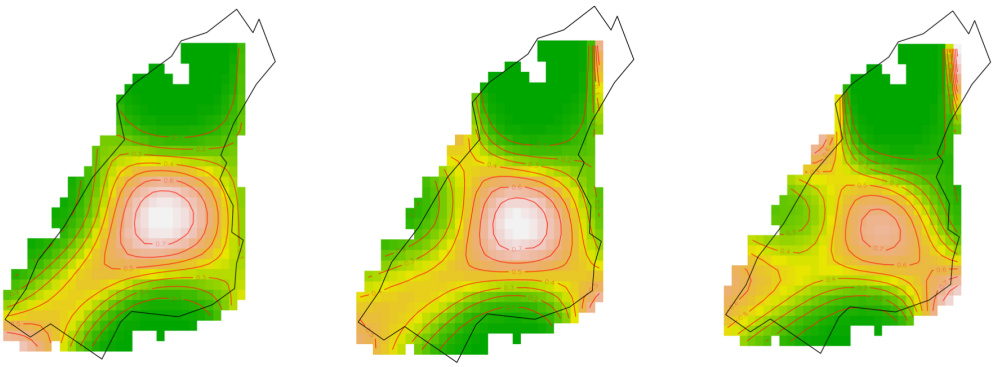

FIG 4.23: Plot of estimated probabilities of being a case from the fit `gam_quarter_nat`. Left to right: levels `2`, `3` and `4` of `quarter`. Green to red: probability values from 0 to 1.

density kriging. Firstly, the part focused on the road network enabled to handle the road network on the software. On the other hand, the study also gave the road segments that were the riskiest using a kernel estimation of the intensity of the point process that generated the road crashes point pattern of Besançon. More specifically a strong hypothesis on road segments near the bridges *Pont de la République* and *Pont Robert Schwint* to be risky has been made. Secondly, the traffic density data have been predicted on a given grid using kriging. The data have been produced in order to be considered as the intensity of a point process that generated point patterns on the road network. These point patterns constituted the group of controls for our case-control study. Finally, space and time analyses have been fitted using semiparametrics models, based on Generalized Additive Model (GAM). Maps of probabilities being a case have been produced and enable to conclude that the riskiest zones of the city of Besançon are a circle centred on the train station *Gare Viotte* and the departmental road *D*673.

Discussions This chapter gives only preliminary analyses and allows to make assumptions on potential spatio-temporal dependences or eventual future investigations. It could be interesting to model the road crashes point pattern on the road network using point processes (Baddeley et al., 2021a). Then for the spatio-temporal analysis, the focus on the GAM has to be set. Indeed, the GAMs need to be improved. These models are composed of hyperparameters that can be tuned and there are several propositions of covariate combinations to be tested.

4.6 Supplementary material: areal interpolation

This section aims at giving the interpolation methods used in order to form the grid of covariate values `grid_bes_sf` used in Section 4.4.1.

The information through covariates in GAMs is associated this time to a point, instead of a polygon as seen in previous chapters. The process proposed is as follows: use the same grid that was used for kriging for the interpolation methods and then make the intersection

between the cells and the points that will keep the data. Note that the choice of this grid of cells 100×100 meters was motivated by having the information at a finer scale than in the previous chapters (given at cells of 650×650 meters).

As almost all the steps for the interpolation methods are similar to ones used in Section 2.3.2, only the steps for the interpolation of socio-demographic data will be presented below. Indeed, remind that in Section 2.3.2 the socio-demographic data were available on a grid of cells 200×200 meters, named as source zones. The target zones, that was a grid of cells 650×650 meters, were bigger than the source zones. Hence, the interpolation method used was simply the point-in-polygon method. However, the target zones now are cells of 100×100 meters, which are smaller than the source zones. Hence, in order to be as rigorous as possible, the method point-in-polygon should be avoided (Do et al., 2021) and we used instead the *areal weighting interpolation* (DAW).

Remind some notations given in Section 2.3.2. The set of target and source zones are respectively $t_1, \ldots, t_T$ and $s_1, \ldots, s_S$. The value of the variable Z (variable of interest to be interpolated) on the the target zone t_i and the source zone s_j is denoted respectively as Z_{t_i} and Z_{s_j}. The intersection zone between zones t_i and s_j is I_{t_i,s_j}. The value of Z on the intersection zone I_{t_i,s_j} is denoted as Z_{t_i,s_j}. The DAW method is defined as follows

$$Z_{t_i} = \sum_{j=1}^{S} Z_{t_i,s_j} = \sum_{j=1}^{S} \frac{|I_{t_i,s_j}|}{|s_j|} Z_{s_j}$$

where $|I_{t_i,s_j}|$ and $|s_j|$ denote respectively the area of the intersection zone I_{t_i,s_j} and the source zone s_j, $i = 1, \ldots, T$, $j = 1, ..., S$.

In other words, in the case where for example a source zone overlaps two target zones, the DAW method disaggregates the values of the variable Z to be interpolated between the two target zones, proportionally to the area of the intersected zones.

First for the sake of clarity, a copy of the object `pred_grid_sf` is made, named `grid_bes_sf`, in order to use another named object as their uses have different goals. New socio-demographic datasets are loaded and named `ngrid_insee_sf`. In the date of research work written in Chapter 2 and Chapter 3, the socio-demographic data dated to 2015 and have been provided by INSEE in July 2019. The website INSEE made socio-demographic data available, similar as ones used in previous chapters, dated to 2017 and provided in March 2022. Hence, we decided to use the up-to-date data in order to be as accurate as possible.

The data of `ngrid_insee_sf` are the same as `grid_insee_sf` from Section 2.3.2:

- `Ind` : the number of individuals ;
- `Ind_18_24` : number of individuals between 18 and 24 years old ;
- `Ind_65_79` : number of individuals between 65 and 79 years old ;
- `Ind_80p` : number of individuals more than 80 years old.

```
> nrow(grid_bes_sf)
```

```
> sum(st_area(grid_bes_sf))
> nrow(ngrid_insee_sf)
> sum(st_area(ngrid_insee_sf))

[1] 18225
182250000 [m^2]
[1] 900
34054543 [m^2]
```

The grid where the information has to be interpolated is composed of $T = 18\,225$ target zones that cover an area of 182 250 000 squared meters. A pre-processing of `ngrid_insee_sf` data (command-lines not shown here) consisted in making the intersection between the INSEE cells and the study window `owin_bes_sf`. The `sf` object `ngrid_insee_sf` is composed of $S = 900$ sources zones that cover an area of 34 054 543 squared meters.

The interpolation of these four variable values on the grid `grid_bes_sf` using the DAW method is computed as follows:

```
> grid_bes_sf$id <- seq(1, nrow(grid_bes_sf))
> grid_bes_sf$id <- paste0('target', grid_bes_sf$id)
> ngrid_insee_sf$id <- seq(1, nrow(ngrid_insee_sf))
> ngrid_insee_sf$id <- paste0('source', ngrid_insee_sf$id)

> st_crs(ngrid_insee_sf) <- st_crs(grid_bes_sf)
> grid_bes_sf <- aw_interpolate(grid_bes_sf, tid = id,
+                   source = ngrid_insee_sf, sid = id,
+                   weight = "sum", output = "sf",
+                   extensive = c("Ind", "Ind_18_24",
+                                 "Ind_65_79", "Ind_80p"))
```

The function for applying the DAW method is `aw_interpolate` from the package `areal`. This function requires first : a `sf` object that represents the target zones, the column name that contains the identification of the target zones, a `sf` object that represents the source zones and the column name that contains the identification of the source zones. Hence, the first four command-lines above consist in the creation of the identifications required. Then, the target variables are specified in the attribute `extensive`.

Finally, the proportions of people aged between 18 and 24 years old, and over 65 years old are computed as follows:

```
> grid_bes_sf <- grid_bes_sf %>%
+  mutate(prop18 = Ind_18_24/Ind,
+       prop65 = (Ind_65_79 + Ind_80p)/Ind) %>%
+  dplyr::select(-Ind, -Ind_18_24, -Ind_65_79, -Ind_80p)
> grid_bes_sf[is.na(grid_bes_sf$prop18), ]$prop18 <- 0
> grid_bes_sf[is.na(grid_bes_sf$prop65), ]$prop65 <- 0
```

The remaining covariates *health, school, college, shop, station, gasoline, leisure, intersection, radars, municipal length* and *national length* do not suffer from the smaller cell size of 100×100 meters and are, hence, interpolated as seen in Section 2.3.2. The final `sf` object `grid_bes_sf` can be exported as follows:

```
> st_write(obj = grid_bes_sf, "DATA/grid_bes_sf.shp", delete_layer =
    TRUE)
```

Note that, as used in Section 3.4, the data values have been normalized using the min-max normalization.

Appendix A

Spatial road crashes and related factors data handling

This appendix corresponds to the supplementary materials of Chapter 2. In the following, the reader may find plots of each variable values that composed the object `grid_cagb_sf` created in Section 2.3.3. The following plots show the range of *prop18*, *prop65*, *health*, *school*, *college*, *station*, *gasoline*, *leisure*, *intersection*, *radars*, *municipal_length* and *national_length* values respectively given in Fig A.1, Fig A.2, Fig A.3, Fig A.4, Fig A.5, Fig A.6, Fig A.7, Fig A.8, Fig A.9, Fig A.10, Fig A.11 and Fig A.12. Note that the plot of the values of *shop* has been taken in example and has been produced in Fig 2.9.

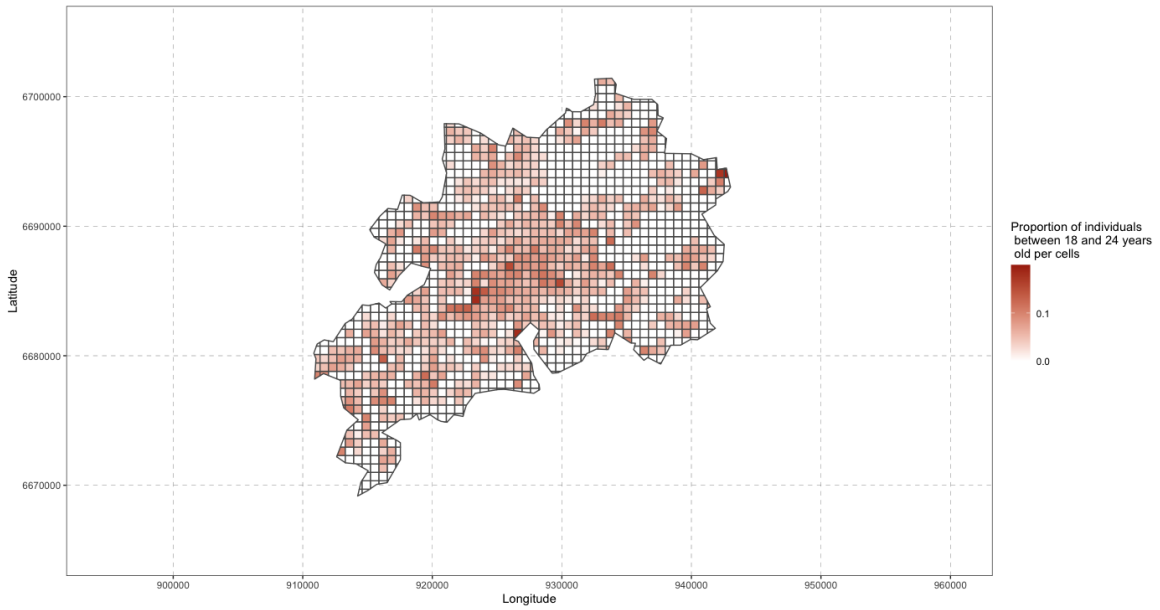

Fig A.1: Plot of *prop18* variable values, between 0 and 0.20, per cells of `grid_cagb_sf`.

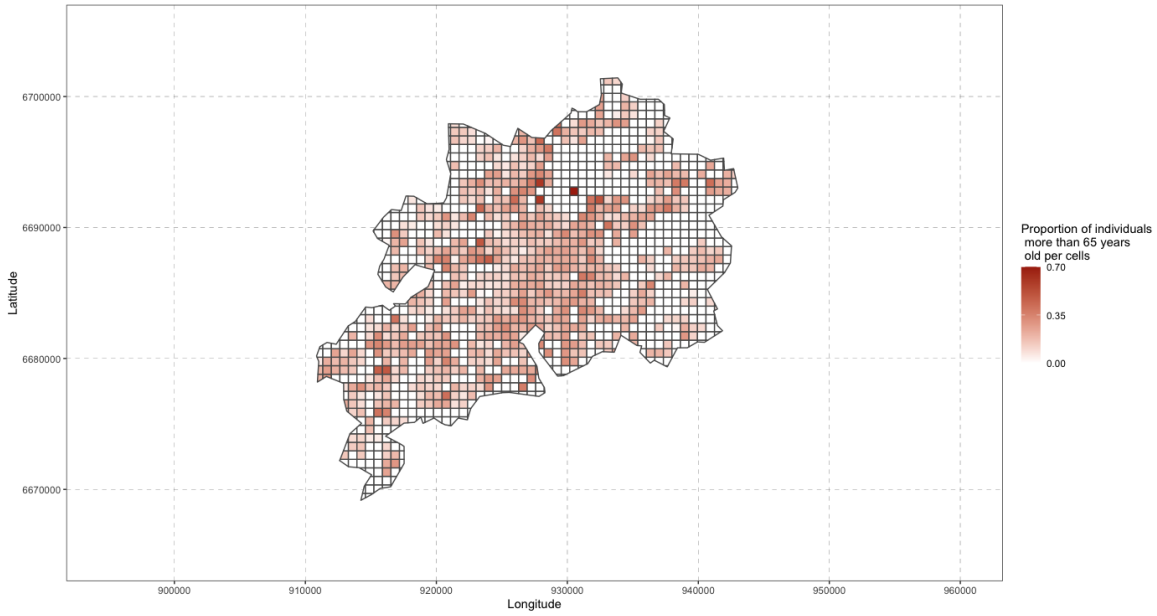

FIG A.2: Plot of *prop65* variable values, between 0 and 0.70, per cells of `grid_cagb_sf`.

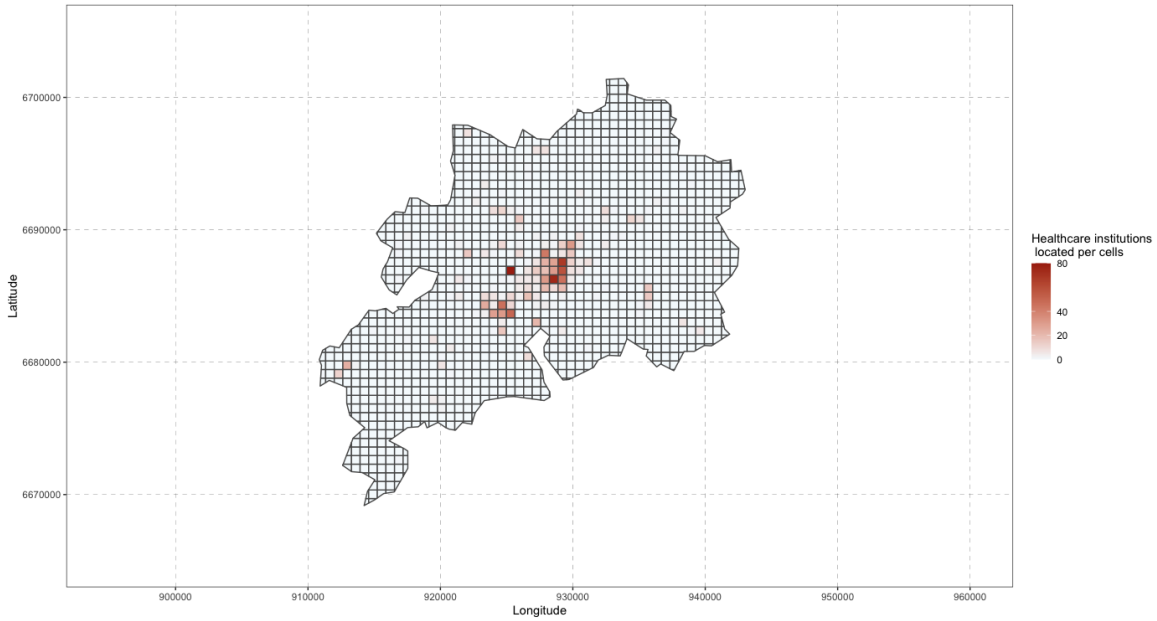

FIG A.3: Plot of *health* variable values, between 0 and 80, per cells of `grid_cagb_sf`.

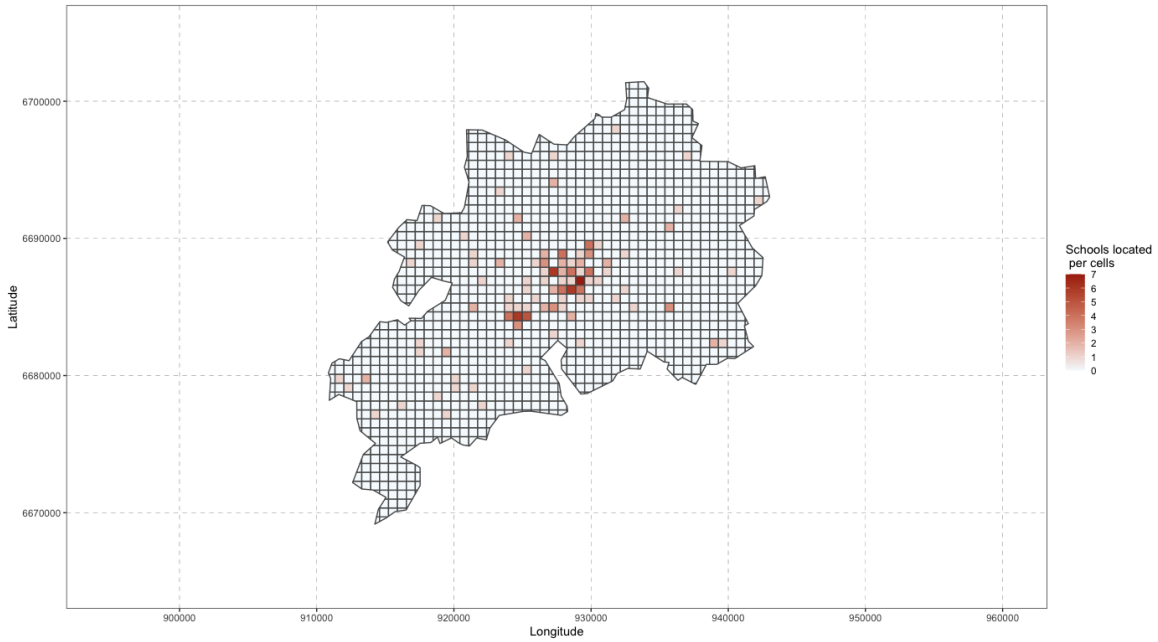

FIG A.4: Plot of *school* variable values, between 0 and 7, per cells of `grid_cagb_sf`.

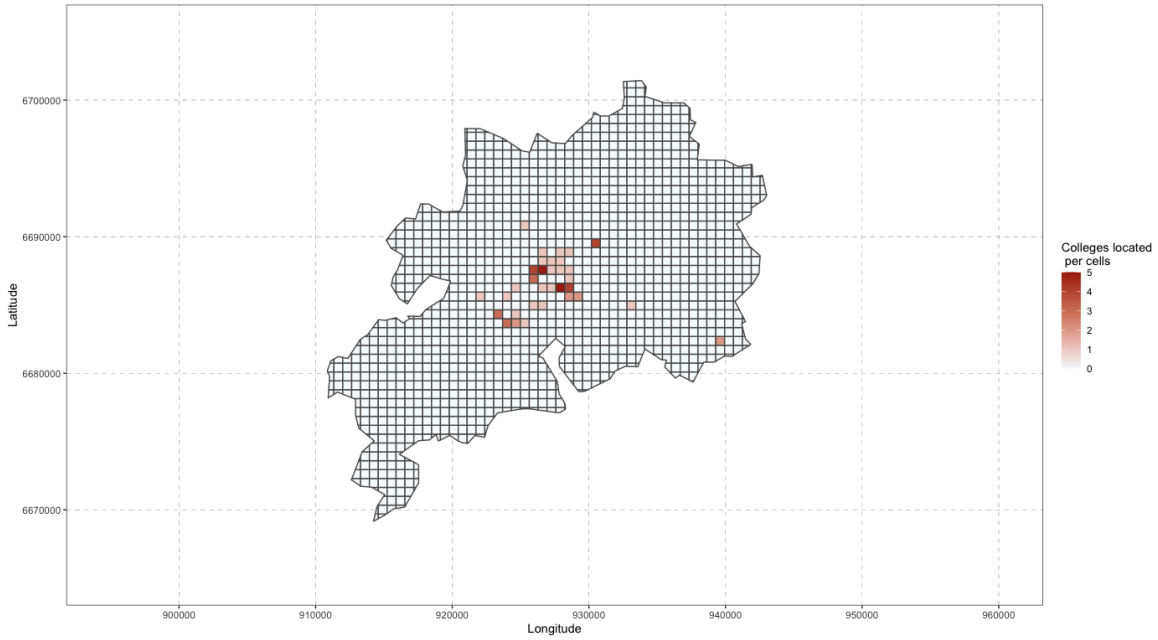

FIG A.5: Plot of *college* variable values, between 0 and 5, per cells of `grid_cagb_sf`.

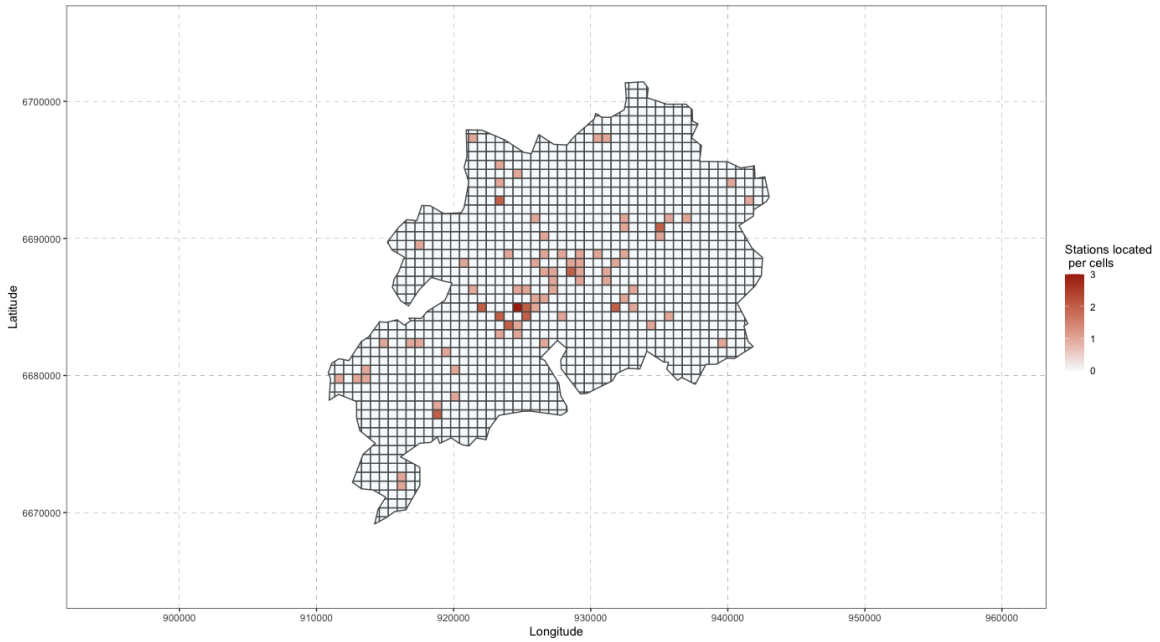

FIG A.6: Plot of *station* variable values, between 0 and 3, per cells of `grid_cagb_sf`.

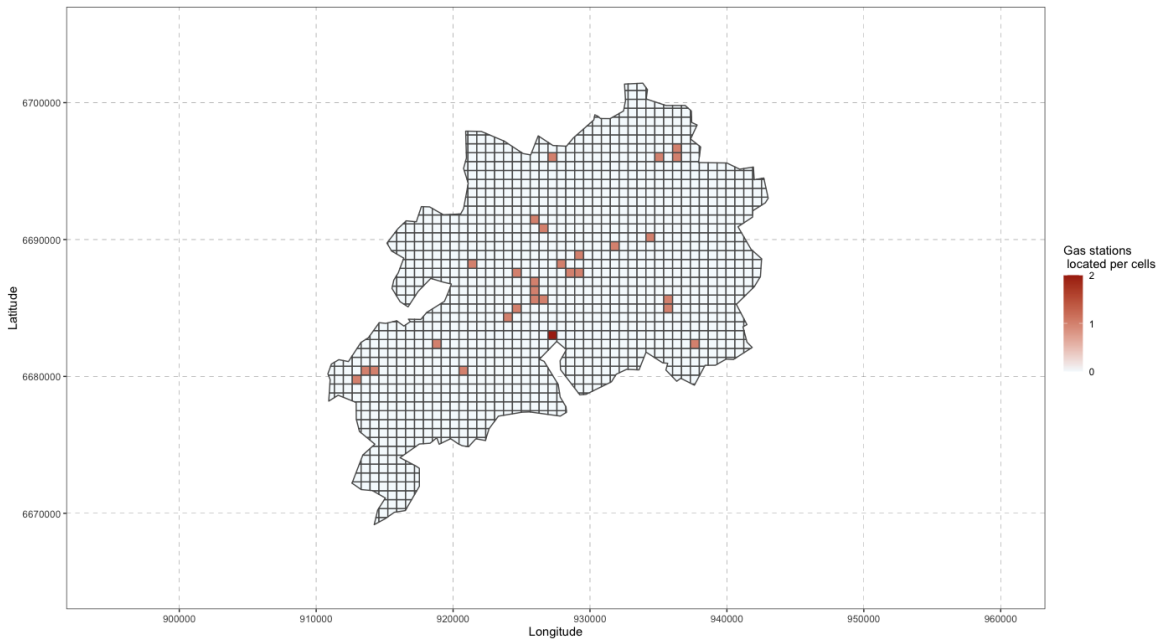

FIG A.7: Plot of *gasoline* variable values, between 0 and 2, per cells of `grid_cagb_sf`.

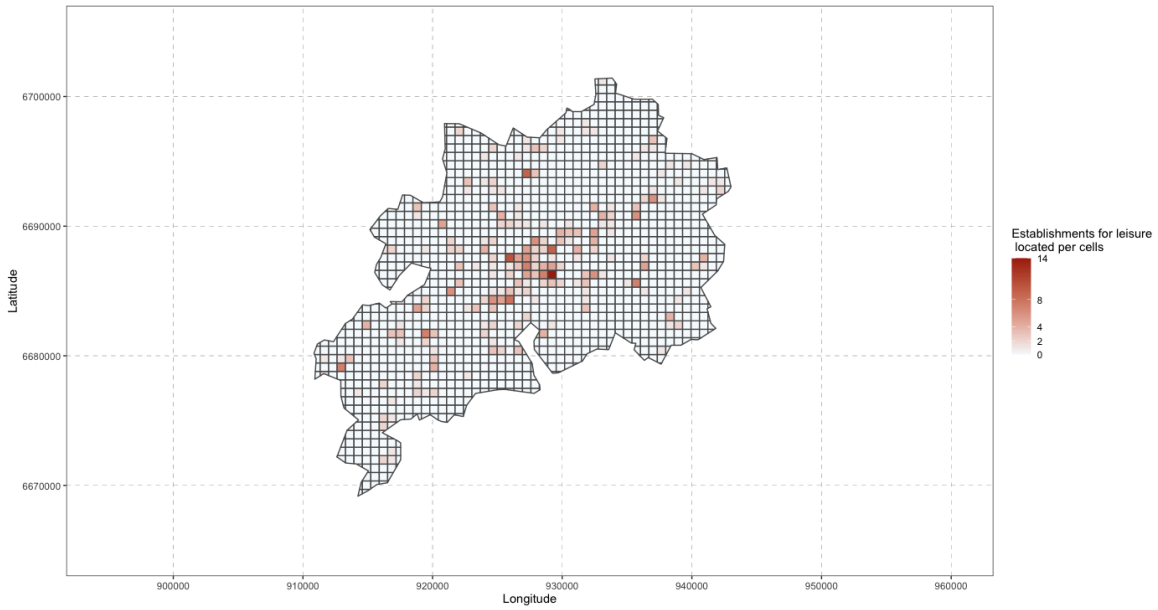

FIG A.8: Plot of *leisure* variable values, between 0 and 14, per cells of `grid_cagb_sf`.

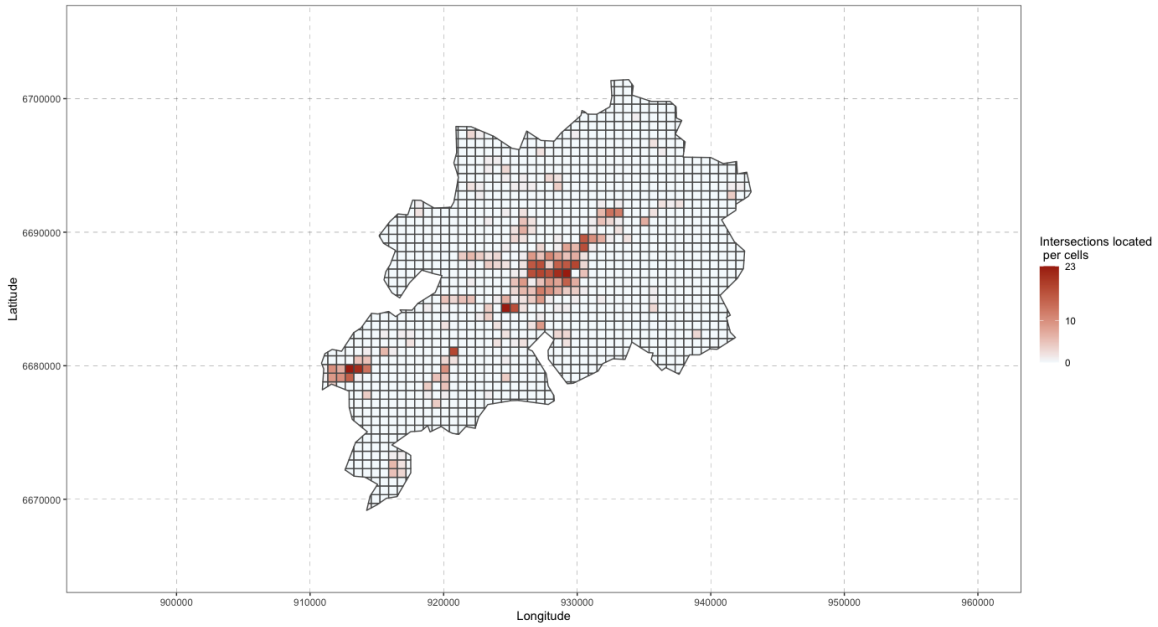

FIG A.9: Plot of *intersection* variable values, between 0 and 23, per cells of `grid_cagb_sf`.

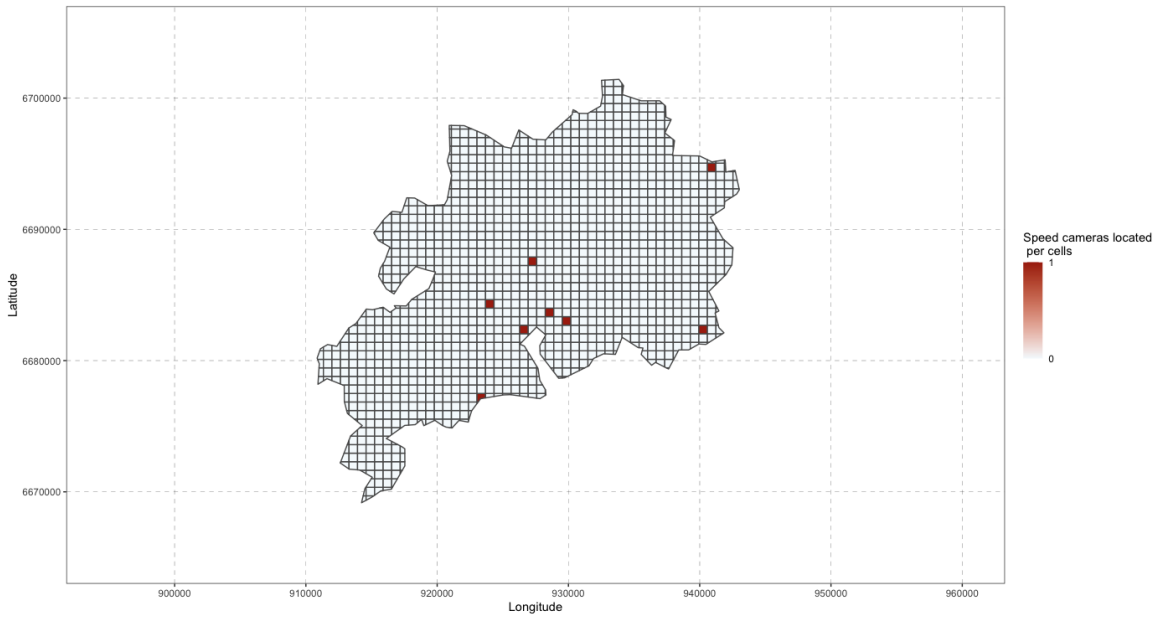

FIG A.10: Plot of *radars* variable values, between 0 and 1, per cells of `grid_cagb_sf`.

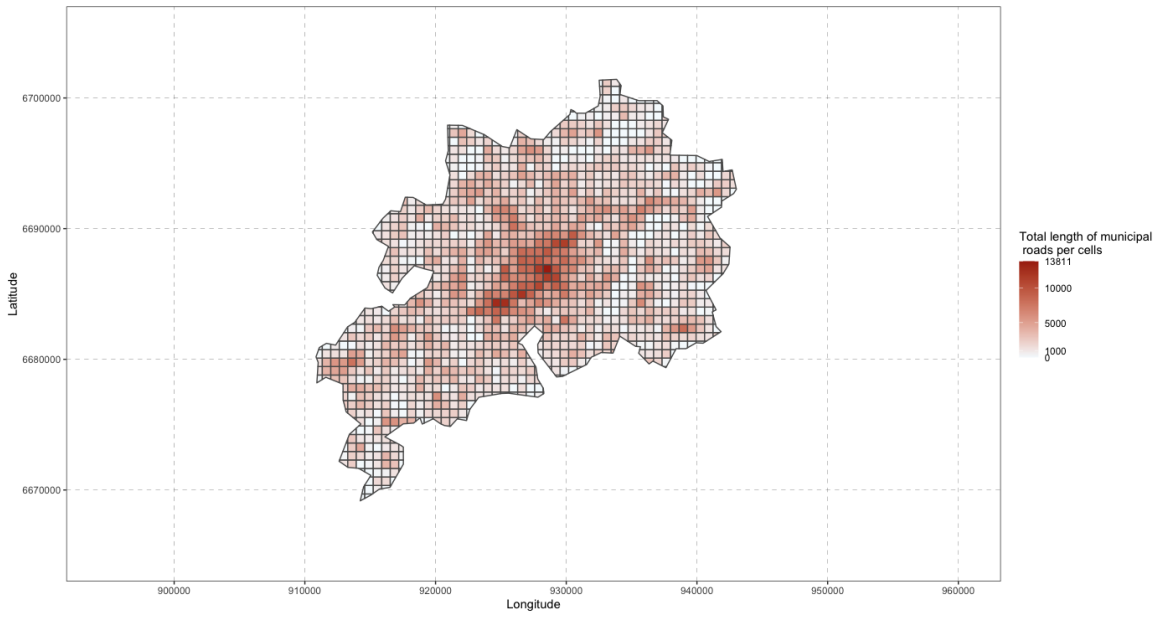

FIG A.11: Plot of *municipal_length* variable values, between 0 and around 13 812, per cells of `grid_cagb_sf`.

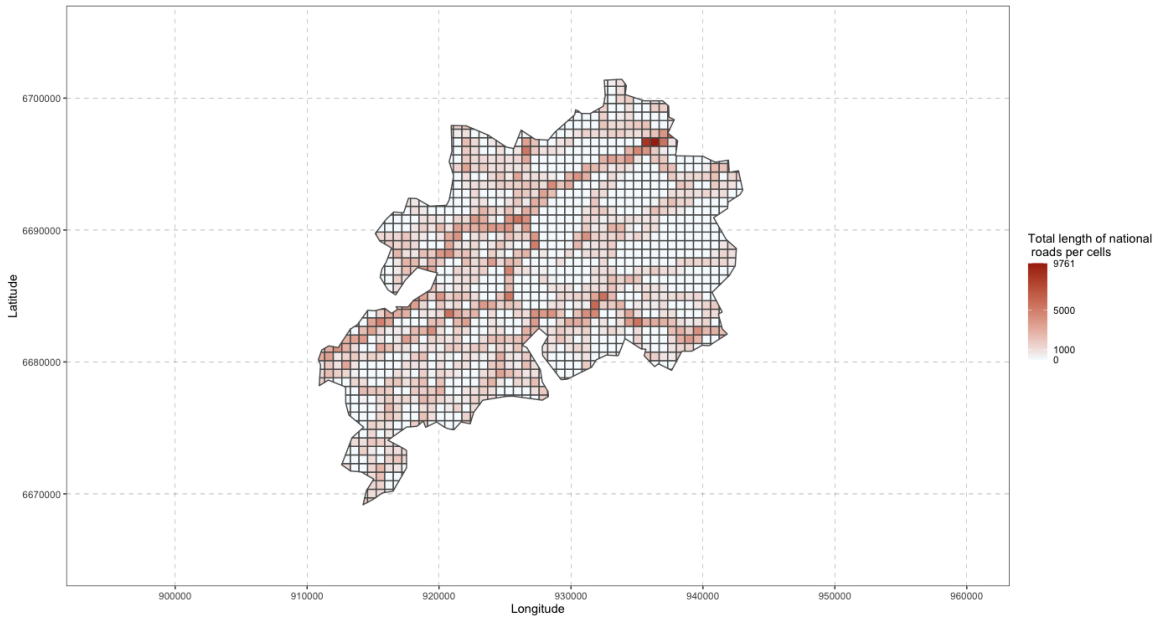

FIG A.12: Plot of *national_length* variable values, between 0 and around 9 762, per cells of `grid_cagb_sf`.

Appendix B

Spatial modelling road accidents in the urban community of Besançon using log-Gaussian Cox processes

This appendix corresponds to the supplementary materials of Chapter 3. In the following, the reader may find a table that summarizes the result of the variable selection methods fitted, variable importance plots for the Poisson models aggregation and Random Forest methods, trace plots and autocorrelation plots of the parameters of the model **LGCP3**.

The variable selection methods have been fitted on various training samples : first a training sample based on the SMOTE method ; second a training sample where the distribution has been preserved ; finally on the whole dataset. The results are given in Tab B.1.

The variable importance plot for the Poisson models aggregation where six, eight, ten and all the covariates have been randomly selected at each iteration is given respectively in Fig B.1, Fig B.2, Fig B.3 and Fig B.4. Note that the variable importance plot for four covariates has been taken in example and has been produced in Fig 3.10. Then, the reader may find the variable importance plot of the Random Forest fitted with eight covariates randomly selected at each iteration in Fig B.5.

Finally, trace plots and autocorrelation plots for the model **LGCP3** are given respectively in Fig B.6 and Fig B.7.

TAB B.1: Mean squared error (MSE) and R-squared values of various statistical models fitted on the road crash data. The statistical models have been fitted with different samples of the data: on a training set based on the SMOTE method (which corresponds simply to `selection_train`), on a training set where the observed distribution has been preserved and finally on the whole dataset (corresponds to no split rule).

Model fitted	**Number of covariates introduced**	**Split method**		**MSE**	**R-squared**
Poisson regression	13	SMOTE	train	1.14	0.70
			test	0.42	0.69
		Distribution preserved	train	0.75	0.50
			test	0.41	0.70
		No split rule		0.61	0.58
Poisson Regression with AIC	7	SMOTE	train	1.19	0.68
			test	0.45	0.67
		Distribution preserved	train	0.75	0.50
			test	0.41	0.70
		No split rule		0.63	0.56
Poisson regressions aggregation	4	SMOTE	train	1.39	0.63
			test	0.58	0.57
		Distribution preserved	train	0.81	0.45
			test	0.40	0.70
		No split rule		0.71	0.51
	6	SMOTE	train	1.16	0.69
			test	0.46	0.66
		Distribution preserved	train	0.83	0.44
			test	0.44	0.67
		No split rule		0.67	0.53
	8	SMOTE	train	1.09	0.71
			test	0.42	0.69
		Distribution preserved	train	0.94	0.36
			test	0.60	0.55
		No split rule		0.76	0.57
	10	SMOTE	train	1.07	0.72
			test	0.41	0.70
		Distribution preserved	train	1.45	0.01
			test	0.84	0.37
		No split rule		0.83	0.43
	13	SMOTE	train	1.18	0.69
			test	0.42	0.69
		Distribution preserved	train	2.40	-0.62
			test	0.91	0.32
		No split rule		1.09	0.25
Random Forest	8	SMOTE	train	0.28	0.92
			test	0.51	0.62
		Distribution preserved	train	0.36	0.76
			test	0.51	0.62
		No split rule		0.31	0.78

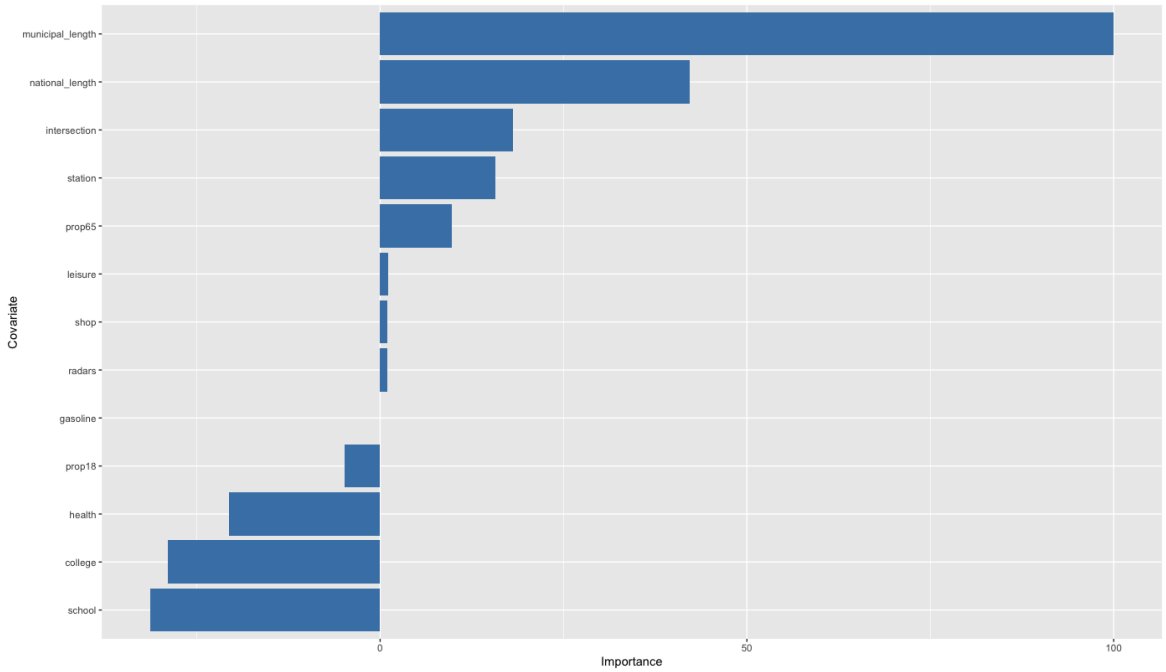

FIG B.1: Variable importance plot for Poisson models aggregation with six covariates randomly included at each iteration.

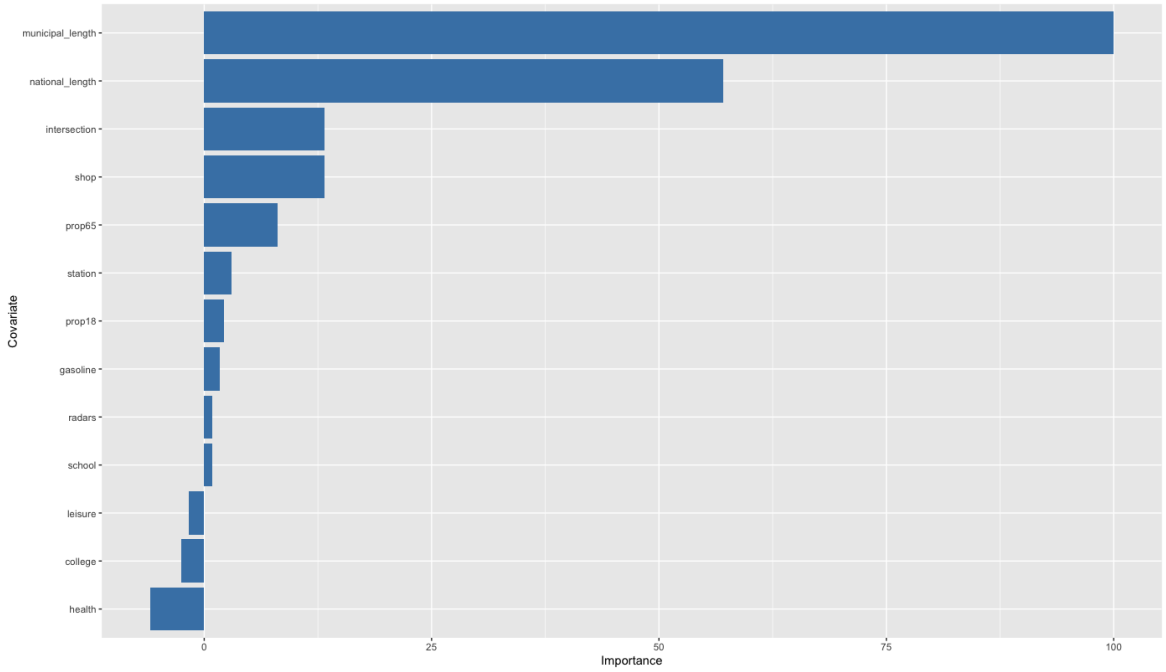

FIG B.2: Variable importance plot for Poisson models aggregation with eight covariates randomly included at each iteration.

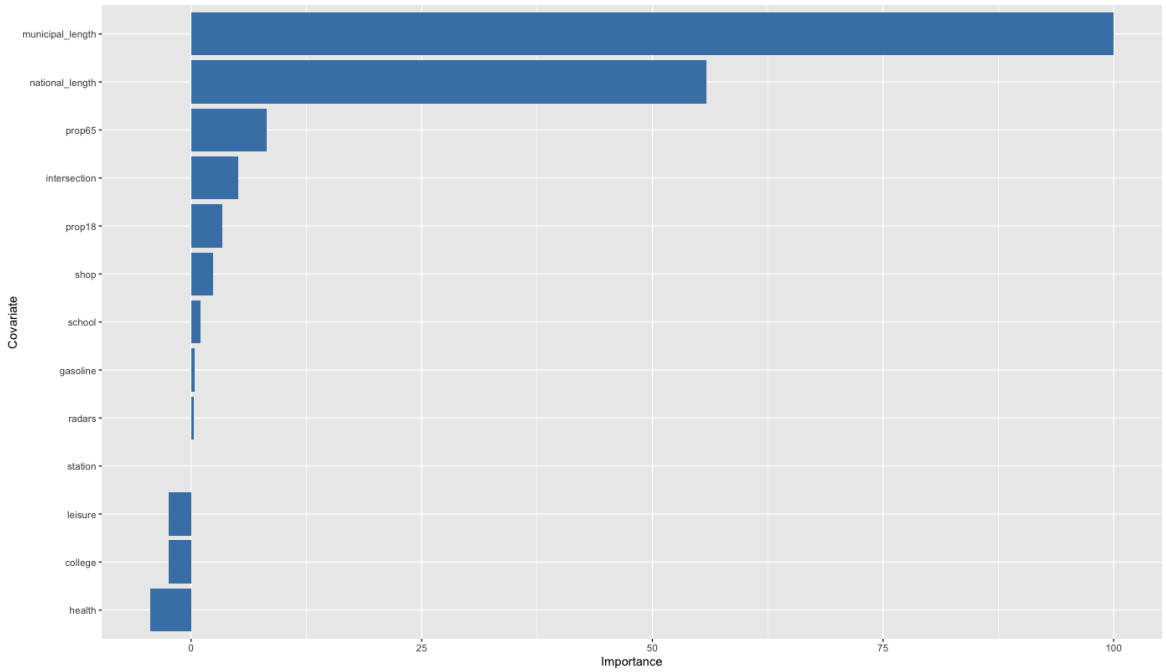

FIG B.3: Variable importance plot for Poisson models aggregation with ten covariates randomly included at each iteration.

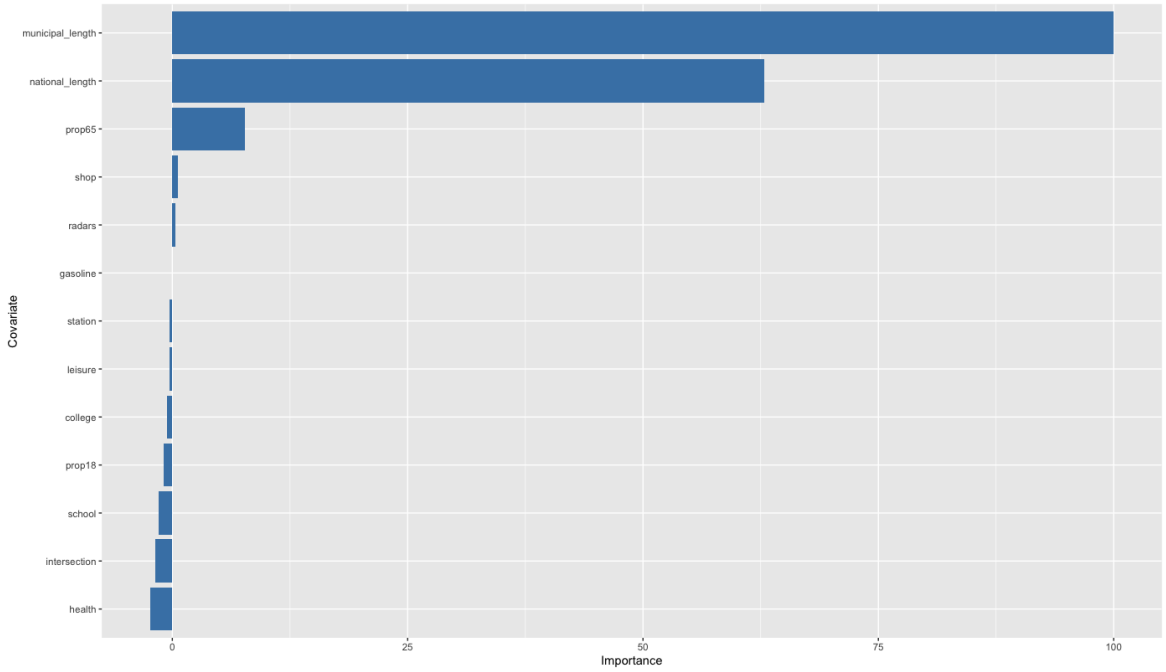

FIG B.4: Variable importance plot for Poisson models aggregation with all the covariates (thirteen) randomly included at each iteration.

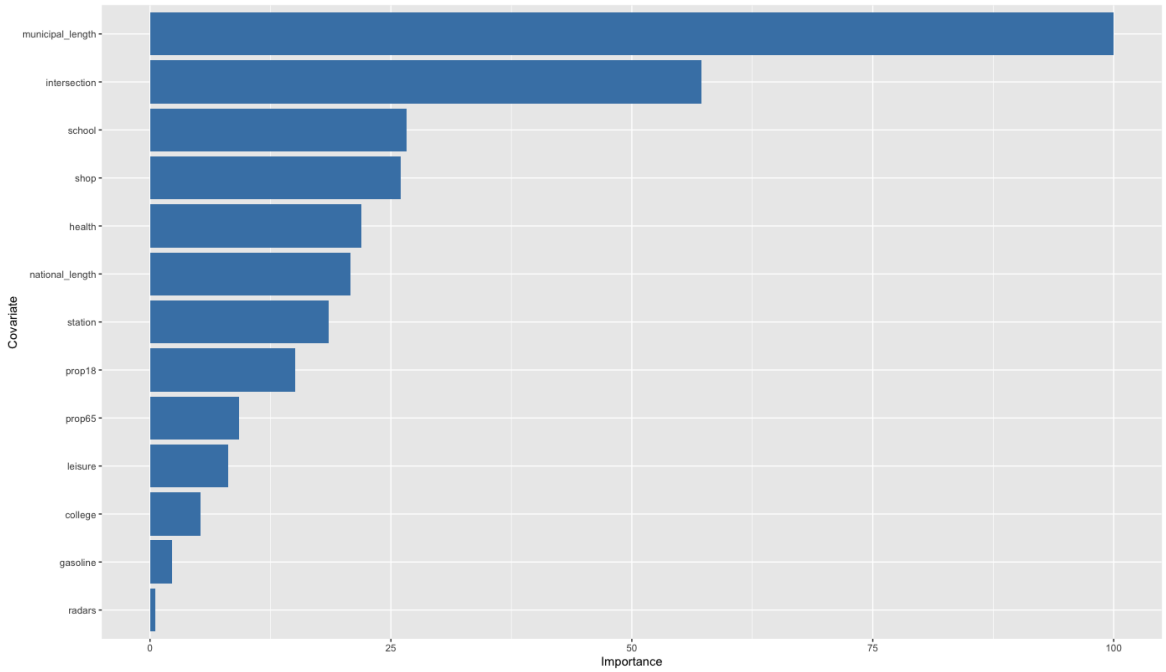

FIG B.5: Variable importance plot for the Random Forest model with eight covariates randomly included at each iteration.

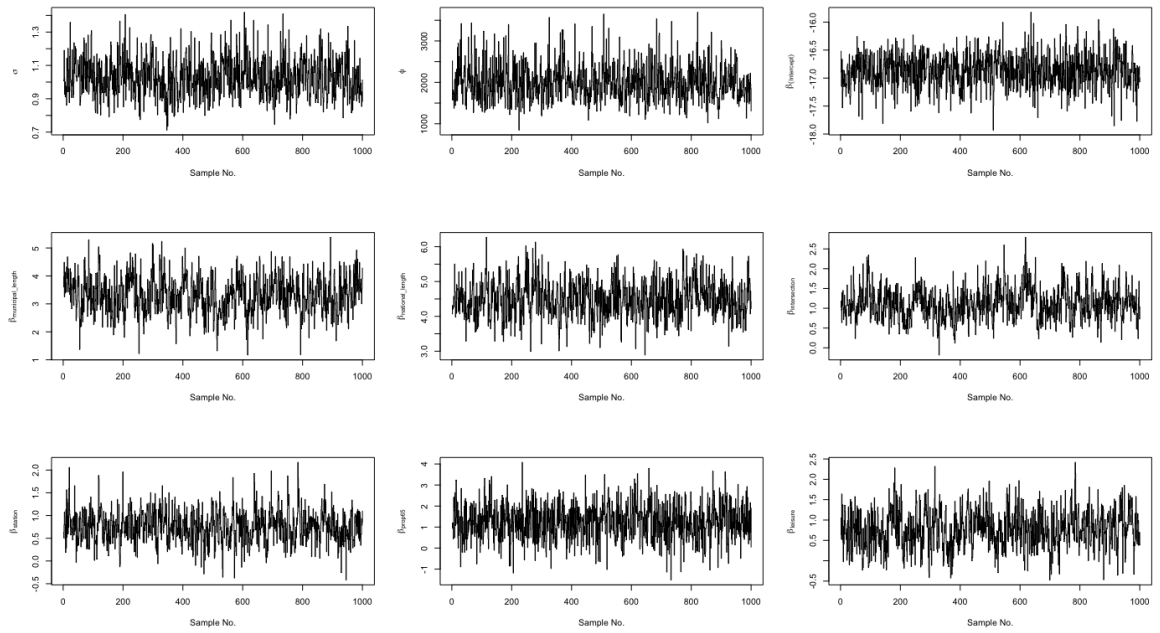

FIG B.6: Trace plots for the parameters σ, ϕ and β from `lgcp3`.

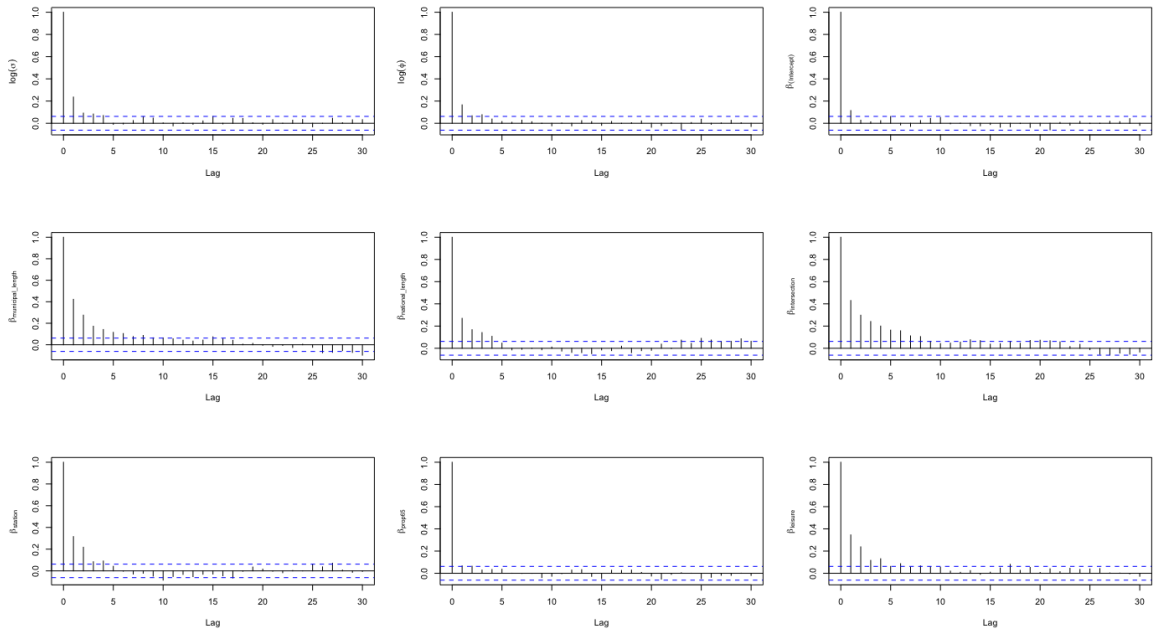

FIG B.7: Autocorrelation plots for the parameters ϕ and β from `lgcp3`.

Appendix C

Road accident space and time preliminary analyses of the city of Besançon

This appendix corresponds to the supplementary materials of Chapter 4. In the following, the reader may find the plot of the semivariogram analysis of traffic density data of the year 2019 and tables that summarize the results of the semiparametric space and time analyses fitted on the road crash data of the city of Besançon.

The empirical semivariogram and the Gaussian semivariogram fit of traffic density data of the year 2019 given in Fig C.1.

The results of the space and time semiparametric models fitted on the datasets `bes_c_1` and `bes_c_4`, that respectively corresponds to the dataset of cases and controls from which the matching ratio is 1:1 and 1:4, are respextively given in Tab C.1 and C.2.

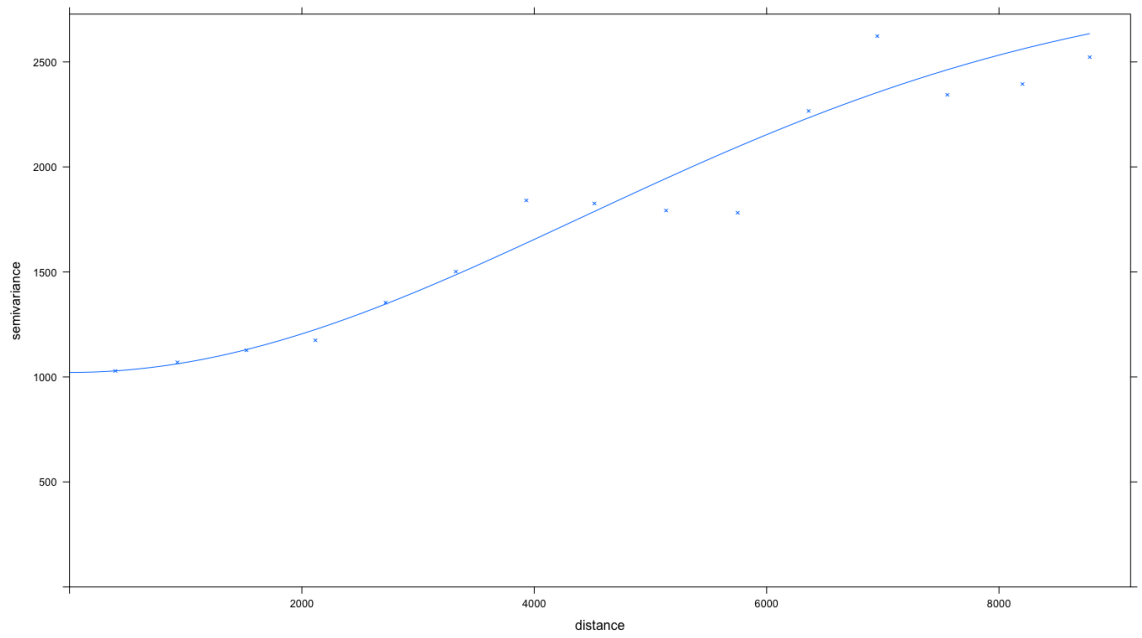

FIG C.1: Fit of Gaussian semivariogram model (solid line) and empirical semivariogram (crosses) of traffic density data of 2019. Arguments of `fit.variogram` : `model = "Gau"`, `psill = 3200`, `range = 2200` and `nugget = 1000`.

TAB C.1: Results of GAMs fitted on the dataset `bes_c_1`. The item * means that the p-value of the nullity coefficient test is less than 0.05.

Environment covariate included	Estimate	p-value		Smoothing term included	p-value	
				te(X, Y, quarter)	1.29e-04	*
				te(X, Y, month)	2.48e-06	*
				te(X, Y, day)	2.16e-06	*
	-0.19	0.92		te(X, Y, quarter)	1.73e-04	*
prop18	-0.68	0.78		te(X, Y, month)	0.07	
	0.57	0.79		te(X, Y, day)	1.33e-04	*
	0.96	0.31		te(X, Y, quarter)	7.16e-04	*
prop65	1.01	0.41		te(X, Y, month)	0.12	
	1.03	0.33		te(X, Y, day)	2.87e-04	*
	1.95	0.24		te(X, Y, quarter)	6.58e-04	*
health	0.69	0.44		te(X, Y, month)	0.03	*
	2.35	0.19		te(X, Y, day)	3.12e-04	*
	1.95	0.24		te(X, Y, quarter)	6.58e-04	*
school	1.68	0.27		te(X, Y, month)	0.08	
	2.13	0.17		te(X, Y, day)	1.52e-04	*
	-2.25	0.32		te(X, Y, quarter)	1.39e-04	*
college	-1.67	0.51		te(X, Y, month)	0.07	
	-1.01	0.66		te(X, Y, day)	1.24e-04	*
	4.75	0.04	*	te(X, Y, quarter)	8.58e-04	*
shop	5.37	0.04	*	te(X, Y, month)	0.11	
	5.69	0.02	*	te(X, Y, day)	3.64e-04	*
	2.95	0.12		te(X, Y, quarter)	1.03e-04	*
station	1.89	0.33		te(X, Y, month)	0.07	
	2.89	0.14		te(X, Y, day)	1.09e-04	*
	0.92	0.20		te(X, Y, quarter)	1.59e-04	*
gasoline	1.45	0.09		te(X, Y, month)	0.06	
	1.19	0.12		te(X, Y, day)	7.3e-05	*
	-1.14	0.37		te(X, Y, quarter)	1.21e-04	*
leisure	-1.28	0.37		te(X, Y, month)	0.06	
	-0.67	0.62		te(X, Y, day)	1.05e-04	*
	1.63	3.28e-03	*	te(X, Y, quarter)	2.04e-03	*
intersection	1.44	0.02	*	te(X, Y, month)	0.13	
	1.71	3.13e-03	*	te(X, Y, day)	6.70e-04	*
	0.39	0.74		te(X, Y, quarter)	1.54e-04	*
radars	0.22	0.89		te(X, Y, month)	0.06	
	0.63	0.62		te(X, Y, day)	1.19e-04	*
	-0.11	0.86		te(X, Y, quarter)	1.51e-04	*
municipal length	0.07	0.93		te(X, Y, month)	0.06	
	-0.06	0.93		te(X, Y, day)	1.24e-04	*
	1.90	4.53e-04	*	te(X, Y, quarter)	2.14e-05	*
national length	1.32	0.04	*	te(X, Y, month)	0.04	*
	1.30	0.03	*	te(X, Y, day)	6.14e-05	*

TAB C.2: Results of GAMs fitted on the dataset `bes_c_4`. The item * means that the p-value of the nullity coefficient test is less than 0.05.

Environment covariate included	Estimate	p-value		Smoothing term included	p-value	
				te(X, Y, quarter)	1.28e-06	*
				te(X, Y, month)	3.19e-03	*
				te(X, Y, day)	2.16e-06	*
	0.82	0.57		te(X, Y, quarter)	2.92e-06	*
prop18	-0.01	0.99		te(X, Y, month)	0.02	*
	0.56	0.73		te(X, Y, day)	4.09e-06	*
	-0.49	0.48		te(X, Y, quarter)	3.58e-06	*
prop65	-0.81	0.30		te(X, Y, month)	0.03	*
	-0.81	0.28		te(X, Y, day)	3.70e-06	*
	1.18	0.16		te(X, Y, quarter)	2.92e-06	*
health	0.69	0.44		te(X, Y, month)	0.03	*
	0.65	0.45		te(X, Y, day)	4.69e-06	*
	0.12	0.85		te(X, Y, quarter)	1.45e-06	*
school	-0.11	0.86		te(X, Y, month)	0.02	*
	-0.25	0.70		te(X, Y, day)	2.31e-06	*
	0.01	0.99		te(X, Y, quarter)	1.28e-06	*
college	-0.17	0.93		te(X, Y, month)	0.02	*
	-0.09	0.96		te(X, Y, day)	2.14e-06	*
	3.35	0.01	*	te(X, Y, quarter)	4.67e-04	*
shop	2.66	0.04	*	te(X, Y, month)	0.08	
	2.48	0.07		te(X, Y, day)	3.47e-05	*
	1.75	0.09		te(X, Y, quarter)	1.53e-06	*
station	1.40	0.19		te(X, Y, month)	0.02	*
	1.89	0.08		te(X, Y, day)	2.21e-06	*
	1.18	0.02	*	te(X, Y, quarter)	1.38e-06	*
gasoline	1.12	0.04	*	te(X, Y, month)	0.02	*
	0.96	0.07		te(X, Y, day)	3.16e-06	*
	-1.46	0.19		te(X, Y, quarter)	2.83e-06	*
leisure	-1.72	0.13		te(X, Y, month)	0.02	*
	-1.74	0.13		te(X, Y, day)	1.70e-06	*
	1.73	1.15e-06	*	te(X, Y, quarter)	6.77e-05	*
intersection	1.58	1.70e-05	*	te(X, Y, month)	0.06	
	1.57	1.44e-05	*	te(X, Y, day)	3.78e-05	*
	1.14	0.22		te(X, Y, quarter)	2.19e-06	*
radars	1.49	0.14		te(X, Y, month)	0.02	*
	1.29	0.17		te(X, Y, day)	2.76e-06	*
	-0.18	0.71		te(X, Y, quarter)	1.29e-06	*
municipal length	-0.18	0.72		te(X, Y, month)	0.02	*
	-0.22	0.65		te(X, Y, day)	2.01e-06	*
	1.93	1.74e-07	*	te(X, Y, quarter)	<2e-16	*
national length	1.70	2.78e-05	*	te(X, Y, month)	4.84e-03	*
	1.71	1.73e-05	*	te(X, Y, day)	1.16e-06	*

Bibliography

Abdel-Aty, M. A. and Abdelwahab, H. T. (2000). Exploring the relationship between alcohol and the driver characteristics in motor vehicle accidents. *Accident Analysis and Prevention*, pages 473–482.

Abdel-Aty, M. A., Chen, C., and Schott, J. R. (1998). An assessment of the effect of driver age on traffic accident involvement using log-linear models. *Accident Analysis and Prevention*, 30(6):851–861.

Agresti, A. (1990). *Categorical Data Analysis*. Wiley.

Agresti, A. (2002). *Categorical Data Analysis*. Wiley, second edition.

Agresti, A. (2013). *Categorical Data Analysis*. Wiley-Blackwell, 3nd edition.

Andersen, E. (1980). *Discrete Statistical Models with Social Science Applications*. North-Holland.

Baddeley, A., Nair, G., Rakshit, S., McSwiggan, G., and Davies, T. M. (2021a). Analysing point patterns on networks — a review. *Spatial Statistics*, 42:100435.

Baddeley, A., Rubak, E., and Turner, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. Chapman & Hall/CRC.

Baddeley, A., Turner, R., and Rubak, E. (2021b). *spatstat: Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests*. R package version 2.3-0.

Bao, J., Liu, P., and Ukkusuri, S. V. (2019). A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data. *Accident Analysis & Prevention*, 122:239–254.

Bayisa, F. L., Ådahl, M., Rydén, P., and Cronie, O. (2020). Large-scale modelling and forecasting of ambulance calls in northern sweden using spatio-temporal log-gaussian cox processes. *Spatial Statistics*, 39:100471.

Benzécri, J.-P. (1982). *Histoire et Préhistoire de l'Analyse des Données*. Paris: Dunod.

Benzécri, J.-P. e. a. (1973). *L'Analyse des Données: L'Analyse des Correspondences*. Paris: Dunod.

Bièvre, D. (2017). Acteurs de contingence et insécurité routière: les limites de la statistique descriptive. Technical report.

Bivand, R., Keitt, T., Rowlingson, B., Pebesma, E., Sumner, M., Hijmans, R., Baston, D., Rouault, E., Warmerdam, F., Ooms, J., and Rundel, C. (2021a). *rgdal: Bindings for the 'Geospatial' Data Abstraction Library*. R package version 1.5-28.

Bivand, R., Lewin-Koh, N., Pebesma, E., Archer, E., Baddeley, A., Bearman, N., Bibiko, H., Brey, S., Callahan, J., Carrillo, G., Dray, S., Forrest, D., Friendly, M., Giraudoux, P., Golicher, D., Gómez Rubio, V., Hausmann, P., Ove Hufthammer, K., Jagger, T., Johnson, K., Lewis, M., Luque, S., MacQueen, D., Niccolai, A., Perpiñán Lamigueiro, O., Plunkett, E., Rubak, E., Short, T., Snow, G., Stabler, B., Stokely, M., and Turner, R. (2021b). *maptools: Tools for Handling Spatial Objects*. R package version 1.1-2.

Bivand, R., Rundel, C., Pebesma, E., Stuetz, R., Ove Hufthammer, K., Giraudoux, P., Davis, M., and Santilli, S. (2021c). *rgeos: Interface to Geometry Engine - Open Source ('GEOS')*. R package version 0.5-9.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Brix, A. and Diggle, P. J. (2001). Spatiotemporal prediction for log-gaussian cox processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(4):823–841.

Brooks, S., Gelman, A., Jones, G., and Meng, X. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press.

Chawla, N. V., Bowyer, K. W., and Kegelmeyer, W. P. (2002). Smote : Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Cheng, J., Karambelkar, B., Xie, Y., Wickham, H., Russell, K., Johnson, K., Schloerke, B., jQuery Foundation, Agafonkin, V., CloudMade, contributors, L., Copeland, B., Dietrich, J., Becquet, B., AS, N., Voogdt, L., Montague, D., AB, K., Kajic, R., Mapbox, and Bostock, M. (2021). *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. R package version 2.0.4.1.

Christensen, R. H. B. (2019). ordinal: Regression models for ordinal data. R package version 2019.12-10. https://CRAN.R-project.org/package=ordinal.

Cressie, N. (1993). *Statistics for Spatial Data*. John Wiley & Sons, revised edition.

Cressie, N. and Wikle, C. K. (2011). *Statistics for Spatio-Temporal Data*. John Wiley & Sons.

Das, S., Avelar, R., Dixon, K., and Sun, X. (2018). Investigation on the wrong way driving crash patterns using multiple correspondence analysis. *Accident Analysis and Prevention*, 11:43–55.

Das, S. and Sun, X. (2015). Factor association with multiple correspondence analysis in vehicle-pedestrian crashes. *Trasportation Research Record: Journal of the Transportation Research Board*, 2519:95–103.

Das, S. and Sun, X. (2016). Association knowledge for fatal run-off-road crashes by multiple correspondence analysis. *IATSS Research*, 39:146–155.

Davies, T. M. and Bryant, D. J. (2013). On circulant embedding for gaussian random fields in r. *Journal of Statistical Software*, 55(9).

Dekoninck, L. and Severijnen, M. (2022). Correlating traffic data, spectral noise and air pollution measurements: Retrospective analysis of simultaneous measurements near a highway in the netherlands. *Atmosphere*, 13(5).

Diggle, P. J. (2013). *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. Chapman & Hall/CRC, third edition.

Diggle, P. J., Moraga, P., Rowlingson, B., and Taylor, B. M. (2013). Spatial and spatio-temporal log-gaussian cox processes: Extending the geostatistical paradigm. *Statistical Science*, 28(4):105848.

Do, V. H., Vanhems, A., and Laurent, T. (2021). Guidelines on areal interpolation methods. In Daouia, A. and Ruiz-Gazen, A., editors, *Advances in Contemporary Statistics and Econometrics*, chapter 20, pages 385–408. Springer.

Dong, N., Huang, H., and Zheng, L. (2015). Support vector machine in crash prediction at the level of traffic analysis zones: assessing the spatial proximity effects. *Accident Analysis & Prevention*, 82:192–198.

Eme-Ziri, C. (2022). Besançon : l'interdiction du pont de la république aux voitures suscite inquiétude et polémique. https://france3-regions.francetvinfo.fr/bourgogne-franche-comte/doubs/besancon/.

Escofier, B. and Pagès, J. (2008). *Analyses factorielles simples et multiples*. Dunod, 4ème edition.

Feng, C. (2022). Spatial-temporal generalized additive model for modeling COVID-19 mortality risk in Toronto, Canada. *Spatial Statistics*, 49:100526.

Fort, E., Gadegbeku, B., Gat, E., Pelissier, C., Hours, M., and Charbotel, B. (2019). Working conditions and risk exposure of employees whose occupations require driving on public roads – factorial analysis and classification. *Accident Analysis and Prevention*, 131:254–267.

Gamerman, D. and Lopes, H. F. (2006). *Markov Chain Monte Carlo - Stochastic Simulation for Bayesian Inference*. Chapman & Hall/CRC, second edition.

Genuer, R. and Poggi, J. (2019). *Les forêts aléatoires avec R*. Presses universitaires de Rennes.

Goodman, L. A. (1986). Some useful extensions of the usual correspondence analysis approach and the usual log-linear models approach in the analysis of contingency tables. *International Statistical Review*, 54:243–309.

Greenacre, M. (1989). The carroll-green-schaffer scaling in correspondence analysis: a theoretical and empirical appraisal. *Journal of Marketing Research*, 26:358–365.

Greenacre, M. (2006). From simple to multiple correspondence analysis. In Greenacre, M. and Blasius, J., editors, *Multiple Correspondence Analysis and Related Methods*. Chapman & Hall/CRC.

Greenacre, M. and Blasius, J. (2006). *Multiple Correspondence Analysis and Related Methods*. Chapman & Hall/CRC.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, 1(3):297–318.

Hijmans, R. J., van Etten, J., Sumner, M., Cheng, J., Baston, D., Bevan, A., Bivand, R., Busetto, L., Canty, M., Fasoli, B., Forrest, D., Ghosh, A., Golicher, D., Gray, J., Greenberg, J. A., Hiemstra, P., Hingee, K., Ilich, A., Karney, C., Mattiuzzi, M., Mosher, S., Naimi, B., Nowosad, J., Pebesma, E., Perpinan Lamigueiro, O., Racine, E. B., Rowlingson, B., Shortridge, A., Venables, B., and Wueest, R. (2022). *raster: Geographic Data Analysis and Modeling*. R package version 3.5-15.

Hothorn, T. and Everitt, B. S. (2014). *A Handbook of Statistical Analyses Using R*. CRC Press, third edition.

Husson, F., Lê, S., and Pagès, J. (2016). *Analyse des données avec R*. Presses Universitaires de Rennes, 2ème edition.

INSEE (2018). Recensement de la population - populations légales en vigueur à compter du 1er janvier 2018.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.

Kassambara, A. and Mundt, F. (2019). *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. R package version 1.0.6.

Lê, S., Josse, J., and Husson, F. (2008). FactoMineR: A package for multivariate analysis. *Journal of Statistical Software*, 25(1):1–18.

Le Roux, B. and Rouanet, H. (2004). *Geometric Data Analysis: From Correspondence Analysis to Structured Data*. Dordrecth: Kluwer.

Li, P., Abdel-Aty, M., and Yuan, J. (2020). Real-time crash risk prediction on arterials based on lstm-cnn. *Accident Analysis & Prevention*, 135:105371.

Loader, C. (1999). *Local Regression and Likelihood*. Springer.

Ma, X., Tao, Z., Wang, Y., Yu, H., and Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman & Hall/CRC, second edition.

Mekonnen, B. (2018). Risk factors of road traffic accidents and its severity in north shewa zone, amhara region, ethiopia. *American Journal of Theoretical and Applied Statistics*, 7(4):163–166.

Menardi, G. and Torelli, N. (2018). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1):92–122.

Mohler, G. (2014). Marked point process hotspot maps for homicide and gun crime prediction in chicago. *International Journal of Forecasting*, 30(3):491–497.

Moller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log gaussian cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482.

Murotani, T., Igarashi, S., and Koto, H. (2019). Distribution analysis and modeling of air voids in concrete as spatial point processes. *Cement and Concrete Research*, 115:124–132.

Nychka, D., Furrer, R., Paige, J., Sain, S., Gerber, F., and Iverson, M. (2021). *fields: Tools for Spatial Data*. R package version 13.3.

Okabe, A., Satoh, T., and Sugihara, K. (2009). A kernel density estimation method for networks, its computational method and a gis-based tool. *International Journal of Geographical Information Science*, 23(1):7–32.

Olsbo, V., Myllymäki, M., Waller, L. A., and Särkkä, A. (2013). Development and evaluation of spatial point process models for epidermal nerve fibers. *Mathematical Biosciences*, 243(2):178–189.

ONISR (2019). *La sécurité routière en France – Bilan de l'accidentalité de l'année 2018*. ONISR, Paris.

Opitz, T., Bonneu, F., and Gabriel, E. (2020). Point-process based bayesian modeling of space–time structures of forest fire occurrences in mediterranean france. *Spatial Statistics*, 40:100429.

Papagiotakos, D. B. and Pitsavos, C. (2004). Interpretation of epidemiological data using multiple correspondence analysis and log-linear models. *Journal of Data Science*, 2:75–86.

Park, H., Haghani, A., Samuel, S., and Knodler, M. A. (2018). Real-time prediction and avoidance of secondary crashes under unexpected traffic congestion. *Accident Analysis & Prevention*, 112:39–49.

Pebesma, E., Bivand, R., Racine, E., Sumner, M., Cook, I., Keitt, T., Lovelace, R., Wickham, H., Ooms, J., Müller, K., Lin Pedersen, T., Baston, D., and Dunnington, D. (2022). *sf: Simple Features for R*. R package version 1.0-6.

Pebesma, E., Bivand, R., Rowlingson, B., Gomez-Rubio, V., Hijmans, R., Sumner, M., MacQueen, D., Lemon, J., Lindgren, F., O'Brien, J., and O'Rourke, J. (2021). *sp: Classes and Methods for Spatial Data*. R package version 1.4-6.

Pebesma, E. and Graeler, B. (2022). *gstat: Spatial and Spatio-Temporal Geostatistical Modelling, Prediction and Simulation*. R package version 2.0-9.

Prener, C., Revord, C., and Fox, B. (2022). *areal: Areal Weighted Interpolation*. R package version 0.1.8.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rakshit, S., Davies, T., Moradi, M. M., McSwiggan, G., Gopalan, N., Mateu, J., and Baddeley, A. (2019). Fast kernel smoothing of point patterns on a large network using two-dimensional convolution. *International Statistical Review*, 87(3):531–556.

Ramírez, A. F. and Valencia, C. (2021). Spatiotemporal correlation study of traffic accidents with fatalities and injuries in bogota (colombia). *Accident Analysis & Prevention*, 149:105848.

Rezapour, M. and Ksaibati, K. (2018). Application of multinomial and ordinal logistic regression to model injury severity of truck crashes, using violation and crash data. *Springer*.

Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society B*, 39(2):172–192.

Robert, C. (1996). *Méthodes de Monte Carlo par chaînes de Markov*. Economica.

Robert, C. P. (2007). *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer, second edition.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer.

Setia, M. (2016). Methodology series module 2: Case-control studies. *Indian Journal of Dermatology*, 61(2):146.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall.

Spinu, V., Grolemund, G., Wickham, H., Vaughan, D., Lyttle, I., Costigan, I., Law, J., Mitarotonda, D., Larmarange, J., Boiser, J., and Lee, C. H. (2021). *lubridate: Make Dealing with Dates a Little Easier*. R package version 1.8.0.

Spychala, C., Armand, J., Dombry, C., and Goga, C. (2021). Multivariate statistical analysis for exploring road crash-related factors in the Franche-Comté region of France. *Communications in Statistics: Case Studies, Data Analysis and Applications*, 7(3):442–474.

Tang, D., Yang, X., and Wang, X. (2020). Improving the transferability of the crash prediction model using the TrAdaBoost.R2 algorithm. *Accident Analysis & Prevention*, 141:105551.

Taylor, B. (2021). *miscFuncs: Miscellaneous Useful Functions Including LaTeX Tables, Kalman Filtering and Development Tools*. R package version 1.5-2.

Taylor, B. M., Davies, T. M., Rowlingson, B. S., and Diggle, P. J. (2013). lgcp: An r package for inference with spatial and spatio-temporal log-gaussian cox processes. *Journal of Statistical Software*, 52(4).

Taylor, B. M., Davies, T. M., Rowlingson, B. S., and Diggle, P. J. (2015). Bayesian inference and data augmentation schemes for spatial, spatiotemporal and multivariate log-gaussian cox processes in r. *Journal of Statistical Software*, 63(7).

Taylor, B. M., Davies, T. M., Rowlingson, B. S., Diggle, P. J., Pebesma, E., and Schumacher, D. (2021). *lgcp: Log-Gaussian Cox Process*. R package version 1.7.

Taylor, B. M. and Diggle, P. J. (2013). Inla or mcmc ? a tutorial and comparative evaluation for spatial prediction in log-gaussian cox processes. *Journal of Statistical Computation and Simulation*, 84(10):2266–2284.

Tovo, A. and Favretti, M. (2018). The distance decay of similarity in tropical rainforests. a spatial point processes analytical formulation. *Theoretical Population Biology*, 120:78–89.

Van der Heijden, P., De Falguerolles, A., and De Leeuw, J. (1989). A combined approach to contingency table analysis using correspondence analysis and log-linear analysis. *Applied Statistics*, 38:249–292.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

Wang, X. and Brown, D. E. (2012). The spatio-temporal modeling for criminal incidents. *Security Informatics*, 1(1).

Wickham, H. (2021). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.3.1.

Wickham, H., Chang, W., Henry, L., Lin Pedersen, T., Takahashi, K., Wilke, C., Woo, K., Yutani, H., and Dunnington, D. (2021). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 3.3.5.

Wood, S. (2022). *mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation*. R package version 1.8-40.

Wood, S. N. (2017). *Generalized Additive Models - An introduction with R.* Chapman & Hall/CRC, second edition.

Yannis, G., Golias, J., and Papadimitriou, E. (2005). Driver age and vehicle engine size effects on fault and severity in young motorcyclists accidents. *Accident Analysis and Prevention*, pages 327–333.

Zammit-Mangion, A. and Sainsbury-Dale, M. (2022). *FRK: Fixed Rank Kriging.* R package version 2.0.3.

Résumé. Dans cette thèse, nous nous intéressons à l'analyse statistique des données d'accidents routiers dans la région Franche-Comté. Plusieurs problématiques sont étudiées selon que l'on adopte un point de vue rétrospectif (analyse des facteurs de risques des accidents déjà produits) ou prédictif (position spatiale et période temporelle les plus à risque). Pour se faire, nous avons employé des méthodes de prédiction plutôt classiques d'apprentissage statistique et des outils de prédiction spatiale et spatio-temporelle. La première partie de la thèse est centrée sur l'analyse statistique de la gravité des accidents. De nombreux facteurs (tels que la consommation d'alcool et/ou drogues, le département, le moment de la journée) sont mis en relation avec la gravité de l'accident via des approches non-supervisées ou supervisées d'apprentissage statistique. La dépendance entre ces éléments est modélisée par des modèles log-linéaires proposés suite à une analyse préliminaire de correspondances multiples et également par une régression logistique ordinale. Ces deux modélisations permettent de quantifier l'effet des différents facteurs de risques sur la gravité des accidents. La deuxième partie de la thèse porte sur la prédiction spatiale ou spatio-temporelle de la survenue d'un accident. Les données d'accidents sont modélisées par des processus de Cox log-Gaussiens (LGCP) basés sur les coordonnées géolocalisées des accidents ainsi que sur des covariables socio-démographiques et d'infra-structures routières afin d'identifier les zones géographiques les plus critiques. Nous proposons également une méthode de sélection de variables basée sur une aggrégation de modèles de Poisson et un critère d'importance de variables pour sélectionner les variables les plus importantes à incorporer dans le modèle LGCP. Le meilleur modèle est utilisé pour déterminer les zones les plus risquées. Dans un deuxième temps, la composante temporelle est ajoutée à l'étude et une modélisation semi-paramétrique par modèle additif généralisé est proposée afin d'identifier les zones et périodes critiques. La modélisation s'inspire des études cas-contrôles épidémiologiques et prend en compte la structure du réseau routier ainsi que des données spatio-temporelles de trafic.

Mots-clés: analyse des correspondances multiples; interpolation de données surfaciques; krigeage; modèle généralisé additif; processus de Cox log-Gaussien; régression logistique ordinale; régression log-linéaire; réseau linéaire.

Abstract. In this thesis, we focus on the statistical analysis of road accident data. Several issues are studied depending on whether the accident has already occurred (retrospective point of view) or before the accident occurs (predictive point of view). In order to do that, we use classical machine learning statistical methods and spatial and spatio-temporal prediction models. The first part of the thesis focuses on the statistical analysis of the accident severity. Many factors such as alcohol and/or drug consumption, department, time of day are related to accident severity through unsupervised or supervised statistical learning approaches. The dependence between these factors is modelled by using first log-linear models based on the associations highlighted by preliminary analyses of multiple correspondences and next, by ordinal logistic regression. These two types of modelling allow to quantify the risks associated with the factors analysed in relation to the severity of the accidents. The second part of the thesis focuses on the accident occurrence. The accident data are modelled by log-Gaussian Cox processes (LGCP) based on the geolocated coordinates of the accidents as well as on socio-demographic and road infrastructure covariates in order to identify the most critical geographical areas. We propose a variable selection method based on an aggregation of Poisson models and a variable importance criterion to select the most important covariates to be then used in an LGCP type model. The best model is then used to determine the riskiest areas. Secondly, the temporal component is introduced in the model and semiparametric generalized additive models are proposed in order to identify the critical zones and time periods. The modelling is inspired by case-control studies from epidemiology and takes into account the road network structure and space-time traffic data.

Keywords: areal interpolation; log-linear models; generalized additive models; kriging; log-Gaussian Cox processes, linear networks; multiple correspondence analysis; ordinal logistic regression.