Advances in Count Time Series Monitoring for Public Health Surveillance

Maëlle Salmon

Barcelona 2016

Advances in Count Time Series Monitoring for Public Health Surveillance

Maëlle Salmon

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Maëlle Salmon
aus Vannes, Frankreich

Barcelona, den 7. März 2016

Erstgutachter: Assoc. Professor Dr. Michael Höhle, Stockholm University
Zweitgutachter: Prof. Dr. Roland Fried, Technische Universität Dortmund
Drittgutachter: Prof. Dr. Volker Schmid, Ludwig-Maximilians-Universität
Prüfungsdatum: 29. Juni 2016

Zusammenfassung

Diese Arbeit beschäftigt sich mit statistischen Algorithmen zur Ausreißererkennung in Zeitreihen von Zähldaten, insbesondere in den Fallzahlen von Infektionskrankheiten. Das Ziel dieser Doktorarbeit war die Entwicklung und Anwendung von modernen statistischen Algorithmen zur Ausbrucherkennung in routinemäßig gemeldeten Fallberichten. Diese stammten aus Surveillance-Daten des Robert Koch-Instituts (RKI), dem deutschen nationalen Institut für öffentliche Gesundheit, welches für die Surveillance von menschlichen Infektionskrankheiten zuständig ist, und wo diese Arbeit angefertigt wurde. Obwohl bereits viele Algorithmen zur Ausreißererkennung in routinemäßig gesammelten Daten im Bereich der öffentlichen Gesundheit veröffentlicht wurden, haben sich diese neueren Methoden in der Praxis noch nicht durchgesetzt. Diese Doktorarbeit leistet Beiträge in drei Bereichen zur Erweiterung der Anwendung von statistischen Algorithmen zur Ausreißererkennung.

Erstens wurden die Open-Source-Implementierungen von statistischen Algorithmen zur Ausreißererkennung in dem R-Paket `surveillance` und ihre Dokumentierung zusammen mit der Vorstellung der dazuzugehörigen Theorie durchgeführt. Zweitens wurde im Rahmen dieser Doktorarbeit ein neuer Ausbrucherkennungsalgorithmus entwickelt, der das Problem der Rechtstrunkierung der Daten behandelt. Diese Anforderung war insbesondere durch das deutsche Surveillancesystem motiviert, wo ein zeitlicher Verzug zwischen der Erkrankung und dem Eingang der Fallmeldung am RKI zu beobachten ist. Schließlich unterstützte die vorliegende Doktorarbeit die Gestaltung und Entwicklung eines Surveillancesystems, das die Arbeit der Epidemiologen und Epidemiologinnen unterstützt, indem es automatische Berichte mit Signalen aus der Anwendung moderner Ausbruchserkennungsalgorithmen produziert.

In dieser Doktorarbeit wurden Statistische Algorithmen für Ausbrucherkennung bevorzugt, die auf Regressionsmodellen basieren. Dies erlaubt die solide Berücksichtigung der Schätz- und Beobachtungsunsicherheit. Diese Arbeit konzentriert sich dabei auf generalisierte lineare Modelle (GLM), sowie generalisierte additive Modelle (GAM), die gewählt wurden, da vor allem solche Modelle auf die Zähldatenstruktur der Zeitreihen eingehen können.

In dieser Dissertation werden sowohl neue theoretische, als auch praktische Entwicklungen vorgestellt. Die Struktur ist wie folgt: Kapitel 1 gibt eine Einleitung zum statistischen und praktischen Kontext dieser Arbeit und stellt den generellen Rahmen von Zähldatenzeitreihen und passenden Analysemethoden, die auf Regressionsmodellen basieren, vor. Im Kapitel 2 werden statistische Ausreißererkennung und dann eine Auswahl von statistischen Algorithmen zur Ausreißererkennung vorgestellt, die in der Routinesurveillance schon implementiert sind oder implementiert werden könnten. Kapitel 3 befasst sich mit dem Problem der Rechtstrunkierung von Daten. In diesem Kapitel werden existierende Methoden für Nowcasting und Ausreißererkennung bei bestehenden Übermittlungs- und Meldeverzügen präsentiert. Anschließend wird für dieses Problem ein neuer Bayesianischen Algorithmus vorgestellt, der in Simulationstudien untersucht wurde, auf der Zeitreihe von wöchentlichen Fallzahlen von *Salmonella* Newport aufgewandt wurde und für die Routineanwendung am RKI zum Zeitpunkt der Fertigstellung der Dissertation getestet wird. Kapitel 4 beschreibt den Beitrag der Dissertation für die Routineanwendung von statistischen Algorithmen zur Ausreißererkennung: das Kapitel stellt das neue automatisierte Surveillancesystem für Infektionskrankheiten am RKI vor, das zusammen mit Informatikern und Epidemiologen gestaltet und entwickelt wurde, und erklärt, wie man mit dem R-Paket `surveillance` ein einfacheres Surveillancesystem entwickeln könnte. Kapitel 5 gibt einen Ausblick über mögliche methodische Weiterentwicklungen zur Ausreißererkennung. Abschließend werden in Kapitel 6 die Resultate dieser Dissertation zusammengefasst und diskutiert.

Summary

This thesis deals with statistical algorithms for aberration detection in time series of counts, in particular counts of reported cases of infectious diseases. The goal was to develop and apply modern statistical algorithms for the detection of outbreaks in routinely reported case notifications, utilizing surveillance data from the Robert Koch Institute (RKI), which is the German national public health institute in charge of the surveillance of human infectious diseases, and where this PhD work was conducted. A multitude of different outbreak detection algorithms for routinely collected public health data has already been published, but their routine application is not entirely successful yet. This thesis brings three different types of contributions for extending the practical use of statistical algorithms for outbreak detection.

First, we contributed to the open-source implementation of aberration detection algorithms in the R package `surveillance` and to their documentation along with the description of the corresponding theory. Furthermore, we provided a methodological development for a commonly encountered issue: we made a regression based proposal for handling right-truncation of the data. This is motivated by the German surveillance system for infectious diseases, where there is a delay between disease onset and arrival of the case report at the RKI. Lastly, we participated to the design and development of a new system for automatic outbreak detection at the RKI which now produces automatic reports with alarms from modern aberration algorithms for supporting epidemiologists' work.

Importantly, in this thesis we advocate for regression-based aberration detection: most algorithms for outbreak detection as presented in this thesis are based on regression models. This allows a solid treatment of the estimation and observation uncertainty inherent to such models when used for prediction. The present work mainly concentrates on algorithms based on regression models for count data, Generalized Linear Models (GLMs) and Generalized Additive Models (GAMs). We chose such models because they adequately handle the count nature of surveillance data.

In the thesis, both the theoretical and practical advances for statistical outbreak detection brought by this work are presented. The structure is as follows. Chapter 1 contains an introduction to the statistical and practical context of applied disease surveillance, and presents the framework of count time series as well as suitable analysis methods using regression models. Chapter 2 introduces the statistical framework for outbreak detection, before exposing chosen algorithms for outbreak detection that are or could be applied to routine outbreak detection, including two algorithms whose implementation has been added to the surveillance package during this work. Chapter 3 deals with the problem of right-truncation of the data: after offering a short review of existing methods for nowcasting and outbreak detection in the presence of reporting delays, this Chapter presents a novel Bayesian algorithm that was tested in simulation studies and applied to the time series of the number of weekly cases of *Salmonella* Newport in Germany, and that is currently being evaluated at the Robert Koch Institute for routine application. Chapter 4 is a complete synthesis of our practical work in favour of routine application of state-of-the-art statistical algorithms for outbreak detection: this Chapter presents the new automatic surveillance system at the RKI that was designed and developped along with informaticians and epidemiologists, and provides guidance for implementing a simpler surveillance system using the R package surveillance. Chapter 5 contains an outlook of possible future methodological developments for aberration detection. Finally, Chapter 6 concludes the thesis.

Contents

Chapter 1

Introduction

The fight against infectious diseases nowadays does not only require treating patients and setting up measures for prevention once an outbreak has taken its toll. It also demands the timely recognition of emerging outbreaks in order to avoid their expansion. Along these lines, German public health authorities collect and store information about the occurrence of notifiable diseases – typically represented as individual case reports. This data collection, which is regulated by the 2001 Infection Protection Act (Niemer, 2001; Heudorf et al., 2013), enables situational awareness in general and in particular the timely detection of aberrant counts: for any specific aggregation of characteristics of events, such as adults becoming sick with salmonellosis in Germany, data can be represented as time series of counts with *e.g.* weeks as time units of the aggregation. Abnormally high or low values at a given time point can reveal critical issues such as an outbreak of the disease or a malfunction of data transmission. Thus, identifying aberrations in the collected data is decisive.

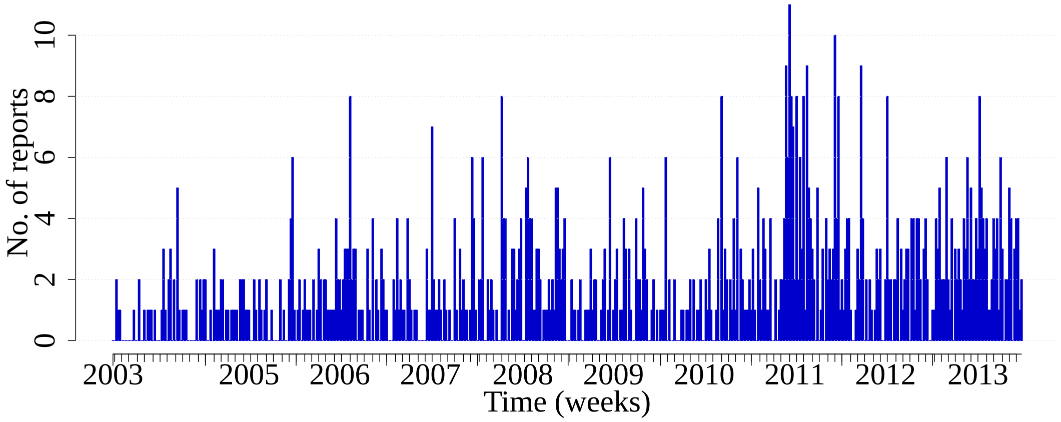


Figure 1.1: Weekly number of cases of enterohemorrhagic E. coli in Saxony aggregated by week of report, 2004-2013. One can see the outbreak in 2011 (Altmann et al., 2011; Bernard et al., 2014) and more cases afterwards compared to before the outbreak.

This thesis deals with modern statistical methods for aberration detection in count time series, also called integer valued time series, of surveillance data. For this purpose we mostly consider algorithms based on generalized linear or additive models, with inference performed in frequentist or Bayesian frameworks. We restrict our analysis to univariate time surveillance, in contrast to multivariate time surveillance and to (multivariate) space-time surveillance. However, the developments brought by this work could be the basis of improvements in (multivariate) space-time surveillance. We moreover focus on the detection on aberrations that are high counts, disregarding the detection of counts that are lower than expected, since our primary motivation is to detect outbreaks of infectious diseases rather than problems of data transmission or of diagnosis.

Beside presenting suitable statistical methods dealing with both estimation and observation uncertainties we put an emphasis on the the implementation of such methods in a public health institution. This work was performed at the Robert Koch Institute which is the German national public health institute in charge of the surveillance of human infectious diseases. Nevertheless, our developments could be generalized to other countries and surveillance systems.

Within this introduction we first give a brief outline of the public health context of this work including an overview of the methodological interest and aims of this thesis. Afterwards we provide a condensed review of the statistical framework of this thesis. The outline of the present thesis will conclude this introduction.

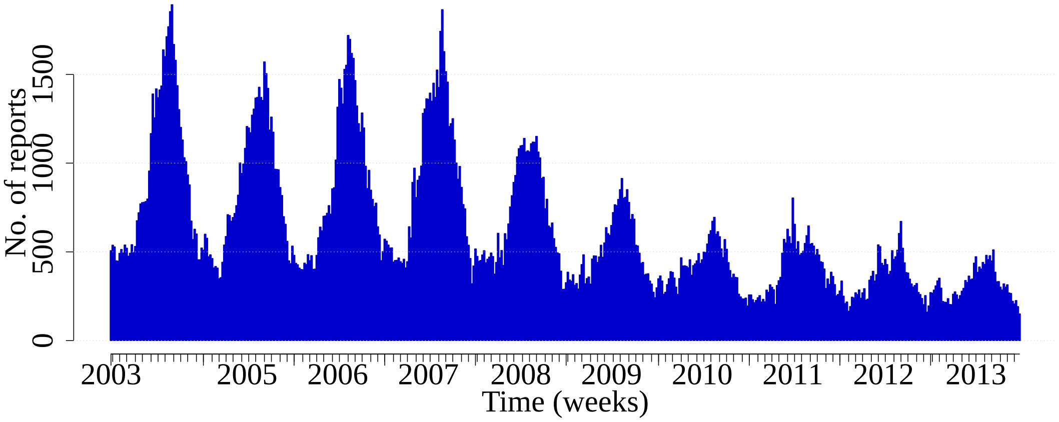


Figure 1.2: Weekly number of cases of *Salmonella* in Germany aggregated by date of disease onset, 2004-2013. The seasonality and a negative trend over time are quite easy to spot.

1.1 Public health context and methodological scope of this work

The work on thesis was funded by the Robert Koch Institute with the aim to improve its outbreak detection system, and to provide general developments for aberration detection in count time series that can be used in other institutions and contexts. In this section we explain what outbreak detection at an institute such as the Robert Koch Institute is, and present the the practical and scientifical goals of this work.

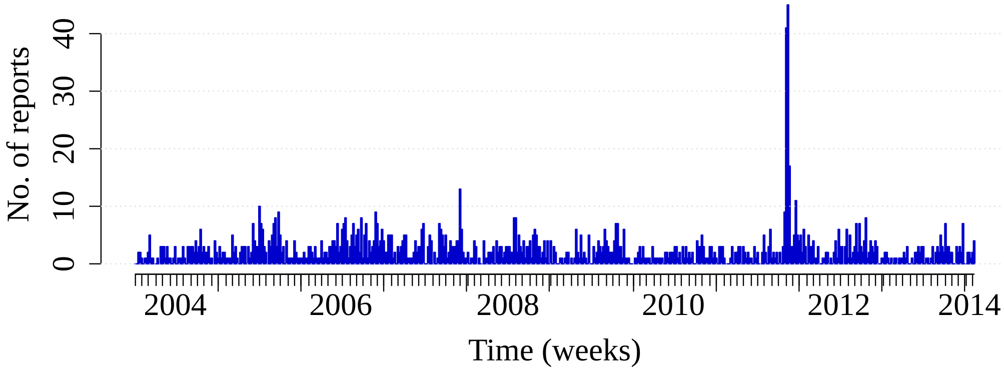


Figure 1.3: Weekly number of cases of S. Newport in Germany aggregated by week of report, 2004-2013. There was a known outbreak in 2011 (Bayer et al., 2014).

1.1.1 Statistical aberration detection at a public health institute

Cases of notifiable infectious diseases can be aggregated into univariate time series according to their date of declaration or of diagnosis. One thus gets a time series of, say, the weekly number of reported cases of enterohemorrhagic E. coli in Saxony or of *Salmonella* in Germany as shown, respectively, in Figure 1.1 and in Figure 1.2. Defining time series for all notifiable pathogens of a country, and for subsets such as pathogen subtypes or federal states where the case was diagnosed, creates many time series. Figure 1.3 shows the weekly number of reported cases of the *Salmonella* Newport serotype, and Figure 1.4 shows this weekly number for the federal states of Bavaria and Berlin on their own. An institute such as the Robert Koch Institute in Berlin needs to use such time series on the one hand to increase situational awareness – for instance having a rough idea of the counts when the situation is in control – and on the other hand to detect outbreaks early. Because they are more than 80 reporting categories in Germany, with many possible different analysis levels such as age-groups, sex or places, aberration detection has to be made automatic, and therefore supported by sound statistical methods.

1.1.2 Ambitions of this work

The surveillance time series created at a national public health institute display diversity in their patterns of seasonality, mean number of cases, etc. (Enki et al., 2013), as one can already see in the four Figures 1.1, 1.2, 1.3 and 1.4. This diversity complicates the development of algorithms for outbreak detection. One could either choose to develop a model for each time series or to develop a model that works fairly well for all time series. This approach called *one size fits all* is the one that is currently often chosen (Noufaily et al., 2013). The present work has a strong focus on methods that can be used on a multitude of time series without *fine-tuning* and without too much computing time.

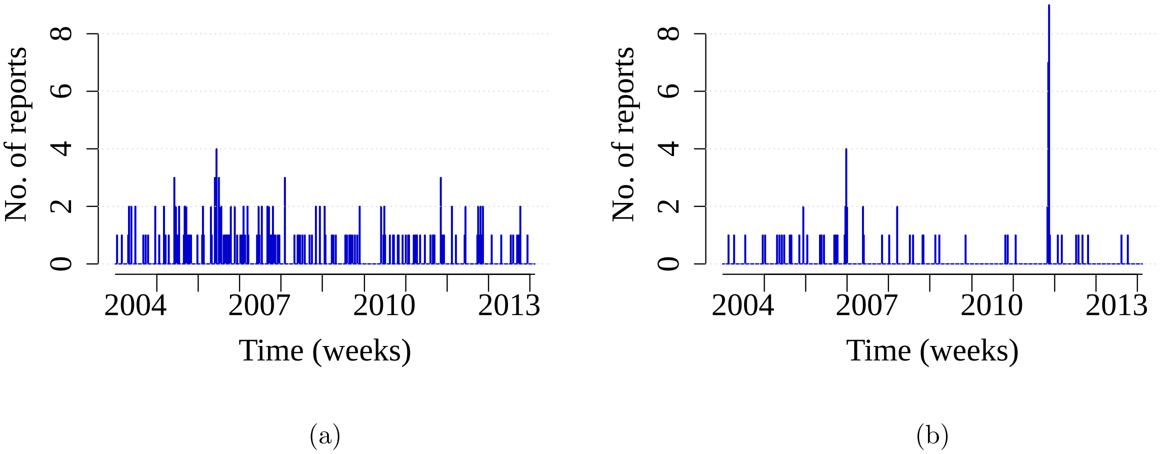


Figure 1.4: Weekly count of S. Newport in the German federal states (a) Bavaria and (b) Berlin aggregated by week of report, 2004-2013. Both time series display small counts.

This work puts an emphasis on implementation of methods for outbreak detection in practice, for getting automatic tools for recognizing and thus fighting against outbreaks early on. Systems for automatic aberration detection are implemented in several European countries (Hulth et al., 2010) where they help to get aware of unusual numbers of cases possibly indicating emerging outbreaks of infectious diseases. The Robert Koch Institute receives reports of notifiable infectious diseases that are transmitted from local health authorities *via* federal state health authorities. The Robert Koch Institute wanted to improve its system for outbreak detection which is why this work was funded. Part of the tasks related to this thesis was offering statistical counseling for a new automatic system

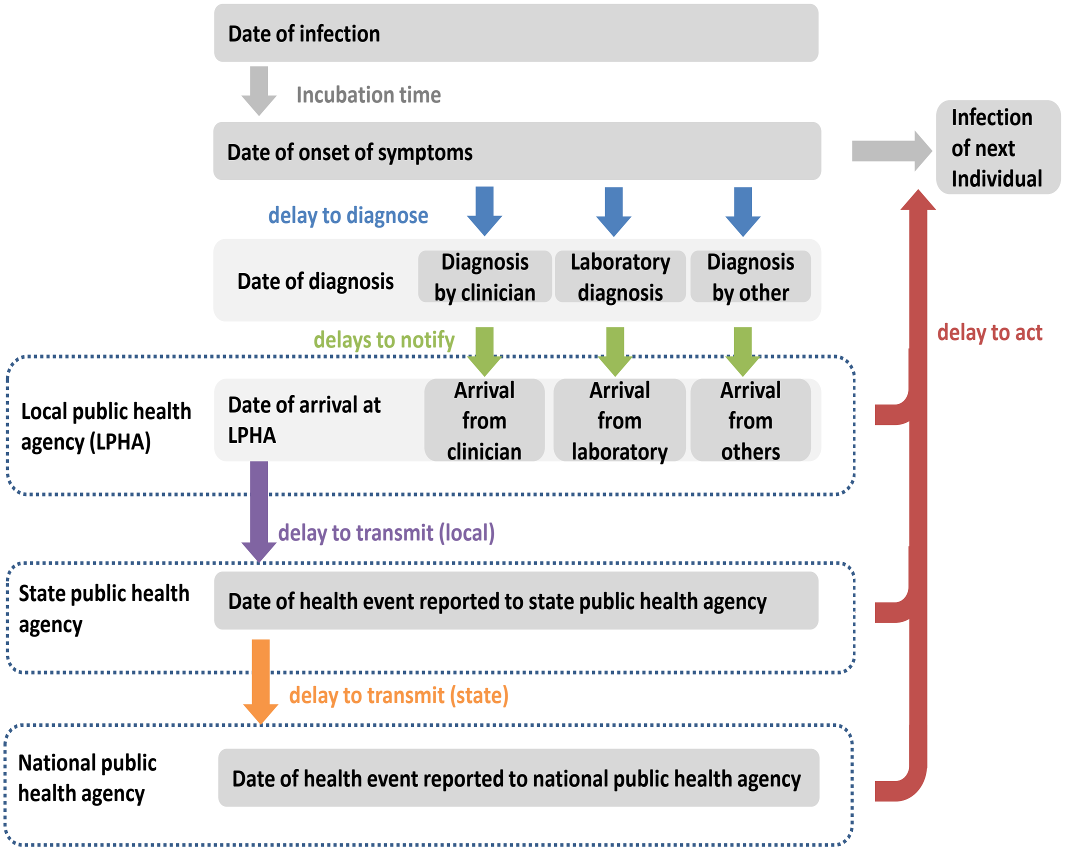

Date of infection
Incubation time
Date of onset of symptoms
Infection of next Individual
delay to diagnose
Date of diagnosis
Diagnosis by clinician
Laboratory diagnosis
Diagnosis by other
delays to notify
delay to act
Local public health agency (LPHA)
Date of arrival at LPHA
Arrival from clinician
Arrival from laboratory
Arrival from others
delay to transmit (local)
State public health agency
Date of health event reported to state public health agency
delay to transmit (state)
National public health agency
Date of health event reported to national public health agency

Figure 1.5: German reporting system, figure taken from Schumacher et al. (2016).

for outbreak detection at the Robert Koch Institute, that was created during the work on this thesis (Salmon et al., 2016) and that will be the topic of Chapter 4.

Moreover, we intented to be receptive to common complications of routine surveillance and therefore developed an algorithm for aberration detection taking reporting delays into account. Figure 1.5 that was taken from Schumacher et al. (2016) shows the German reporting system. Each case needs to first be diagnosed by a clinician, a laboratory or other, depending on the case definition. The information about the case needs to be transmitted to the local public health authority. Once this is done the case will be transmitted first to the State public health agency and then to the national public health agency, the Robert Koch Institute. Outbreak detection and case management are performed at the local, state and national public health agency, but each agency only sees cases diagnosed on its territory, which makes the role of the national public health agency crucial in that it receives information about cases from the whole country. Figure 1.5 thus illustrates delays due to diagnosis and transmission. Because of these delays, information about cases is not available right away at any level of the reporting system. Therefore, the observed current number of count is not complete yet. Every local and federal state health authority does some sort of data analysis, including automatic tools in some institutions (Läubrich et al., 2011), but we concentrate on aberration detection performed at the national level at the Robert Koch Institute. Only when cases information arrives at the RKI can one notice a country-wide outbreak since each federal state only has information about cases diagnosed on its territory.

There was already work aiming at predicting the current number of count despite reporting delays, the so-called *nowcasting* (Höhle and an der Heiden, 2014), but no statistical framework for producing a threshold for an incomplete current count. To our knowledge, the only other statistical work reporting such an effort was found in Noufaily et al. (2015). Taking right-truncation of the data into account when defining the decision threshold demands adding a new time dimension: not only do we have time of *e.g* diagnosis but also time of *e.g.* report. For dealing with this difficulty, we used statistical literature in the field of epidemiology (Lawless, 1994; Höhle and an der Heiden, 2014) but also borrowed concepts from the field of insurance mathematics (Schmidt and Wünsche, 1998). We describe the algorithm thus created during this doctoral work in Chapter 3, along with an evaluation study on simulated data.

Beside the routine implementation of methods at the Robert Koch Institute, this work also was motivated by the perspective of simplifying the comparison and use of state-of-the-art algorithms for outbreak detection in general. Therefore, a strong effort was put on adding algorithms into the R package `surveillance` which offers tools for outbreak detection and on documenting them. We chose the statistical programming language `R` (R Core Team, 2015) for all implementations because it is an open-source software with many dedicated statistical tools. Part of the work of this thesis was moreover directed at writing an article that covers theoretical and practical aspects of the package use (Salmon et al., 2016). The manuscript is a condense introduction to aberration detection based on regression models, with a strong focus on explaining the use of code and the routine implementation of such methods. Therefore, this work does not only support outbreak detection at the Robert Koch Institute but also contributes to open-source software for aberration detection.

1.2 Count time series models for monitoring

In this section we lay out the theoretical framework of the work performed in this thesis. We will first give a short introduction to the model-based approach of aberration detection, and then present the characteristics of count time series as the ones found in the context of the surveillance of infectious diseases, and also present the corresponding statistical distributions before introducing regression models adapted to these characteristics. Using such regression models, we want to build control charts for aberration detection, *e.g.* defining an aberrant count as any count above a quantile of the predictive distribution of the current count. First attempts to review and compare aberration detection methods by dividing each method into two subsequent steps (first forecasting and then detection) can be found in Murphy and Burkom (2008), and, with an even greater decomposition – starting from ”getting baseline data” and ”transforming data” – in Buckeridge et al. (2008). Such a review approach is different from, the one in Unkel et al. (2012).

1.2.1 Motivation for a regression-based approach for aberration detection

Using appropriate regression models to make predictions for aberration detection is a quite recent development in the field since traditional methods such as Stroup's method introduced in Stroup et al. (1989) or EARS methods explained in Fricker et al. (2008) do not rely on such techniques. We assume that observations $\{y_t,\ t = 1, 2, \ldots\}$ are realizations of random variables that follow the same parametric distribution but with different parameters, *e.g.* in the case when they have different means. We shall use regression, with time being one of the covariates, for handling time trends and dependences. Therefore, we deem the aforementioned methods, Stroup's method and EARS methods, to be less statistically correct: in this thesis we advocate for regression-based aberration detection. Some current surveillance systems still use very basic methods for calculating the threshold, for instance the mean of historic values plus two standard deviations in Abat et al. (2015).

We believe that aberration detection should be supported by using regression models adapted to count time series in the frequent cases when the surveillance time series display autocorrelation, seasonality, time trends and presence of past outbreaks in the records. Regression itself needs to be performed on historic values of the time series, that we assume to be informative regarding the normal and abnormal behaviours of the time series.

1.2.2 Statistical distributions for count time series

Surveillance count time series display various characteristics that should be taken into account when modelling them. They are time series of non-zero integers and often overdispersed: in empirical data one would often notice that the variance is higher than their mean. They can display, as we previously mentioned, autocorrelation, seasonality, a time trend, and past outbreaks or aberrations in the records. A count time series can be represented as a Poisson or negative binomial distributed variable with a time-varying mean and realizations $\{y_t,\ t = 1, 2, \ldots\}$. Thus we either assume that $y_t \sim \text{Po}(\mu_t)$ or that $y_t \sim \text{NB}(\mu_t, \nu)$. The probability mass function (pmf) of a Poisson variable with mean μ_t is

$$P(y_t = k) = \frac{\mu_t^k}{k!} e^{-\mu_t}$$

so that $E(y_t) = \text{Var}(y_t) = \mu_t$. The pmf of a negative binomial distributed variable with mean μ_t and overdispersion parameter ν is

$$P(y_t = k) = \frac{\Gamma(k + \nu)}{\Gamma(\nu) \cdot k!} \cdot \frac{\mu_t^k \cdot \nu^\nu}{(\mu_t + \nu)^{k+\nu}}$$

so that $E(y_t) = \mu_t$ and $\text{Var}(y_t) = \mu_t(1 + \mu_t/\nu)$. The negative binomial distribution is called NBII model in Hilbe (2011) because the variance depends on the squared mean. The negative binomial distribution allows to account for overdispersion in the data. The Poisson distribution, and the negative binomial distribution with fixed ν are distributions belonging to the one-parameter exponential family. In this thesis we do not consider the variations of the Poisson or negative binomial distributions such as the zero-inflated distributions (Hilbe, 2011).

1.2.3 Regression models for count time series

In this thesis, we concentrate on algorithms based on regression models that offer the possibilities of accounting for the characteristics of count time series. Such models then easily form the basis of various control charts (Höhle and Paul, 2008; Noufaily et al., 2013; Manitz and Höhle, 2013). The review of models for count time series in Jung and Tremayne (2011) states that no complete nomenclature of the models exists. A classical nomenclature is to define observation- vs. parameter-driven models following Cox et al. (1981). However, in this thesis, we rather differentiate models regarding how they represent overdispersion, seasonality, past outbreaks and time trends.

We are interested in regression models such that

$$y_t \sim \text{Po}(\mu_t)$$

or

$$y_t \sim \text{NB}(\mu_t, \nu)$$

and

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

where g is a known link function, often chosen to be log for count time series and η_t is called the linear predictor. We therefore use (quasi) Generalized Linear Models (GLMs) and Generalized Additive Models (GAMs) (Fahrmeir et al., 2013). The difference between GLMs and GAMs is that in a GAM the linear predictor η_t can be linear to smooth functions of some continuous covariates, *e.g.* $\eta_t = \beta_0 + f(t)$ where f is a smooth function. In contrast to the linear predictor of a GLM, this linear predictor would be able to describe linear trend more complex than a log-linear one, which can for instance come in handy when there was first an increase and then a decrease in counts over the time period for which one applies the regression. The linear predictor can include auto-regression on previous values of the mean or of the observation (Liboschik et al., 2016) or a time-varying intercept to account for auto-regression (Heisterkamp et al., 2006; Manitz and Höhle, 2013). Seasonality can be modelled *e.g.* as a sum of sinusoidal components (Höhle and Paul, 2008) or as a possibly penalized spline (Noufaily et al., 2013; Manitz and Höhle, 2013).

On top of the negative binomial distribution, as another extension of the GLM and GAM framework one can use the quasi-Poisson family which is a Poisson regression but with a supplementary parameter ϕ defined so that $\text{Var}(y_t) = \phi \cdot \mu_t$ (Hilbe, 2011). This model is called NBI model in Hilbe (2011). There is no such thing as a quasi-Poisson distribution. Nonetheless, one can parameterize a negative binomial variable such that it matches the quasi-Poisson specification of the mean and variance and hence could be used to sample quantiles with the mean-variance relationship corresponding to the quasi-Poisson model.

We do not use models defining transitions between unobserved states, such as state-space models (De Jong, 1988; Dunsmuir and Scott, 2015) and hidden Markov models as used for influenza surveillance in Martínez-Beneito et al. (2008); Conesa et al. (2015), or models that make a regression on other distribution parameters than the mean (Rigby and Stasinopoulos, 2005; Stasinopoulos and Rigby, 2007).

1.2.4 Model inference

In this thesis we describe algorithms whose model inference is performed either in a frequentist or in a Bayesian framework. We are not interested in point-predictions, but rather in probabilistic predictions. A very important aspect we want to underline here is uncertainty: Inference for aberration detection should take sources of uncertainty into account. Thus, the decision regarding whether an observed count is aberrant can have a higher specificity. We include two sources of uncertainties: the estimation uncertainty and the observation uncertainty. In this work, we indeed choose to disregard the uncertainty related to the choice of the regression model. The observation uncertainty is often taken into account by defining a threshold for the current count as a quantile of its predictive distribution. The estimation uncertainty is disregarded by some existing algorithms (Höhle and Paul, 2008; Noufaily et al., 2013) but can be taken into account very naturally by Bayesian methods (Manitz and Höhle, 2013; Salmon et al., 2015). Disregarding estimation uncertainty makes aberration detection less specific. In this section, we shortly introduce the main inference methods used in this thesis.

In this thesis we describe algorithms whose model inference is performed either in a frequentist or in a Bayesian framework. We are not interested in point-predictions, but rather in probabilistic predictions. A very important aspect we want to underline is uncertainty: Inference for aberration detection should take sources of uncertainty into account. Thus, the decision regarding whether an observed count is aberrant can be more specific. We include two sources of uncertainties: the estimation uncertainty and the observation uncertainty. In this work, we indeed choose to disregard the uncertainty related to the choice of the regression model. The observation uncertainty is often taken into account by defining a threshold for the current count as a quantile of its predictive distribution. The estimation uncertainty is disregarded by some existing algorithms (Höhle and Paul, 2008; Noufaily et al., 2013) but can be taken into account very naturally by Bayesian methods (Manitz and Höhle, 2013; Salmon et al., 2015). Disregarding estimation uncertainty makes aberration detection less specific. In this section, we shortly introduce the main inference methods used in this thesis.

In the case of the Poisson GLM, for which we have $\eta_t = \boldsymbol{\beta X}$ with $\boldsymbol{X}$ the matrix of independent variables and $\boldsymbol{\beta}$ the vector of parameters, the regression parameters are estimated by Maximum Likelihood using the Iterative Weighted Least Squares (IWLS) algorithm (Zeileis et al., 2008). When using a quasi-Poisson family, the overdispersion parameter is estimated separately in a second step as

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

where n is the sample size, p the number of estimated parameters and $\hat{\mu}_i$ the Maximum Likelihood Estimator of μ_i (calculated based on $\boldsymbol{X}$ and on $\hat{\boldsymbol{\beta}}$). The model inference is no longer a likelihood but instead a quasi-likelihood approach (Fahrmeir et al., 2013). We have to write *extended* GLMs because the inference is different from the one for *e.g.* the Poisson model. In `R`, frequentist GLM of the Poisson and quasi-Poisson families can be fitted using the `glm` function of the `stats` package (R Core Team, 2015) and GAM using the `gam` package (Hastie, 2015).

The log-likelihood of a negative binomial extended GLM depending on the observations and the parameters $\boldsymbol{\beta}$ and ν is

$$l = \sum_{i=1}^{n}\left(y_i\log(\mu_i) + \nu log(\nu) - (\nu + y_i)\log(\nu+\mu_i) + \log\left(\frac{\Gamma(\nu+y_i)}{\Gamma(\nu)}\right) - \log(y_i!)\right).$$

The regression coefficients of the linear predictor of $\log(\mu_t)$ and ν can be iteratively estimated separately (Zeileis et al., 2008) based on the first derivatives of the log-likelihood that are (Lawless, 1987)

$$\frac{\delta l}{\delta\beta_r} = \sum_{i=1}^{n}\frac{x_{ir}(y_i-\mu_i)}{1+\mu_i/\nu},\ r = 1,\ldots,p$$

with $\boldsymbol{X}$ the matrix of independent variables, p the number of parameters and $\boldsymbol{\beta}$ the parameters vector; and

$$\frac{\delta l}{\delta\nu} = \sum_{i=1}^{n}\left(\psi(\nu+\mu_i) - \psi(\nu) + \log(\nu) - \log(\nu+\mu_i) - \frac{\nu+y_i}{\nu+\mu_i} + 1\right),$$

where ψ is the digamma function, *i.e*

$$\psi(x) = \frac{d}{dx}\ln\Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}.$$

More precisely, the iterative process is such that

1. $\boldsymbol{\beta}$ is estimated for a fixed value of ν using a standard GLM fit (with a fixed ν the negative binomial distribution belongs to a one-parameter exponential family exponential family);

2. then for a fixed $\boldsymbol{\beta}$ ν is estimated with a Newton-Raphson iterative scheme.

These steps are repeated until convergence is obtained. It was proven that $\boldsymbol{\beta}$ and ν are asymptotically independent (Lawless, 1987) so that the standard errors are correctly estimated by the iterative procedure. The negative binomial extended GLMs and GAMs can be fitted in a frequentist framework using the `MASS` package (Venables and Ripley, 2002) and the `gam` package, respectively.

In this thesis, for Bayesian inference of GLMs and GAMs we use integrated nested Laplace approximations (INLA) as described in Rue et al. (2009) and implemented in the `INLA` package (Rue et al., 2015). We actually do not fit GAMs in a frequentist framework in this work. Here we briefly present the principle of INLA, which is an Bayesian inference method much faster than Markov chain Monte Carlo (MCMC). INLA is an inference method for latent Gaussian models. A negative binomial GLM can be viewed as a latent Gaussian model. The parameters $\boldsymbol{\beta}$ are then described as a p-dimensional Gaussian field (with p the number of parameters). The likelihood model for $y_i|\boldsymbol{\beta}$ is the negative binomial likelihood that is controlled by the unknown hyperparameter ν. The marginal posterior density of each of the parameters is:

$$\pi(\beta_i|y) = \int_\nu \pi(\beta_i|\nu, y)\pi(\nu|y)d\nu.$$

The INLA method consists in using Laplace approximations for both $\pi(\nu|y)$ and $\pi(\beta_i|\nu, y)$, which explains the name of the method, and in approximating the integral by a sum. More details are given in Rue et al. (2009). In this work, we use INLA inference in Section 2.4 and Chapter 3.

Past outbreaks or aberrations can have a very high influence on inference. This can be accounted for by including a covariate for past outbreaks when they are known, *e.g.* as a binary variable with additive effect on η_t (Manitz and Höhle, 2013). In Fried et al. (2015) outliers are detected and modelled explicitely in a Bayesian framework. Or one can use standardized Anscombe residuals of a first fit of the model to down-weight outlying observations in a second fit of the model, which is the makeshift solution used in Farrington et al. (1996) and Noufaily et al. (2013). A future development of count time series model would be to offer robust inference such as Aeberhard et al. (2014) for the negative binomial distribution, but tailored to the auto-regression structure of some time series models (Elsaied and Fried, 2014). These investigations were not part of this thesis but such inference methods could easily replace the current ones in existing control charts as described and used in this work.

1.2.5 Model assessment

In this section, we shortly discuss how regression models as those presented in this thesis can be assessed, without, however, aiming at providing an exhaustive review of model selection. Good references on this subject are *e.g.* Fahrmeir et al. (2013) and Held and Sabanés Bové (2014). In this thesis, we are more particularly interested in assessing probabilistic predictions offered by a regression model. In this regard, two important notions are *calibration* and *sharpness*. *Calibration* means, roughly said, that we wish the predictive distribution to give a high probability to the observed count: a very bad predictive distribution would classify all observed counts as extreme values. *Sharpness* means that the predictive distribution should be concentrated: this property, contrary to the calibration, does not depend on the observed counts. Obviously, one looks for a trade-off between calibration and sharpness: having a high sharpness is tantamount to betting everything on a value, which can be very wrong.

Regression models for count time series can be assessed using standard tools such as the AIC (Akaike's Information Criterion) or the BIC (Bayesian Information Criterion) that take into account the likelihood of the observations but also the number of parameters. The predictive capabilities of regression models for count time series can be also assessed using proper scoring rules (Christou and Fokianos, 2015; Czado et al., 2009). Scoring rules are a function of the predictive distribution and of the observed count; they often are reported as a mean of this function over all observed counts. In Jung and Tremayne (2011) a proper scoring rule is defined: it is a scoring rule that gives best scores for forecasters when they predict ”according to their true belief about the predictive distribution”. Thus, the better the score, the better the prediction. In Czado et al. (2009) it is explained that a *proper scoring rule* is proper when the score given to the true predictive distribution is always *higher that or equal to* the score given to any other predictive distribution. *Strictly proper scoring rules* are the ones for which one can write *higher than* instead of *higher that or equal to*. Czado et al. (2009) argue that scoring rules allow to investigate *calibration* and *sharpness* of a predictive distribution. There exist tests based on scoring rules such as the permutation test in Paul and Held (2011) that outputs a Monte Carlo p-value which helps making a decision about which model to use for a time series. Other tests, though only defined for independent forecasts, are presented in Wei and Held (2014). In any case, it is best to use several scoring rules when assessing models (Czado et al., 2009).

Beside AIC, BIC and scoring rules using predictions, an important step when assessing a model for count time series is to look at the ACF (auto-correlation function) of the transformed or untransformed residuals in order to judge whether the regression model accounts for the autoregression structure of the data. Offering a general comparison of possible models for count time series, and assessing them on many time series, is beyond the scope of this thesis. However, in Section 5.1.1 we show an example of assessment of several models based on ACF and proper scoring rules. The methodological development presented in Chapter 3 could actually be used with another regression model than the one we applied.

1.3 Outline of the thesis and contributions

This thesis is structured as follows. Chapter 2 offers an overview of representative algorithms for outbreak detection that illustrate well the state of the art in aberration detection. Chapter 3 presents the main methodological advance brought by this work: an algorithm for outbreak detection that accounts for right-truncation of the data while including both estimation and observation uncertainty. Chapter 4 is a description of the design choices and of the implementation of a routine system for aberration detection at the Robert Koch Institute. In Chapter 5 we present possible future methodological developments of algorithms described in the thesis and mention remaining challenges for aberration detection at a national public health institute. Lastly, Chapter 6 summarizes the thesis.

Part of the work presented in the thesis has been submitted and accepted for publication in peer-reviewed scientific journals. They form the basis of several chapters and sections. For this thesis the respective manuscripts were adapted in order to obtain a consistent notation and to eliminate redundancy. The articles are:

- **M. Salmon**, D. Schumacher, M. Höhle. Monitoring Count Time Series in R: Aberration Detection in Public Health Surveillance, *Journal of Statistical Software*, accepted for publication.

  Michael Höhle created and maintains the package `surveillance` of which **Maëlle Salmon** has been an active contributer: three algorithms and two functions for simulating time series with reporting delays were added to `surveillance` as a result of her work. **Maëlle Salmon** conceived and prepared the manuscript. **Maëlle Salmon** wrote the manuscript with contributions from Dirk Schumacher and Michael Höhle. **Maëlle Salmon**, Dirk Schumacher and Michael Höhle edited the manuscript.

  Chapter 2 is mostly based on this article, with more details e.g. about standardized Anscombe residuals in Section 2.3 and about a detection algorithm in Section 2.5.1 and a whole new section, Section 2.2. Moreover, the last part of the JSS manuscript, which explains how to set up a simple surveillance system using the package, is used for Section 4.4.

- **M. Salmon**, D. Schumacher, K. Stark, M. Höhle. Bayesian Outbreak Detection in the Presence of Reporting Delays, *Biometrical Journal*, *57* (6), 1051-1067, 2015.

  Maëlle Salmon conceived and implemented the proposed algorithm. **Maëlle Salmon**, together with Dirk Schumacher and Michael Höhle, designed the simulation study and prepared the manuscript. **Maëlle Salmon** wrote the manuscript. **Maëlle Salmon**, Dirk Schumacher, Klaus Stark and Michael Höhle edited the manuscript.

  The first sections of Chapter 3 are based on this paper, but Section 3.5, the description of the proof in 3.3.1, and Section 3.6 are new content compared to the article, and present the code created for this part of the thesis and a related method published by other authors in Noufaily et al. (2015), respectively.

- **M. Salmon**, D. Schumacher, H. Burmann, C. Frank, H. Claus, M. Höhle. A system for automated outbreak detection of communicable diseases in Germany, *Eurosurveillance*, accepted for publication.

  Dirk Schumacher implemented the system, of which Hendrik Burmann is the new maintainer and developer. **Maëlle Salmon** participated to the system design, was involved in meetings with users, and has provided statistical counseling during the system development and afterwards, including for testing the new algorithm taking reporting delays into account developped during this work. Dirk Schumacher prepared and wrote the first version of the manuscript with the help of **Maëlle Salmon**. **Maëlle Salmon** revised the manuscript. **Maëlle Salmon**, Dirk Schumacher, Christina Frank, Hermann Claus and Michael Höhle edited both versions of the manuscript.

The first sections of Chapter 4 are based on this manuscript, while Section 4.4 is based on the JSS manuscript, but Section 4.3 presents ongoing work started after the publication of both manuscripts.

The manuscripts will be cited again at the beginnings of the respective sections.

Chapter 2

Count Time Series Monitoring for Infectious Diseases Surveillance

In this chapter we aim at presenting the statistical framework for aberration detection, while underlining that it is based on statistical prediction. Our first goal is to offer a critical survey of the field, pointing at weaknesses and strengths of existing methods. Our second goal is to present implementation aspects of algorithms, including code that we have written and added to a public open-source software package. We shall present a representative set of methods for aberration detection that are or could be used in public health surveillance.

Note that this chapter is an extended version of the article ***M. Salmon****, D. Schumacher, M. Höhle. Monitoring Count Time Series in R: Aberration Detection in Public Health Surveillance,* Journal of Statistical Software*, accepted for publication, of which it partially uses text passages.*

2.1 Statistical framework for aberration detection

We introduce the framework for aberration detection by considering an univariate time series of counts with observations $\{y_t,\ t = 1, 2, \ldots\}$. Surveillance aims at prospectively – as opposed to retrospectively – detecting an *aberration*, that is to say, an important change in the process occurring at an unknown time τ. The prospective aspect of monitoring means that the decision made at timepoint s is solely based on the counts $\{y_t,\ t < s\}$. This change can be a step increase of the counts of cases or a more gradual change (Sonesson and Bock, 2003): one could imagine a surge in salmonellosis cases based on the punctual availability of an infected food item on the market, or a slower and more persistent increase of measles numbers due to decreasing vaccination rates. Based on the possibility of such a change, for each time t we want to differentiate between the two states *in-control* and *out-of-control*, which actually is a classification problem. At any timepoint $s \geq 1$, the available information – *i.e.*, past counts – is defined as $\boldsymbol{y}_s = \{y_t\ ;\ t \leq s\}$. Detection is based on a statistic $r(\cdot)$ with resulting alarm time $T_A = \min\{s \geq 1 : r(\boldsymbol{y}_s) > g\}$ where g is a known threshold. Functions for aberration detection use past data to compute $r(\boldsymbol{y}_s)$, and compare it to the threshold g, above which the current count can be considered as suspicious and thus doomed as *out-of-control*. Note that an important distinction to be made is between one timepoint detection and detection over several timepoints. In the former, only (a transformation of) the current value of y_t is compared to a threshold, whereas in the latter, detection can be based on accumulated deviation of subsequent response variables compared to an expectation. All methods presented in this thesis except the CUSUM of Höhle and Paul (2008) perform one-timepoint detection.

Aberration detection in public health surveillance has the same goal as aberration detection in industrial contexts: detecting a change in a monitored process, such as a surge in the number of defect items in a production line. Therefore, public health surveillance uses tools from *Statistical Process Control* (SPC) which is a field aiming at creating control charts that actually are algorithms for aberration detection, originally for the industry (Unkel et al., 2012). Both fields, public health surveillance and SPC, also have independent developments due to different characteristics of the data, but can enrich each other (Woodall, 2006). A review of SPC methods for count time series, however disregarding recent work such as Liboschik et al. (2014) and Höhle and Paul (2008), can be found in Weiß and Lu (2015). A very simple control chart is the Shewart control chart (Shewhart, 1931) that is actually very close to the method we shall present in the next paragraphs.

We will illustrate the basic principle of aberration detection by using the `earsC` algorithm that implements the EARS (Early Aberration Detection System) methods of the CDC (Center for Disease Control and Prevention of the USA) as described in Fricker et al. (2008). We start the chapter by introducing this algorithm for didactical reasons: even if the method is flawed, it illustrates the decision process of prospective aberration detection and is commonly used. This algorithm is especially convenient in situations when little historic information is available. It offers three variants called C1, C2 and C3. Here we shall focus on C1 for which the baseline are the 7 timepoints before the assessed timepoint s, that is to say $(y_{s-7}, \ldots, y_{s-1})$. The expected value is the mean of the baseline. The method is based on a statistic called C_s defined as $C_s = {}^{(y_s - \bar{y}_s)}/_{s_s}$, where

$$\bar{y}_s = \frac{1}{7} \cdot \sum_{i=s-7}^{s-1} y_i \quad \text{and} \quad s_s^2 = \frac{1}{7-1} \cdot \sum_{i=s-7}^{s-1} (y_i - \bar{y}_s)^2 .$$

Under the null hypothesis of no outbreak, it is assumed that $C_s \overset{H_0}{\sim} N(0,1)$. This is tantamount to assuming a normal distribution of the disease counts which is an oxymoron since the normal distribution is a continuous distribution. Moreover, the method assumes that the distribution of the number of cases has a constant distribution from the timepoints $s-7$ to s which would hardly be the case for any seasonal disease or, in case of a daily aggegation, in any disease with day-of-the-week effects on diagnosis.

The upperbound U_s is defined by the $(1-\alpha)$-quantile of the normal distribution of the disease count $y_t, t \leq s$: $U_s = \bar{y}_s + z_{1-\alpha} s_s$ where $z_{1-\alpha}$ is the $(1-\alpha)$-quantile of the standard normal distribution. An alarm is raised if $y_s > U_s$. In Fricker et al. (2008) it is stated that an alarm is raised if C_s is higher than 3, which roughly corresponds to $\alpha = 0.001$.

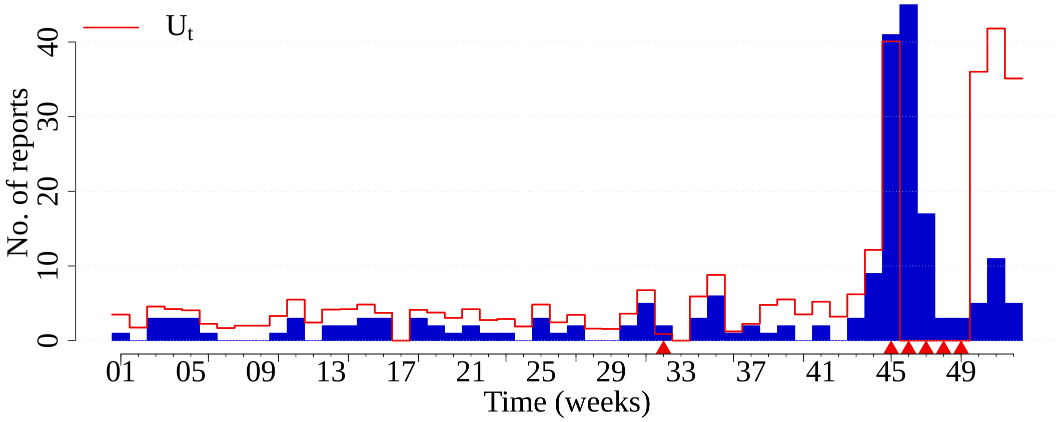


Figure 2.1: Weekly reports of S. Newport in Germany in 2011 monitored by the EARS C1 method. The line represents the upperbound calculated by the prospective algorithm. Triangles indicate alarms that are the timepoints where the observed number of counts is higher than the upperbound.

We implemented the three methods C1, C2 and C3 in the `R` package `surveillance` as the function `earsC` in which one can choose which method to apply to the time series stored in a `sts`-object, which range of timepoints to monitor, and which confidence level α to use (Salmon et al., 2016). Below is R code where one monitors the counts of *Salmonella* Newport for Germany, `salmNewportGermany`, chooses the range of timepoints to monitor as `in2011`, specifies this range in `control` along with the chosen method, C1, and the chosen value for α, 0.05. The output of the `earsC` function is stored in the `sts`-object `surv` shown in Figure 2.1.

```
# Aggregate counts over Germany
R> salmNewportGermany <- aggregate(salmNewport, by = "unit")
# Range for the monitoring
R> in2011 <- which(isoWeekYear(epoch(salmNewport))$ISOYear == 2011)
# Choose parameters
R> control <- list(range = in2011, method = "C1", alpha = 0.05)
# Apply earsC function
R> surv <- earsC(salmNewportGermany, control = control)
```

Figure 2.1 shows the upperbound as a solid line and the alarms – timepoints where the upperbound has been exceeded – as triangles. The four last alarms correspond to a known outbreak in 2011 due to sprouts (Bayer et al., 2014). One sees that the upperbound right after the outbreak is affected by the outbreak: it is very high, so that a smaller outbreak would not be detected.

The EARS methods C1, C2 and C3 are simple in that they only use information from the very recent past. This is appropriate when data has only been collected for a short time or when one expects the count to be fairly constant. However, data from the less recent past often encompasses relevant information about *e.g.*, seasonality and time trend, that one should take into account when estimating the expected count and the associated threshold. For instance, ignoring an decreasing time trend due to reasons other than an outbreak could decrease sensitivity. Inversely, overlooking an annual surge in counts during the summer could decrease specificity. Therefore, it is advisable to use detection methods whose underlying models incorporate essential characteristics of time series of disease count data such as overdispersion, seasonality, time trend and presence of past outbreaks in the records (Unkel et al., 2012; Shmueli and Burkom, 2010). Moreover, the EARS methods do not compute a proper prediction interval for the current count, as the upperbound is based on the assumption that the number of cases is normally distributed. However, since these methods were already implemented in surveillance systems (Fricker et al., 2008) our implementation can be useful for comparing newer methods to these commonly used methods. Moreover, it is the first implementation of these methods that is widely available. Sounder statistical methods will be reviewed in this chapter, after we explain how aberration detection algorithms can be evaluated.

2.2 Evaluation of detection

2.2.1 Measures of performance

Statistical algorithms for aberration detection can be evaluated with respect to their capacity to detect the most real aberrations (*sensitivity*), to produce the least amount of false alarms (*specificity*) and to do this quickly, *e.g.* detecting the begin rather than the peak of an outbreak (*timeliness*). Measures of performance necessitate the knowledge or assumption of timepoints corresponding to *in-control* or *out-of-control* situations, that is, knowing whether the count in each week corresponds to an outbreak. Only when having this information, one can know whether an alarm is a false alarm. Each measure can be defined for each time series to be monitored. In a public health institute as the RKI, epidemiologists tend to think at the scale of a pathogen, so that they would like to have given values of measures of specificity, sensitivity and timeliness for a given pathogen.

Specificity can be characterized by the false positive rate (FPR) defined in Noufaily et al. (2013) as the number of alarms in weeks without outbreaks divided by the number of weeks in the monitored period. An equivalent measure in statistical process control literature is the ARL0, the average run length of the algorithm in absence of any aberration before it gives an alarm. The longer the ARL0, the more specific the algorithm. Conversely, sensitivity can be characterized by the probability of detection (POD) which is the number of detected outbreaks divided by the number of outbreaks in the monitored time series. An outbreak is deemed detected is there is at least an alarm corresponding to one of the outbreak weeks. Sensitivity can also be characterized by the ARL1, the average run length of the algorithm in the presence of an aberration before it gives an alarm. The shorter the ARL1, the more sensitive the algorithm. Note that in real life, one hardly knows the distribution of *e.g.* changepoints so that any value of a measure actually is an average of values of the measure over several monitored time series with different characteristics, instead of an expectation which would be their mathematical definition as explained in (Frisén, 2003). At the RKI, epidemiologists rather use POD and FPR than ARL0 and ARL1 which are measures from the field of SPC. Lastly, measures of sensitivity and specificity could be combined for producing positive or negative predictive values similar to the ones defined for diagnostic tests, or for producing utility functions if one were to define costs associated to missed outbreaks and false alarms (Frisén, 2003). These are not issues that we shall pursue in this thesis.

Timeliness could be defined as the time after the beginning of an outbreak at which an alarm was produced if an alarm was produced (Salmon et al., 2015) or as binary variable indicating whether an alarm was produced in a pre-defined time (Unkel et al., 2012).

2.2.2 Data for evaluation

Evaluating an algorithm for aberration detection does not simply demand the definition of performance measures but also the use of time series on which to apply the methods to be compared. Finding data for assessing an algorithm is not straightforward, as discussed in Unkel et al. (2012). On the one hand simulated time series do not necessarily reproduce all features of real time series, including the form of outbreaks, but on the other hand in real time series one does not know any or all outbreaks. When using simulated time series, one tends to pre-validate given assumptions of the methods one wants to test: in Fricker et al. (2008) data are simulated from continuous distributions for testing the EARS C methods disregarding the count nature of surveillance data, in Noufaily et al. (2013) and Salmon et al. (2015) data are simulated without autocorrelation for testing algorithms disregarding autocorrelation. This is quite logical: if a researcher knows a characteristics of surveillance data and deems it to be important, they are more likely to include them in the aberration detection algorithm. Using real time series and adding simulated outbreaks is also suggested in Unkel et al. (2012); Buckeridge et al. (2008) but simulated outbreaks still are simulated and the real time series used as in-control baseline may contain outbreaks. Therefore, there is no ideal solution. In any case, any study assessing an algorithm should thoroughly present the data used and, if applicable, simulation mechanisms, since their characteristics can partly explain the measured performance of any algorithm.

There is up to now no public data set that would serve as a reference for method developers. The data used in Hutwagner et al. (2005) was no longer available online at the link provided in the corresponding article at the time we did the work of Chapter 3. However, the dataset is actually downloadable from an archive website[1]. But tracing the dataset was not straightforward, therefore it would be hard to establish this dataset as a

[1] https://web.archive.org/web/20130503125219/http:/www.bt.cdc.gov/surveillance/ears/datasets.asp

reference. The website of the Project Mimic that aimed at ”making realistic biosurveillance data available for researchers and to enable others to do the same” has not been updated since 2008[2]. The data used in Noufaily et al. (2013) and slightly modified in Salmon et al. (2015) represents a diversity of time series but disregard autocorrelation when simulating the time series. In conclusion, there is a lack of reference surveillance data sets.

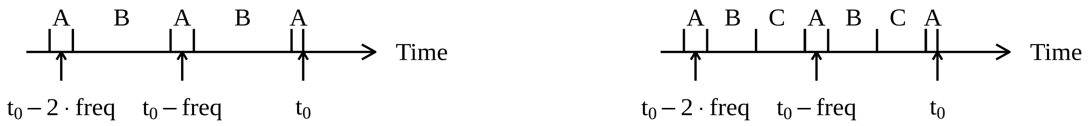
Assessing algorithms does not simply serve the role of determining how a new method compares to previous methods but can also guide the choice of parameters values, as shall be seen in Section 2.5. Moreover, these assessment criteria characterize the signals produced by an algorithm, but algorithms for aberration detection only make a difference in practice if they are implemented in a system offering good communication of the signals. An algorithm can be very good, but if its code is not easy to reproduce and/or not available, it will not be used. Moreover, even if an algorithm is very good and freely available, it will hardly make a difference in practice if it is not integrated into a system encompassing data queries, computation of signals, and delivery of reports to persons, *e.g.* epidemiologists, that can interpret the signals and act upon them. See Chapter 4 for a presentation and discussion of such a system.

In the following parts of this chapter, we present a set of representative algorithms, which are already in routine application at several public health institutions or which we think have the potential to become so. First we describe the Farrington method introduced by Farrington et al. (1996) together with the improvements proposed by Noufaily et al. (2013). As a Bayesian counterpart to these methods we present the BODA method published by Manitz and Höhle (2013) which allows the easy integration of covariates. All these methods perform one-timepoint detection in that they detect aberrations only when the count at the currently monitored timepoint is above the computed threshold, without memory of the comparison of the counts at the previous timepoints to corresponding thresholds. Hence, no accumulation of evidence takes place. As an extension, we introduce an implementation of the negative binomial cumulative sum (CUSUM) of Höhle and Paul (2008) that allows the detection of sustained shifts by accumulating evidence over several timepoints.

2.3 One size fits them all for count time series

The Farrington method (Farrington et al., 1996) along with its improved version (Noufaily et al., 2013) are currently *the* methods of choice at European public health institutes (Hulth et al., 2010; Enki et al., 2013; Salmon et al., 2016). Both methods are implemented in the R package `surveillance`. First, the 1996 method as described in Farrington et al. (1996) is implemented as the function `farrington`. Its use was already described in Höhle (2007) and in Höhle and Mazick (2010). Now, the newly implemented function `farringtonFlexible` that we added to the `surveillance` package supports the use of this *original method* as well as of the *improved method* built on suggestions made by Noufaily

[2] http://www.projectmimic.com/



(a) `noPeriods` $= 2$

(b) `noPeriods` $= 3$

Figure 2.2: Construction of the `noPeriods`-level factor to account for seasonality, depending on the value of the half-window size w and of the freq of the data. Here the number of years to go back in the past b is 2. Each level of the factor variable corresponds to a period delimited by ticks and is denoted by a character. The windows around s are respectively of size $2w + 1$, $2w + 1$ and $w + 1$. The segments between them are divided into the other periods so that they have the same length up to rounding.

et al. (2013) for improving the specificity without reducing the sensitivity.

2.3.1 Definition of the GLM

In both cases the steps of the algorithm are the same. In a first step, an overdispersed Poisson generalized linear model with log link is fitted to the reference data $\boldsymbol{y}_s \subseteq \{y_t\, ;\, t \leq s\}$, where $\mathrm{E}(y_t) = \mu_t$ with $\log \mu_t = \alpha + \beta t$ and $\mathrm{Var}(y_t) = \phi \cdot \mu_t$ and where $\phi \geq 1$ is ensured. The original method took seasonality into account by using a subset of the available data as reference data for fitting the GLM: `w` timepoints centred around the timepoint located $1, 2, \ldots, b$ years before s, amounting to a total $b \cdot (2w + 1)$ reference values. However, Noufaily et al. (2013) found that the algorithm performs better when using more historical data. In order to do so while taking seasonality into account, the authors introduced a zero order spline with 11 knots, which can be conveniently represented as a 10-level factor. We have extended this idea in our implementation so that one can choose an arbitrary number of periods in each year. Thus, $\log \mu_t = \alpha + \beta t + \gamma_{c(t)}$ where $\gamma_{c(t)}$ are the coefficients of a zero order spline with `noPeriods` $+ 1$ knots, which can be conveniently represented as a `noPeriods`-level factor that reflects seasonality. Here, $c(t)$ is a function indicating to which season or period of the year t belongs.

The algorithm uses the parameters `w`, `b` and `noPeriods` to deduce the length of periods so they have the same length up to rounding. An exception is the reference window centred around s. Figure 2.2 shows an example, where each character corresponds to a different period. Note that setting `noPeriods` $= 1$ corresponds to using the original method with only a subset of the data: there is only one period defined per year, the reference window around s and other timepoints are not included in the model. Moreover, Noufaily et al. (2013) found that it is better to exclude the last 26 weeks before s from the baseline in order to avoid reducing sensitivity when an outbreak has started recently before s. In the `farringtonFlexible` function, one controls this by specifying `pastWeeksNotIncluded`, which is the number of last timepoints before s that are not to be used. The default value is 26.

2.3.2 Definition of the threshold

In a second step, the expected number of counts μ_s is estimated for the current timepoint s using this GLM. An upperbound U_s is calculated based on this estimated value and its variance. The two versions of the algorithm make different assumptions for this calculation.

The original method assumes that a transformation of the prediction error $g\left(y_s - \hat{\mu}_s\right)$ is normally distributed. There is an optional power-transformation for skewness correction and variance stabilisation. When using the identity transformation $g(x) = x$ one obtains

$$y_s - \hat{\mu}_0 \sim \mathcal{N}(0, \mathrm{Var}(y_s - \hat{\mu}_0)).$$

The upperbound of the prediction interval is then calculated based on this distribution. First we have that

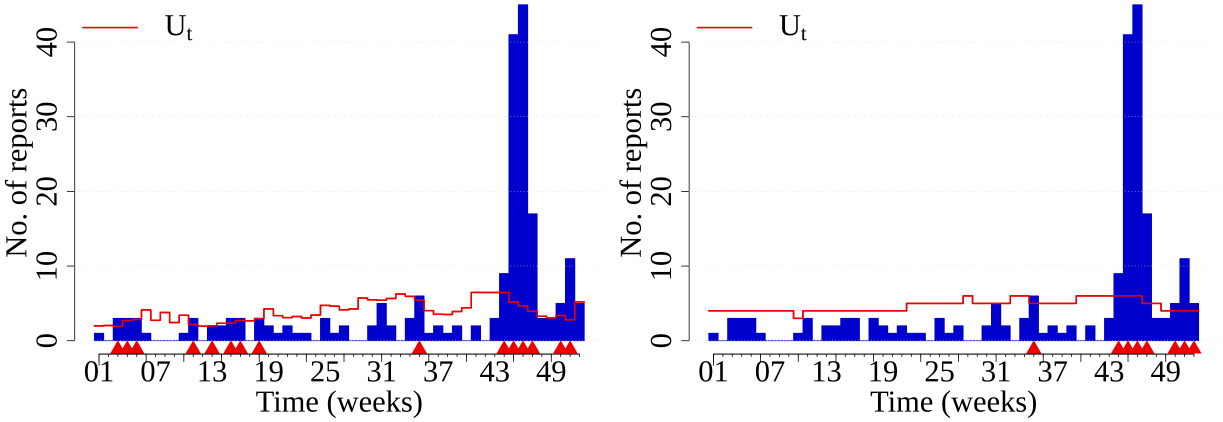
$$\mathrm{Var}(y_s - \hat{\mu}_s) = \mathrm{Var}(y_s) + \mathrm{Var}(\hat{\mu}_s) = \phi\mu_0 + \mathrm{Var}(\hat{\mu}_s)$$

with $\mathrm{Var}(y_s)$ being the variance of an observation and $\mathrm{Var}(\hat{\mu}_s)$ being the variance of the estimate. The threshold, defined as the upperbound of a one-sided $(1-\alpha)\cdot 100\%$ prediction interval, is then

$$U_s = \hat{\mu}_0 + z_{1-\alpha}\widehat{\mathrm{Var}}(y_s - \hat{\mu}_s).$$

However, a weakness of this procedure is the normality assumption itself, so that an alternative was presented in Noufaily et al. (2013). The central assumption of this approach is that, although using a quasi-Poisson GLM, Noufaily et al. (2013) assume that $y_s \sim \mathrm{NB}\left(\mu_s, \nu\right)$, with μ_s the mean of the distribution and $\nu = {}^{\mu_s}/_{\phi - 1}$ its overdispersion parameter. In this parameterization, we still have $\mathrm{E}(y_t) = \mu_t$ and $\mathrm{Var}(y_t) = \phi \cdot \mu_t$ with $\phi > 1$ – otherwise a Poisson distribution is assumed for the observed count. The threshold is defined as a quantile of the negative binomial distribution with *plug-in* estimates $\hat{\mu}_s$ and $\hat{\phi}$. Only minor differences in performance were noticed in Noufaily et al. (2013) when using a negative binomial GLM. Note that the *plug-in* definition of the threshold disregards the estimation uncertainty in $\hat{\mu}_s$ and $\hat{\phi}$.

As a consequence, we implemented a third method in the `farringtonFlexible` function, called with the value `muan` (*mu* for μ and *an* for asymptotic normal) for the `thresholdMethod`, that tries to solve the problem by using the asymptotic normal distribution of $(\hat{\alpha}, \hat{\beta})$ to derive the upper $(1-\alpha)$-quantile of the asymptotic normal distribution of $\hat{\mu}_s = \hat{\alpha} + \hat{\beta}s$. Note that this does not reflect all estimation uncertainty because it disregards the estimation uncertainty of $\hat{\phi}$. Note also that for time series where the variance of the estimator is large, the upperbound also ends up being very large. Thus, the method described by Noufaily et al. (2013) for calculating the threshold seems to provide information that is easier to interpret by epidemiologists but with the one we proposed being more statistically correct.



(a)

(b)

Figure 2.3: S. Newport in Germany in 2011 monitored by (a) the original method and (b) the improved method. In both cases we turned off the option that the threshold is only computed if there were more than 5 cases during the 4 last timepoints including s.

2.3.3 Inference and monitoring

In a last step, the observed count y_s is compared to the upperbound U_s and an alarm is raised if $y_s > U_s$. In both cases the fitting of the GLM involves three important steps. First, the significance of the time trend is checked. The time trend is included only when significant at a chosen level, when there are more than three years reference data and if no overextrapolation occurs because of the time trend. Then, past outbreaks are reweighted based on their standardized Anscombe residuals. The Anscombe residual associated to the observation y_t with estimated mean μ_t is (Anscombe, 1953; McCullagh and Nelder, 1983):

$$r_t = \frac{A(y_t) - A(\mu_t)}{A'(\mu_t)\sqrt{V(\mu_t)}}, \quad A(\mu_t) = \int_{x=0}^{\mu_t} V(x)^{-1/3} dx$$

where V is the variance function and where $A'(\mu_t)$ is the derivative of $A(\mu_t)$ with respect to μ_t. For standardizing them, one has to divide them by $\sqrt{\phi(1-h_{tt})}$ where h_{tt} is a diagonal element of the hat matrix of the regression.

For a quasi-Poisson GLM, since we have $V(\mu_t) = \phi\mu_t$, we get

$$r_t = \frac{3}{2}\frac{y_t^{2/3} - \mu_t^{2/3}}{\mu_t^{1/6}\sqrt{\phi(1-h_{tt})}}.$$

In `farringtonFlexible` the limit L_r for reweighting past counts can be specified by the user. If the standardized Anscombe residual of a count is higher than the limit it is reweighted accordingly in a second fitting of the GLM. More mathematically, in the second fit a weight w_t is associated to each observation y_t depending on its standardized Anscombe residual r_t from the first fit:

$$w_t = \gamma r_t^{-2\mathbb{1}(r_t>L_r)}, \quad \gamma = \frac{N}{\sum_{i=1}^{N} r_t^{-2I(r_t>L_r)}}$$

where $\mathbb{1}$ is the indicator function and with N the number of observations used for fitting the model. Farrington et al. (1996) used $L_r = 1$ whereas Noufaily et al. (2013) advise a value of $L_r = 2.56$ so that the reweighting procedure is less drastic, because it also shrinks the variance of the observations. In our case, as we want to get a quantile of the predictive distribution of the current count, estimating the overdispersion adequately is quite crucial. Therefore, this reweighting procedure could be replaced by directly using robust inference in a single fit of the model (Aeberhard et al., 2014; Elsaied and Fried, 2014). Lastly, note that as a protection against alarms related to small counts that are unlikely to be interesting to epidemiologists, the threshold is only computed if there were more than 5 cases during the 4 last timepoints including s.

2.3.4 Implementation of the algorithm

In the function `farringtonFlexible` that we implemented during this thesis work, one can choose to use the original method or the improved method by specification of appropriate options in the `control` slot. In the example below, `control1` corresponds to the use of the original method and `control2` indicates the options for the improved method.

```
R> control1 <- list(range = in2011, noPeriods = 1, b = 4, w = 3,
   weightsThreshold = 1, pastWeeksNotIncluded = 3, pThresholdTrend = 0.05,
   thresholdMethod = "delta")
R> control2 <- list(range = in2011, noPeriods = 10, b = 4, w = 3,
   weightsThreshold = 2.58, pastWeeksNotIncluded = 26, pThresholdTrend = 1,
   thresholdMethod = "nbPlugin")
R> salm.farrington <- farringtonFlexible(salmNewportGermany, control1)
R> salm.noufaily <- farringtonFlexible(salmNewportGermany, control2)
```

Note that in the new implementation a population offset can be included in the GLM by setting `populationBool` to `TRUE` and supplying the possibly time-varying population size in the `population` slot of the `sts`-object, but this will not be discussed further here. A good offset in the example of monitoring the weekly number of salmonellosis cases could be the weekly number of diagnosis tests performed, if it were known.

2.3.5 Practical usage of the algorithm

The original method is widely used in public health surveillance (Hulth et al., 2010). It is implemented at the Robert Koch Institute in the automatic system for outbreak detection (Salmon et al., 2016) described in Chapter 4. The reason for its success is primarily that it does not need to be *fine-tuned* for each specific pathogen. It is hence easy to implement it for scanning data for many different pathogens. Furthermore, it does tackle classical issues of surveillance data: overdispersion, presence of past outbreaks that are reweighted, seasonality that is taken into account differently in the two methods. An example of use of the function is shown in Fig. 2.3. Note that the newer method uses more data in the regression, and produces integer threshold values. One gets less alarms with the most recent method and still does not miss the outbreak in the summer. Simulations performed by Noufaily et al. (2013) support the use of the improved method instead of the original method.

2.4 An algorithm based on a Bayesian GAM

The next algorithm we shall present has several interesting features, in that it is more tailored at the time series aspect of monitoring, and in that its model is defined in a Bayesian framework. It is called BODA (`boda` in the R package `surveillance`) for *Bayesian outbreak detection algorithm*. It is a regression-based method where the regression model is a GAM fitted with integrated nested Laplace approximations (INLA) (Rue et al., 2009; Rue et al., 2015), and was presented in Manitz and Höhle (2013).

2.4.1 Definition of the GAM

The GAM in BODA is based on the negative binomial distribution with time-varying expectation and time constant overdispersion parameter, *i.e.*, the parameterization is such that

$$y_t \sim \mathrm{NB}(\mu_t, \nu)$$

with μ_t the mean of the distribution and ν the dispersion parameter (Lawless, 1987). The density of the distribution is $P(y_t = x) = \Gamma(x+\nu)/(\Gamma(\nu)x!)p^{\nu}(1-p)^{x}$ where $p = \nu/(\nu+\mu_t)$. One can interpret y_t as the number of failures occurring in a sequence of Bernoulli trials with probability of success p before a target number of successes, ν, is reached, although ν does not need to be an integer.

Hence, we have $\mathrm{E}(y_t) = \mu_t$ and $\mathrm{Var}(y_t) = \mu_t \cdot (1 + \mu_t/\nu)$. The linear predictor of the GAM is given by

$$\log(\mu_t) = \alpha_{0t} + \beta t + \gamma_t + \boldsymbol{x}_t^{\top}\boldsymbol{\delta} + \xi z_t, \quad t = 1, \ldots, s.$$

Here, the time-varying intercept α_{0t} is described by a random walk. Three models are proposed for describing the random walk prior with precision λ_α such that $\alpha_{0t} \sim N(g_m(t), \lambda_\alpha^{-1})$:

- stationary model $g_m(t) = \alpha_0$,
- neighbour model $g_m(t) = \alpha_{0,t-1}$,

- linear model $g_m(t) = 2\alpha_{0,t-1} - \alpha_{0,t-2}$.

The time-varying intercept in the neighbour and the linear models was inspired by the work of Heisterkamp et al. (2006) on hierarchical time series models, and accounts for temporal autocorrelation which is ignored in the Farrington method.

Then, γ_t denotes a seasonal effect (as implemented in `INLA`) with period s equal to the periodicity of the data, defined such that $\sum_{i=0}^{s-1} \gamma_i \sim N(0, \lambda_\gamma^{-1})$ with λ_γ^{-1} the precision of the seasonality effect. Furthermore, β characterizes the effect of a possible linear trend (on the log-scale) and ξ is the effect of previous outbreaks which one can include if one knows for each timepoint if there was an outbreak at this timepoint. Typically, z_t is a binary process denoting if there was an outbreak in week t, but more involved adaptive and non-binary forms are imaginable. Finally, $\boldsymbol{x}_t$ denotes a vector of possibly time-varying covariates, which influence the expected number of cases. For instance, if we expected the number of influenza cases to be naturally driven by temperature, we could add temperature as a covariate in order to monitor the excess cases of influenza instead of both temperature-related and excess influenza cases.

2.4.2 Definition of the threshold

As before, data from the previous timepoints $1, \dots, s-1$ is used to determine the posterior distribution of all model parameters and subsequently the posterior predictive distribution of y_s is computed. If the actual observed value of y_s is above the $(1-\alpha)$-quantile of the predictive posterior distribution an alarm is flagged for s. Using the posterior predictive distribution of y_s allows to take into account both estimation and observation uncertainty. Inference for the posterior is performed with INLA as described in Rue et al. (2009) using the R INLA package (Rue et al., 2015).

The example in Manitz and Höhle (2013) is the time series of weekly reported Campylobacter cases in Germany as represented in Figure 2.4. The authors state that there is a strong association between the number of cases and humidity so that humidity was added to the model as a covariate to control for its effect on the number of cases. If one were not sure whether the seasonality in the number of cases is induced by the seasonality in humidity, one could decompose the time series of humidity so that one only includes the season-adjusted time series in the model, as proposed by Erbas and Hyndman (2000). Then one would not get a spurious relationship between the response variable (number of cases, or number of hospital admissions in Erbas and Hyndman (2000)) and a dependent variable that originally has a collinear seasonality component. For decomposing time series of dependent variables, Erbas and Hyndman (2000) use the R function `stl` that applies the Loess smoother for decomposing the time series in trend, season and irregular component. Note that such an analysis is beyond the scope of this thesis. Moreover, Manitz and Höhle (2013) did not question the relation between the time series of weekly reported Campylobacter cases and the time series of humidity, including seasonality.

Other covariates are three binary variables. The first one indicates whether the timepoint is during the 2011 STEC O104:H4 outbreak. The motivation to do so is that for all

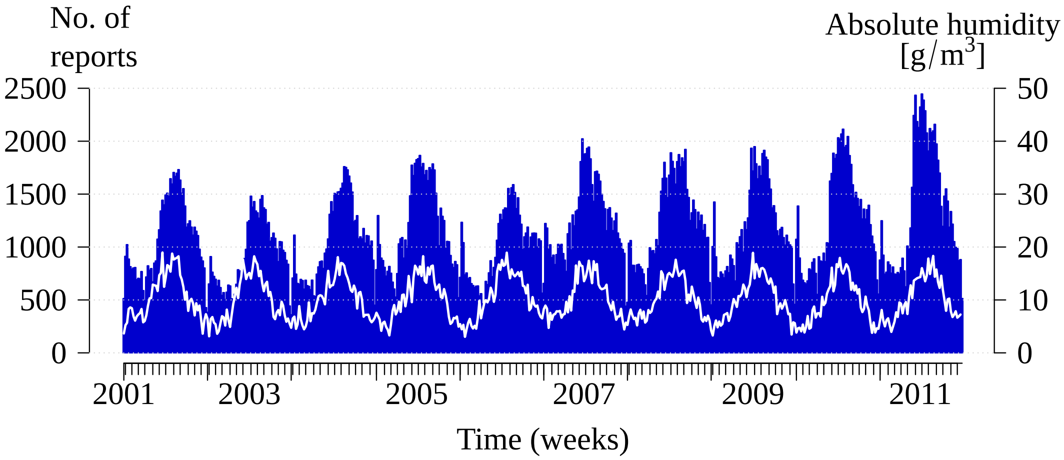


Figure 2.4: In black are shown the weekly number of reported campylobacteriosis cases in Germany 2002-2011 as vertical bars. In addition, the corresponding mean absolute humidity time series is shown as a white curve.

gastro-intestinal diseases there were more diagnosed cases in this period, probably because of increased awareness and testing (Bernard et al., 2014). The authors add two other binary variables indicating whether the timepoint is situated during the two last weeks of the year or the two first week of the years. The later two variables are needed, because there is a systematically changed reporting behaviour at the turn of each year due to holidays delaying visits to the doctor to the beginning of the following year: one does not want to get alarms for an usual peak whose origin is known. In this model, no additional correction for past outbreaks is made since there is no reliable information about this. We show the results of this monitoring, with a stationary model for the intercept, on Figure 2.5.

2.4.3 A Bayesian algorithm

This algorithm uses a Bayesian GAM and outputs a threshold taking both estimation and observation uncertainty into account, thus being a step towards more Bayesian thinking in the field of aberration detection. In the SPC literature, works such as Lee and Apley (2011) or Psarakis et al. (2014) also support the use of Bayesian methods for taking estimation uncertainty into account, thus reducing the rate of false alarms in control charts. Taking estimation uncertainty into account makes monitoring more conservative, since the predictive posterior distribution is wider than it would be if disregarding estimation uncertainty.

However, this algorithm could take even more advantage of Bayesian methodological developments. We are going to cite two examples. Inference could be made sequential so that the threshold calculation at each timepoint would not be made from scratch, following

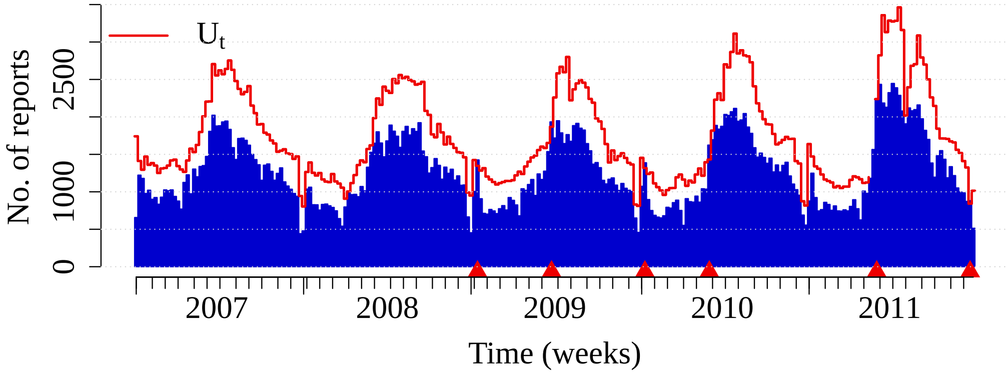


Figure 2.5: Weekly reports of Campylobacter in Germany in 2007-2011 monitored by the BODA method with covariates. The line represents the upperbound calculated by the algorithm. Triangles indicate alarms, *i.e.*, timepoints where the observed number of counts is higher than the upperbound.

approaches such as the one described by Bhattacharya and Wilson (2015) that is not advanced enough yet to tackle the inference of a model with as many parameters as the one we presented here. Then, one could reformulate the decision making with the use of utility or loss functions (Berger, 2013). Such developments are, though, beyond the scope of this thesis.

2.4.4 Practical usage of the algorithm

The BODA method is to be seen as a step towards more Bayesian thinking in aberration detection. However, it has two prohibitive characteristics as regards routine application: one would need to carefully choose an appropriate model for each time series and the procedure is quite slow even if INLA is faster than MCMC inference for example. As a response Salmon et al. (2015) introduce a method which has two advantages: it allows to adjust outbreak detection for reporting delays and includes an approximate inference method much faster than the INLA inference method. However, its linear predictor is more in the style of Noufaily et al. (2013) not allowing for additional covariates or time-varying intercept. We present this other method in Chapter 3.

2.5 An algorithm that goes beyond one-timepoint detection

The GLMs or GAMs used in the Farrington or the BODA method are suitable for the purpose of aberration detection since their regression approach adjusts counts for known phenomena such as trend or seasonality in surveillance data. Nevertheless, they only perform one-timepoint detection. In some contexts it can be more relevant to detect sustained shifts early, *e.g.*, an outbreak could be characterized at first by counts slightly higher than usual in subsequent weeks without each weekly count being flagged by one-timepoint detection methods. We shall first present a method implemented at the CDC, before moving on to a more modern and statistically sounder method.

2.5.1 A simple but flawed method

The EARS C3 method allows to perform three-timepoint detection. It is quite different from the two other methods – C1 defined in Section 2.1 and C2. For C3, the baseline are timepoints t-11 to t-3.

The statistic $C_3(t)$ is the sum of discrepancies between observations and predictions.

$$C_3(t) = \sum_{i=t}^{t-2} \max(0, C_2(i) - 1).$$

$C_2(i)$ is defined by

$$C_2(i) = \frac{y(i) - \bar{y}_2(i)}{S_2(i)},$$

where

$$\bar{y}_2(i) = \frac{1}{7} \sum_{j=i-3}^{i-9} y(j)$$

and

$$S_2^2(i) = \frac{1}{6} \sum_{j=i-3}^{i-9} [y(j) - \bar{y}_2(j)]^2.$$

The fact that one adds $C_2(i)-1$ instead of $C_2(i)$ must be due to the parallel that Hutwagner et al. (2003) draw between the C3 method and a method based on a statistic A_t such that

$$A_t = \max\left(0, A_{t-1} + \frac{y_t - (\bar{y}_t + kS_t)}{S_t}\right)$$

where k is called ”the detectable shift from the mean”. Thus, the 1 in the formula of $C_3(t)$ could correspond to $k = 1$, although $C3$ only has memory of deviations from the

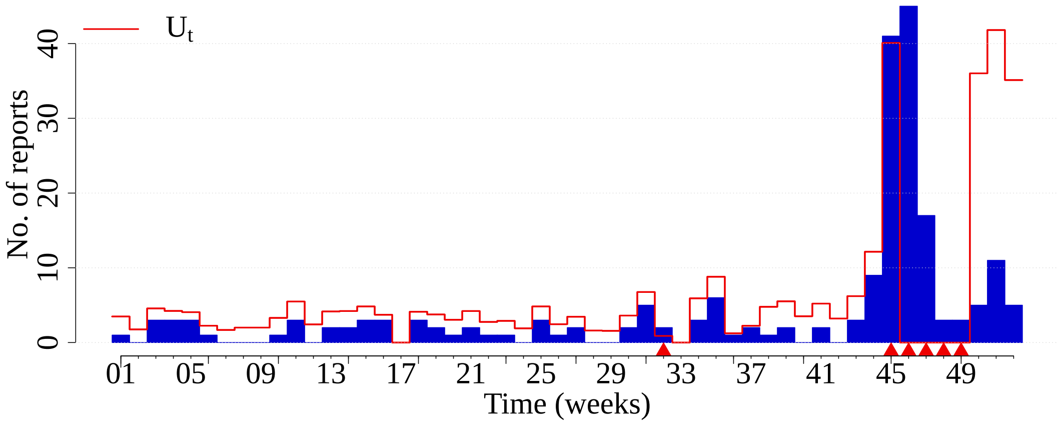


Figure 2.6: Weekly reports of S. Newport in Germany in 2011 monitored by the EARS C3 method.

expectation over 3 timepoints, whereas a CUSUM generally adds deviations until one gets an alarm. Please note that the EARS methods are not very well documented.

Then, under the null hypothesis of no outbreak, one assumes that

$$C_3(t) \sim N(0, 1).$$

An alarm is raised if $C_3(t) \geq z_{1-\alpha}$, with $z_{1-\alpha}$ the $(1-\alpha)$-quantile of the centered reduced normal law. The upperbound $U_3(t)$ is then defined by:

$$U_3(t) = \bar{y}_2(t) + S_2(t)\left(z_{1-\alpha} - \sum_{i=t-1}^{t-2} \max(0, C_2(i) - 1)\right).$$

An example is shown on Figure 2.6. This method does indeed go beyond one-timepoint detection which is a wish often expressed by epidemiologists, and has a quite intuitive interpretation: how different from the moving average are the observations of the last three timepoints. Yet, beside having the downsides of EARS methods mentioned at the beginning of the chapter such as using only little information from the past, disregarding time trend and seasonality and and using a normal distribution to describe the outcome of a count variable, it uses an arbitrary number of timepoints over which to cumulate deviations from an expectation, which is not optimal. Therefore, we now present a sounder method created by Höhle and Paul (2008).

2.5.2 Definition of a modern method based on a control chart

Cumulative control charts inspired by statistical process control (SPC) *e.g.*, cumulative sums (CUSUM), would allow the detection of sustained shifts. A control chart defined in Hawkings and Olwell (1998) was already created for overdispersed count data $\{y_t,\ t = 1, 2, \ldots\}$ following the negative binomial distribution parameterized by the parameters r and c so that its mean is $\mu = r/c$ and so that is variance is $\text{Var}(y_t) = \mu(1+1/c)$. Hawkings and Olwell (1998) assume that ν is constant and monitor c for detecting a change from level c_0 to level c_1. Here we shall concentrate on the downward shifts of c since they indicate a surge in counts. One has to estimate c_0 and to choose c_1. The authors define a CUSUM in recursive form

$$l_0 = 0 \quad \text{and} \quad l_n = \max(0, l_{n-1} + y_n + k^-), \quad n \geq 1,$$

where

$$k^- = \frac{\nu \log\left(\frac{c_1(1+c_0)}{c_0(1+c_1)}\right)}{\log\left(\frac{1+c_0}{1+c_1}\right)}.$$

The stopping time for alarm is given by

$$N = \inf\{n : l_n \geq \texttt{c.ARL}\}$$

where `c.ARL` is chosen to ensure a given ARL0.

Yet this control chart is not tailored to the specific characteristics of surveillance data including a time-varying mean because of *e.g.* seasonality. The method presented in Höhle and Paul (2008) conducts a synthesis of both worlds, *i.e.*, traditional surveillance methods and SPC. The method is implemented in the package as the function `glrnb`. It allows to monitor a negative binomial variable with time-varying mean but constant ν overdispersion parameter as in the previous method from Hawkings and Olwell (1998). However, here the monitored parameter is μ_t, not c.

For the control chart, two distributions are defined, one for each of the two states of the system: *in-control* and *out-of-control*. The *in-control* distribution $f_{\boldsymbol{\theta}_0}(y_t|\boldsymbol{z}_t)$ with the covariates $\boldsymbol{z}_t$ is formulated by a GLM of the Poisson or negative binomial family with a log link, depending on the overdispersion of the data. In this context, the standard model for the *in-control* mean is

$$\log \mu_{0,t} = \beta_0 + \beta_1 t + \sum_{s=1}^{S}\left[\beta_{2s}\cos\left(\frac{2\pi st}{\texttt{Period}}\right) + \beta_{2s+1}\sin\left(\frac{2\pi st}{\texttt{Period}}\right)\right],$$

where S is the number of harmonic waves to use and `Period` is the period of the data, for instance 52 for weekly data. However, more flexible linear predictors, *e.g.*, containing splines, concurrent covariates or an offset could be used on the right hand-side of the equation. The GLM could therefore be made very similar to the one used by Noufaily et al. (2013), with reweighting of past outbreaks and various criteria for including the time trend, or one could use elements from time series models with feedback mechanism as proposed by Liboschik et al. (2014) for better accounting for autocorrelation in the time series.

The parameters of the *in-control* and *out-of-control* models are respectively given by $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$. The *out-of-control* mean is defined as a function of the *in-control* mean, either with a multiplicative shift (additive on the log-scale) whose size κ can be given as an input or re-estimated at each timepoint $t > 1$, $\mu_{1,t} = \mu_{0,t} \cdot \exp(\kappa)$, or with an unknown autoregressive component as in Held et al. (2005), $\mu_{1,t} = \mu_{0,t} + \lambda y_{t-1}$ with unknown $\lambda > 0$.

Timepoints are divided into two intervals: phase 1 and phase 2. The *in-control* mean and overdispersion are estimated with a GLM fitted on phase 1 data, whereas surveillance operates on phase 2 data. When λ is fixed, one uses a likelihood-ratio (LR) and defines the stopping time for alarm as

$$N = \min \left\{ s \geq 1 : \max_{1 \leq t \leq s} \left[\sum_{u=t}^{s} \log \left\{ \frac{f_{\boldsymbol{\theta}_1}(y_u|\boldsymbol{z}_u)}{f_{\boldsymbol{\theta}_0}(y_u|\boldsymbol{z}_u)} \right\} \right] \geq \texttt{c.ARL} \right\},$$

where c.ARL is the threshold of the CUSUM. This formula can be given in a recursiv form

$$l_0 = 0 \quad \text{and} \quad l_n = \max(0, l_{n+1} + \log \left\{ \frac{f_{\boldsymbol{\theta}_1}(y_n)}{f_{\boldsymbol{\theta}_0}(y_n)} \right\}), \quad n \geq 1,$$

with stopping rule

$$N = \inf \left\{ n : l_n \geq \texttt{c.ARL} \right\}.$$

When λ is unknown and with the autoregressive component one has to use a generalized likelihood ration (GLR) with the following stopping rule to estimate them on the fly at each time point so that

$$N_G = \min \left\{ s \geq 1 : \max_{1 \leq t \leq s} \sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \left[\sum_{u=t}^{s} \log \left\{ \frac{f_{\boldsymbol{\theta}}(y_u|\boldsymbol{z}_u)}{f_{\boldsymbol{\theta}_0}(y_u|\boldsymbol{z}_u)} \right\} \right] \geq \texttt{c.ARL} \right\}.$$

Thus, one does not make any hypothesis about the specific value of the change to detect, but this GLR is more computationally intensive than the LR.

2.5.3 Practical usage of the algorithm

For using such an algorithm one has two choices to make. First, one has to choose an *in-control* model that will be fitted on phase 1 data. The choice of the exact *in-control* model depends on the data under surveillance. Performing model selection is a compulsory step in practical applications. Then, one needs to tune the surveillance function itself, for one of the two possible change forms. One can choose either to set theta to a given value and thus perform LR instead of GLR. The value of theta has to be adapted to the specific context in which the algorithm is applied: how big are shifts one wants to detect optimally? Is it better not to specify any and use GLR instead?

The threshold c.ARL also has to be specified by the user. As explained in Höhle and Mazick (2010) one can compute the threshold for a desired run-length in control through direct Monte Carlo simulation or a Markov chain approximation. Lastly, as mentioned in

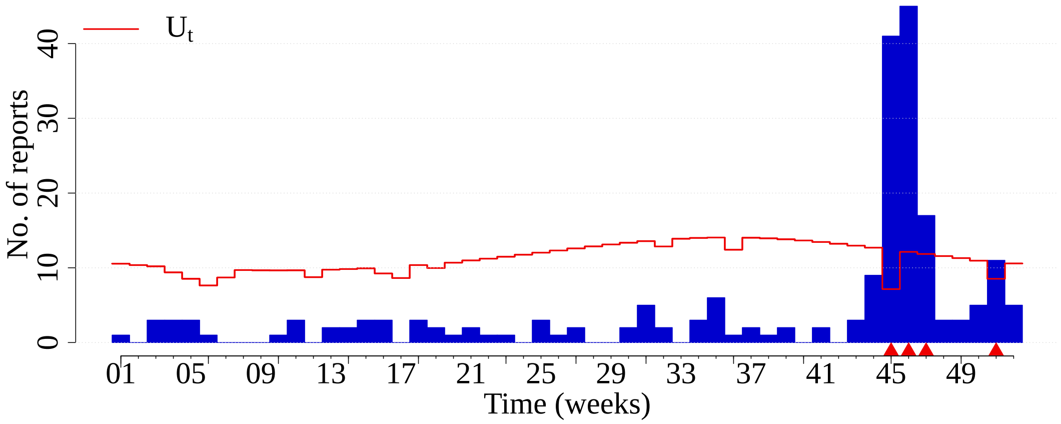


Figure 2.7: S. Newport in Germany in 2011 monitored by the `glrnb` function.

Höhle and Paul (2008), a window-limited approach of surveillance, instead of looking at all the timepoints until the first observation, can make computation faster.

In Salmon et al. (2016) we applied `glrnb` to the time series of report counts of *Salmonella Newport* in Germany by assuming a known multiplicative shift of factor 2 and an *in-control* model with one harmonic for seasonality and a trend. This model is refitted after each alarm, but first we used data from the years before 2011 as reference or `phase1`, and the data from 2011 as data to be monitored or `phase2`. The threshold `c.ARL` was chosen to be 4, as it was found with the same Monte Carlo approach as Höhle and Mazick (2010) that it made the probability of a false alarm within one year smaller than 0.1. This approach means drawing observations from the estimated *in-control* distributions and performing monitoring using different possible values of `c.ARL` over a grid, storing results for several draws at each value of `c.ARL`. Of course, it would be computationally intensive to perform such a Monte Carlo study for each time series monitored at a public health institute, so that one would maybe extrapolate a value for `c.ARL` from analyses of time series with similar characteristics. Figure 2.7 shows the results of this monitoring with `c.ARL` equal to 4.

The implementation of `glrnb` on individual time series was already thoroughly explained in Höhle and Mazick (2010). In Salmon et al. (2016) we moreover provide practical tips for the implementation of this function on huge amounts of data in public health surveillance applications. Issues of computational speed become important in such a context with thousands of time series. Our proposal to reduce the computational burden incurred by this algorithm is to compute the *in-control* model for each time serie (pathogen, subtype, subtype in a given location, *etc.*) only once a year and to use this estimation for the computation of a threshold for each time series. An idea to avoid starting with an initial value of zero in the CUSUM is to use either $(1/2) \cdot$ `c.ARL` as a starting value (fast initial response CUSUM as presented in Lucas and Crosier (1982)) or to let surveillance run with the new *in-control* model during a buffer period and use the resulting CUSUM as an initial value. One could also choose the maximum of these two possible starting values as a starting value. During the buffer period alarms would be generated with the model from the previous year. Lastly, using GLR is much more computationally intensive than using LR, whereas LR performs reasonably well on shifts different from the one indicated by `theta` as seen in the simulation studies of Höhle and Paul (2008). Therefore one should use LR with a reasonable predefined `theta`. The amount of historical data used each year to update the model, the length of the buffer period and the value of `theta` have to be fixed for each specific application, *e.g.*, using simulations and/or discussion with experts.

It is worth noting that even if it is well adapted for surveillance time series, the method that we have just presented is very different from the other ones presented in this chapter. Not only does it detect prolonged shifts instead of one-timepoint aberrations, but it is also based on a likelihood-ratio and calibrated for obtaining a given ARL0 rather than a given sensitivity. Thus, it stands out at a SPC method adapted to surveillance data.

2.6 Conclusion

In this chapter, we presented aberration detection and three methods that are representative of the state-of-the-art in the field of aberration detection: using generalized models tailored to time series for making predictions, taking both estimation and observation uncertainties into account, adapting tools from statistical process control. These methods are aimed at aberration detection in practice, *e.g.* in public health institutes.

Criteria for selecting a specific method in practice do not only include its performance and of the availability of an implementation, but also numerous other practical factors (Salmon et al., 2016). First one needs to ponder on the amount of historical data at hand – for instance the EARS methods (Fricker et al., 2008) only need data for the last timepoints whereas the Farrington and Noufaily methods (Noufaily et al., 2013; Farrington et al., 1996) use data up to b years in the past. Then one should consider the amount of past data used by the algorithm – historical reference methods use only a subset of the past data, namely the timepoints located around the same timepoint in the past years, whereas other methods use all past data included in the reference data. This can be a criterion of choice since one can prefer using all available data. It is also important to decide whether one wants to detect large and potentially short aberrations within one timepoint or whether one wants to be able to detect smaller and more prolonged shifts. Moreover, running time per time series is an important issue when analysing thousands of time series. Lastly, an important criterion is how much work needs to be done for finetuning the algorithm for each specific time series. Note that this decision must be coupled to planning the data queries and signal communication associated to the algorithm implementation in practice, because without such a system, output from any algorithm is useless. In Chapter 4 we present the system for routine aberration detection in place at the RKI.

On top of these practical considerations, methodological challenges remain for improving existing aberration detection algorithms, such as adding feedback mechanisms in GLMs used for monitoring to account for autocorrelation and adding even more Bayesian thinking in the field. Theoretical topics of aberration detection for count time series such as autocorrelation are discussed in Chapter 5. In the next chapter, Chapter 3, we present an extension of this work for dealing with another common problem of surveillance data, that is right-truncation because of reporting delays.

Chapter 3

Delay Correction for Monitoring of Infectious Diseases Counts

*This chapter up to Sect. 3.4.1 corresponds to the work presented in the article **M. Salmon**, D. Schumacher, K. Stark, M. Höhle. Bayesian Outbreak Detection in the Presence of Reporting Delays,* Biometrical Journal*, 57 (6), 1051-1067, 2015.*

Since 2001 the Protection against Infection Act regulates which infectious diseases are notifiable in Germany so that anonymised information about each diagnosed case is sent from local health authorities to the Robert Koch Institute (RKI) *via* federal health authorities (Faensen et al., 2006; Krause et al., 2007). The resulting database does not only support the exploration of disease counts for situational awareness and yearly summaries for health statistics, but also the detection of emerging outbreaks by an automated data analysis system (Salmon et al., 2016). However, case reports do not arrive immediately at the RKI because of inherent reporting delays, due to a time lag between disease onset and reporting date as well as to the decentralized structure of the reporting system. This can be a hindrance to the timely detection of outbreaks. For example Jones et al. (2014) and Noufaily et al. (2014) have analysed reporting delays in European reporting systems, respectively for Salmonella in France and 12 infections in the UK. Furthermore, Altmann et al. (2011) and Höhle and an der Heiden (2014) have analysed reporting delays during the German EHEC O104:H4 outbreak in particular, while Schumacher et al. (2016) have analysed the effect of a law change on transmission time. In this chapter, we investigate how to improve the real-time automated outbreak detection by adequately addressing reporting delays.

The available notification reports can be aggregated over time units (*e.g.* weeks) and other sublevels (*e.g.* pathogen subtype or age group) in order to obtain time series of reported incidence counts, which often display characteristics such as seasonality, overdispersion and presence of past outbreaks. Statistical algorithms for aberration detection can fit adequate models to these time series of counts and thus derive a predictive distribution for the current count. If the observed current count lies above a suitable quantile of this distribution, an alarm is flagged, prompting further checks by epidemiologists. A variety of statistical algorithms for aberration detection exist: see *e.g.*, the reviews in Buckeridge et al. (2005) and Unkel et al. (2012) and the selection presented in Chapter 2. Nearly all of them are of frequentist nature, but recently more Bayesian oriented proposals have emerged (Höhle, 2007; Martínez-Beneito et al., 2008; Conesa et al., 2015; Manitz and Höhle, 2013). One advantage of a Bayesian approach when calculating a predictive distribution is that this distribution takes into account both the uncertainty from estimation and the stochasticity of the model, that is on the one hand the uncertainty in the model parameters which we estimate based on a limited sample, and on the other hand the uncertainty resulting from natural fluctuations of the number of cases. In the present work, we shall build upon these strengths and hence try to move towards Bayesian thinking in the field of outbreak detection, especially because prediction is natural in a Bayesian framework.

The RKI automated system for aberration detection (Salmon et al., 2016), which will be the main motivation of our work, helps uncovering outbreaks of infectious diseases based on the German mandatory reporting data. In this system, aberration detection is performed on a daily basis for the weekly disease counts of hundreds thousands of time series and is based on the state-of-the-art algorithm proposed by Noufaily et al. (2013) which is an improvement of the widely adopted algorithm by Farrington et al. (1996) and which was presented in Section 2.3. Nevertheless, the response time of the system when there is an outbreak, a.k.a. the system's timeliness, does not only depend on the implemented algorithm but also on data quality. In Germany, each new case of a notifiable infectious disease has to pass several stages before arriving at the RKI database. This includes onset of disease symptoms in the patient, the patient's visit to the doctor, a diagnostic laboratory analysis and a report of the case to the local health authority followed by its notification to the federal health authority before it is finally notified at the RKI (see Figure 1.5). Moreover, each step within the reporting system requires the fulfilment of certain quality criteria before further transmission. Although technical, legal and managerial efforts are made in order to reduce these delays, reporting delays remain an issue.

Adjusting time series of public health events for reporting delays became important in biostatistics as part of modelling of the AIDS epidemic (Brookmeyer and Damiano, 1989; Kalbfleisch and Lawless, 1989; Zeger et al., 1989), but applications are also found in non-infectious modelling such as cancer registry data (Midthune et al., 2005) or mortality monitoring (Lin et al., 2008). Moreover, the adjustment for such occurred-but-not-yet-reported events (Lawless, 1994) has strong links to actuarial sciences, where it is part of claims reserve modelling (England and Verrall, 2002; Hess and Schmidt, 2002). Recently, such adjustments have re-emerged in public health settings under the name of *nowcasting* (Donker et al., 2011; Höhle and an der Heiden, 2014). In outbreak detection, we aim at comparing the current number of cases to a threshold computed from past data in order to see if the current number of cases is *unexpectedly high*. Using *nowcasting* would allow us to correct the current number of cases before the comparison to a threshold (Gergonne et al., 2011; Heisterkamp et al., 2006). In the present work we choose an alternative approach: we correct the threshold for reporting delays and hence compare the observed current number of cases to this threshold. This approach allows to have all sources of estimation uncertainties – from the prediction error as well as from the delay correction – on the same side of the comparison rather than to have on the one side a nowcasted number of observed cases with its uncertainty and on the other side a predicted number of cases with its own uncertainty, possibly correlated with each other. Additionally, the use of the actual observed number, rather than of some hypothetical estimate associated with uncertainty, is psychologically advantageous since epidemiologists do not need to make further investigations about the imaginary cases that a nowcasted number of cases automatically entails. Another recent work, which we shall present in Section 3.6, also follows this strategy of correcting the threshold, and uses a statistical test on a cumulative sum of discrepancies between the observed number of counts and a threshold corrected for reporting delays (Noufaily et al., 2015).

Altogether, our blueprint for a Bayesian outbreak detection algorithm taking reporting delays into account is a synthesis of the outbreak detection algorithms of Noufaily et al. (2013) and Manitz and Höhle (2013) where we extend the regression approach to the so-called reporting triangle, in which information about reporting delays is stored. The structure of the chapter is as follows. First, we state the problem of reporting delays in routine surveillance data, introduce adequate notation and illustrate it using RKI salmonellosis data in Sect. 3.2. The proposed algorithm, including numerical aspects of its implementation, is explained in Sect. 3.3 before being tested on simulated data in Sect. 3.4.1 and illustrated on real data in Sect. 3.4.2. We then present the R implementation of the algorithm in Sect. 3.5. Finally, we present another algorithm aiming at solving the same problem in Sect. 3.6, and a discussion in Sect. 3.7 rounds off the chapter.

3.1 Introducing reporting delays

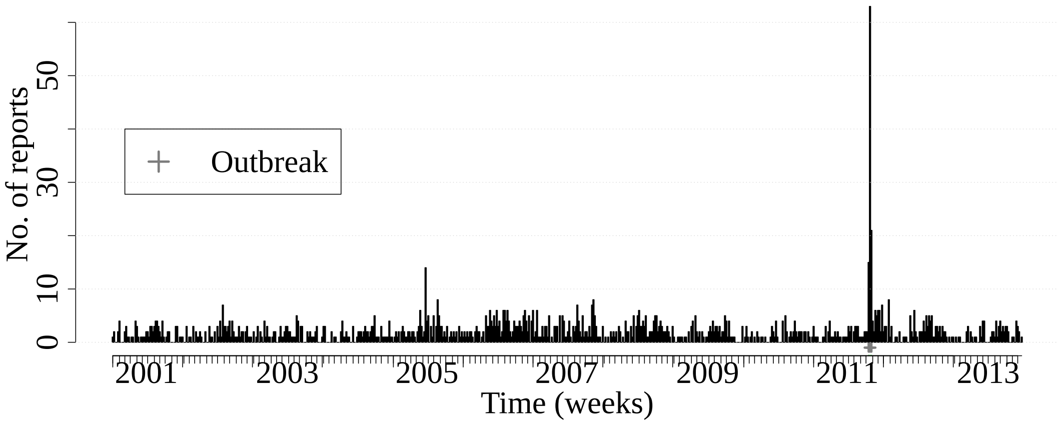


Figure 3.1: Weekly time series for all *Salmonella* Newport cases reported in Germany 2002-2013 by onset of disease (as available now in retrospect).

Throughout our work, we shall use notification data for *Salmonella enterica* subsp.

enterica serovar Newport (*Salmonella* Newport) to illustrate matters. Salmonellosis is a bacterial caused gastrointestinal disease, with symptoms such as diarrhea, fever and vomiting (Sánchez-Vargas et al., 2011). The Newport serovar is rather uncommon in Germany: in 2010 only 0.3% of all 25,307 notified *Salmonella* infections were of this serovar. Nevertheless, during the winter of 2011 there was a strong increase in the number of reported cases as shown in Fig. 3.1, which was due to an outbreak linked to mung bean sprouts (Bayer et al., 2014). We shall use this outbreak as illustration and motivation throughout our work.

Currently, monitoring at the RKI is performed on time series aggregated by date of report arrival at the local health authority. Nonetheless, as a supplement, there is an interest in monitoring cases based on their symptoms onset date: for Salmonella this would typically be the onset of diarrhea. One advantage is that onset dates, despite being self-reported by patients, provide a more precise description of the temporal extension of a possible outbreak. Unlike dates of report they escape noise introduced by delays, *i.e.* due to patients going to the doctor, shipping of samples to laboratories, as well as reporting artefacts such as the late discovery of cases through interviews of other cases. In the RKI S. Newport data, onset date is available for about 80% of all cases. Because of the additional delay between symptoms onset and reporting to local health authority, taking reporting delays into account becomes even more important when trying to monitor available time series based on this data. Hence, only a delay adjustment method could justify the additional costs incurred by this supplementary monitoring. For a case report i let

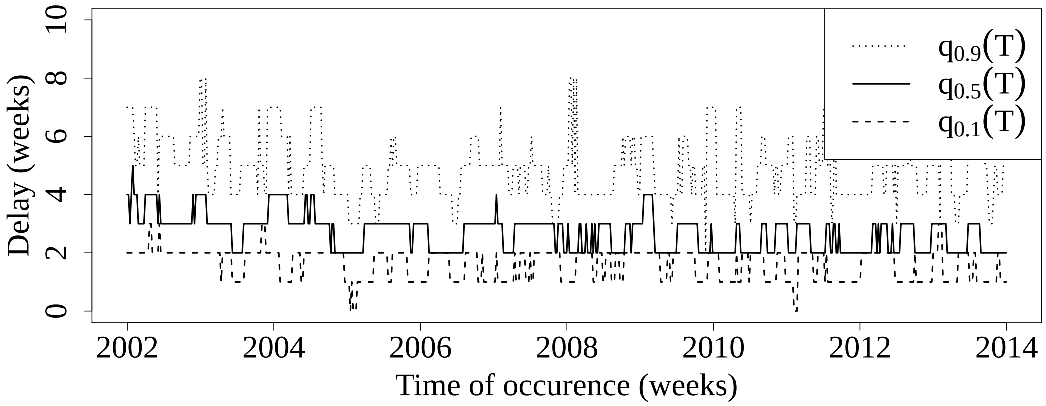


Figure 3.2: Median reporting delay between symptoms onset and arrival at the RKI (as well as 0.1 and 0.9 quantiles) of the smoothed empirical delay distribution as a function of time of occurrence, for all *Salmonella* Newport cases reported in Germany 2002-2013 (as available now in retrospect).

the tuple $\left(t_i^E, t_i^R\right)$ denote the time of the $e$vent of interest ($e.g.$ disease onset or receipt of case information at local health authority) and the time the notification of this event a$r$rives at the RKI, respectively. Let $d_i = t_i^R - t_i^E$ be the delay between these two events. We place our analysis in a discrete time setting, with units being, *e.g.*, days or weeks and assume that the d_i are independent variables drawn from a distribution with probability mass function f_d that has support $\{0, 1, 2, \ldots, D\}$, where D is the maximal relevant delay. This can for example be the delay after which sufficiently many of the observations have become available or the maximum delay back in time where interventions are still relevant. Reports with a delay larger than D are ignored. For the convenience of the analysis, we shall furthermore assume that the delay distribution is stable over time. This means we ignore increase or decrease of the delays during, *e.g.*, past outbreaks. For an outbreak to be detected, the delay can thus be assumed to be the same as for non-outbreak situations, since no awareness of the outbreak exists yet. Fig. 3.2 contains an illustration of the weekly median as well as the 0.1 and 0.9 quantiles of the smoothed empirical delay distribution for *Salmonella* Newport cases in Germany, obtained from tabulating all cases occurring within a moving window of $t-4, \ldots, t+4$ weeks. The distribution can be assumed stable until March 2013 where an amendment to the Protection against Infection Act made it compulsory to notify cases on a daily basis to the RKI.

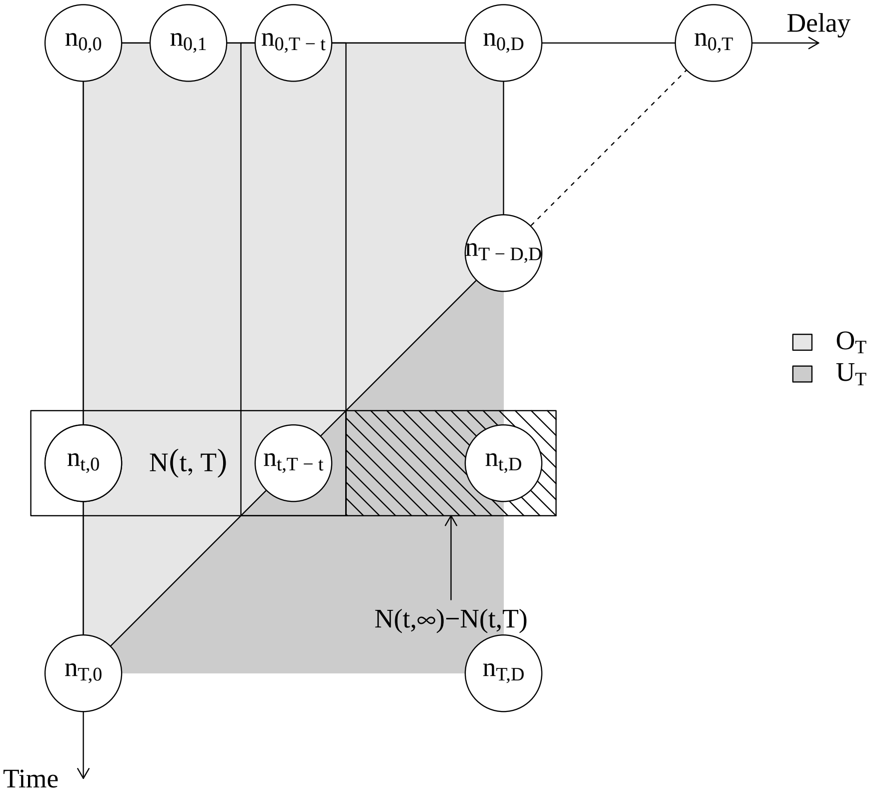


Figure 3.3: Reporting triangle at time T. Available observations are those in the right-angled trapezoid O_T spanned by $n_{0,0}$, $n_{T,0}$, $n_{T-D,D}$ and $n_{0,D}$. Occurred-but-not-yet-reported events are those in the triangle U_T spanned by $n_{T,1}$, $n_{T-D+1,D}$ and $n_{T,D}$.

At fixed observation time T = now which in our context is a specific week, we only observe a case occurred at time t if it was reported with a delay at most equal to $T-t$: the data is thus right-truncated. Following the notation from Lawless (1994), let $n_{t,d}$ be the number of cases of the disease occurred at time t and reported with a delay of d time units. Not all $n_{t,d}$ for $t \in \{0, \dots, T\}$ and $d \in \{0, \dots, D\}$ are available at time T. This is illustrated in Fig. 3.3: each row represents counts of cases occurred at time t, counts for cells located to the right of $\min(T, D-t)$ are not available at time T. For the subsequent analysis we define three index sets: $A_T = \{(t,d) : 0 \leq t \leq T, 0 \leq d \leq D\}$ contains indices for *all* observations $n_{t,d}$ while $O_T = \{(t,d) : 0 \leq t \leq T, 0 \leq d \leq \min(D, T-t)\}$ only contains indices for *observed* $n_{t,d}$ at time T and $U_T = A_T \setminus O_T$ only contains indices for *occurred-but-not-yet-reported* $n_{t,d}$ at time T. Note that O_T and U_T respectively are indicated as the light grey trapezoid and as the darker grey triangle on Fig. 3.3. Counts corresponding to those index sets are denoted by $\boldsymbol{n}_{O_T}$ and $\boldsymbol{n}_{U_T}$, respectively. At time T, we only observe events reported by time T so that $N(t,T) = \sum_{d=0}^{\min(D,T-t)} n_{t,d}$.

For situational awareness at time T, we would like to know the row sum $N_t = \sum_{d=0}^{\infty} n_{t,d}$, *i.e.* the total number of cases occurred at time t, where $0 \leq t \leq T$. One would eventually observe N_t after waiting for an infinite time or for at least a time equal to the maximal possible delay D. Because of right truncation, $N(t,T)$ is possibly still smaller than N_t at time $T < t + D$, as seen in Fig. 3.3.

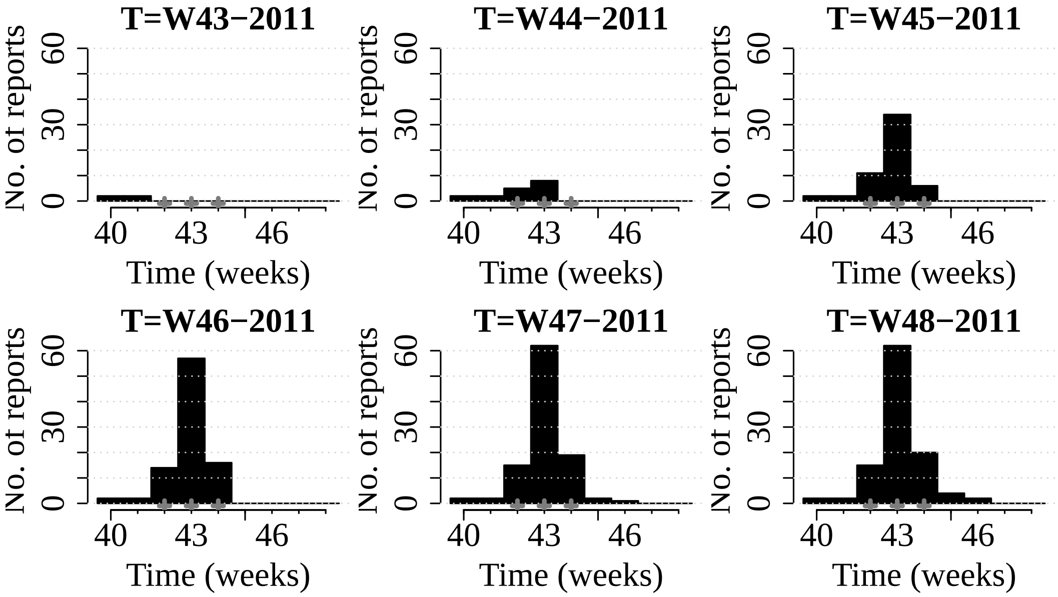


Figure 3.4: Weekly counts $N(t,T)$ of *Salmonella* Newport cases in the RKI database aggregated by date of disease onset at different observation times T during the outbreak. The six plots show t = W40-48 in 2011 as of T = W43-48 in 2011. The grey '+' symbols indicate the three outbreak weeks.

The practical consequences of this right truncation for the S. Newport outbreak associated with sprouts are illustrated on Fig. 3.4: at week 45 of 2011 the RKI could only observe 34 cases for week 43, *i.e.* $N(\text{W43-2011}, \text{W45-2011}) = 34$, but the occurred number of cases that week was 63, *i.e.* $N_{\text{W43-2011}} = 63$. Such incomplete observations during monitoring are an obstacle to situational awareness and outbreak detection.

3.2 Statistical treatment of reporting delays

In the context of aberration detection, we want to compare to a threshold the available number of events $N(s,T)$ occurred at time $s \leq T$ and observed at fixed time $T = \text{now}$. Typically, s is the current week, i.e. $s = T$.

When $D > 0$, the right truncation of the data creates two complications. On the one hand, for each observation time T the counts from the previous time units can have increased due to reporting delays and become aberrant so that one should not only monitor the current timepoint $s = T$ but also timepoints before T. For $s < T - D$, we have $N(s,T) = N_s$ so that these counts do not change anymore and hence do not need to be monitored again. Assume therefore, than in the presence of reporting delays, at each observation timepoint T we monitor the set of timepoints $M_T = \{T - D, \ldots, T\}$. If this is repeated for several subsequent observation timepoints then each timepoint gets monitored $D + 1$ times. This is illustrated in Fig. 3.5.

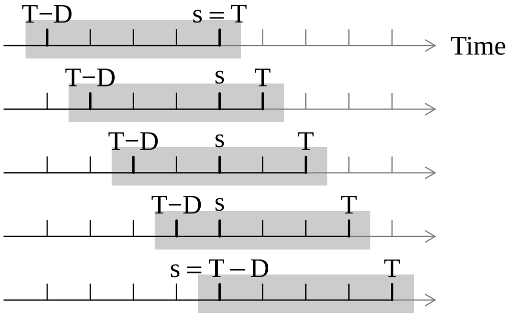


Figure 3.5: Sets of timepoints M_T monitored for five subsequent observation times T, among which we see a fixed timepoint s. In this illustration, $D = 4$. From top to bottom time goes by so that at the end the timepoint s has been monitored $D + 1 = 5$ times.

On the other hand, one needs to adjust the detection algorithm for the incomplete observations because otherwise incomplete counts are compared to complete counts. When $D > 0$ the use of the predictive distribution for N_s *in control* is meaningful only if $s \leq T - D$ so that the observations for s are complete. Otherwise, we make a biased comparison, at the risk of not getting any alarm in the cases where $N(s,T)$ is smaller than the threshold not because N_s itself is smaller than the threshold, but simply because of reporting delays. Current algorithms for aberration detection output the same threshold $Q_{s,T}$ no matter how close s is to T. When correcting for right-truncation in the threshold calculation we thus hope to be able to spot aberrations for time s at an earlier observation timepoint T. We do this as follows: we aim at inferring a predictive distribution for $N(s,T)$, instead of a predictive distribution for N_s. The actually observed $N(s,T)$ is then compared to a quantile $Q_{s,T}$ from this distribution and an alarm is flagged if this threshold is exceeded, i.e. $a_{s,T} = I\left(N(s,T) > Q_{s,T}\right)$.

3.3 Proposal of an algorithm for aberration detection in presence of reporting delays

Our model is a synthesis of the modelling presented by Manitz and Höhle (2013) and of the modelling of Noufaily et al. (2013) now extended to account for reporting delays while keeping the treatment in a Bayesian framework. From the HIV/AIDS literature it is well known that the observations in the reporting triangle can be formulated as an inference problem in an incomplete contingency table (Zeger et al., 1989; Kalbfleisch and Lawless, 1989).

3.3.1 Distributional assumptions

The model we define is inspired by the so-called multinomial model for claim counts (Schmidt and Wünsche, 1998), *i.e.* we assume the following hierarchical model:

$$\begin{aligned} N_t &\sim \mathrm{NB}(\mu_t, \nu), \quad 0 \leq t \leq T, \\ (n_{t,0}, n_{t,1}, \ldots, n_{t,D}) \mid N_t, \boldsymbol{p} &\sim \mathrm{M}(N_t, \boldsymbol{p}), \end{aligned} \tag{3.1}$$

with $E(N_t) = \mu_t$ and $\mathrm{Var}(N_t) = \mu_t(1 + \mu_t/\nu)$. Furthermore, $\mathrm{M}(N_t, \boldsymbol{p})$ is the multinomial distribution with size parameter N_t and $\boldsymbol{p} = (p_0, p_1, \ldots, p_D)$ is the probability mass function (pmf) of the delay distribution. Then, from Schmidt and Wünsche (1998) we know that the marginal distribution of the $n_{t,d}$ is

$$n_{t,d} \sim \mathrm{NB}(\mu_t \cdot p_d, \nu), \quad (t,d) \in O_s. \tag{3.2}$$

The distribution of the $n_{t,d}|N_t, \boldsymbol{p}$ stems from Lemma 1, where the negative binomial distribution is parameterized with size and probability parameters instead of mean and size.

Lemma 1. *If*

$$N_t \sim \mathrm{NB}(\nu, \upsilon_t)$$

with size parameter ν and probability υ_t so that $\mu_t = \nu\upsilon_t/(1-\upsilon_t)$ and $\mathrm{Var}(N_t) = \mu_t(1 + \mu_t/\nu)$*, and*

$$(n_{t,0}, n_{t,1}, \ldots, n_{t,D}) \mid N_t, \boldsymbol{p} \sim \mathrm{M}(N_t, \boldsymbol{p}),$$

then for any tuple of integers (i,j) with $0 \leq i \leq j \leq D$,

$$P\left[\bigcap_{d=i}^{j}\{n_{t,d}=y_{t,d}\}\right]=\frac{\Gamma(\nu+\sum_{d=i}^{j}y_{t,d})}{\Gamma(\nu)\prod_{d=i}^{j}y_{t,d}!}\left(\frac{1-\upsilon_t}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{\nu}\prod_{d=i}^{j}\left(\frac{\upsilon_t p_d}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{y_{t,d}}.$$

The proof for the case where the $n_{t,d}$ are binomial variables in Schmidt (2006), who writes that the proof for the negative binomial distribution is very similar. Here, we make this claim more explicit, because Lemma 1 plays a central role in our algorithm, as we will see in Section 3.3.1 where we discuss its consequences for our regression model. The proof will be in two parts: first we shall prove Lemma 1 for a tuple of integers $(i,j)=(0,D)$ and then prove the recurrence relation: if we assume Lemma 1 to be true for $(i,j)\in\{0,1,\ldots,D\}$ with $i<j$ then it is true for $(i,j-1)$ (the proof of the Lemma then being true for $(i+1,j)$ would be similar). The two parts together will have proven Lemma 1 in the negative binomial case. Afterwards, we will discuss the consequences of Lemma 1 for our algorithm, before giving an urn problem representation of Lemma 1.

Proof. First we prove Lemma 1 for a tuple of integers $(i,j)=(0,D)$, for which we have, as written in Schmidt (2006):

$$\begin{aligned}P\left[\bigcap_{d=0}^{D}\{n_{t,d}=y_{t,d}\}\right] &= P\left[\bigcap_{d=0}^{D}\{n_{t,d}=y_{t,d}\}\cap\{N_t=n_t\}\right]\\ &= P\left[\bigcap_{d=0}^{D}\{n_{t,d}=y_{t,d}\}\,|\,\{N_t=n_t\}\right]\cdot P\left[\{N_t=n_t\}\right]\\ &= \left(\frac{n_t!}{\prod_{d=0}^{D}y_{t,d}!}\prod_{d=0}^{D}p_d^{y_{t,d}}\right)\cdot P\left[\{N_t=n_t\}\right],\end{aligned}$$

where n_t is a realization of N_t. Because of the definition of a negative binomial distribution parameterized with size parameter ν and probability υ_t, then

$$P\left[\{N_t=n_t\}\right]=\frac{\Gamma(\nu+n_t)}{n_t!\Gamma(\nu)}(1-\upsilon_t)^{\nu}\upsilon_t^{n_t}\cdot$$

Thus,

$$\begin{aligned}P\left[\bigcap_{d=0}^{D}\{n_{t,d}=y_{t,d}\}\right] &= \left(\frac{n_t!}{\prod_{d=0}^{D}y_{t,d}!}\prod_{d=0}^{D}p_d^{y_{t,d}}\right)\cdot\frac{\Gamma(\nu+n_t)}{n_t!\Gamma(\nu)}(1-\upsilon_t)^{\nu}\upsilon_t^{n_t}\\ &= \frac{\Gamma(\nu+n_t)}{\Gamma(\nu)\prod_{d=0}^{D}y_{t,d}!}\left(\prod_{d=0}^{D}p_d^{y_{t,d}}\right)\upsilon_t^{n_t}(1-\upsilon_t)^{\nu}\cdot\end{aligned}$$

Note that

$$\left(\prod_{d=0}^{D} p_d^{y_{t,d}}\right) \upsilon_t^{n_t}(1-\upsilon_t)^{\nu} = (1-\upsilon_t)^{\nu} \prod_{d=0}^{D} (p_d \upsilon_t)^{y_{t,d}},$$

since

$$n_t = \sum_{d=0}^{D} y_{t,d}.$$

So we get

$$P\left[\bigcap_{d=0}^{D} \{n_{t,d} = y_{t,d}\}\right] = \frac{\Gamma(\nu + n_t)}{\Gamma(\nu)\prod_{d=0}^{D} y_{t,d}!}(1-\upsilon_t)^{\nu} \prod_{d=0}^{D} (p_d \upsilon_t)^{y_{t,d}},$$

which proves Lemma 1 for $(i, j) = (0, D)$, since

$$\sum_{d=0}^{D} \upsilon_t p_d = \upsilon_t.$$

The proof is now based on a recurrence relation. Let us assume that Lemma 1 is true for $(i, j) \in \{0, 1, \ldots, D\}$ with $i < j$. Now we are going to prove it for $(i, j-1)$. We can write that

$$\begin{aligned} P&\left[\bigcap_{d=i}^{j-1} \{n_{t,d} = y_{t,d}\}\right] \\ &= \sum_{y_{t,j}=0}^{\infty} P\left[\bigcap_{d=i}^{j} \{n_{t,d} = y_{t,d}\}\right] \\ &= \sum_{y_{t,j}=0}^{\infty} \frac{\Gamma(\nu + \sum_{d=i}^{j} y_{t,d})}{\Gamma(\nu)\prod_{d=i}^{j} y_{t,d}!} \left(\frac{1-\upsilon_t}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{\nu} \prod_{d=i}^{j} \left(\frac{\upsilon_t p_d}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{y_{t,d}} \end{aligned} \quad (3.3)$$

since we assume Lemma 1 to be true for (i, j).

In order to prove Lemma 1 for $(i, j-1)$ we shall re-write Equation 3.3. This will show that if Lemma 1 is true for (i, j) then it is true for $(i, j-1)$. We shall decompose the formula in several parts and re-write them separately, before putting the parts together again.

So first let us write that

$$\frac{\Gamma(\nu + \sum_{d=i}^{j} y_{t,d})}{\prod_{d=i}^{j} y_{t,d}!} = \frac{\Gamma(\nu + \sum_{d=i}^{j} y_{t,d})}{\prod_{d=i}^{j} y_{t,d}!} \cdot \underbrace{\frac{\prod_{d=i}^{j-1} y_{t,d}!}{\Gamma(\nu + \sum_{d=i}^{j-1} y_{t,d})} \cdot \frac{\Gamma(\nu + \sum_{d=i}^{j-1} y_{t,d})}{\prod_{d=i}^{j-1} y_{t,d}!}}_{=1},$$

where we see that

$$\frac{\Gamma(\nu+\sum_{d=i}^{j} y_{t,d})}{\prod_{d=i}^{j} y_{t,d}!} \cdot \frac{\prod_{d=i}^{j-1} y_{t,d}!}{\Gamma(\nu+\sum_{d=i}^{j-1} y_{t,d})} = \frac{\Gamma(\nu+\sum_{d=i}^{j} y_{t,d})}{y_{t,j}!\Gamma(\nu+\sum_{d=i}^{j-1} y_{t,d})},$$

so that

$$\frac{\Gamma(\nu+\sum_{d=i}^{j} y_{t,d})}{\prod_{d=i}^{j} y_{t,d}!} = \frac{\Gamma(\nu+\sum_{d=i}^{j} y_{t,d})}{y_{t,j}!\Gamma(\nu+\sum_{d=i}^{j-1} y_{t,d})} \frac{\Gamma(\nu+\sum_{d=i}^{j-1} y_{t,d})}{\prod_{d=i}^{j-1} y_{t,d}!}.$$

Furthermore, in Equation 3.3, we also have

$$\begin{aligned} \left(\frac{1-\upsilon_t}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{\nu} &= \left(\frac{1-\upsilon_t}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{\nu} \cdot \underbrace{\left(\frac{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}\right)^{\nu}}_{=1} \\ &= \left(\frac{1-\upsilon_t}{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}\right)^{\nu} \cdot \left(\frac{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{\nu}. \end{aligned}$$

Moreover, in Equation 3.3, we have

$$\begin{aligned} &\prod_{d=i}^{j}\left(\frac{\upsilon_t p_d}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{y_{t,d}} \\ &= \left(\prod_{d=i}^{j-1}\left(\frac{\upsilon_t p_d}{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}\right)^{y_{t,d}}\right) \cdot \left(\frac{\upsilon_t p_j}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{y_{t,j}} \cdot \left(\frac{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{\sum_{d=i}^{j-1} y_{t,j}}. \end{aligned}$$

All in all, for Equation 3.3 we now have

$$\begin{aligned} &P\left[\bigcap_{d=i}^{j-1}\{n_{t,d}=y_{t,d}\}\right] \\ &= \sum_{y_{t,j}=0}^{\infty} \frac{\Gamma(\nu+\sum_{d=i}^{j} y_{t,d})}{y_{t,j}!\Gamma(\nu+\sum_{d=i}^{j-1} y_{t,d})} \left(\frac{\upsilon_t p_j}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{y_{t,j}} \cdot \left(\frac{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{\nu+\sum_{d=i}^{j-1} y_{t,j}} \cdot \\ &\quad \frac{\Gamma(\nu+\sum_{d=i}^{j-1} y_{t,d})}{\Gamma(\nu)\prod_{d=i}^{j-1} y_{t,d}!} \left(\frac{1-\upsilon_t}{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}\right)^{\nu} \prod_{d=i}^{j-1}\left(\frac{\upsilon_t p_d}{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}\right)^{y_{t,d}}, \end{aligned}$$

where

$$\sum_{y_{t,j}=0}^{\infty} \frac{\Gamma(\nu+\sum_{d=i}^{j} y_{t,d})}{y_{t,j}!\Gamma(\nu+\sum_{d=i}^{j-1} y_{t,d})} \left(\frac{\upsilon_t p_j}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{y_{t,j}} \cdot \left(\frac{1-\upsilon_t+\sum_{l=i}^{j-1}\upsilon_t p_l}{1-\upsilon_t+\sum_{l=i}^{j}\upsilon_t p_l}\right)^{\sum_{d=i}^{j-1} y_{t,j}}$$

is the sum of the pmf for all possible values of a negative binomial variable with size ν and probability parameter

$$\frac{\upsilon_t p_j}{1 - \upsilon_t + \sum_{l=i}^{j} \upsilon_t p_l}$$

and is thus equal to 1. Therefore,

$$P\left[\bigcap_{d=i}^{j-1} \{n_{t,d} = y_{t,d}\}\right] = \frac{\Gamma(\nu + \sum_{d=i}^{j-1} y_{t,d})}{\Gamma(\nu)\prod_{d=i}^{j-1} y_{t,d}!}\left(\frac{1 - \upsilon_t}{1 - \upsilon_t + \sum_{l=i}^{j-1} \upsilon_t p_l}\right)^{\nu} \prod_{d=i}^{j-1}\left(\frac{\upsilon_t p_d}{1 - \upsilon_t + \sum_{l=i}^{j-1} \upsilon_t p_l}\right)^{y_{t,d}},$$

which proves Lemma 1 for $(i, j-1)$. We have thus proven Lemma 1 in the negative binomial case. □

Consequence of Lemma 1

Let us assume that $i = j = d$. Then we have

$$P\left[n_{t,d} = y_{t,d}\right] = \frac{\Gamma(\nu + y_{t,d})}{\Gamma(\nu) y_{t,d}!}\left(\frac{1 - \upsilon_t}{1 - \upsilon_t + \upsilon_t y_{t,d}}\right)^{\nu}\left(\frac{\upsilon_t p_d}{1 - \upsilon_t + \upsilon_t p_d}\right)^{y_{t,d}}$$

which means that $n_{t,d}$ has a negative binomial distribution with parameters size ν and probability

$$\frac{\upsilon_t p_d}{1 - \upsilon_t + \upsilon_t p_d}.$$

Moreover, because of the grouping property of the multinomial distribution, we know that $(N(t,T), N_t - N(t,T))$ is a multinomial (binomial) variable. It is therefore straightforward that $N(t,T)$ has a negative binomial distribution with parameters size ν and probability

$$\frac{\upsilon_t \sum_0^{\min(T-t,D)} p_d}{1 - \upsilon_t + \upsilon_t \sum_0^{\min(T-t,D)} p_d}.$$

We write this with the mean-size parameterization of the negative-binomial distribution:

$$N(t,T) \sim \text{NB}\left(\sum_{d=0}^{\min(T-t,D)} p_d \mu_t, \nu\right).$$

Interpretation of Lemma 1 On top of the previous mathematical proof that the $n_{t,d}$ marginally have a negative binomial distribution with the same size parameter, we come up with an urn problem representation of the corresponding binomial experiment, when ν, $\nu\upsilon_t/(1-\upsilon_t)$ and $p_d\nu\upsilon_t/(1-\upsilon_t)$ are integers for all $d, 0 \leq d \leq D$.

Let us say we have an urn with ν white balls and $\nu\upsilon_t/(1-\upsilon_t)$ balls of D different colours. The number of balls of each colour d is $p_d\nu\upsilon_t/(1-\upsilon_t)$. All balls have the same probability of being drawn.

The experiment that represents the arrival and number of cases corresponds to drawing a ball from the urn, writing its colour, putting it back and drawing again until one has gotten ν white balls. Coloured balls are cases, each colour corresponding to a specific delay.

The number of colour balls one has gotten in the meanwhile is a negative binomial variable of size ν and of probability of failure (getting a colour ball) υ_t.

What about the distribution of the number of, say, d colour, *e.g.* green balls? If we are only interested in the number of green balls, we could say that we do not write down the colour of the ball if it is not either white or green. Thus, balls of other colours are non-existent for us. Therefore, in practice we only have $\nu(1+p_d\upsilon_t/1-\upsilon_t)$ balls, which gives us a Bernoulli experiment where the events (getting a white ball, getting a green ball) have respective probabilities

$$\frac{1-\upsilon_t}{1-\upsilon_t+\upsilon_t p_d}$$

and

$$\frac{\upsilon_t p_d}{1-\upsilon_t+\upsilon_t p_d},$$

adding up to 1. Hence the number of green balls is a negative binomial variable of size ν and of probability of failure

$$\frac{\upsilon_t p_d}{1-\upsilon_t+\upsilon_t p_d}.$$

For illustrating this idealized mental exercise with numbers, let us assume that:

- We have $\nu = 2$ white balls,
- $D = 1$, $\upsilon_t = 0.75$, $p_0 = 1/3$ and $p_1 = 2/3$. Therefore we have $p_0\nu\upsilon_t/(1-\upsilon_t) = 2$ green balls and $p_1\nu\upsilon_t/(1-\upsilon_t) = 4$ purple balls.

We represent the urn problem in Figure 3.6, for the case when one is interested in the marginal distribution of $n_{t,0}$. One draws a ball out of the urn. If it is purple, one simply puts it back into the urn. If it is white or green, one counts the draw before putting it back into the urn. The experiment stops when one has drawn $\nu = 2$ white balls, $n_{t,0}$ is the number of green balls one has gotten before the end of the experiment. It is a negative binomial variable with size $\nu = 2$ and probability parameter

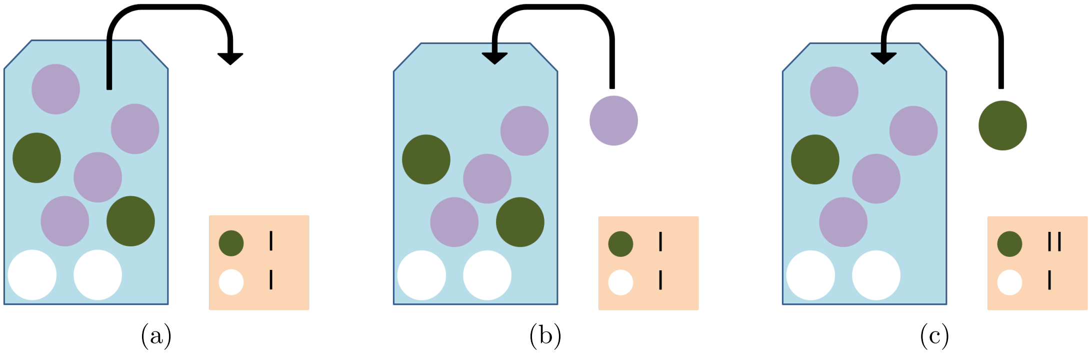
$$\frac{\upsilon_t p_d}{1-\upsilon_t+\upsilon_t p_d} = 0.5.$$

Now that we have presented the main distributional assumptions of the hierachical model we use, we shall present its structure and inference.

3.3.2 Model structure

Synthesizing the elements of Noufaily et al. (2013) for the modelling of $\log(\mu_t)$ in the above we obtain

$$\log(\mu_t) = \beta_0 + \beta_1 \cdot t + \gamma_{c(t)}.$$



(a) (b) (c)

Figure 3.6: Urn problem representation of the marginal distribution of the $n_{t,d}$. Cases are coloured balls, reported with no delay (green – dark – balls) or the maximal delay of one week (purple – light – balls).

We parametrize the delay on $0, \ldots, D$ as $\log(p_d) = \alpha_d$. Altogether, the parameter vector for the linear predictor is $\boldsymbol{\theta} = (\beta_0, \beta_1, \gamma_0, \ldots, \gamma_{S-1}, \alpha_0, \ldots, \alpha_D)'$ and the whole parameter vector is $\boldsymbol{\psi} = (\boldsymbol{\theta}, \nu)'$. Hence, we can write that

$$\log(\mu_t \cdot p_d) = \eta_{t,d} = \boldsymbol{z}'_{t,d}\boldsymbol{\theta}$$

with $\boldsymbol{z}_{t,d}$ the vector of covariates corresponding to the linear predictor $\eta_{t,d}$. As in Manitz and Höhle (2013) we assume the observations $n_{t,d}$ given the model structure (3.2) are independent, so that the likelihood becomes

$$f(\boldsymbol{n}_{O_s} \mid \boldsymbol{\psi}) = \prod_{(t,d) \in O_s} f(n_{t,d} \mid \boldsymbol{\psi}),$$

where $f(n_{t,d} \mid \boldsymbol{\psi})$ is the probability mass function of the negative binomial distribution presented in equation 3.2. From this, using Bayes' theorem, we get

$$f(\boldsymbol{\psi} \mid \boldsymbol{n}_{O_s}) \propto f(\boldsymbol{n}_{O_s} \mid \boldsymbol{\psi}) \cdot \pi(\boldsymbol{\psi}).$$

For infering the predictive posterior distribution we now propose two methods: a fully Bayesian method using integrated nested Laplace approximations (INLA) (Rue et al., 2009), and a method using an asymptotic normal approximation for the posterior that is much faster and thus better adapted to future routine applications of the algorithm.

3.3.3 Inference of the predictive posterior distribution with INLA

Priors are chosen as in Manitz and Höhle (2013) where possible. This means, e.g., $\beta_i \sim \mathrm{N}(0, \lambda_{\beta_i}^{-1})$, $i = 1, 2$, where the λ_{β_i}'s indicate precision parameters. We choose $\nu \sim \log\mathrm{N}(0, 100)$ for not imposing a too high constraint on overdispersion. For the period effects we use independent priors, i.e. $\gamma_i \sim \mathrm{N}(0, \lambda_{\gamma_i}^{-1})$. Furthermore, for each level α_d of the factor variable accounting for delay we use independent priors $\alpha_d \sim \mathrm{N}(0, \lambda_{\alpha_d}^{-1})$. For ensuring identifiability we add two corner constraints for the blocks of parameters for season and delay: $\alpha_0 = 0$ and $\gamma_0 = 0$. For all fixed effects, we used the default value for the precision, 0.001. Finally, we assume independence of the priors so that the prior $\pi(\boldsymbol{\psi})$ is the product of the parameters' priors. Inference for the posterior is then performed with INLA as described in Rue et al. (2009) and Rue et al. (2015).

3.3.4 Asymptotic inference of the predictive posterior distribution

In this method, we first fit a negative binomial GLM type regression model with log-link to the data in O_s according to equation 3.2, providing a frequentist estimation of the parameters. This results in the estimators $(\hat{\eta}_{t,d}, \hat{\nu})'$. We then use the asymptotic normal distribution of the parameters for approximating the posterior $f(\boldsymbol{\psi} \,|\, \boldsymbol{n}_{O_s})$. As explained in, *e.g.*, Held and Sabanés Bové (2014) the posterior distribution is under suitable regularity conditions asymptotically Gaussian with mean equal to the maximum-likelihood estimator and covariance equal to the inverse observed Fisher information matrix, no matter the prior choice:

$$\boldsymbol{\psi} \mid \boldsymbol{n}_{O_s} \overset{a}{\sim} \mathrm{N}(\hat{\boldsymbol{\psi}}, \boldsymbol{I}(\hat{\boldsymbol{\psi}})^{-1}).$$

Moreover, due to the estimation method of the negative binomial model, we assume that $\eta_{t,d}$ and ν are information-orthogonal. With these assumptions, $\eta_{t,d}$ is the linear combination of Gaussian variables so that we can write that

$$\eta_{t,d} \overset{a}{\sim} \mathrm{N}\left(\boldsymbol{z}_{t,d}'\hat{\boldsymbol{\theta}} \, , \, \boldsymbol{z}_{t,d}'\boldsymbol{I}(\hat{\boldsymbol{\theta}})^{-1}\boldsymbol{z}_{t,d}\right),$$

where $\boldsymbol{I}(\hat{\boldsymbol{\theta}})$ is the block related to $\hat{\boldsymbol{\theta}}$ in $\boldsymbol{I}(\hat{\boldsymbol{\psi}})$. Note that with $b = 4$ years data, $S = 10$ and $D = 10$ we have over $150 \cdot (D + 1)$ observations for inference, that is to say there are over 1600 elements in O_s for the 24 parameters to be estimated. We hence expect the difference between the asymptotic method and the INLA method to be small.

3.3.5 Predictive posterior distribution integration and threshold calculation

One can directly compute the posterior marginal distribution of $N(t, T)$ from the inferred distribution of the parameters, because marginally

$$N(t, T) \mid N_t, \boldsymbol{p} \sim \mathrm{Bin}\left(N_t, \sum_{d=0}^{\min(T-t,D)} p_d\right).$$

Therefore, as N_t is a negative binomial variable, we obtain (Schmidt and Wünsche, 1998),

$$N(t,T) \sim \mathrm{NB}\left(\sum_{d=0}^{\min(T-t,D)} p_d \mu_t, \nu \right).$$

The joint posterior distribution of the current observations and of the parameter $\boldsymbol{\theta}_T$ can then be computed. For getting the predictive posterior distribution for the $n_{t,d}$ we have to integrate with respect to $\boldsymbol{\psi}$,

$$\begin{aligned} f(N(s,T) \mid \boldsymbol{n}_{O_s}) &= \int f(N(s,T), \boldsymbol{\psi} \mid \boldsymbol{n}_{O_s}) d\boldsymbol{\psi} \\ &= \int f(N(s,T) \mid \boldsymbol{\psi}) f(\boldsymbol{\psi} \mid \boldsymbol{n}_{O_s}) d\boldsymbol{\psi}. \end{aligned}$$

The threshold $Q_{s,T}$ is defined as the $(1-\alpha)$-quantile from this distribution. Then the alarm indicator is $a_{s,T} = I\left(N(s,T) > Q_{s,T}\right)$. This comparison encompasses both estimation and observation uncertainties. Independent of whether the INLA or asymptotic method was used we shall proceed as in Manitz and Höhle (2013) and use Monte Carlo simulation to obtain the desired quantile of the posterior predictive, as explained in Algorithm 1.

```
Data: $\boldsymbol{n}_{O_s}$
Result: $\hat{Q}_{s,T}$
Determine $f(\boldsymbol{\psi} \mid \boldsymbol{n}_{O_s})$ using either the INLA method or the asymptotic method.
for $1 \leq r \leq R$ do
    Sample $\boldsymbol{\psi}^{(r)} \sim f(\boldsymbol{\psi} \mid \boldsymbol{n}_{O_s})$.
    Sample the $N(s,T)^{(r)} \sim f(N(s,T) \mid \boldsymbol{\psi}^{(r)})$.
end
Determine $\hat{Q}_{s,T}$ as the empirical $(1-\alpha)$-quantile of the responses
  $N(s,T)^{(r)}, r \in \{1,\ldots,R\}$.
```

Algorithm 1: Algorithm for computing the threshold in the Bayesian setting.

3.4 Application of the proposed algorithm

3.4.1 Simulation study

This section explores the performance of the proposed delay-adjusting algorithm by first using simulated data. Section 3.4.2 contains an analysis on the S. Newport surveillance time series.

Simulated data

We used simulated time series built as follows to evaluate the algorithm: Baseline counts B_t are drawn from a negative binomial distribution with size parameter ν and mean μ_t for each timepoint. The structure for the mean is the same as in Noufaily et al. (2013), *i.e.*

$$\log(\mu_t) = \beta_0 + \beta_1 t + \sum_{j=1}^{m} \left\{ \gamma_{2j-1} \cos\left(\frac{2\pi j t}{52}\right) + \gamma_{2j} \sin\left(\frac{2\pi j t}{52}\right) \right\}.$$

In this expression β_0 is the baseline frequency of reports, β_1 is the time trend, and if $m > 0$ then the γ's specify the annual ($m = 1$) and biannual ($m = 2$) seasonality with $\gamma_1 = \gamma_3$ and $\gamma_2 = \gamma_4$. We used the 42 scenarios of Noufaily et al. (2013) but had to change their parameterisation of the variance which was $\text{Var}(B_t) = \phi\mu_t$, *i.e.* the variance structure adapted to a quasi-Poisson regression model. In order to keep the variety of the scenarios and to have comparable scenarios, we chose the values for ν so that the variance obtained for $\mu^* = \exp(\beta_0)$ would be the same in both cases. This gives, for each scenario, $\nu = \mu^*/(\phi - 1)$.

Outbreaks were added as in Noufaily et al. (2013). Their size is fixed by the parameter σ, the outbreak *size coefficient*, so that the total number of cases of the outbreak starting at t is $\text{Po}(\sigma \cdot \text{Var}(B_t))$. The outbreak cases are distributed in time according to a discretized lognormal distribution with mean 0 and standard deviation 0.5. At each timepoint, we have $N_t = B_t + K_t$ with K_t the outbreaK count which is possibly zero.

We also simulated the reporting of these cases with the delay distributed according to the empirical distribution of reporting delays of Salmonella cases to the RKI in 2011 and with maximal delay set to $D = 10$ weeks. Cases with delays longer than 10 are assigned the maximal delay. The probability associated with each possible value of the delay from 0 to 10 is given by $\boldsymbol{p} = (0.035, 0.369, 0.357, 0.139, 0.049, 0.020, 0.01, 0.005, 0.004, 0.002, 0.009)'$. Using this delay distribution we are able to generate the reporting triangles and hence the partial time series at a given timepoint T, *i.e.* $N(t, T)$ instead of N_t.

The functions we have written for simulating the time series in `R` are presented in Section 3.5.3.

Specification of the algorithms

We compare the algorithm of Noufaily et al. (2013) and the algorithm proposed in Sect. 3.3 with and without delay correction, i.e. using $D = 10$ and $D = 0$. In all cases we use $S = 10$ periods for the seasonal effect, $b = 4$ and $w = 3$. We generate time series of 350 weeks. Since we wish to study an algorithm correcting for reporting delays we need to monitor each timepoint at different dates as shown in Fig. 3.5: for each time series monitored, each timepoint is monitored $D + 1 = 11$ times. This means that our simulation study involves many more computations than studies ignoring reporting delays and only monitoring complete counts. For methods that do not correct for reporting delays, all upperbounds for a fixed s are equal – although the proposed method with $D = 0$ can produce slightly different thresholds because of sampling uncertainty. Thus, when using a non-reporting delay adjusting method for monitoring, if $a_{s,T} = 1$ then $a_{s,T+1} = 1$, *i.e.* alarms do not disappear.

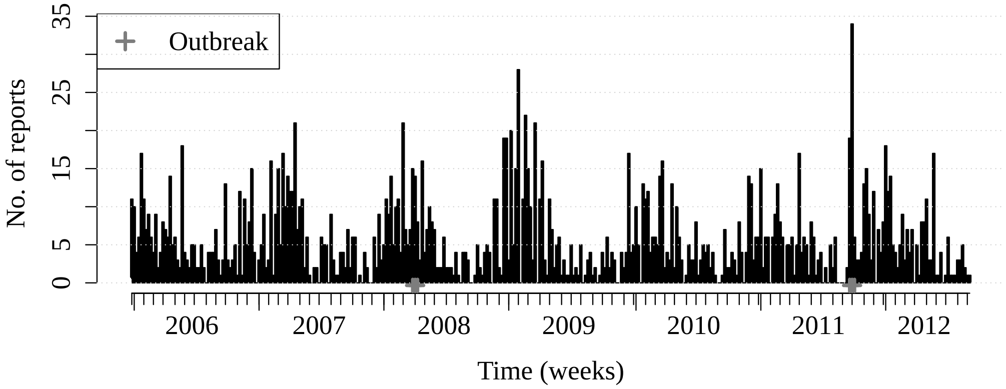


Figure 3.7: Example of a simulated time series with outbreaks of size coefficient $\sigma = 2$ and $\sigma = 5$ starting at week 13 of 2008 and at week 38 of 2011, respectively. The code used for simulating such time series is presented in Section 3.5.3.

Evaluation measurements

We used the same evaluation measurements as in Noufaily et al. (2013): the probability of detection (POD) and the false positive rate (FPR). The FPR is the number of alarms in weeks without outbreaks divided by the number of weeks in the monitored period. The POD is estimated as the number of detected outbreaks, that is to say when a alarm corresponds to one of the outbreak weeks, divided by the number of outbreaks. Since we only include one outbreak in the monitored period of time, each simulation gives a value of either 1 or 0 for a given algorithm, whose mean among several simulations gives the POD. Note that the POD and the FPR do not measure timeliness. Therefore, for each tested algorithm, we calculate the POD and the FPR for each possible value of $T - s \in \{0, \dots, D\}$. In methods that do not correct for reporting delays, the threshold is only valid for complete observations so that we expect the FPR to be controlled only for the $N(s, T)$ with $s = T - D$. Moreover, as the methods that do not correct for reporting delays make the same guess for every $N(s, T)$ no matter how close s is to T, they are also less likely to detect an outbreak for small $T - s$. Thus we expect our algorithm to have a fairly constant FPR over different values of $T - s$ and to detect outbreaks for smaller $T - s$ in comparison with methods that do not correct for reporting delays.

We also introduce a measure of timeliness: the reaction time (RT) needed for the algorithm to produce an alarm for an outbreak after its beginning if the outbreak was detected. In other words, if there is an alarm for the outbreak starting at t_{out} and spanning over l_{out} time units, $\text{RT}(t_{\text{out}}, l_{\text{out}})$ is the difference between t_{out} and the smallest observation time T at which we get an alarm for any $s \in \{t_{\text{out}}, \dots, t_{\text{out}} + l_{\text{out}}\}$. If the outbreak is not detected at all we set $\text{RT}(t_{\text{out}}, l_{\text{out}})$ to NA.

Results

We start by comparing the results of the two methods of inference as regards estimation of the posterior $\boldsymbol{\psi}$. In Fig. 3.8 we show two contourplot examples for two randomly selected simulation scenarios. Ideally one should use the INLA method for performing full Bayesian inference, but as our goal was the creation of an algorithm also suitable for routine use, we chose to evaluate the asymptotic method which is 15 times faster than the INLA approach. In Section 3.4.2 we also compare the alarm threshold of the two methods for the S. Newport data.

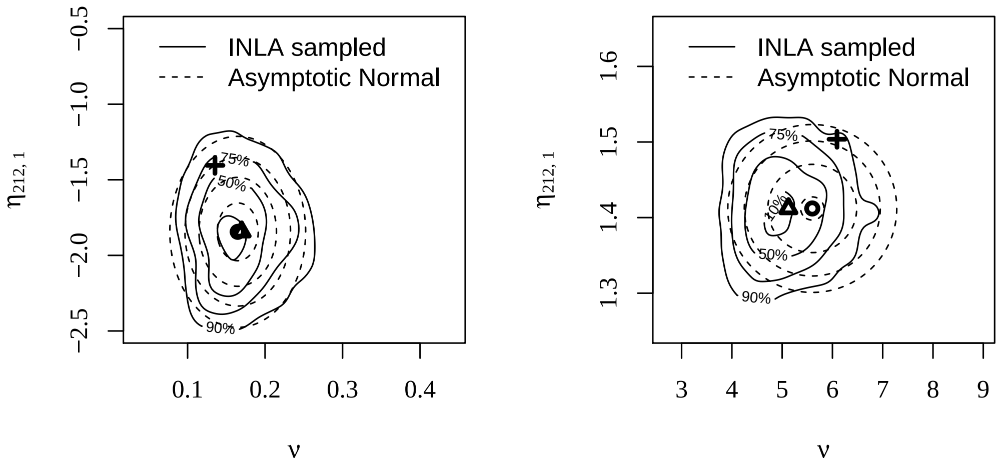


Figure 3.8: Contourplots of the sampled posterior for $(\eta_{t,d}, \nu)'$ obtained from the negative binomial model for the 212'th timepoint of a time series simulated with the scenarios (a) 12 and (b) 25 of the simulation study and $d = 1$, with data available as of the 350'th timepoint of the time series. True values of the parameters are represented with a cross, while the mean values of the posterior distributions of the parameters are represented with a triangle and a circle respectively for the INLA method and the asymptotic method.

In a first study of the efficiency of the algorithm, we explore the specificity of the new algorithm and thus only simulate time series with no outbreak. We generate 10 time series for each of the 42 scenarios and monitor each of them over one year, *i.e.* 52 weeks. Our goal with this study is to explore the FPR of our method compared to the same algorithm without reporting delays correction and compared to the established Noufaily algorithm. We performed the exploration of the FPR of our algorithm with $\alpha = 0.05$. As the algorithm proposed by Noufaily et al. (2013) uses a $(1-\alpha/2)$-quantile, we used $\alpha = 0.1$ for this algorithm.

The results are shown in Fig. 3.9 with the mean of the FPR obtained over all 420 time series, and the standard deviation of the mean of FPR over the 42 scenarios, for each possible value of $T-s$. One cannot deduce from Fig. 3.9 that the FPR does vary with the scenario since we only had 10 time series for each scenario. One would need more repetitions by scenario to be able to see whether different scenarios lead to different performance of the algorithms. This may be due to the fact that how well the parameters are estimated depends on the parameters values (Lloyd-Smith, 2007). As expected the new method adjusting for delay ($D=10$) has a higher FPR for small values of $T-s$. The FPR of the two other methods is smaller for these small values of $T-s$ because the threshold of the two other methods, defined for complete counts, is nearly always too high for having false alarms on very incomplete counts. For all three methods, the average FPR is never higher than the nominal level.

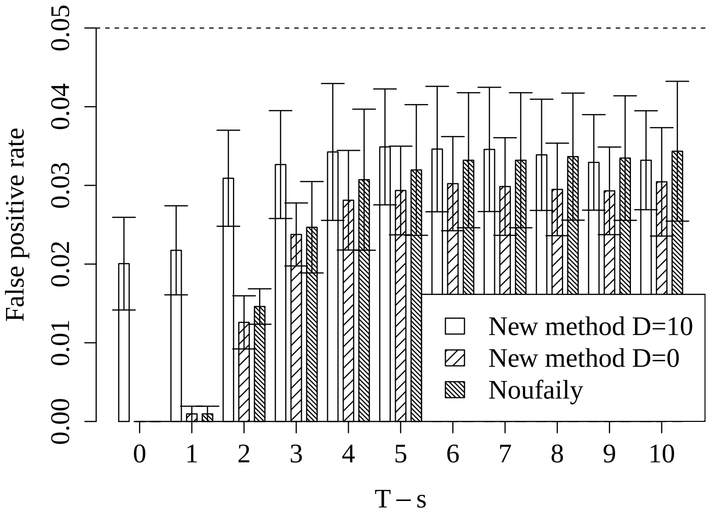


Figure 3.9: FPR for the 11 possible values of $T-s \in \{0,\ldots,D\}$ for the three algorithms tested and using $\alpha = 0.05$. The horizontal bars represent the mean of the FPR ± its standard deviation over the 42 scenarios.

In a second series of simulations, we investigated how the three methods compare at detecting outbreaks. For this, we again use $\alpha = 0.05$. We generated 420 time series, 10 per scenario, and added outbreaks as in Noufaily et al. (2013) but with only 3 outbreaks in the baseline of size coefficient $\sigma \in \{2,3,5,10\}$, and an outbreak in the monitored weeks with size coefficient $\sigma \in \{1,\ldots,10\}$. The starting dates of the outbreaks were randomly drawn. We only monitored the weeks where the outbreak was and record whether an alarm was produced by the algorithm and if so for which s and which T.

Fig. 3.11 shows the reaction time RT for the three methods. One sees that outbreaks are detected earlier with the proposed algorithm with $D=10$ than with the other methods. The associated POD are represented on Fig. 3.10. With the delay distribution we use, after 3 weeks the new method with $D=10$ offers no advantage to the algorithm developped by Noufaily et al. (2013). Note that with the chosen delay distribution, after 3 weeks about 76% of the cases have already been reported.

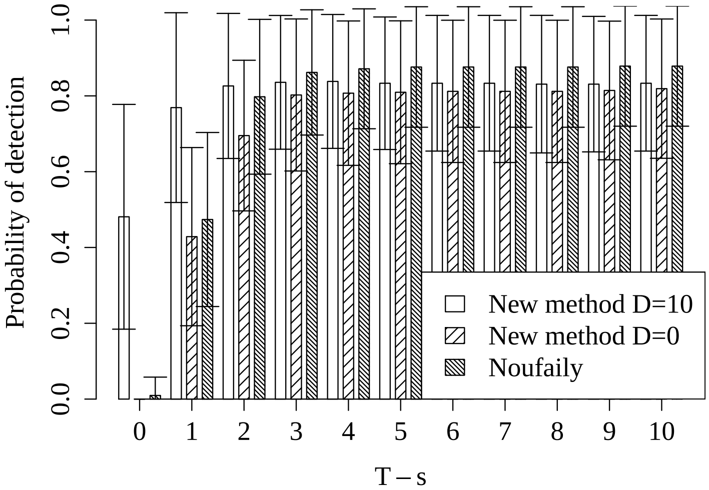


Figure 3.10: POD for the 11 possible values of $T - s \in \{0, \ldots, D\}$ for the three algorithms tested and using $\alpha = 0.05$. The horizontal bars represent the mean of the POD ± its standard deviation over the 42 scenarios.

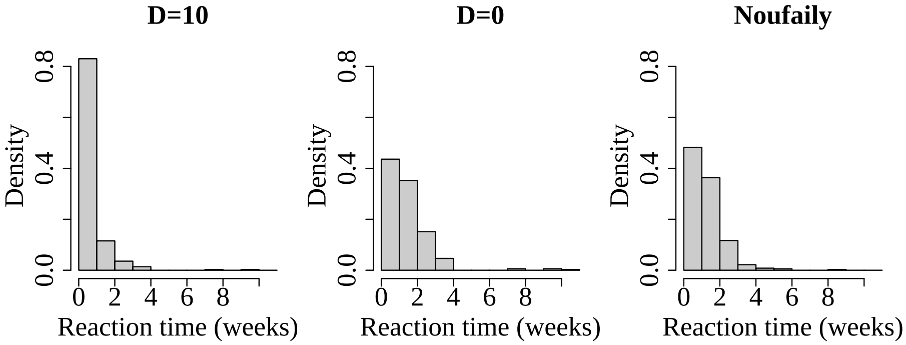


Figure 3.11: Histogram of the reaction time RT for the three compared methods measured on 420 time series with simulated outbreaks. The RT can only be calculated if the outbreak was indeed detected. The medians are respectively 1, 2 and 2 days.

3.4.2 Application to *S.* Newport

As an illustration of how the proposed algorithms operate we applied them to the time series of *Salmonella* Newport cases shown in Sec. 3.2. We used data available as of $T = \text{W43-2011}, \text{W44-2011}, \text{W45-2011}$ (shown in Fig. 3.12), with cases aggregated by date of disease onset as in Fig. 3.4. We present the results for the method of Noufaily et al. (2013) and for the proposed algorithm with $D = 10$ for the two inference methods. When using the method of Noufaily et al. (2013) the threshold is not adjusted for delay and does not change depending on the observation time: for each observation time T we get a new threshold value for $s = T$ while the threshold values for monitored timepoints $s < T$ do not change. In contrast, the threshold computed by delay-adjusting algorithms (D=10) can change for all timepoints $s \leq T$: in Fig. 3.12 one sees for instance that when T = W43-2011 the threshold for week s = W42-2011 is 1 with both inference methods, and when T = W44-2011 it is 2 with the asymptotic approximation and 3 with the INLA inference method. In other words, the threshold was adapted to the number of cases we could have expected to see for this week (s = W42-2011) *by then* (T = W43-2011, T = W44-2011). Adjusting the threshold for right-truncation could make the alarm being sounded sooner: a one week earlier detection could already make a difference for an outbreak.

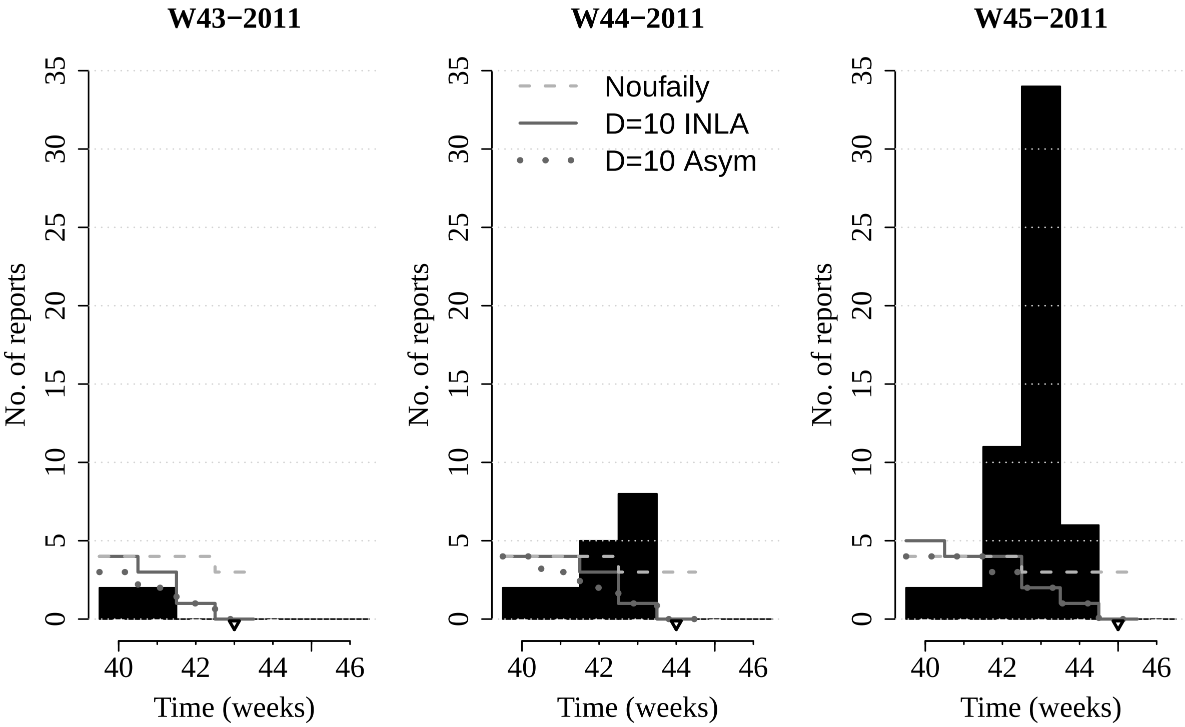


Figure 3.12: Weekly counts $N(s,T)$ of *Salmonella* Newport cases in the RKI database aggregated by date of disease onset at different observation times T = W43-W45 2011 during the 2011 outbreak and for s = W40-46 2011. The three plots correspond to different values of T whose position is indicated on the x-axis by a black triangle. We present the results for the Noufaily method and for the proposed method with D = 10 using both inference methods.

A slightly disappointing result is that for this specific outbreak, the new methods adjusting for delay do not provide an earlier alarm: for all three methods considered, the first time that the number of observed cases is higher than the threshold is week 44. The number of cases reported during week 44 was so high that N(W43-2011, W44-2011) was above the threshold computed by all investigated methods, whereas N(W42-2011, W43-2011) and N(W43-2011, W43-2011) were not above the threshold computed by any of the investigated methods – even the ones adjusting for delay. The only difference between the three alarms for $N(\text{W43-2011}, \text{W44-2011})$ is that the threshold of the methods correcting for delay is smaller, so that if one were to compute a probability for the current count, it would be smaller, so that the alarm would be bigger. However, at present alarms are often only binary indicators. Back in 2011, the outbreak was manually detected by the National Reference Centre for Salmonella because of a cluster of S. Newport isolates from a clinic in Northern Germany, not by automatic outbreak detection. These algorithms were not as thoroughly implemented then as compared to now. Hence, the example shows more the virtue of automatic detection and not so much the added value of adjustment for reporting delays. Furthermore, the peak was huge so that despite reporting delays the number of cases was quickly high enough to be detected by any algorithm. However, with another pattern of cases diagnosis and reporting for an outbreak, delay adjustment could have made the difference for getting an earlier alarm. The present example thus shows that the usefulness of the method depends on the pattern of cases reporting, on the form of the outbreak but also on the baseline counts.

As regards the comparison of the two inference methods for $D = 10$, the two thresholds computed are very similar. On our computer the inference using INLA needed 170 seconds for providing the results for $T = \text{W45-2011}$, whereas the other method only needed 11 seconds, which is about 15 times faster. This makes the implementation of the algorithm based on the asymptotic approximation more realistic for practical routine use.

3.5 Presentation of the `R` implementation of the algorithm

As part of this thesis work we have implemented the algorithm presented in Section 3.3 as a function, `bodaDelay`, which has become part of the `surveillance` package. It performs Bayesian inference either by assuming Gaussian posteriors, or, like the `boda` function that corresponds to the work of Manitz and Höhle (2013), by using integrated nested Laplace approximations, which are implemented in the R INLA package (Rue et al., 2015). In this Section we will first explain the use of the `bodaDelay` function, then compare it to the `boda` function, and finally we will present other smaller functions implemented in the `surveillance` package for doing the work presented in Section 3.4.1.

3.5.1 How to use `bodaDelay`

The function `bodaDelay` works like the other `surveillance` functions presented in Chapter 2: it takes a `sts`-object as an input, and outputs another one with threshold and alarm values computed using the options given in `control`. The only difference with other algorithms provided in the package is that it demands that the input `sts`-object contains an additional `data.frame` describing the arrival of the cases that got sick at time t, *i.e* all $n_{t,d}$ instead of only N_t. We shall illustrate the function by using data for the reported number of cases of Salmonella in Germany 2001-2014 aggregated by data of disease onset.

```
# Aggregate counts over Germany
R> data(salmAllOnset)
```

The additional `data.frame` giving the $n_{t,d}$ has the structure shown below in Table 3.1.

```
R> head(salmAllOnset@control$reportingTriangle$n, n = 12)
```

	d=0	d=1	d=2	d=3	d=4	d=5	d=6	d=7	d=8	d=9	d=10
2001-01-01	0	0	0	0	377	52	62	37	13	68	45
2001-01-08	0	0	0	315	88	81	18	20	81	57	15
2001-01-15	0	0	158	147	152	29	35	88	41	16	3
2001-01-22	0	24	92	190	83	51	78	33	17	5	18
2001-01-29	2	22	121	131	136	118	52	28	8	19	2
2001-02-05	3	27	68	189	196	48	46	12	11	4	2
2001-02-12	1	13	133	334	133	90	25	28	4	5	5
2001-02-19	1	25	210	191	163	54	44	10	7	8	5
2001-02-26	1	53	146	286	124	56	26	12	22	4	7
2001-03-05	9	32	136	220	142	50	13	18	5	6	14
2001-03-12	6	44	175	224	100	42	25	11	12	18	8
2001-03-19	3	43	175	163	116	60	29	13	24	10	4

Table 3.1: $n_{t,d}$ for Salmonella cases in Germany, aggregated by week of disease onset t, for the first weeks of 2001. The left column indicates t as the date of the Monday of the week.

Each line corresponds to the division of the N_t by delay. The N_t corresponding to the `data.frame` above are

```
R> head(observed(salmAllOnset), n = 12)
```

	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12
observed1	654	675	669	591	639	606	771	718	737	645	665	640

We can actually check that the sum of each row of this `data.frame` is equal to N_t. The following command outputs ”TRUE”.

```
R>  all(rowSums(salmAllOnset@control$reportingTriangle$n)
       == observed(salmAllOnset), na.rm = TRUE)
```

Now that we have shown the structure of the data our algorithm works with, we shall present an example of its use. In the code below we produce two `sts`-object based on the same input data, using the normal approximation for the posterior distribution of the parameters, for weeks 410 to 412. The first output ignores delays – `delay=FALSE` – while the second output makes full use of our methodological development – `delay=TRUE`. Figure 3.12 was produced with such a code.

```
R> rangeTest <- 410:412
R> alpha <- 0.05
# Control slot for the proposed algorithm with D=0 correction
R> controlNoDelay <- list(range = rangeTest, b = 4, w = 3,
R>                       pastAberrations = TRUE, mc.munu=10, mc.y=10,
R>                       verbose = FALSE,
R>                       alpha = alpha, trend = TRUE,
R>                       limit54=c(0,50),
R>                       noPeriods = 10, pastWeeksNotIncluded = 26,
R>                       delay = FALSE, inferenceMethod = "asym")
# Control slot for the proposed algorithm with D=10 correction
R> controlDelay <-  list(range = rangeTest, b = 4, w = 3,
R>                       pastAberrations = TRUE, mc.munu = 10, mc.y = 10,
R>                       verbose = FALSE,
R>                       alpha = alpha, trend = TRUE,
R>                       limit54=c(0,50),
R>                       noPeriods = 10, pastWeeksNotIncluded = 26,
R>                       delay = TRUE, inferenceMethod = "asym")
R> salm <- bodaDelay(salmAllOnset, controlNoDelay)
R> salm.delay <- bodaDelay(salmAllOnset, controlDelay)
R> par(mfrow=c(1,2))
R> plot(salm.Normal)
R> plot(salm.delay)
```

Now that we have explained how to use `bodaDelay` we shall compare this function with the `boda` function that corresponds to the work of Manitz and Höhle (2013) and that was implemented by the authors of the method.

3.5.2 Differences between `boda` and `bodaDelay`

The function `bodaDelay` we added to the package is not only an extension of `boda` to account for reporting delays in surveillance data. Indeed, it currently has a simpler intercept but one could add a time-varying intercept to `bodaDelay` in future releases of the `surveillance` package. There are three main advantages of `bodaDelay` over `boda` as regards implementation. First, we make better use of `INLA`. In the first versions of `boda`, the Monte Carlo computing of the threshold contained drawing from the parameters marginals instead of their joint posterior distribution. When the parameters indeed are independent, such sampling is fine, but in the case where μ_t and ν are dependent, it is better to sample from their joint posterior distribution. Since writing the code for `bodaDelay`, we modified the code of `boda` so that when one chooses the option `joint` as sampling method, parameters values are drawn from the joint posterior distribution of the parameters. Another difference between the two algorithms is that `bodaDelay` offers a much faster inference method based on the asymptotic normal approximation for the posterior distributions of the parameters, using the R MASS package. This asymptotic method would allow for routine applications of the algorithm thanks to the gain in computing time. Last, but not least, the code of bodaDelay is more modular than the code of boda. There are two functions in boda, one main function and one for fitting the GLM, whereas bodaDelay is composed of five functions: one main function as well, one for preparing the data.frame of data for regression from the provided sts-object, one for writing the formula for the GLM, one for fitting the GLM and one for computing the threshold. This makes bodaDelay easier to test with unit tests. Moreover, we think that a modular code is easier to read and understand, which also supports checks or further developments of the algorithm by new contributers. Furthermore, pieces of a modular code can more easily be copied and pasted into new code. We therefore think that our implementation bodaDelay constitutes original work that allows to use our methodological developments thanks to sound code. We present an improvement of both boda and bodaDelay, which makes the quantile computation faster, in Section 5.2.1.

3.5.3 Other functions written as part of this work

As of the work on the Bayesian algorithm taking reporting delays into account, we have also written two functions for the simulation study presented in Section 3.4.1, sts_creation that simulates time series of counts that follow a negative binomial distribution with optional seasonality and time trend, and sts_observation that takes a sts-object with all $n_{t,d}$ as input and outputs a sts-object corresponding to the same time series observed earlier in time.

Below we first illustrate the use of sts_creation. We start by fixing the random seed for reproducibility. One then has to provide the parameters of the time series, the vector of dates at which observations are to be created.

```
R> set.seed(12345)
# Time series parameters
R> scenario4 <- c(1.6,0,0.4,0.5,2)
R> theta <- 1.6
R> beta <- 0
R> gamma1 <- 0.4
R> gamma2 <- 0.5
R> overdispersion <- 2
R> m <- 1
# Dates
R> firstDate <- "2006-01-01"
R> lengthT=350
R> dates <- as.Date(firstDate,origin='1970-01-01') + 7 * 0:(lengthT - 1)
```

One then prepares information about the delay distribution.

```
# Maximal delay in weeks
R> D <- 10
# Delay distribution
R> data("salmAllOnset")
R> in2011 <- which(formatDate(epoch(salmAllOnset), "%G") == 2011)
R> rT2011 <- control(salmAllOnset)$reportingTriangle$n[in2011,]
R> densityDelay <- apply(rT2011,2,sum, na.rm = TRUE)/sum(rT2011,
R>                                                       na.rm = TRUE)
```

The next step is defining the starting dates and sizes of the outbreaks that will be added to the baseline. Note that these dates and sizes could be randomly drawn as in our simulation study.

```
# Dates and sizes of the outbreaks
R> datesOutbreak <- c(as.Date("2008-03-30"),
R>                    as.Date("2011-09-25", origin = "1970-01-01"))
R> sizesOutbreak <- c(2,5)
# alpha for the threshold
R> alpha <- 0.05
```

The last step is providing all these parameters as arguments of the sts-creation function, for getting a sts-object that one can *e.g.* plot which would reproduce Figure 3.7.

```
# Create the sts with the full time series
R> stsSim <- sts_creation(theta = theta,beta = beta,
R>                        gamma1 = gamma1, gamma2 = gamma2,
R>                        m = m, overdispersion = overdispersion,
R>                        dates = dates, sizesOutbreak = sizesOutbreak,
R>                        datesOutbreak = datesOutbreak,
R>                        delayMax = D,
R>                        densityDelay = densityDelay,
R>                        alpha = alpha)
R> plot(stsSim)
```

The function sts_observation only has three arguments: the sts-object that has to be truncated, the new date of observation – that has to be smaller or equal to the last date of the time series – and a Boolean indicating if the timepoints located after the new date of observations should be simply erased or keeped and filled with NA, which could be useful for plotting purposes for instance.

```
R> data("salmAllOnset")
R> salmAllOnsety2013m01d20 <- sts_observation(salmAllOnset,
R>  dateObservation = "2014-01-20", cut = FALSE)
```

Using such a code, one can create illustrative figures such as Figure 3.4. Thanks to these two functions added to the R package `surveillance`, one can better understand, reproduce or extend our simulation study. We feel that making these two functions available was an important step towards reproducibility.

Before concluding this chapter and reflect on our own contribution, we shall present the contribution of other researchers to the issue of aberration detection in the presence of reporting delays.

3.6 Another algorithm correcting for reporting delays

Noufaily et al. (2015) proposed another algorithm for aberration detection tailored to dealing with right-truncated data, defined in a frequentist framework. In this Section, we present the rationale of their method in a condense form, while using our own notation for simplifying the comparison. They propose to use the following test statistic:

$$Q = \sum_{t=T-m}^{T} N(t,T) - \zeta_T(m)$$

where m is the size of a moving window and where

$$\zeta_T(m) = \sum_{t=T-m}^{T} \left(\mu_t \sum_{d=0}^{\min(D,T-t)} p_d \right).$$

Figure 3.13 shows the zone used for calculating Q: Q is the difference between the observations $n_{t,d}$ and the estimator of their mean in the triangle of counts for event time and delay respectively spanning between $t = T$ and $d = 0$, $t = T - m$ and $d = 0$, and $t - m$ and $\min(D, m)$.

They estimate μ_t using a quasi-Poisson GLM similar to the one used in Farrington's and Noufaily's method (Farrington et al., 1996; Noufaily et al., 2013) with overdispersion parameter ϕ. They assume that the delay is overdispersed with respect with a multinomial distribution of parameters $p_0, p_1, \ldots, p_D$ so that the variance of $n_{t,d}$ is $\psi N_t p_d(1 - p_d)$ and so that the covariance of $n_{t,d}$ and $n_{t,d'}$ is $-\psi N_t p_d p_{d'}$. For the delays they use a quasi-multinomial model.

With these assumptions they obtain

$$\text{Var}(Q) = \psi \zeta_T(m) + (\phi - \psi)\delta_T(m)$$

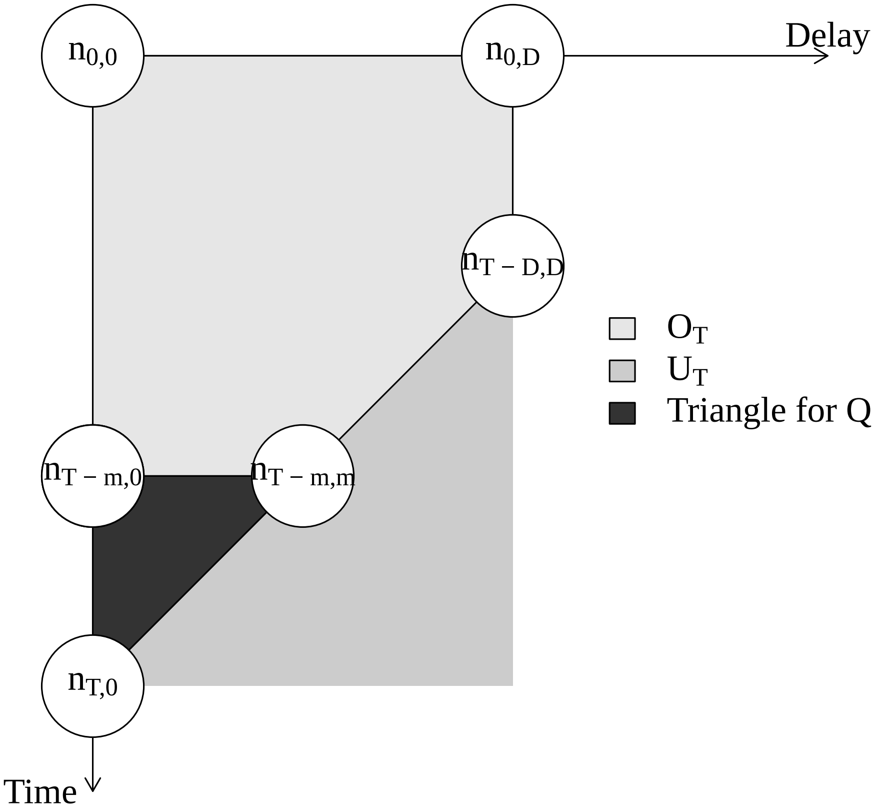


Figure 3.13: Reporting triangle at time T, similar to Figure 3.3 except this one shows the zone used for calculating Q in the method presented in Section 3.6: the triangle of counts for event time and delay respectively spanning between $t = T$ and $d = 0$, $t = T - m$ and $d = 0$, and $t = T - m$ and $\min(D, m)$.

where

$$\delta_T(m) = \sum_{t=T-m}^{T} \mu_t \left(\sum_{d=0}^{\min(D,T-t)} p_d \right)^2 .$$

They advocate for a 2/3 power-transformation so that they actually use

$$Q^* = \left(\sum_{t=T-m}^{T} N(t,T) \right)^{\frac{2}{3}} - \zeta_T(m)^{\frac{2}{3}}$$

as a test statistic, and assume it is normally distributed with

$$\mathrm{Var}(Q^*) \simeq \frac{4}{9} \zeta_T(m)^{\frac{1}{3}} \left\{ \psi + \frac{(\phi - \psi)\delta_T(m) + \mathrm{Var}(\zeta_T(m))}{\zeta_T(m)} \right\} .$$

Finally they define the exceedance score Z^* as

$$Z* = \frac{T^*}{z_\alpha \left\{\text{Var}(T^*)\right\}^{\frac{1}{2}}}$$

where z_α is the $(1-\alpha)$-quantile of the standard normal density. An alarm is defined for T if $Z^* > 1$.

They describe how to choose m in order to ensure good specificity, sensitivity and timeliness, and then proceed on to testing the algorithm on simulated time series with known injected outbreaks first with a constant mean $\mu_t = \mu_0$ and then with seasonality so that $\mu_t = \exp(\beta_0 + \beta_2 \sin((2\pi t)/52 + \beta_3))$; and lastly on real time series from the UK national surveillance system of infectious diseases. These time series contain unknown outbreaks.

On top of taking reporting delays and their possible overdispersion into account, the method of Noufaily et al. (2015) has the advantage of summing deviations over several timepoints, but m has to be chosen and is fixed. Moreover, the delay distribution and the baseline counts are estimated separately, although this has the advantage of allowing to use less old counts for estimating the delay distribution. This might make the method more robust to changes in the delay distribution, which was not tested in Noufaily et al. (2015). Lastly, this algorithm however a strong disadvantage: it disregards estimation uncertainty. This could lead to a higher FPR which is known to lead to user fatigue.

Since there was no implementation of the method available, we could not compare its performance to our method. This could be the goal of further work.

3.7 Conclusion

In this Chapter, we introduced a novel regression based statistical algorithm for aberration detection in public health surveillance data if reporting delays are present. For dealing with the issue of reporting delays, our method corrects the alarm threshold, rather than the observed counts, which we think is a better way of communicating the delay adjustment. Moreover, we developped our method in a Bayesian framework, so that the derivation of the decision-supporting threshold encompasses both estimation and observation uncertainty. Finally, the proposed algorithm was implemented in the R package `surveillance` (Höhle, 2007; Salmon et al., 2015) as the function `bodaDelay`.

In our simulation study the suggested method did detect outbreaks earlier without producing more false alarms than the nominal level. However, further work remains to be done regarding the application of our algorithm in practice, including tests at a public health institute, which we present in Section 4.3.1. If adjusting aberration detection for reporting delays in routine use, for each time unit of aggregation (week, day) to be monitored one could get many alarms in a row. This can be quite cumbersome: each time unit is monitored until no new case is reported for this week, *i.e.* $D+1$ times. We reckon that one must develop an efficient routine so that the alarms that have already been marked once do not lead to user fatigue, while at the same time accounting for the possible increase of evidence for an aberration. Lastly, a possible improvement could be a correction for delay at the daily level, but this would make the workload even higher. Such an improvement would have to take day-of-the-week effects into account.

Besides, the way we simulate (annual and biannual) seasonality with sinus-cosinus functions may be quite artificial, but we chose this model because it was used successfully by Noufaily et al. (2013) and we think that it does not invalidate our comparison of algorithms with and without delay corrections. We hope that with real data, the seasonality model of the algorithm itself, which is a 10-level factor variable, would capture more complicated annual seasonality. However, we think the inclusion of more realistic models for seasonality in the simulations as well as in the detection algorithm itself should be the goal of future research.

The use of a Bayesian framework offered a very flexible framework since the right-truncation can be viewed as a missing data problem and there is no fundamental distinction between parameters and data in Bayesian settings. This would also allow for the integration of more flexible delay distribution models in case this is needed. In the future, we aim at exploring two features of such models: how feasible and how much an improvement it is to integrate time-dependent delay distributions, and how the maximal delay chosen and the number of classes in the multinomial distribution of delay impact the results. Furthermore, we currently do not try to on-line identify aberrant past counts, which is a feature one could later add for increasing the probability of detection. In Manitz and Höhle (2013) this was handled by having an outbreak indicator in the model, for which one has to either know which time units contained outbreaks, or to spot outliers. Since one rarely knows all past outbreaks, one could spot outliers based on the predictive mid-p value instead, which was recommended for count data in Held et al. (2010), or use the same approach as Fried et al. (2015) for spotting outliers.

It is worth pointing out that reporting delays do not only include cases being notified later but also information such as pathogen subtype to be informed later than other case details. We think that including such features of case reports, possibly by the use of multi-state models, would be yet another step in bringing a more statistical modelling perspective into the analysis of the data transmission mechanisms underlying surveillance systems. In the next Chapter, we present the application of aberration detection algorithms at the Robert Koch Institute.

Chapter 4

Routine Aberration Detection at the Robert Koch Institute

This chapter except Section 4.3 and 4.4 corresponds to the description of the RKI aberration detection system in the article ***M. Salmon****, D. Schumacher, H. Burmann, C. Frank, H. Claus, M. Höhle. A system for automated outbreak detection of communicable diseases in Germany,* Eurosurveillance*, accepted for publication. Section 4.3 is original work, and Section 4.4 corresponds to a Section of the article* ***M. Salmon****, D. Schumacher, M. Höhle. Monitoring Count Time Series in R: Aberration Detection in Public Health Surveillance,* Journal of Statistical Software*, accepted for publication.*

In recent years, more and more data has been collected for the routine surveillance of infectious diseases. For instance, in Germany the Robert Koch Institute (RKI) implemented a national electronic surveillance system (SurvNet@RKI) in 2001 as a response to the then newly enacted Protection against Infection Act that requires regular data collection on a number of notifiable diseases (Faensen et al., 2006). Cases are first reported by laboratories or physicians to local health authorities that may perform further investigations, and then transmitted to the RKI via federal state health authorities.

In addition to increasing data collection, a multitude of different outbreak detection algorithms for routinely collected public health data has been published (Unkel et al., 2012). Nonetheless, the added value of applying statistical methods for aberration detection at public health institutions is still subject to discussion because of several challenges, among which automating the data analysis and identifying signals without producing a plethora of signals. For instance, as of October 2015 the SurvNet@RKI database contained approximately 6.0 million case notifications in 88 different reporting categories such as Salmonella or Norovirus, while outbreaks often become apparent when inspecting certain subsets of the data, e.g., within a specific geographical area or even a specific age group (Koch et al., 2005). The problem is therefore to promptly identify these relevant subsets in the haystack of data. One statistical approach to this problem is to regularly analyse the data as multiple univariate time series in order to detect unexpected aberrations in specific subsets.

Nowadays, in many public health institutions, a semi-automatic monitoring system is in operation (for examples in Europe see Hulth et al. (2010)). But because of too many signals or because of a misalignment between users' needs and signal presentation, the system output often has little impact on the practical work of these institutions. First attempts to focus more on the user perspective of monitoring systems are, however, found in Cakici et al. (2010) and Kling et al. (2012). Our goal was to develop and establish an automatic information system that supports epidemiologists at the RKI in timely detecting potential outbreaks of communicable diseases.

In this chapter, we present the implementation of a novel automated monitoring system at the Robert Koch Institute in Germany. The new system is now in routine use at the RKI for most of the reporting categories. In the following, we shall describe the architecture of the system and our design decisions in the first section. Its functioning and first results with its routine use are reported in the next section. Finally, we round off with a discussion of current and future improvements. In sharing our experiences we aim to provide valuable information to others working on similar surveillance systems.

4.1 System design

4.1.1 Defining features of the system

Altogether, we wanted to obtain constant quality results as well as a standard procedure for the routine surveillance workflow in our organisation. This objective lead to specific requirements for the system that were largely in line with the checklist for computer-supported outbreak detection systems formulated by Hulth et al. (2010). This article contains a dozen recommendations such as user-friendliness and tight integeration with the database. The development of the system and the refinements of the requirements were then conducted iteratively. Based on the rapid prototyping philosophy we initially focused on building a first prototype for one reporting category, namely Salmonella with its many serotypes.

Once the prototypes of the components had produced first results, we started discussing the output of the system with a few users for Salmonella. Once the system produced satisfactory results for this reporting category, we progressively scaled up the system to 48 reporting categories which account for roughly 80% of all received cases. Our goal has always been to create a general system for a variety of diseases instead of highly disease-specific solutions. In addition to the one-on-one discussions with the system users, we received more and more feedback and feature requests as the system grew.

4.1.2 System design

The system consists of two components: an automatic component routinely monitors the data and a manual component enriches data queries with ad hoc aberration detection; Figure 4.1 depicts this structure.

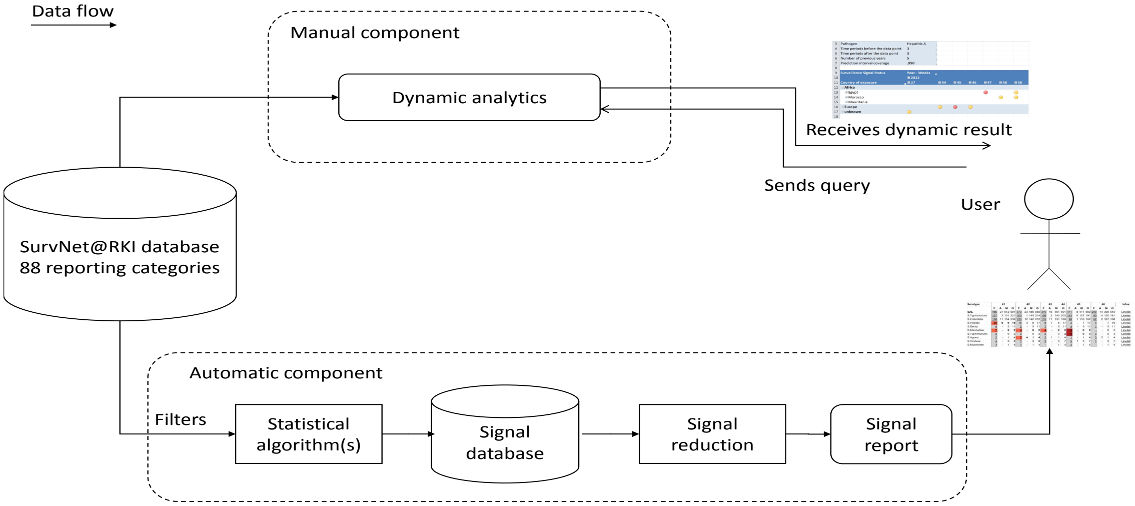


Figure 4.1: Structural overview of the automatic surveillance system.

4.1.3 Automated analytical process

As shown in Figure 4.1, the automatic component consists of three subsystems: an analytical process, a signal database and a signal interface. The analytical process analyses the data with aberration detection algorithms and in case of an unusually high number of cases produces signals which are stored in the signal database and communicated to the user through the signal interface.

The analytical process monitors the SurvNet@RKI case counts of the last weeks on a daily basis for all reporting categories selected for aberration detection. Since outbreaks can occur within specific subsets of the population, e.g., at a specific location and within an age group, we monitor in parallel numerous time series corresponding to the respective subsets of the population in order to be able to see signals that would be invisible when analysing the whole population. In particular, we stratify the time series by pathogen subtype (e.g. Salmonella serotype such as Salmonella (S.) Infantis) or symptom (e.g. pneumonia), location (federal states, counties), age group, sex, place of exposure and mortality status. This stratification yields a set of univariate time series for each reporting category aggregated on a weekly or monthly scale. The number of (negative) diagnostic tests performed – biological samples analyzed for looking for instance for *Salmonella* but whose analysis in the laboratory revealed no *Salmonella* presence – is not part of the mandatory reporting system. Therefore, the analysis of the numbers is sensitive to variations due to, e.g., changes in laboratory procedures or doctor requesting patterns.

The system applies the algorithm of Noufaily et al. (2013) to each time series for getting a threshold for each observed count. The last four years of historic data are used as reference values for the algorithm, which accounts for seasonality, time trend and presence of past outbreaks in the record, to provide a threshold specific to each monitored week. A signal is generated for time t_0 if the observed number of cases exceeds the threshold. We refer to Noufaily et al. (2013) or Section 2.3 of this thesis for a more detailed description of the algorithm. To address reporting delay we monitor the last six weeks, i.e. it is possible to obtain a signal for one of the last weeks given the current data.

The automated analytical process was initially implemented solely in the statistical programming language R using the `surveillance` package (Höhle, 2007; Salmon et al., 2016) for the detection part and other R packages (Ripley and Lapsley, 2012; Dowle et al., 2013; Wickham, 2013) for the data pre- and post-processing steps as well as support for behaviour driven software development. As in other systems (Reis et al., 2007; Cakici et al., 2010), the automated component was built in a modular way so the detection component can incorporate different detection algorithms. R was chosen over other programming languages as it allowed us to directly use a variety of statistical detection algorithms and visualization procedures out of the box and because of its ability to rapidly prototype statistical procedures. During subsequent developments we ported large parts of the data management components to the programming language C# to harmonize the system with existing IT infrastructure at the RKI.

4.1.4 Signal database

The signal database stores signals generated by the analytical process. A signal corresponds to statistical evidence that the case count in a given subset of the data is higher than we would expect it to be based on historic data. A signal combines two types of information. On the one hand, a signal contains information about that data segment in which case counts were detected by a statistical algorithm, i.e. a filter on the data with a set of attributes; for example ”Hepatitis A; Week 25 of 2013”. On the other hand, a signals contains about the algorithm itself, its configuration and its output, e.g. the detection threshold.

This definition can be used directly to store the signals in the signal database and enables subsequent processing of the signals. This has several direct advantages over analysis and communication as a combined step: the signals can have an age, they can be more or less important, they can be similar to each other and they can disappear over time due to new data being received. In addition, signals can be communicated differently based on aspects such as user preferences.

4.1.5 Signal interface and communication

Signals are communicated to the user through predefined report templates for each reporting category. The respective reports display relevant signals found for that category within a given period. In addition to these main reports, several small reports display new signals found recently, line lists and a spatial visualisation of the cases. The main reports are archived as Microsoft Excel files once a day and are sent by email to epidemiologists in charge of specific reporting categories once a week. Such a push/pull principle of communication was inspired by other monitoring systems such as the one described in Reis et al.

(2007).

The signal interface uses Microsoft SQL Server Reporting Services (Microsoft Corp., 2012) as a technological basis, mainly because it is already used at the RKI for various other tasks. It allows quick development of the reports that can be accessed from the Intranet through a web-browser and supports the exportation of the reports as Microsoft Excel files. Furthermore, in order to support the decision on whether a signal is relevant, the user can click on any case count in the report to directly see the associated list of cases from the SurvNet@RKI database (line list).

4.1.6 Signal abstraction

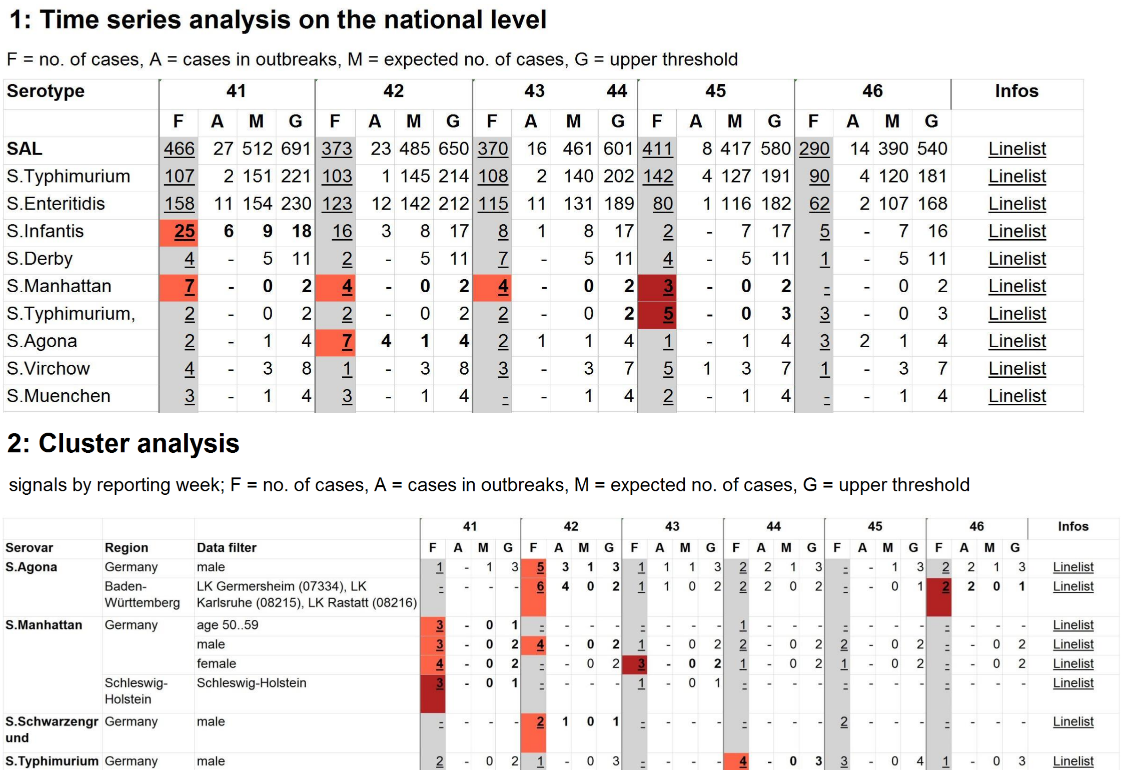
However, during reporting a problem arises due to the monitoring of the many time series aggregated in different ways for a reporting category: given a set of closely related signals, what signals should be shown to the user? Closely related signals could be signals for Salmonella in week 22 in Bavaria, for Salmonella in week 22 in Munich, for Salmonella in week 22 in Munich for males. We developed a method to reduce the number of signals for reporting categories with a high number of signals, such as Salmonella.

The procedure utilizes the fact that each signal is associated with a filter which has a set of attributes, e.g. geographic location, temporal location, sex and age group. Now, given a set of signals that is available for reporting, first we determine similar signals by partitioning the original set of signals into a set of signal groups. All signals within a specific group have equal values for a number of filter attributes. For example, we could group all signals by week so that each signal group consists of signals with the same reporting week; e.g. 2013 week 42. In the system at the RKI we group by all attributes except for sex, age group and reporting location of the signal. Thus all signals within a group will not necessarily have the same values for sex, age group and location.

In a second step, within each of these groups we filter out signals which do not add much information to the report. This filtering is done by so called filter-relations, which allow us to rank and compare signals according to a predefined metric. We use three different relations: *more specific than*, *more general than* and *more specific on the location and more general on age and sex*. The user can select between having no reduction, one of the three relations or a combination of the first two relations. In our example the most general signal would be the signal for Salmonella in week 22 in Bavaria, whereas the most specific signal would be the signal for Salmonella in week 22 in Munich for males. It is hence possible to focus the analysis of the signals on specific aspects, e.g. locating a centre of a possible outbreak by displaying only the most specific signals in terms of their filter attributes.

4.1.7 Manual analytical component

In addition to the automatic tool for outbreak detection we also implemented a detection tool that can be applied to almost any subset of the data defined by the user, allowing users to screen very specific time series on demand, which was a wish expressed during our meetings. This component monitors specific subsets of the data, for example case counts



1: Time series analysis on the national level

F = no. of cases, A = cases in outbreaks, M = expected no. of cases, G = upper threshold

Serotype	41				42				43			44	45				46				Infos
	F	A	M	G	F	A	M	G	F	A	M	G	F	A	M	G	F	A	M	G	
SAL	466	27	512	691	373	23	485	650	370	16	461	601	411	8	417	580	290	14	390	540	Linelist
S.Typhimurium	107	2	151	221	103	1	145	214	108	2	140	202	142	4	127	191	90	4	120	181	Linelist
S.Enteritidis	158	11	154	230	123	12	142	212	115	11	131	189	80	1	116	182	62	2	107	168	Linelist
S.Infantis	25	6	9	18	16	3	8	17	8	1	8	17	2	-	7	17	5	-	7	16	Linelist
S.Derby	4	-	5	11	2	-	5	11	7	-	5	11	4	-	5	11	1	-	5	11	Linelist
S.Manhattan	7	-	0	2	4	-	0	2	4	-	0	2	3	-	0	2	-	-	0	2	Linelist
S.Typhimurium,	2	-	0	2	2	-	0	2	2	-	0	2	5	-	0	3	3	-	0	3	Linelist
S.Agona	2	-	1	4	7	4	1	4	2	1	1	4	1	-	1	4	3	2	1	4	Linelist
S.Virchow	4	-	3	8	1	-	3	8	3	-	3	7	5	1	3	7	1	-	3	7	Linelist
S.Muenchen	3	-	1	4	3	-	1	4	-	-	1	4	2	-	1	4	-	-	1	4	Linelist

2: Cluster analysis

signals by reporting week; F = no. of cases, A = cases in outbreaks, M = expected no. of cases, G = upper threshold

			41				42				43				44				45				46				Infos
Serovar	Region	Data filter	F	A	M	G	F	A	M	G	F	A	M	G	F	A	M	G	F	A	M	G	F	A	M	G	
S.Agona	Germany	male	1	-	1	3	5	3	1	3	1	1	1	3	2	2	1	3	-	-	1	3	2	2	1	3	Linelist
	Baden-Württemberg	LK Germersheim (07334), LK Karlsruhe (08215), LK Rastatt (08216)	-	-	-	-	6	4	0	2	1	1	0	2	2	2	0	2	-	-	0	1	2	2	0	1	Linelist
S.Manhattan	Germany	age 50..59	3	-	0	1	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	Linelist
		male	3	-	0	2	4	-	0	2	1	-	0	2	2	-	0	2	2	-	0	2	-	-	0	2	Linelist
		female	4	-	0	2	-	-	0	2	3	-	0	2	1	-	0	2	1	-	0	2	-	-	0	2	Linelist
	Schleswig-Holstein	Schleswig-Holstein	3	-	0	1	-	-	-	-	1	-	0	1	-	-	-	-	-	-	-	-	-	-	-	-	Linelist
S.Schwarzengrund	Germany	male	-	-	-	-	2	1	0	1	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	Linelist
S.Typhimurium	Germany	male	2	-	0	2	1	-	0	3	-	-	-	-	4	-	0	3	3	-	0	4	1	-	0	3	Linelist

Figure 4.2: Excerpt of the Salmonella report for weeks 41-46 of 2013: Time series analysis on the national level and cluster analysis.

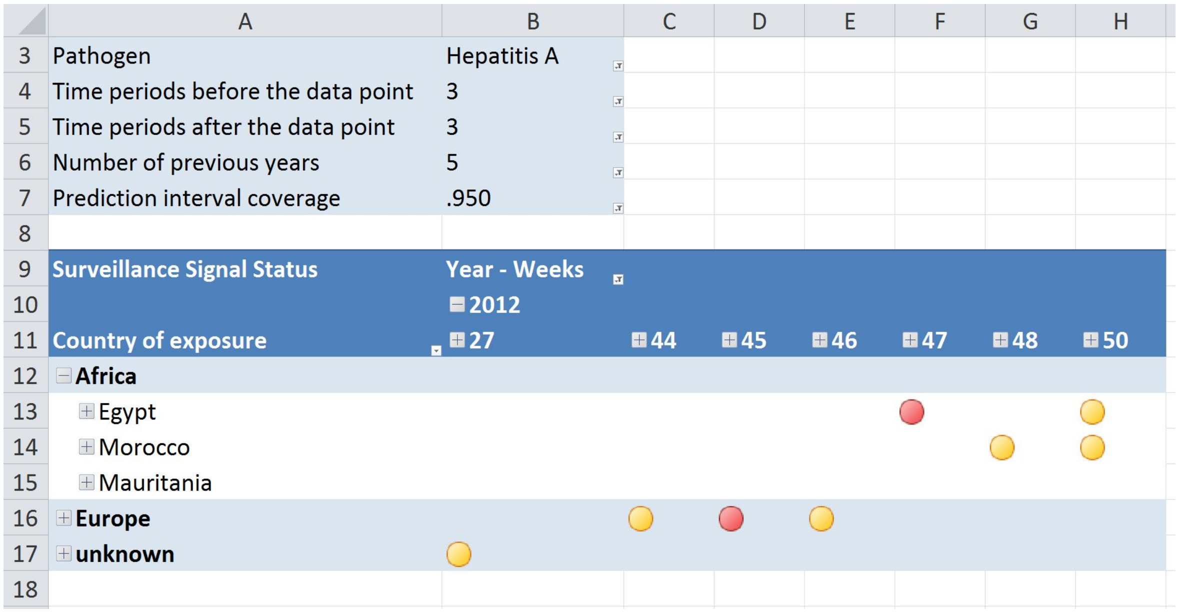
of Hepatitis A in Berlin within the last 6 weeks, through the comparison of the current counts to past data, using a method similar to the algorithm of Stroup et al. (1989).

4.2 System use

4.2.1 Report interface

As of October 2015, 62 users within the RKI and federal state health authorities receive weekly reports from the automated component and interact with the reports.

The tables on Figure 4.2 correspond to an excerpt of the Excel based report for Salmonella reported cases from weeks 41 - 46 of 2013. The report contains two data tables with a similar structure. For each week t we report the number of cases y_t, the estimated expected case count μ_t, the threshold U_t and the number cases that were manually marked as being part of an outbreak o_t in the SurvNet@RKI database. Cases are sometimes identified as a cluster by local health authorities, e.g., a cluster of Norovirus cases



	A	B	C	D	E	F	G	H
3	Pathogen	Hepatitis A						
4	Time periods before the data point	3						
5	Time periods after the data point	3						
6	Number of previous years	5						
7	Prediction interval coverage	.950						
8								
9	**Surveillance Signal Status**	**Year - Weeks**						
10		**⊟2012**						
11	**Country of exposure**	**⊞27**	**⊞44**	**⊞45**	**⊞46**	**⊞47**	**⊞48**	**⊞50**
12	⊟ **Africa**							
13	⊞ Egypt					🔴		🟠
14	⊞ Morocco						🟠	🟠
15	⊞ Mauritania							
16	⊞ **Europe**		🟠	🔴	🟠			
17	⊞ **unknown**	🟠						
18								

Figure 4.3: Example of the results of a dynamic data query in Microsoft Excel with aberration detection of hepatitis A cases associated with the country of exposure in 2012.

after a common meal. Coloured cells indicate signals for the respective week. Signals that were detected seven or more days ago are marked orange while newer signals are marked red. The first table of the report, as illustrated in Figure 4.2 (table 1), corresponds to the reported number of cases per serotype for the last 6 weeks with e.g. a signal for S. Infantis in week 41, while the second table, as illustrated in Figure 4.2 (table 2), displays the results of the stratified analysis described in the previous section. In this example we see a cluster of female S. Manhattan cases in week 41. Some of these signals prompt further checks by epidemiologists, helped by a direct link between the signal and the corresponding cases (line list). The number of signals in a report is an interplay between the number of filters with cases for the disease, the algorithm settings for the disease, and whether signal reduction is performed. From January to October 2015, the median number of signals over all filters in the weekly Salmonella report was 62.

Figure 4.3 shows an example output of the manual component for a query of hepatitis A cases for the year 2012 by country of exposure, which is not a time series routinely analysed by the automatic component, but could be of interest in particular situations. The table displays weeks of 2012 and countries where the number of cases exceeds the upper limit of the prediction interval.

4.2.2 Experiences from operation

Since 2013 the monitoring system has been widely adopted within the RKI. Although it has not been formally evaluated yet, we can observe a positive user acceptance. Further-

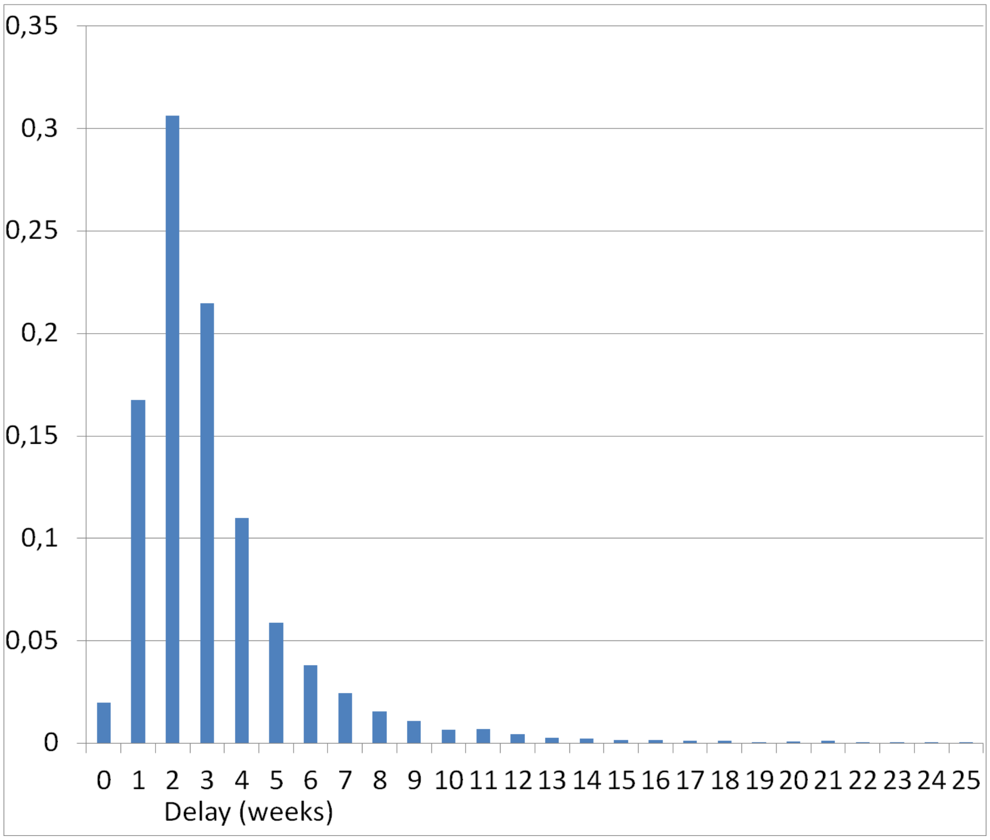


Figure 4.4: Distribution of delay for hepatitis A from date of disease onset to date of report arrival at the RKI.

more, the system already helped to trigger several outbreak investigations. For example, it detected a large local outbreak of cryptosporidiosis in August 2013 (Gertler et al., 2015). Apart from outbreak detection, we also experienced that the tool helped to provide situational awareness to the epidemiologists, especially those monitoring trends in frequently notified infections prone to causing outbreaks. Moreover, the aberration detection tool for dynamic data queries on case counts is well appreciated since it is not always straightforward to visually assess whether the numbers of a time series plot are higher than usual. The manual component now provides a statistically informed decision for this

4.3 Current work and remaining challenges

4.3.1 Testing a new algorithm

Since the current system is now in operation and appreciated by epidemiologists at the Robert Koch Institute and in federal states and local health authorities, it became possible to add new features. We have started to test `bodaDelay` that was presented in Chapter 3, first on a single disease, hepatitis A. Starting the test on only one disease is the same strategy as the one used for implementing the system itself at first. We wanted to see which data queries were needed for testing `bodaDelay` and how this new tool was received by epidemiologists before broadening its use. We can already report on some steps that had to be performed and shed light into what should be taken into account when using `bodaDelay` in practice.

When applying the algorithm on a new pathogen one has to decide which date should be chosen for aggregating the cases and which value to give to the maximal delay D. In the current system, cases are aggregated by date of report which is the date at which a case was reported to the local health authority. Indeed this date is known for all cases and there is only a short delay between date of report and date of arrival of the information at the Robert Koch Institute (Schumacher et al., 2016). However, a date such as the date of disease onset might make more sense for epidemiological investigations. Thus, when deploying `bodaDelay` for a test implementaiton, there was interest in using it for analysing time series where cases are aggregated by date of disease onset. For hepatitis A, date of disease onset is unknown for 31% of all cases. The two epidemiologists in charge of hepatitis A deemed that it was however reasonable to use this date as an aggregation date. Note that during the test the other surveillance reports using Noufaily's method (Noufaily et al., 2013) and the date of report as aggregation date were still produced so that eventually all reported cases were monitored anyway. When applying the algorithm on a new pathogen one has to decide which date should be chosen for aggregating the cases and which value to give to the maximal delay D. In the current system, cases are aggregated by date of report which is the date at which a case was reported to the local health authority. Indeed this date is known for all cases and there is only a short delay between date of report and date of arrival of the information at the Robert Koch Institute (Schumacher et al., 2016). However, a date such as the date of disease onset might make more sense for epidemiological investigations. Thus, when deploying `bodaDelay` for a test implementaiton, there was interest in using it for analysing time series where cases are aggregated by date of disease onset. For hepatitis A, date of disease onset is unknown for 31% of all cases. The two epidemiologists in charge of hepatitis A deemed that it was however reasonable to use this date as an aggregation date. Note that during the test the other surveillance reports using Noufaily's method (Noufaily et al., 2013) and the date of report as aggregation date were still produced so that eventually all reported cases were monitored anyway.

In real life, there is no real maximal delay, since sometimes cases are still reported a very long time after disease onset, *e.g.* if the report was lost. These are, however, rare occurrences. Moreover, in any case, one needs to define a maximal delay D. This is important for not having too many columns in the report sent to epidemiologists. It is justified to ignore cases that arrive after a time longer than D weeks if the distribution of delays is such that a quite high proportion of cases has a delay smaller or equal to D; and also because an alarm for cases that were diagnosed or became ill a very long time ago can be considered irrelevant as regards outbreak investigations or management measures. On Figure 4.4 we show the delay distribution for hepatitis A. 91% of all cases for which a date of disease onset is known have a delay smaller or equal to 6 weeks. Therefore, the

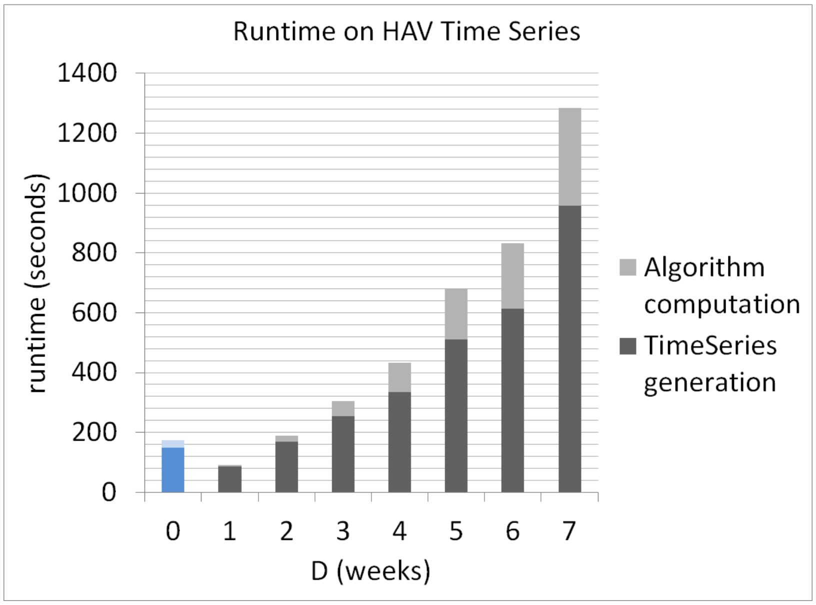


Figure 4.5: Running time for analysing all time series for hepatitis A in the RKI system with `farringtonFlexible` (grey) or `bodaDelay` (blue)

epidemiologists decided to test `bodaDelay` with $D = 6$ weeks. Since cases with a delay greater than 6 weeks are thus ignored, and since we only know the date of disease onset for 79% of cases, 72% of reported cases are monitored by the algorithm correcting for reporting delays.

There are other factors to weigh in when using `bodaDelay`. One is the running time. The running time of `bodaDelay` is higher than the running time of `farringtonFlexible`, as illustrated in Figure 4.5. This is due to the subdivision of counts according to delay that makes both time series generation from the database and computing by the algorithm itself longer. Using `bodaDelay` instead of or on top of `farringtonFlexible` therefore uses more computing resources. Another important factor are human resources if one decides to make two algorithms run in parallel: epidemiologists then need more time because they have to analyse two reports.

The test of `bodaDelay` on hepatitis A was not very conclusive, since the epidemiologists preferred to work with the default report produced with the algorithm of Noufaily et al. (2013) which they deemed more sensitive. A first positive result was however that the new algorithm is now working with the appropriate data query from the RKI database. Moreover, epidemiologists are interested in the results. A next step will be to expand its test on other diseases, which might happen in the course of the renewal of the surveillance system through the project DEMIS (German electronic reporting system for infectious diseases).

4.3.2 How to include several algorithms into one system?

Right now, the whole RKI system uses only one algorithm for aberration detection, presented in Section 2.3. But as routine aberration detection is now well established, there is interest into including new algorithms in the system, such as `bodaDelay` with tests going on as described in the previous Section, or the algorithm presented in Section 2.5 since epidemiologists express the wish to get a CUSUM method. This leads to the question of defining a strategy for using several algorithms in one surveillance system. How to include several algorithms into one system? Questions related to this issue are *e.g.*: should one have several algorithms running on the same time series, or choose one best or preferred method for each time series or pathogen? How should results be presented to avoid users fatigue in the presence of many signals originating from different algorithms?

In the CASE system implemented in the Swedish national public health institute, several algorithms can be applied to the same pathogen (Cakici et al., 2010). Having the same algorithm(s) for all time series corresponding to one pathogen may on the one hand seem very reasonable, since often one person will review all these signals and may wish to get signals corresponding to the same detection method. On the other hand, though, the regression model for the mean in the algorithm may have to be different for, say, the time series of *Salmonella* Typhimurium (very frequent subtype of *Salmonella*) in whole Germany and the time series of *Salmonella* Agona (rarer subtype) in the (rather small) federal state of Hamburg.

If one were to choose a very specific algorithm for a pathogen, or each subtype, or any category of time series, how should the decision be made? One could use model selection based on the tools presented in Section 1.2.5, and exemplified in Section 5.1.1, for choosing the best regression model. As regards the detection method itself, Bayesian *vs.* frequentist or single-timepoint *vs.* CUSUM, this could be decided upon users preferences and/or measures of performance of the algorithms on the considered time series, maybe with superimposed simulated outbreaks, which is the subject of Section 2.2. This selection could *e.g.* be made once a year so that all reports from one year can be comparable.

Last, but not least, if one is to apply several algorithms to the same time series, one will get more signals including false alarms. This demands a very good presentation of the reports and makes the link to the cases database even more important since users cannot spend much time analysing each signal. How exactly should an ideal presentation look like could be a further research issue, although one could be guided by the Swedish experience desbribed in Cakici et al. (2010). Sorting signals based on *e.g.* the posterior probability of the observed time count could be a way to make reports readable, but how do such probabilities originating from different algorithms compare?

These are all open questions to which there are probably no *right* or *wrong* answers. These thoughts should rather guide a discussion with the epidemiologists when adding new algorithms. The development of the RKI system until now provides a good example of users involvement: statistical counseling was provided as part of this thesis work to explain possibilities and limitations of algorithms to users. Furthermore, the design and development of the system were supported by informatic counseling so that the best profit was taken from epidemiology, statistics and informatics.

4.4 A simpler surveillance system

In this Section we reproduce a part of the article ***M. Salmon***, *D. Schumacher, M. Höhle. Monitoring Count Time Series in R: Aberration Detection in Public Health Surveillance,* Journal of Statistical Software*, accepted for publication, where we explain how to build a surveillance system with little resources. This text shows our efforts towards the wide applicability of algorithms for aberration detection.*

Suppose you have a database with surveillance time series but little resources to build a surveillance system encompassing all the above stages. Using `R` and `Sweave` or `knitr` for LaTeX you can still set up a simple surveillance analysis without having to do everything by hand. You only need to input the data into `R` and create `sts`-objects for each time series of interest as explained thoroughly in Höhle and Mazick (2010). Then, after choosing a surveillance algorithm, say `farringtonFlexible`, and feeding it with the appropriate `control` argument, you can get a `sts`-object with upperbounds and alarms for each of your time series of interest over the `range` supplied in `control`. For defining the range automatically one could use the R-function `SysDate()` to get today's date. These steps can be introduced as a `chunk` in a `Sweave` or `knitr` document that will be translated it into a report that you can send to the epidemiologists in charge of the respective pathogen whose cases are monitored.

Below is an example of a short segment showing the analysis of the *S. Newport* weekly counts of cases in the German federal states Baden-Württemberg and North Rhine-Westphalia with the improved method implemented in `farringtonFlexible`. The package provides a `toLatex` method for `sts` objects that produces a table with the observed number of counts and upperbound for each column in `observed`, where alarms can be highlighted by for instance bold text. The resulting table is shown in Tab. 4.1.

```
R> data("salmNewport")
R> today <- which(epoch(salmNewport) == as.Date("2013-12-23"))
R> rangeAnalysis <- (today - 4):today
R> in2013 <- which(isoWeekYear(epoch(salmNewport))$ISOYear == 2013)
R> algoParameters <- list(range = rangeAnalysis, noPeriods = 10,
                          populationBool = FALSE,
                          b = 4, w = 3, weightsThreshold = 2.58,
                          pastWeeksNotIncluded = 26, pThresholdTrend = 1,
                          thresholdMethod = "nbPlugin", alpha = 0.05,
                          limit54 = c(0, 50))
R> results <- farringtonFlexible(salmNewport[, c("Baden.Wuerttemberg",
                                               "North.Rhine.Westphalia")],
                       control = algoParameters)
R> start <- isoWeekYear(epoch(salmNewport)[range(range)[1]])
```

```
R> end <- isoWeekYear(epoch(salmNewport)[range(range)[2]])
R> caption <- paste("Results of the analysis of reported S. Newport
            counts in two German federal states for the weeks W-",
            start$ISOWeek, "-", start$ISOYear, " - W-", end$ISOWeek,
            "-", end$ISOYear, " performed on ", Sys.Date(),
            ". Bold upperbounds (UB) indicate weeks with alarms.",
            sep="")
R> toLatex(results, caption = caption)
```

Year	Week	Baden-Wuerttemberg	Threshold	North-Rhine-Westphalen	Threshold
2013	48	0	5	0	4
2013	49	1	3	0	3
2013	50	1	3	0	3
2013	51	2	3	0	3
2013	52	**3**	2	1	3

Table 4.1: Results of the analysis of reported S. Newport counts in two German federal states for the weeks W-48 2013 - W-52 2013 performed on 2015-06-04. Bold upperbounds (thresholds) indicate weeks with alarms.

The advantage of this approach is that it can be made automatic. The downside of such a system is that the report is not interactive, for instance one cannot click on the cases and get the linelist. Nevertheless, this is a workable solution in many cases – especially when human and financial resources are narrow. The RKI was able to build a more elaborate system. Nonetheless, part of the work related to this thesis was contributing to free software and the associated documentation in order to allow anyone to take advantage of state-of-the-art detection algorithms, instead of having to resort to *e.g.* building interactive Excel files from scratch.

4.5 Conclusion

We developed a system that provides results that are fairly easy to understand and to use, while being based on sound statistical methods, with disease- and user-specific adjustments. The implemented system was the result of an interdisciplinary collaboration between computer scientists, statisticians and epidemiologists combining the best of their respective worlds (user focused system design, proper treatment of uncertainty and infectious disease knowledge) to obtain a decision support tool useful for everyday practice. Targeting the users helped designing the software and will help sustaining it while institutionalising knowledge about routine aberration.

Although the system already produces valuable results for the routine work at the RKI, a number of future improvements could be tackled. We could work on the problem of comparing frequently incomplete first-version data (e.g. where a pathogen subtype and a possible travel history of the case may now be known yet) to historic more complete last-version data (e.g. where subtype and likely country of infection have been added). Moreover, we could handle reporting delays with specific detection algorithms (Noufaily et al., 2015; Salmon et al., 2015). Furthermore, we are currently only able to detect outbreaks when case numbers are above the threshold in at least one week; i.e. if an outbreak emerges very slowly over several weeks it might not be detected quickly. Here, CUSUM-oriented procedures could be better at picking up the signal (Höhle and Paul, 2008). On a geographical level, only a fixed set of regions is monitored: Germany as a whole, federal states, counties, and each county together with its adjacent neighbours (which may overlap state borders). Thus we are also only able to geographically detect outbreaks that are visible in one of these predefined county clusters. However, the architecture of the system would allow us to include more sophisticated space-time methods such as in the works of Tango et al. (2011); Neill (2012); Kulldorff (1997) into the surveillance process. Additionally, performing multiple tests on overlapping data leads to a high number of false alarms. Currently, we accept rather high rates of false positive signals, but offer the epidemiologists tools to delve deeper into the data generating the signals in order to better understand the context. A framework for controlling overall false alarm rates for each user in combination with the signal abstractions could further improve user acceptance.

We think that these accounts of successful implementations at public health institutions are an important contribution to the surveillance of infectious diseases, because automatic detection systems are much needed in the current big data environments arising from routine surveillance data collection. Our aim is to explain the RKI development strategy and user-focus in order to gain acceptance. Finally, a more technical article about the R package `surveillance` describing the algorithmic functionality of the package exists (Salmon et al., 2016), which more technically inclined readers of the Eurosurveillance manuscript might seek for details. Chapter 2 and Section 4.4 of this thesis were based on this manuscript.

The amount of data held by public health institutes will certainly continue to grow in the near future. As a consequence, automatic outbreak detection systems, as the one presented here, are becoming increasingly important. At the same time, care is needed to integrate such a system into the workflow and hence take further steps towards actual user acceptance. From an organisational point of view, a challenge here is to design effective guidelines on how the generated signals are to be handled in a standardized way. This could range from signals being considered only as an additional resource for surveillance to a procedure where each signal has to be explicitly checked by an epidemiologist. Here, the resources available for such investigations play an important role. With a tool in place it becomes possible to tailor the detection even more to the needs of the users, e.g., by actively including user feedback in the statistical detection algorithms. Including user feedback could start by collecting appropriate data about users' reaction to each signal. As of now, our article provided enough insight into our own experience with an automatic surveillance system to motivate the development and maintenance of similar decision support tools in other countries. Moreover, this Chapter completed the article by showing how to build a simpler surveillance system, and by presenting the current work on the system at the RKI, which should always remain a work in progress. In the next Chapter, we discuss remaining theoretical challenges for outbreak detection algorithms.

Chapter 5

Future methodological challenges for outbreak detection

In this Chapter, we give an outlook on potential and ongoing future research topics within the scope of count time series monitoring, in particular aimed at public health surveillance. We first present two main challenges for regression models when applied to surveillance count time series: autocorrelation and the need for robust distributional assumptions. After this, we focus on issues related to computing the decision-threshold for monitoring: we propose a faster method for computing the quantile of the posterior predictive distribution for the algorithms presented in Sect. 2.4 and 3.3, before presenting a novel way of defining the decision-threshold. In the last Section, we give an overview of our current work for evaluating algorithms at a German national public health institute.

5.1 Better regression models?

In this section we critically look back on the regression models used in this thesis. In particular, we consider their treatment of autogression in the time series, and the consequences of the distributional assumptions that they entail.

5.1.1 Autoregression in count data time series

Count time series originating from public health surveillance data often display auto-correlation – even in the absence of outbreaks. One cause of auto-correlation can be seasonality, possibly due to climatic factors influencing diseases, for instance see Figure 2.4 with time series of counts of campylobacteriosis cases and of humidity in Germany. So far, regression models have been our main tool for handling such influences. If the regression model one uses for defining a quantile does not account for all factors, residuals may be auto-correlated, which is a symptom of a model misspecification (Fahrmeir et al., 2013). Auto-correlation in the residuals may therefore indicate that our prediction is inadequate. This may lead to higher false positive rate (FPR) or smaller probability of detection (POD) in the outbreak detection algorithms because of wrongly estimated quantiles of the predictive distribution of the current count. Therefore, checking auto-correlation in the residuals is an important step for validating the selected models. In this section, we start by showing an example of a time series, then we present models that could account for the autocorrelation, and finally we explain how one could tackle serial dependence present in surveillance data.

The example we use is the time series of weekly number of reported disease cases caused by Escherichia coli in the federal state of North Rhine-Westphalia (Germany) from January 2001 to May 2013, excluding cases of EHEC and HUS, as provided in the R package `tscount` (Liboschik et al., 2016) and shown in Figure 5.1. We chose these data for comparability with a first version of Liboschik et al. (2015). We shall fit count regression models with log link to the weeks W50-2000 to W52-2006:

- *Model 1*, a negative binomial regression model similar to the one used in Noufaily et al. (2013), which does not explicitely address auto-correlation;
- *Models 2 to 4*, three models corresponding to the three different intercept specifications of the algorithm presented in Section 2.4;
- *Model 5*, a model from Liboschik et al. (2016) with inclusion of a regression on the previous mean and observation of the process,
- and finally *Model 6*, a two-component model (Held et al., 2005).

We then compare ACF plots of the resulting residuals from the models and discuss consequences of residuals auto-correlation on prediction.

For all the fitted regression models, we assume that $y_t \sim \text{NB}(\mu_t, \nu)$ and in each model we include a time trend (linear on the log scale) and a general seasonality term, defined by

$$\text{seas}(t) = \sum_{s=0}^{1} \left[\beta_{2s} \cos\left(\frac{2\pi st}{52}\right) + \beta_{2s+1} \sin\left(\frac{2\pi st}{52}\right)\right], \quad t = 50, \ldots, 365,$$

where $(\beta_0, \beta_1, \beta_2, \beta_3)$ are seasonality parameters to be estimated.

Model 1

Linear predictor The most simple model that we fit to the data, inspired by Noufaily et al. (2013), is such that

$$\log(\mu_t) = \beta_0 + \beta_1 t + \text{seas}(t).$$

Inference and software We fit this model using the R package `MASS` (Venables and Ripley, 2002), with the likelihood-based inference method as explained in Section 1.2.4: the regression parameters on one side and the overdispersion parameter on the other side are iteratively fixed and estimated until convergence is obtained.

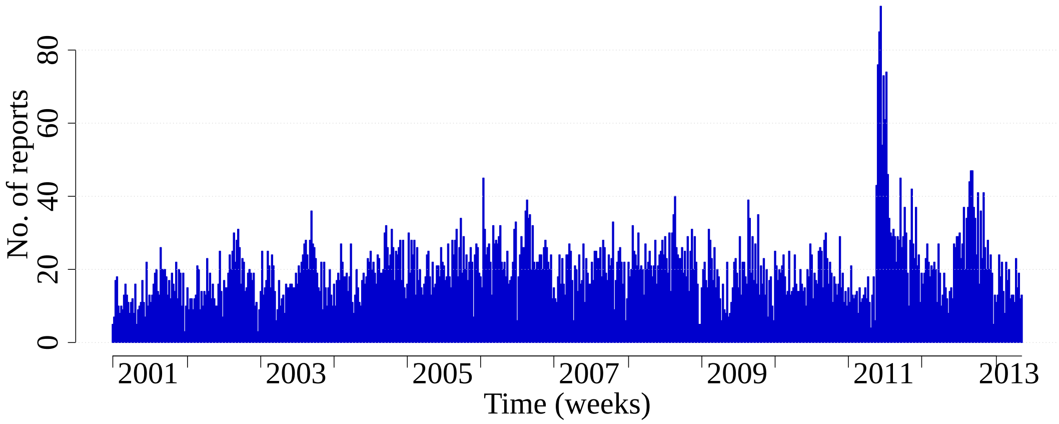


Figure 5.1: Weekly number of reported disease cases caused by Escherichia coli in the federal state of North Rhine-Westphalia (Germany) from January 2001 to May 2013, excluding cases of EHEC and HUS, as provided in the R package `tscount` (Liboschik et al., 2016). One notices the 2011 outbreak (Altmann et al., 2011; Bernard et al., 2014).

Models 2 to 4

Linear predictor We also fit three models as presented in Section 2.4,

$$\log(\mu_t) = \beta_{0,t} + \beta_1 t + \mathrm{seas}(t),$$

with three different specifications for the time-varying intercept $\beta_{0,t}$, see Section 2.4 for more details. The stationary model for the intercept gives a regression model that is similar to Model 1, while the neighbour and linear models account for autoregression on the previous counts (on top of seasonality).

Inference and software For this model, the inference is Bayesian. We use the INLA method for inference (Rue et al., 2009), as implemented in the R package `INLA` (Rue et al., 2015). We briefly presented INLA in Section 1.2.4. Priors are chosen as in Manitz and Höhle (2013) and in Salmon et al. (2015). This means, e.g., $\beta_i \sim \mathrm{N}(0, \lambda_{\beta_i}^{-1})$, $i = 1, 2$, where the λ_{β_i}'s indicate precision parameters.

Model 5

Linear predictor The next model we fit to the data has the following linear predictor (Liboschik et al., 2016):

$$\log(\mu_t) = \beta_0 + \beta_1 t + \mathrm{seas}(t) + \delta \log(y_{t-1}) + \alpha \log(\mu_{t-1})$$

that is, it contains log-transformed versions of both the previous observation y_{t-1} and the previous mean μ_{t-1}, hence it takes auto-correlation into account. In this predictor, the constraints are $|\delta| < 1$, $|\alpha| < 1$ and $|\delta + \alpha| < 1$.

Inference and software We fit this model using the R package `tscount`. The inference method is quite different from the one used for Model 1: it is a quasi-likelihood inference method introduced in Christou and Fokianos (2014). As in the inference for the quasi-Poisson model presented in Section 1.2.4, the regression parameters are estimated first, and after that the overdispersion parameter is estimated. Moreover, in this model the estimated overdispersion parameter is $1/\nu$.

Model 6

Linear predictor The last model we fit to the data contains an AR(1)-like regression on the previous observation. It is a two-component model introduced in Held et al. (2005) such that $\mu_t = \eta_t + \lambda y_{t-1}$ with

$$\log(\eta_t) = \beta_0 + \beta_1 t + \mathrm{seas}(t).$$

Inference and software This models does not fit in the GLM framework but is fitted using a generic optimization of the likelihood using quasi-Newton methods (Held et al., 2005). The method is implemented as the function `hhh4` in the R package `surveillance`.

The aim of the illustrative analysis in this Section is to look at how much residual auto-correlation there is after fitting these regression models, and if so, how the presence of residual auto-correlation leads to different (probabilistic) predictions of the counts in the weeks W1-2007 to W1-2009. Residuals are defined as $r_i = y_i - \hat{\mu}_i$ where $\hat{\mu}_i$ is the estimator of the mean in the corresponding model. For the sake of the comparison of Models 2 to 4 with the other models that are not Bayesian and for keeping things simple in this illustrative example, we define $\hat{\mu}_t$ as the mean of the posterior distribution of μ_t and used this point-estimate to compute residuals. We calculated the empirical auto-correlation function (ACF) of the residuals at different lags using the R function `acf()`. We chose 104 as a maximal lag, because it is equal to twice the period of the time series. We show the ACF plots of the residuals of the six models in Figure 5.2. One sees that Model 1 contains substantial residual autocorrelation at lag 1. This is also the case of Model 2 which has a constant intercept. The other models show nearly no serial dependence in the residuals.

We then concentrate on the model without auto-correlation fitted with `MASS`, *i.e. Model 1*, and fitted with `tscount`, *Model 7*, and on the model with auto-correlation on the past mean and on the past observation fitted with `tscount`, *Model 5*. Models 1 and 7 are thus models with the same linear predictor and likelihood but with different inference methods. We calculate point-predictions of the counts in the weeks W1-2007 to W1-2009, along with 0.95 quantiles of the predictive distribution with plug-in estimates of μ_t and ν, which are shown in Fig. 5.3. For assessing the predictive capabilities of the three models, we calculate

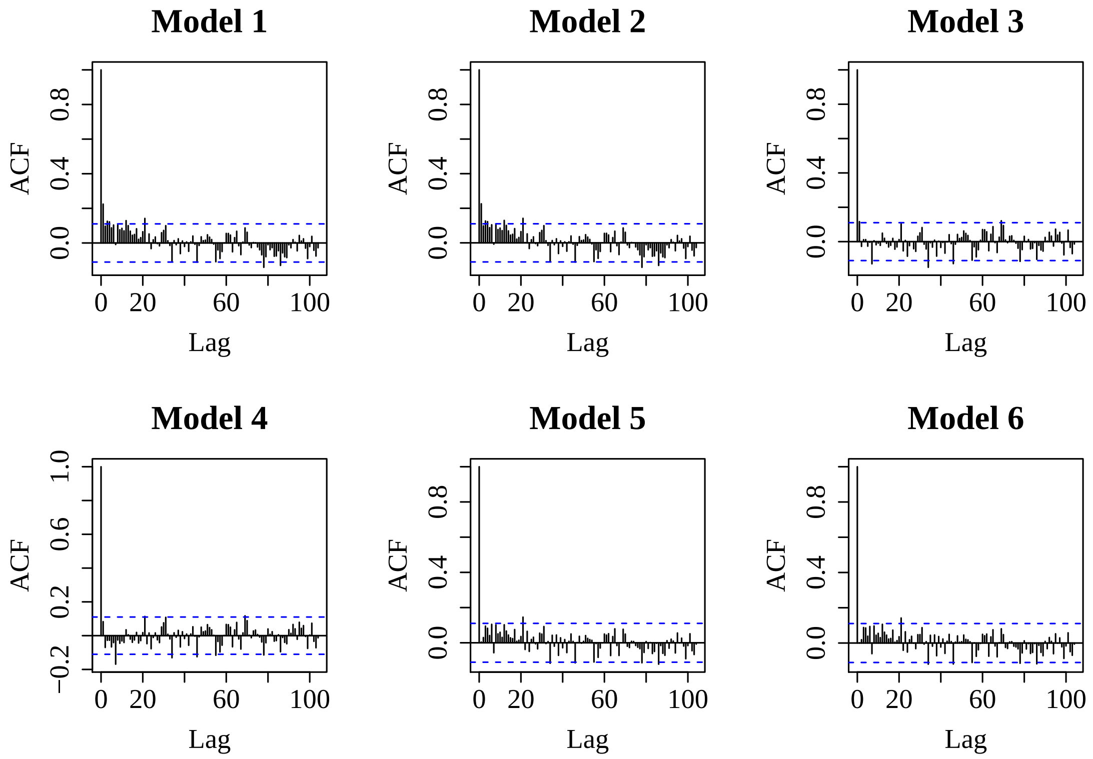


Figure 5.2: Autocorrelation plots of the residuals of the models explained in Section 5.1.1. Models 3 to 6 explicitely account for auto-regression on the previous observed count.

scoring rules as defined in Czado et al. (2009) and presented in Section 1.2.5 for these out-of-sample observed counts and performed permutation tests (Paul and Held, 2011) for each of them for comparing the three models. Here we use three scoring rules. Two of them are examples of two strictly proper scoring rules and one is an example of a proper scoring rules. Say f_t is the predictive distribution of y_t. The *logarithmic score* is (Czado et al., 2009)

$$\text{logs}(f_t, y_t) = -\log(f(y_t)).$$

The *ranked probability score* is (Czado et al., 2009)

$$\text{rps}(f_t, y_t) = \sum_{k=0}^{\infty} \left(f_t(y_t) - \mathbb{1}(y_t \leq k)\right)^2$$

where $\mathbb{1}$ is the indicator function. The RPS-score has an interpretation in terms of expectations if the expectations are assumed to be finite. Let us define Y_t and Y_t' as independent copies of a random variable with distribution f_t. The score is

$$\text{rps}(f_t, y_t) = E_f|Y_t - y_t| - 0.5E_f|Y_t - Y_t'|.$$

This formulation helps understanding the score: the smaller this difference is, the closer the realizations of Y_t and y_t expectedly are compared to realizations of Y_t and Y_t', which means a better calibration. The two scores we have just presented are strictly proper scoring rules. We also use a not-strictly proper scoring rule that is the *normalized squared error score* (Czado et al., 2009),

$$\text{nses}(f_t, y_t) = \left(\frac{y_t - \hat{\mu}_t}{\hat{\sigma}_t}\right)^2,$$

where $\hat{\mu}_t$ is the mean of f_t and $\hat{\sigma}_t$ its variance.

The value which one then actually uses for assessing a predictive distribution $f_{t,1}$ against another predictive distribution $f_{t,2}$ is a mean of the difference between scores over m out-of-sample observed values, *i.e.*

$$\frac{1}{m}\sum_{i=n+1}^{n+m} \text{rps}(f_{t,1}, y_i) - \text{rps}(f_{t,2}, y_i).$$

The results are shown in Table 5.1. The smaller the scoring rule, the better the predictive distribution. The table moreover provides p-values for the permutation tests performed as suggested in Paul and Held (2011). All three considered scoring rules give the same ranking of the three models, with small p-values: One observes that the model with auto-regression, Model 5, performs better than the two other ones, Models 1 and 7. Based on these results we hence conclude that Model 5 could be more appropriate for aberration detection – further simulation studies would be needed for a conclusive answer on that matter. One also sees that the two similar Models 1 and 7 have different scores, which could be because `tscount` makes inference on $1/\nu$ whereas `MASS` makes inference on ν. Actually in Aeberhard et al. (2014) ν is said to have a less numerically stable estimator. Another explanation could be that the quasi-inference method presented in Christou and Fokianos (2014) is better than the inference method presented in Lawless (1987) and Zeileis et al. (2008) and explained in Section 1.2.4. However, a better performance of one inference method vs. the other inference method would have to be checked on more than one time series. A futher difference between the two inference methods for this negative binomial regression model is computing time: fitting Model 7 was 16 times longer than fitting Model 1.

Based on the results we hence conclude that for time series like the one we analysed here, algorithms for aberration detection should be based on a regression model accounting for autocorrelation. Another possibility is that finding a better model for seasonality could make residual autocorrelation disappear in some cases. In any case, the performance of the different models as regards their predictive distributions should be compared. Moreoever, it would be of utmost importance to characterize the degree of autocorrelation in different representative sets of surveillance time series.

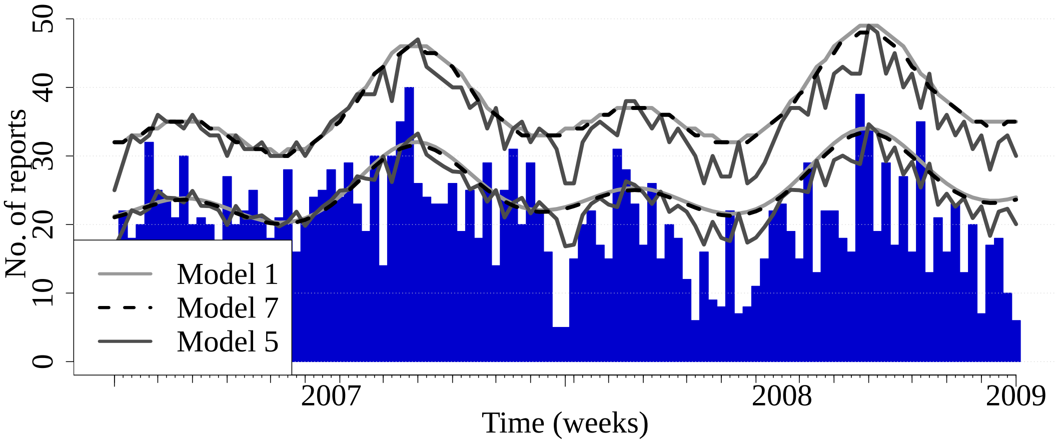


Figure 5.3: Weekly number of reported disease cases caused by Escherichia coli in the federal state of North Rhine-Westphalia (Germany) in the weeks W1-2007 to W1-2009 with point-predictions (bottom lines) and 0.95 quantiles (upper lines) provided by three models 1, 7 and 5.

In a study of mean-variance relationships in UK surveillance data, Enki et al. (2016) produced correlograms for all 1737 investigated time series after seasonality correction, and moreover applied Ljung-Box portmanteau tests (Ljung and Box, 1978) to them, using the p-value of this test as a measure of auto-correlation. Although the residuals of regression models for count time series are not normally or even symmetrically distributed, one can still use the Ljung-Box p-value as a measure of autocorrelation. In UK surveillance data, Enki et al. (2016) found that only a few pathogens display serial dependence after correction for time trend and seasonality, which according to them could be different in other surveillance systems, though.

If we were to look for residual autocorrelation – on top of seasonality – in surveillance data with the aim of taking it into account in regression models, we could not visually inspect every correlogram. Indeed, we would rather store *e.g.* the Ljung-Box p-values. We would need simulation studies or studies involving real data where we concurrently look at scoring rules of regression models with and without auto-correlation and at values of the Ljung-Box p-value of the residuals. This would help us define rules for automatically choosing the right regression model based not only on the values of the Ljung-Box p-value, but also on the autocorrelation for each lag for identifying which lag(s) to use in the regression. Or maybe a different model for seasonality might prove sufficient to suppress auto-correlation in the residuals, but the selection of a good form for seasonality is also an interesting issue for future research. Simulation studies might moreover indicate how much prediction is improved by proper treatment of autocorrelation and how the consequences on the predictive distribution translate into performance measures of aberration detection

	logS	RPS	NSES
Model 1 (NB GLM)	3.68	4.76	1.51
Model 5 (tscount, AR)	3.52	4.25	1.37
Model 7 (tscount, NO AR)	3.64	4.64	1.45
p-value 1 vs 5	0.0002	0.0002	0.017
p-value 1 vs 7	0.0002	0.0001	0.0001

Table 5.1: Scoring rules from the three models (logS: logarithmic score, RPS: ranked probability score, NSES: normalized squared error score), and p-values of permutation tests for each of these scoring rules.

such as probability of detection and false positive rate. Finally, it would show whether the computing time induced by choosing the most appropriate regression model for a time series is worth it.

5.1.2 Validity of the distributional assumptions

Most methods we have presented in this thesis, and most importantly the one we developped and explained in Chapter 3, make parametric assumptions about the distributions of the counts. In this Section, we aim at explaining how one could test the robustness of an algorithm when the distributional assumptions are violated, and how one could improve algorithms accordingly.

Negative binomial distribution of y_t

The number of reported cases in a week is modelled as a random variable with support on $0, 1, 2, \dots$. Since in actual data we often observe overdispersion when applying regression models, one needs to resort to regression models with families allowing overdispersion on top of the typical modelling of seasonality, time trend, and presence of past outbreaks as part of the expectations. The parametric family we chose in Chapter 3 is the negative binomial family. As a reminder, if $y_t \sim \text{NB}(\mu_t, \nu)$ then $\text{Var}(y_t) = \mu_t(1 + \mu_t/\nu)$. In Noufaily et al. (2013) the regression model is of the quasi-Poisson kind, *i.e.* they assume that $\text{Var}(y_t) = \phi\mu_t$. One could imagine many more relationships between the mean and the variance in the presence of overdispersion. If one defined an arbitrary relationship between the mean and the variance, one would not be able to use many existing inference methods and/or implementations. Therefore, there is interest in using the gold standard negative binomial distribution.

However, an important question is: how robust is the estimation of a quantile of the distribution of y_t if the regression model assumes a wrong mean-variance relationship? In Figure 5.4 we show, as an illustration, the true and estimated means as well as 0.95 quantiles of a negative binomial variable y_t defined such that

$$\log(\mu_t) = \log(10) + 0.25\cos\left(\frac{2\pi t}{52}\right) + 0.25\sin\left(\frac{2\pi t}{52}\right)$$

and $\nu = \mu_t/4$ which gives

$$\mathrm{Var}(y_t) = 5\mu_t.$$

For each simulation we fit a negative binomial model to the data, and for each timepoint t we deduce a plug-in estimate of the 0.95 quantile: the 0.95 quantile of the negative binomial distribution with *plug-in* estimates $\hat{\mu}_t$ and $\hat{\nu}$. A first result one can notice in Figure 5.4 is that the mean is quite well estimated by the model, but the quantile appears too high at the peak of the season and too low when they are the least cases. In this case, a quasi-Poisson regression would be more appropriate – which is obvious here given the way we simulated the time series.

In their analyse of UK surveillance data, Enki et al. (2013) started looking at relationships between the mean and the variance in the time series to be monitored. In Enki et al. (2016) they investigate them further and underline the importance of taking the real relationship of the variance and the mean into account when defining regression models for estimating quantiles. Such an analysis with surveillance data from a public health institute would be quite useful, in order to see how the relationship between the mean and the variance is in the monitored data, and whether it differs much from the assumptions implied by negative binomial distribution.

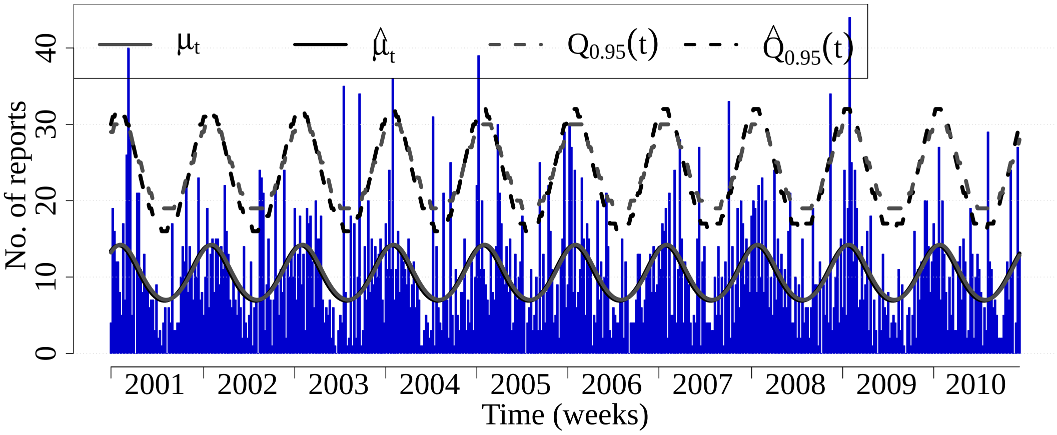


Figure 5.4: Values y_t of a simulated variable of mean μ_t including seasonality, and of variance $\mathrm{Var}(y_t) = 5\mu_t$. Solid lines show the true and estimated means μ_t and $\hat{\mu}_t$ while dotted lines show the true and estimated 0.95 quantiles.

A method for aberration detection without distributional assumptions

The work of Guillou et al. (2014) presents a method based on Extreme Value Theory (EVT) that does not require any assumption about the underlying distribution of the counts. The idea of their method is to associate a return period T_s to every possible observation y_s. This return period is defined such that

$$E\left(\sum_{i=1}^{T_s} I(y_t > y_s)\right) = c$$

where I is the indicator function and c a chosen level. When $c = 1$ one can interpret the T_s in the following way: we expect that on average, the value y_s will be exceeded every T_s time units. The choice of actual value of c, that influences sensitivity and specificity, is subject to current research by the authors of the method. Once return periods are defined, for every new observation y_s one looks backward to see if the same value of the number of cases is to be found in the time interval $(s - T_s, s)$. If this is the case, or if T_s is larger than the sample length, then an alarm is raised for s.

The method proposed in Guillou et al. (2014) uses data from the same periods in the previous years in order to account for seasonality, as it was done in Farrington et al. (1996). This means not using all reference data that one could use. Still, this is an interesting work, since no assumption about the underlying distribution of the counts is needed. However, it remains to be compared with existing methods: the authors underline that their article should be followed by studies comparing the monitoring performance of their algorithm with more established methods.

Assumptions about the delay distribution

In our work presented in Chapter 3, we assumed that the delays follow a multinomial distribution $M(N_t, \boldsymbol{p})$ with time-constant probability vector $\boldsymbol{p}$. This assumption coupled with the assumption that the number of cases follows a negative binomial distribution $\text{NB}(\mu_t, \nu)$ allowed us to use the so-called multinomial model for claim counts (Schmidt and Wünsche, 1998) and to estimate a quantile of the predictive distribution based on a negative binomial regression model. What if the delays follow a different distribution? In this case, our algorithm may not be as efficient as when analyzing simulated data that follow the assumptions we made. We explore two possible violations of the assumptions: the presence of overdispersion in the distribution of the delays, and a time-varying delay distribution instead of a constant one.

For illustrating the possible consequences of a violation of the distributional assumption regarding delays, we simulate vectors $(n_{t,0}, \dots, n_{t,D})$ in a situation where the number of cases N_t follows a negative binomial distribution μ, ν, that is with constant mean. We choose $\mu = 100$, $\nu = 2$ and $D = 4$. For modelling overdispersion in the distribution of delays we use a compound Dirichlet-multinomial distribution (Fahrmeir and Tutz, 2013), in which the delays follow a multinomial distribution with size N_t and probability vector $(p_0, p_1, p_2, p_3, p_4)$ and $(p_0, p_1, p_2, p_3, p_4)$ follows a Dirichlet distribution $\text{Dir}(2, 1, 1, 1, 1)$.

We draw 500 observations, *i.e.* $N_t \sim \text{NB}(\mu, \nu), t = 1, \dots, 500$. We then simulate the arrival of cases for each timepoint,

M1. with a multinomial distribution with constant probability vector $\frac{1}{6}(2, 1, 1, 1, 1)$;

M2. with a compound Dirichlet-multinomial distribution whose probability vector follows a Dirichlet distribution $\mathrm{Dir}(2, 1, 1, 1, 1)$ for simulating overdispersion in the delay distribution;

M3. with a multinomial distribution whose probability vector is $(1/3 - 1/6 \cdot \exp(-t/1000),$ $1/6 + 1/6 \cdot \exp(-t/1000), 1/6, 1/6, 1/6)$, $t = 1, \ldots, 500$, that is, a time-varying delay distribution. We show the corresponding cumulative distribution function of delay in Figure 5.5.

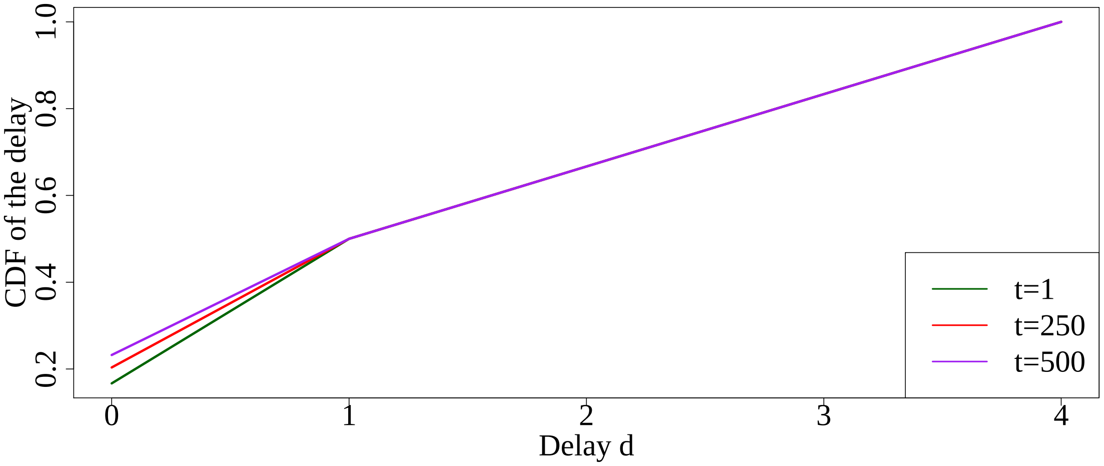


Figure 5.5: Delay distribution at three timepoints for M3.

We show an example of a resulting time series for $n_{t,0}$ in Figure 5.6. We then fit a negative binomial regression model to the data, using the R formula `response` $\sim$ `1 + delay`, similar to what we did in Chapter 3. For this we use the R package `MASS`.

```
R> library("MASS")
R> formulaGLM <- as.formula("response ~ 1 + as.factor(delay)")
R> model1 <- glm.nb(formulaGLM, dataGLM1,
R>                    control = list(epsilon = 1e-8, maxit = 200,
R>                                   trace = FALSE))
```

We show the estimates of the parameters in Table 5.2.

What we observe is that the overdispersion parameter of the negative binomial and the probability vector of multinomial distribution of M1 are quite well estimated, which is logical since in this case the data follow the distribution assumptions of the multinomial model for claim counts (Schmidt and Wünsche, 1998). With the data from M2, the parameters were all well estimated except ν which was under-estimated because of the overdispersion

Simulation	$\hat{\mu}$	$\hat{p}_0$	$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	$\hat{p}_4$	$\hat{\nu}$
True parameters	100	0.33	0.17	0.17	0.17	0.17	2.00
M1	101.2	0.24	0.19	0.19	0.19	0.19	2.03
M2	100.9	0.24	0.19	0.19	0.19	0.19	0.81
M3	98.8	0.20	0.23	0.19	0.19	0.19	2.05

Table 5.2: Output from the three models

in the delay distribution which increases the variance of the $n_{t,d}$ and which our regression model does not tackle. Worringly, a consequence of a too small ν value are overestimated quantiles of N_t. Therefore, the surveillance algorithm presented in Chapter 3 is not robust to the presence of overdispersion in the delay distribution, that would make it too conservative. In consequence, finding a suitable model for dealing with overdispersion in the delay distribution would be important. As regards the data from M3, in this case all parameters but the one related to the delay distribution are well estimated. Actually the structure of the linear predictor does not allow the estimation of a time-varying distribution, so the obtained estimates are a mean of the delay distribution over time. It is possible that real surveillance data contains a time-varying delay distribution for given organisms, especially if one uses a long baseline for computing a threshold. Looking for traces of time-variability in the distribution of delays, and modifying the linear predictor of the model so that it accounts for a time-varying delay distribution, should also be done in the future. If a non-linear evolution of the probability distribution over time were needed, eventually one would resort to GAMs instead of GLMs. One should note that this violation of the assumptions we made in Chapter 3 is less worrying as regards the so-called multinomial model for claim counts (Schmidt and Wünsche, 1998), since at each timepoint t one would still have $N_t \sim \text{NB}(\mu_t, \nu)$ and $(n_{t,0}, n_{t,1}, \ldots, n_{t,D}) \,|\, N_t, \boldsymbol{p}_t \sim \text{M}(N_t, \boldsymbol{p}_t)$ which is sufficient for having $n_{t,d} \sim \text{NB}(\mu_t \cdot p_{d,t}, \nu)$.

Note that in this Section we ignore other complications brought to the surveillance algorithm presented in Chapter 3 if the number of cases were not a negative binomial variable, if ν were not constant, or if the delays were not independent from the total number of cases. In this case, we would of course also no longer be able to use the so-called multinomial model for claim counts (Schmidt and Wünsche, 1998). Therefore, when applying the algorithm presented in 3 on many time series, checking its robustness would be an important first step, and complications of real time series may demand further developments of the algorithm in the future.

5.2 Issues related to the decision threshold in one-timepoint detection

In this section, we discuss two possible modifications of the computation of the decision threshold in one-timepoint detection: first we explain how the algorithms presented in

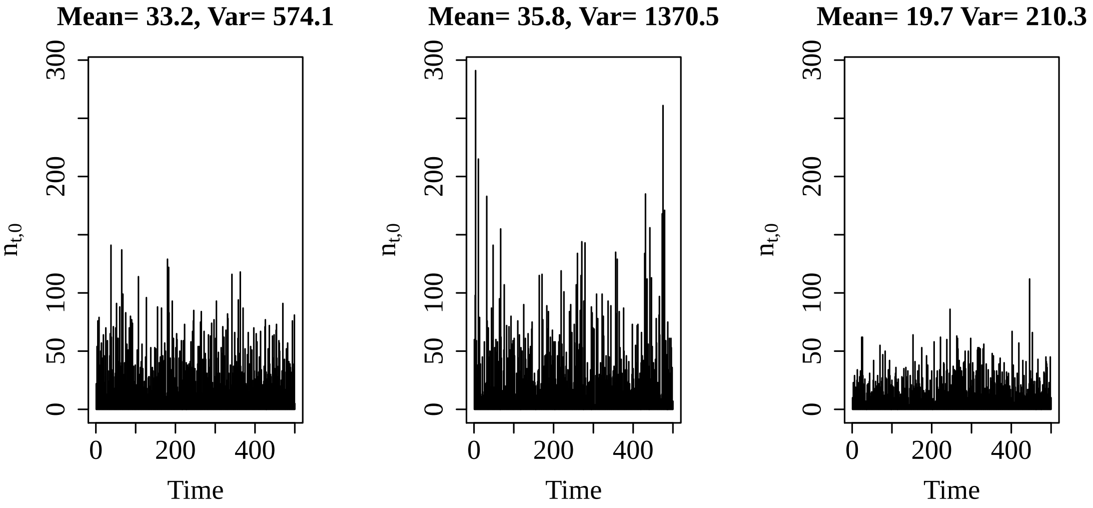


Figure 5.6: Weekly count of the simulated $n_{t,0}, 0 \leq t \leq 500$ corresponding, from left to right, to cases M1, M2 and M3 of the simulation presented in Section 5.1.2, with the empirical mean and variance of $n_{t,0}, 0 \leq t \leq 500$.

Section 2.4 and Chapter 3 could be made faster, and then in Section 5.2.2 we present a different definition of the threshold that may be more adapted to the count nature of surveillance time series than the current definition.

5.2.1 Faster computation of the quantile of a predictive posterior distribution

We are interested in finding the $(1-\alpha)$-quantile of the predictive posterior distribution of $N(s,T)$, for using it as a decision-threshold. We start by explaining the method presented in Section 3.3, in which the threshold is computed by a Monte Carlo sampling method involving many draws. Let us write that

$$\begin{aligned} f(N(s,T) \mid \boldsymbol{n}_{O_s}) &= \int f(N(s,T) \mid \boldsymbol{\psi}) f(\boldsymbol{\psi} \mid \boldsymbol{n}_{O_s}) d\boldsymbol{\psi} \\ &\approx \frac{1}{R} \sum_{r=1}^{R} f(N(s,T) \mid \boldsymbol{\psi}^{(r)}), \boldsymbol{\psi}^{(r)} \sim f(\boldsymbol{\psi} \mid \boldsymbol{n}_{O_s}), \end{aligned}$$

with R being the number of Monte Carlo samples. First, we sample R vectors $\boldsymbol{\psi}^{(r)}, 1 \leq r \leq R$ from $f(\boldsymbol{\psi} \,|\, \boldsymbol{n}_{O_s})$ and for each of them, R' values from $f(N(s,T), \boldsymbol{\psi}^{(r)} \,|\, \boldsymbol{n}_{O_s})$. We then use the empirical $(1-\alpha)$-quantile of the sample of $R \cdot R'$ values as a decision-threshold.

Actually, we could do with less computations using a different method that we would like to present in this Section. The cumulative distribution function (CDF) of the predictive posterior distribution is

$$\begin{aligned}F_{N(s,T)}(y|\boldsymbol{n}_{O_s}) &= P(N(s,T) \leq y|\boldsymbol{n}_{O_s}) = \sum_{x=0}^{y} f(N(s,T)=x|\boldsymbol{n}_{O_s})\\ &\approx \sum_{x=0}^{y} \frac{1}{R}\sum_{r=1}^{R} f(N(s,T)=x|\boldsymbol{\psi}^{(r)}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{x=0}^{y} f(N(s,T)=x|\boldsymbol{\psi}^{(r)})\\ &= \frac{1}{R}\sum_{r=1}^{R} F_{N(s,T)}(y|\boldsymbol{\psi}^{(r)}),\end{aligned}$$

where $F_{N(s,T)}(y|\boldsymbol{\psi}^{(r)})$ is the CDF of the Negative Binomial response distribution with expectation and dispersion parameter calculated from $\boldsymbol{\psi}^{(r)}$. The $(1-\alpha)$-quantile is the smallest y value such that

$$F_{N(s,T)}(y|\boldsymbol{n}_{O_s}) \geq 1-\alpha.$$

Now, for any value y it is quite straightforward to calculate $F_{N(s,T)}(y|\boldsymbol{n}_{O_s})$. In order to efficiently find the quantile based on the CDF from the mixture, one can use a bisectioning method. We implemented this new approach for the two algorithms `bodaDelay` presented in Chapter 3 and `boda` presented in Section 2.4. In the two functions `bodaDelay` and `boda`, if one chooses the option`MM` (for Mixture Model) for the `control` argument `quantileMethod`, the threshold is computed as the $(1-\alpha)$-quantile of the mixture distribution obtained by sampling R vectors from the (joint) posterior distribution of the parameters. More precisely, the quantile is searched by bisectionning, with the most possible extreme values as initial brackets:

- the lower initial bracket is the quantile obtained for the smallest sampled mean, $\min(\mu_t^{(r)}), r \in \{1,\ldots,R\}$ and the highest sampled overdispersion parameter, $\max(\nu^{(r)})$, $r \in \{1,\ldots,R\}$, and
- the upper initial bracket is the quantile obtained for the highest sampled mean, $\max(\mu_t^{(r)}), r \in \{1,\ldots,R\}$ and the smallest sampled overdispersion parameter, $\min(\nu^{(r)})$, $r \in \{1,\ldots,R\}$.

Below we exemplify the use of the `bodaDelay` algorithm with the two different methods for computing the quantile, that is, the one presented in Chapter 3 and Salmon et al. (2015), and the one presented in this Section.

```
# Control slot with D=0 correction and Monte Carlo sampling
R> controlMC <- list(range = 410:412, b = 4, w = 3,
R>                   mc.munu = 1000, mc.y = 1000,
R>                   quantileMethod = "MC",
R>                   alpha = 0.05, trend = TRUE,
R>                   limit54 = c(0,50),
R>                        noPeriods = 10, pastWeeksNotIncluded = 26,
R>                        delay = FALSE,inferenceMethod = "asym")
# Control slot with D = 10 correction and Monte Carlo sampling
R> controlDelayMC <-  modifyList(controlMC, list(delay = TRUE),
R>                                keep.null=TRUE)
# Control slot with D=0 correction and mixture model
R> controlMM <-  modifyList(controlMC, list(quantileMethod = "MM"),
R>                                keep.null = TRUE)
# Control slot with D=10 correction and mixture model
R> controlDelayMM <-  modifyList(controlMM, list(delay = TRUE),
R>                                keep.null = TRUE)
# Calculations
R> salmMC <- bodaDelay(salmAllOnset, controlMC)
R> salm.delay.MC <- bodaDelay(salmAllOnset, controlDelayMC)
R> salmMM <- bodaDelay(salmAllOnset, controlMM)
R> salm.delay.MM <- bodaDelay(salmAllOnset, controlDelayMM)
```

So in each case, mc.munu=1000 samples are drawn from the joint posterior distribution of the parameters. When quantileMethod is MC, mc.y=1000 response values are drawn from the corresponding 1000 distributions and the threshold is the empirical $(1-\alpha)$-quantile of the obtained sample of 10^6 values. When quantileMethod is MM, the quantile is the $(1-\alpha)$-quantile of the mixture distribution, computed using a bisectionning method. The values obtained are very similar, see below.

```
R> t(salmMC@upperbound)
          [,1] [,2] [,3]
observed1 1104 1016  963
R> t(salmMM@upperbound)
          [,1] [,2] [,3]
observed1 1104 1016  963
R> t(salm.delay.MC@upperbound)
          [,1] [,2] [,3]
observed1 1345 1274 1226
R> t(salm.delay.MM@upperbound)
          [,1] [,2] [,3]
observed1 1346 1274 1225
```

The corresponding computing times were 3.36 and 0.48 seconds (without delay correction) and 4.35 and 1.62 seconds (with delay correction), respectively, for the original and the new method for computing the quantile, which is a considerable time gain. The algorithm for the method presented in Sect. 2.4 has been extended to perform quantile inference by this method. This smarter computation scheme makes both algorithms faster, although using INLA for inference still slow them down a lot. Maybe further work on the algorithms could help make inference faster without loosing precision.

5.2.2 A quantile based on mid-p-value as threshold?

The methods presented in this thesis are tailored at the discrete count nature of the surveillance data and use a quantile based on $P(y_s > x)$ to compute the decision-treshold. The quantile we use, x, is such that x is the smallest value for which $P(y_s > x) < \alpha$, or, if we were to formulate the problem using the cumulative distribution function F, the smallest value for which $F(x) \geq 1 - \alpha$. This search also corresponds to a one-sided test, where H_0 is "y_s follows the estimated predictive distribution", and where the type I error should be α: if y_s follows the estimated predictive distribution, the probability of its being higher than x is less than α. Therefore, one can see a parallel between $P(y_s > x)$ and a p-value. Of course, with discrete variables one cannot exactly control the type I error rate α because $P(y_t > x) = \alpha$ often does not have an exact solution x.

A problem one encounters when using $P(y_t > x)$ for detecting aberrations in discrete data is that this probability does not have a uniform distribution under H_0: Berry and Armitage (1995) recommend to use mid-p-values instead, because under H_0 the mid-p-values have a distribution closer to an uniform distribution $U(0, 1)$, which the authors deem as desirable for any significance test. This is also advised by Ma et al. (2011) and Spiegelhalter et al. (2012). The mid-p-value is defined by

$$\text{mid-p}(x) = P(y_s < x)^{\mathbb{1}(x>0)} + \frac{1}{2} P(y_s = x),$$

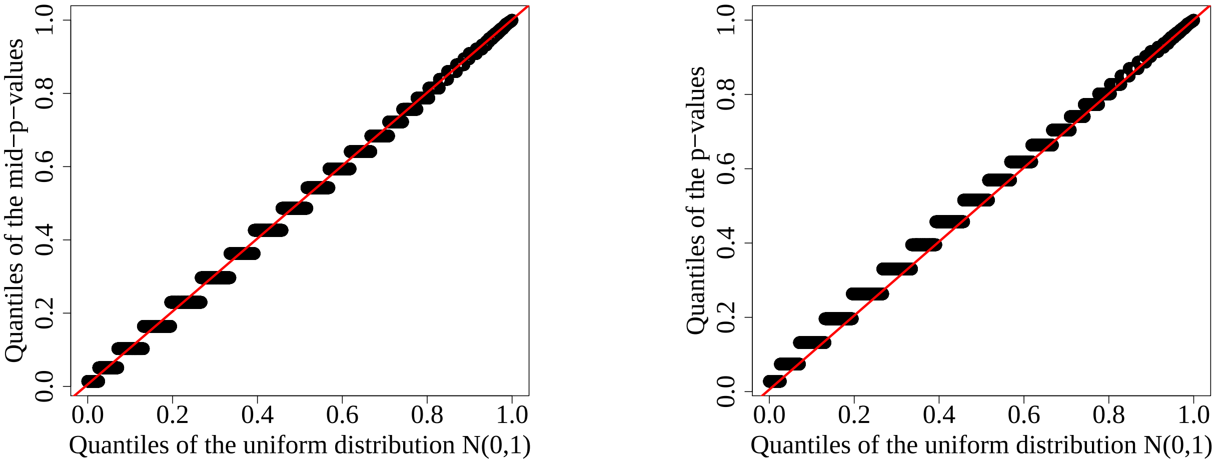
where $\mathbb{1}_{x>0}$ is the indicator variable, equal to 1 if $x > 0$.

For illustrating purposes we drew 10^5 values from the distribution $\text{NB}(\mu, \nu)$ with $\mu = 10$ and $\nu = 2$. We show the corresponding quantile-quantile plots in Figure 5.7 where empirical quantiles from the distribution of the (mid)-p-values of the sample are plotted against quantiles from the uniform distribution $U(0, 1)$. It shows that the mid-p-values have a distribution closer to the uniform distribution than the p-values, although neither of them can have an uniform distribution since we are dealing with count data. This departure seems to depend on the value x: the higher the values of x, the smaller the departure from the uniform distribution.

Probability	0.95	0.99	0.995
p-value-based quantile	25	35	40
Mid-p-value-based quantile	26	36	40

Table 5.3: Quantiles obtained in the simulations using the two computation methods.

For these distributions, the 0.95, 0.99 and 0.995 quantiles based on the p-value and on the mid-p-value are shown in Table 5.3. They are not very different but one should however define upperbounds as quantiles based on mid-p-values in the future, sometimes called *mid-quantiles*. Finding efficient implementations of the search of such quantiles based on the



(a)

(b)

Figure 5.7: Quantile-quantile plots of mid-p-values (a) and of p-values (b) of a sample of 10^5 values from the distribution NB(μ, ν) with $\mu = 10$ and $\nu = 2$ against quantiles from the uniform distribution $N(0, 1)$.

parameters of the negative binomial distribution for the method presented in Section 2.3, or on a sample *e.g.* for the method presented in Section 3.3, would be very important for allowing use of mid-quantiles in aberration detection. The R package `Qtools` (Geraci, 2016) contains functions for calculating mid-cumulative probabilities and mid-quantiles, based on the work of Parzen (2004) and Ma et al. (2011). Furthermore, Jentsch and Leucht (2015) look at the properties of bootstrapping for constructing confidence interval of sample mid-quantiles of discrete data. Therefore mid-quantiles are interesting items to investigate in potential future work involving the computation of the quantile of a discrete distribution as a decision-threshold in count time series monitoring.

5.3 Evaluation of routine applications

After having presented an outlook on theoretical topics of aberration detection for count time series, this Section presents two issues related to routine application of detection algorithms in a public health institute with many time series to monitor, *e.g.* as motivated by our experience at the Robert Koch Institute. While being problems inspired by routine application, they still have a theoretical interest.

5.3.1 Does signal mean alarm?

When evaluating an algorithm as we did in Section 3.4.1, we defined alarms for time-points for which the observed number of counts was higher than the $(1 - \alpha)$-quantile of the predictive posterior distribution of the count. This definition did not take into account the behaviour of epidemiologists when confronted with a weekly surveillance report: what we need for evaluating routine application is a more realistic representation of the actual epidemiological workflow. For an ongoing research project at the RKI on the timeliness of Listeria surveillance in Germany, we started defining what an actual *alarm* is. We call a *signal* the fact that the observed number of counts is higher than the $(1-\alpha)$-quantile of the predictive posterior distribution of the count, and *alarm* a sequence of signals that triggers actions such as the epidemiologist contacting a federal state health authority in order to investigate further. Our motivation is that evaluating sensitivity, specificity and timeliness based on signals instead of alarms might be erroneous. Indeed, not every *signal* is an *alarm*. For instance, a signal related to very few cases might trigger a "wait and see" behaviour of the epidemiologist rather than actual investigation measures. Therefore, we decided to define *signals* and *alarms* while trying to mimic the epidemiologists behaviour.

Epidemiologists act according to the size of the signal and to their experience. An ideal study of an algorithm would be to confront experts with reports and ask them wich signals they consider as alarms. Using such subjective opinion is a quite rare method of evaluation of a system (Drewe et al., 2012) and would cost large human resources for evaluating signals in a big simulation study. Therefore, we need to imitate epidemiologists rather than to have them review signals. Moreover, if one wants to evaluate an algorithm concerning a given pathogen, one cannot use the criteria of decision used by epidemiologists for signals regarding other pathogens, because they might be different. Thus, our definition of alarms had to be validated by epidemiologists experienced in Listeria surveillance at the RKI.

We decided to define two types of signals, based on their size. *Medium signals* are signals associated to the $(1-\alpha)$-quantile but not to the $(1-\alpha')$-quantile with $\alpha' < \alpha$, whereas *high signals* are associated to the the $(1-\alpha')$-quantile. For instance one could choose $\alpha = 0.05$ and $\alpha' = 0.01$. Please note that we use the notation of Section 3.1: $N(t,T)$ is the number of cases diagnosed at time t and known (reported) by time T.

Each week, when receiving a surveillance report from the RKI surveillance system presented in Chapter 4, epidemiologists only see the signals corresponding to the same observation timepoint T. However, they might remember signals from previous weeks that increased their awareness. Moreover, as seen in Figure 4.2, if there was a signal for $N(t,T_1)$ and then again for $N(t,T_2)$ with $T_2 > T_1$, then the cell is coloured orange instead of red in the weekly report so that epidemiologists know that week t was already flagged for a previous observation timepoint.

We define alarms as any of the following sequences:

D1. a high signal,

D2. at least three medium signals for observed counts having the same observation timepoint and subsequent weeks of disease onset: $N(t,T)$, $N(t+1,T)$ and $N(t+2,T)$,

D3. at least two medium signals for observed counts having the same observation timepoint and subsequent weeks of disease onset when for one of these weeks of disease onset there was a medium signal before, *e.g*, a medium signal for $N(t_1, T_1)$ and then medium signals for $N(t_1, T_1 + 1)$ and $N(t_1 + 1, T_1 + 1)$.

These definitions were validated by epidemiologists used to review Listeria weekly surveillance reports at the RKI. Definition D2 is supported by the fact that epidemiologists report looking at subsequent weeks to notice patterns. Definition D3 was created because a signal may make the epidemiologist be more alert when receiving the reports in the subsequent weeks.

The next step of the project will be to evaluate algorithms using this very heuristic definition of signals and alarms, but also using signals as alarms, in order to see how much our defining more complicated sequences of signals as alarms would change the measures of performance of algorithms. We look forward to the first results of this work.

5.3.2 Multiple testing

During generation of weekly reports, many time series are monitored simultaneously, which constitutes a multiple testing problem. Moreover, some of these time series are time series for subgroups of other time series, *e.g.* time series for different age groups of patients. Because time series analysed correspond to different aggregation of several characteristics such as age, sex, location, their hierarchical structure is quite complicated. Taking this structure into account for controlling *e.g.* the false positive rate (Benjamini and Hochberg, 1995) should be a further development and may help decreasing the number of false alarms than can lead to user fatigue. Such an improvement would increase the users satisfaction with the system and would make the detection and investigation of real outbreaks more likely.

In Besag and Newell (1991) the authors present a detection method for cluster of diseases where no adjustment is made for multiple testing because ”this might exclude clusters that are epidemiologically important.” This is a valid argument but in the case of a routine monitoring system the amount of signals that can be checked by epidemiologists is limited.

Another improvement to signal presentation could be to output signals not only as binary variables but to also associate each of them with a (corrected) predictive probability for the observed count, and even to sort signals accordingly.

5.4 Conclusion

In this Chapter, we have mostly listed ideas for future research in count time series monitoring and its applications. We have presented the limitations of most current models used in aberration detection, due to ignoring residual auto-correlation in the time series, and to distributional assumptions regarding the counts and their right-truncation. We have also commented on the definition and computation of the threshold in the case of single-timepoint detection. Regarding the practical applications of algorithms for aberration detection, we have introduced a possible way of mimicking real behaviour for the evaluation of algorithms, and underlined the need for considering multiple testing. All these issues constitute possible directions for future research beyond the work presented in this thesis. In the next Chapter, we conclude this thesis.

Chapter 6

Conclusion

After presenting possible future and current challenges for statistical aberration detection in Chapter 5, this Chapter concludes the thesis: we aimed at improving statistical aberration detection for routine surveillance of infectious diseases, both from a methodological point-of-view and as concerns their application in practice, in particular at the RKI, where this work was conducted. This goal led to a main methodological development for aberration detection and to the design and development of a new automatic surveillance system at the RKI. Contributions were made to open-source software and the corresponding documentation to enable others to use the methods. These three aspects, developping better methods for routine surveillance, supporting their application and making their implementations widely available, are intertwined and necessary for improving aberration detection. In this conclusion, we summarize the results of the thesis for each of these aspects and give future possible directions for research and recommendations.

6.1 Improvement of statistical aberration algorithms

In this Section, we state the methodological results brought by this thesis, including a novel method for outbreak detection and the discussion of existing methods, and discuss future theoretical challenges for aberration detection in count time series.

6.1.1 Summary of methodological results

In Chapter 1 we introduced the statistical and practical context of the work and presented the general framework of count time series. This included a presentation of the Poisson and negative binomial distributions that we used throughout the thesis, and of suitable regression models, i.e. GLMs and GAMs. We also briefly explained how to make inference for such models and how to assess the goodness of fit of such models. Throughout the whole thesis, we focused on the use of aberration detection algorithms based on Generalized Linear Models (GLMs) or Generalized Additive Models (GAMs) as flexible tools that can take into account the (overdispersed) count distribution of surveillance count time series, their characteristics such as seasonality, time trend and possible dependence on concurrent covariate processes.

Chapter 2 presented the statistical framework for outbreak detection and chosen algorithms for outbreak detection that are, or could be, used in routine surveillance. The presented methods included a frequentist method based on a GLM, a Bayesian algorithm based on a GAM including a time-varying intercept acounting for autocorrelation in the time series and using the INLA approximations for fast inference, and a method for monitoring of more than one timepoint. We therefore presented a representative set of statistical algorithms for outbreak detection, with different sources of inspiration: traditional surveillance methods, Bayesian hierarchical time series models and statistical process control.

In Chapter 3 we presented the main methodological statistical development of this thesis. The chapter dealt with the problem of right-truncation of the reporting data: it offered a short review of existing methods for nowcasting and outbreak detection in the presence of reporting delays, and presented a novel Bayesian algorithm that was tested in simulation studies and applied to the time series of weekly counts of *Salmonella* Newport cases in Germany. The method is currently being tested for routine application at the Robert Koch Institute. The Chapter was an enriched version of the article ***M. Salmon**, D. Schumacher, K. Stark, M. Höhle. Bayesian Outbreak Detection in the Presence of Reporting Delays, Biometrical Journal, 57 (6), 1051-1067, 2015.* Compared with the article, we made the proof of a lemma that is central for our algorithm more explicit and we presented the implementation of the algorithm in R. Furthermore, we also presented an alternative approach to the problem of right-truncation of the data that was recently published in Noufaily et al. (2015) but whose implementation is not available.

6.1.2 Remaining methodological challenges

Chapter 5 contained an outlook of future methodological developments for aberration detection. The first issues presented were related to the regression models used, both regarding residual auto-correlation and regarding the validity of distributional assumptions. Improving existing detection algorithms with regard to these challenges could improve their monitoring performance. However every improvement needs to be evaluated and tested in order to see whether it really brings an increase of performance, and at which computational cost. As a further topic we considered switching the definition of the threshold for single timepoint detection, from a quantile of the predictive distribution corresponding to a p-value based quantile, to a quantile based on the mid-p-value. Such mid-quantiles might be more appropriate for count time series. Furthermore, we introduced a change of the sampling method for finding a quantile of the posterior predictive distribution for the two Bayesian detection algorithms presented in Sections 2.4 and 3.3. This could lead to higher speed of calculation, which is an important factor if the method is to be applied in routine surveillance systems in which many time series have to be monitored simultaneously. Another important subject of research that we did not mention would be the areas of space-time and multivariate surveillance, that can at least partly be seen as extensions of methods presented in this thesis.

An important aspect we want to underline in this conclusion is the interdisciplinary aspect of methodological issues encountered in statistical aberration detection in count time series of infectious diseases surveillance. Our work was motivated by routine surveillance data of infectious diseases in Germany, but could be useful in many other contexts as well: properties such as overdispersion, low counts, presence of past outbreaks, apply to a wide range of count and categorical time series in other surveillance contexts such as financial surveillance (Frisén, 2008), occupational safety monitoring (Schuh et al., 2014) or environmental surveillance (Luo et al., 2012). Furthermore, work such as Höhle and Paul (2008) was an effort at conducing a synthesis of traditional surveillance methods using GLMs and statistical process control. Moreover, our novel algorithm presented in Chapter 3 builds on both the aberration detection algorithm of Manitz and Höhle (2013), on the *nowcasting* method of Höhle and an der Heiden (2014), and on the work of Schmidt (2006) in insurance mathematics, thus offering a synthesis of two fields, statistical aberration detection and loss reserving. Altogether, this means that any development in one of these fields could be useful in any of the other fields.

However, in our experience such a mutual enrichment is slowed down by partial ignorance of others' work, which is exemplified by the review of Weiß and Lu (2015) from the SPC field, that disregards recent work such as Liboschik et al. (2014) and Höhle and Paul (2008). Another obstacle to improvement of methods is the use of different notation and terminology. The *run-off triangle* in Schmidt and Wünsche (1998) describes a process similar to the *reporting triangle* in Lawless (1994); Höhle and an der Heiden (2014), but it has a different name and a partially different representation of right-truncation in the data. Our recommendation to researchers of any field where count time series with possible right-truncation are monitored is to strive to be on top of the literature in other fields, and to cite articles from, say loss reserving literature, when writing an outbreak detection article, thus making links between methodological developments easier to detect.

Breaking a practical problem or process down into components such as compound distributions, regression models, etc. beyond the use of field-specific technical terms is actually what is to be expected of a statistician, and can catalyse the development of modern methods for aberration detection.

6.2 Routine aberration detection

In this Section, we discuss our contribution to the improvement of the German surveillance system of infectious diseases.

6.2.1 Summary of the practical results

During this work, we participated to the development and use of the system for routine monitoring of infectious diseases at the RKI, for which we provided statistical counseling. Chapter 4 presented the routine surveillance application at the RKI, along with the challenges encountered. We explained the design strategy and decisions behind the implementation of the system, its structure and its acceptance by the users. The work is explained in the article ***M. Salmon***, *D. Schumacher, H. Burmann, C. Frank, H. Claus, M. Höhle. A system for automated outbreak detection of communicable diseases in Germany, Eurosurveillance, accepted for publication.* The system was developped together with the computer scientists and epidemiologists at the RKI. The inclusion of end-users – epidemiologists – in the design and development of the surveillance system is in our opinion the best way to take into account important practical factors such as workload, user fatigue and to weigh these issues against a systematic insurance for not missing outbreaks.

In Section 4.4 we moreover provided guidance for implementing a simpler surveillance system supported by the `surveillance`. This shows that setting up automatical aberration detection does not need much infrastructure and thus can be applied in a variety of contexts. Even such a such simple system would be a significant improvement compared to not analysing the data at all, or having to do large parts of the analysis by hand.

6.2.2 Recommendations and future research

We also presented ongoing extensions of the RKI automatic surveillance system in Chapter 4: the ongoing test of the algorithm presented in Chapter 3 and thoughts about the possible parallel use of several algorithms for aberration detection in the same system. In our opinion, any surveillance system in a public health institute needs continuous input of statisticians, even after the initial set-up, to ensure the presence of a contact person for statistical questions asked by the users, and the possibility to extend the system using the latest algorithmic developments. The current renewal of the German surveillance system through the project DEMIS (German electronic reporting system for infectious diseases) will bring further opportunities to make methodological improvement to the existing automatic surveillance system.

In Section 5.3.1, we discussed a new scheme for evaluating an algorithm that is more adapted to its use in a real surveillance system by differentiating *signals* and *alarms* for tentatively mimicking epidemiologists' behaviour. A better evaluation of the routine automatic surveillance system of infectious diseases in Germany would produce interesting epidemiological results for the country and similar systems in other countries. Such an evaluation would first need the definition of adapted performance measures. Section 5.3.1 was a first effort in this regard. In Section 5.3.2 we furthermore presented considerations regarding multiple testing, which is an important issue in real surveillance systems monitoring many time series at once, and which would deserve further research.

6.3 Open-source implementation of algorithms

In this Section, we reflect on the open-source contributions of this work: we proposed methodological developments and provided the corresponding implementations in order to facilitate both their comparison with existing methods and their routine use. During this work, we added algorithms to the R package `surveillance`, including the methodological advance presented in Chapter 3. Others algorithms added to the package during this work are `farringtonFlexible` presented in Section 2.3, `earsC` presented in Sections 2.1 and 2.5.1, and `bodaDelay` presented in Section 3.5.

All functionality of the package comes only at the cost of learning how to use the R statistical programming language. To support application by others, Chapter 2 was an enhanced version of the article ***M. Salmon****, D. Schumacher, M. Höhle. Monitoring Count Time Series in R: Aberration Detection in Public Health Surveillance, Journal of Statistical Software, accepted for publication.* Here, we explained the use of the R package `surveillance`.

Our work in favour of open-source software for aberration detection will hopefully also serve future methodological developments for aberration detection, since any new method can easily be compared to the ones we implemented, and since researchers could re-use code instead of re-inventing the wheel. At the same time, our implementation efforts will support the routine use of state-of-the-art method such as the ones presented in Section 2.4 – whose implementation we improved, and Chapter 3 – that we created and implemented. In contrast, the method of Noufaily et al. (2015) presented in Section 3.6 was published without available implementation which in our opinion is a strong obstacle to its use and benchmarking.

Overall, in this thesis, we have made methodological improvements of statistical algorithms for aberration detection in count time series, taking estimation and observation uncertainty into account. Our work also provided a well documented and open-source implementation of such methods. Moreover, we offered statistical counseling for the development and use of a routine surveillance system. Our work allowed the enormous quantities of data collected in the German surveillance system for infectious diseases to get even more informational and managerial value: the output from monitoring algorithms increases awareness and alertness of epidemiologists. Therefore, we would like to transform the quote *"Let my dataset change your mindset"* of the Swedish statistician Hans Rosling[1] into *"Let my toolset change your mindset about your dataset".*

[1] Hans Rosling's TED talk "Let my dataset change your mindset", 2009, https://www.ted.com/talks/hans_rosling_at_state/transcript.

Bibliography

Abat, C., H. Chaudet, P. Colson, J.-M. Rolain, and D. Raoult (2015). Real-Time Microbiology Laboratory Surveillance System to Detect Abnormal Events and Emerging Infections, Marseille, France. *Emerging infectious diseases 21*(8), 1302–1310.

Aeberhard, W. H., E. Cantoni, and S. Heritier (2014). Robust inference in the negative binomial regression model with an application to falls data. *Biometrics 70*(4), 920–931.

Altmann, M., M. Wadl, D. Altmann, J. Benzler, T. Eckmanns, G. Krause, A. Spode, and M. an der Heiden (2011). Timeliness of Surveillance during Outbreak of Shiga Toxin-producing Escherichia coli Infection, Germany, 2011. *Emerging Infectious Diseases 17*(10), 1906–1909.

Anscombe, F. J. (1953). Contribution to the discussion of H. Hotelling's paper. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 15*, 229–230.

Bayer, C., H. Bernard, R. Prager, W. Rabsch, P. Hiller, B. Malorny, B. Pfefferkorn, C. Frank, A. de Jong, I. Friesema, B. Stark, and B. Rosner (2014). An outbreak of *Salmonella* Newport associated with mung bean sprouts in Germany and the Netherlands, October to November 2011. *Eurosurveillance 19*(1), 1–9.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 57*(1), 289–300.

Berger, J. O. (2013). *Statistical decision theory and Bayesian analysis.* Springer Science & Business Media.

Bernard, H., D. Werber, and M. Höhle (2014). Estimating the under-reporting of norovirus illness in Germany utilizing enhanced awareness of diarrhoea during a large outbreak of Shiga toxin-producing E. coli O104:H4 in 2011. *BMC Infectious Diseases 14*(1), 1–6.

Berry, G. and P. Armitage (1995). Mid-p confidence intervals: a brief review. *Statistician 44*, 417–423.

Besag, J. and J. Newell (1991). The detection of clusters in rare diseases. *Journal of the Royal Statistical Society. Series A (Statistics in Society) 154*(1), 143–155.

Bhattacharya, A. and S. Wilson (2015). Sequential Bayesian inference for dynamic state space model parameters. In S. K. Upadhyay, U. Singh, D. K. Dey, and A. Loganathan (Eds.), *Current Trends in Bayesian Methodology with Applications*, Chapter 6, pp. 123–133. Chapman and Hall/CRC.

Brookmeyer, R. and A. Damiano (1989). Statistical methods for short-term projections of AIDS incidence. *Statistics in Medicine 8*(1), 23–34.

Buckeridge, D. L., H. Burkom, M. Campbell, W. R. Hogan, and A. W. Moore (2005). Algorithms for rapid outbreak detection: a research synthesis. *Journal of Biomedical Informatics 38*(2), 99 – 113.

Buckeridge, D. L., A. Okhmatovskaia, S. Tu, M. O'Connor, C. Nyulas, and M. A. Musen (2008). Understanding detection performance in public health surveillance: modeling aberrancy-detection algorithms. *Journal of the American Medical Informatics Association 15*(6), 760–769.

Cakici, B., K. Hebing, M. Grünewald, P. Saretok, and A. Hulth (2010). CASE: a framework for computer supported outbreak detection. *BMC medical informatics and decision making 10*(1), 14.

Christou, V. and K. Fokianos (2014). Quasi-likelihood inference for negative binomial time series models. *Journal of Time Series Analysis 35*(1), 55–78.

Christou, V. and K. Fokianos (2015). On count time series prediction. *Journal of Statistical Computation and Simulation 85*(2), 357–373.

Conesa, D., M. Martínez-Beneito, R. Amorós, and A. López-Quílez (2015). Bayesian hierarchical Poisson models with a hidden Markov structure for the detection of influenza epidemic outbreaks. *Statistical methods in medical research 24*(2), 206–223.

Cox, D. R., G. Gudmundsson, G. Lindgren, L. Bondesson, E. Harsaae, P. Laake, K. Juselius, and S. L. Lauritzen (1981). Statistical analysis of time series: some recent developments [with discussion and reply]. *Scandinavian Journal of Statistics 2*(8), 93–115.

Czado, C., T. Gneiting, and L. Held (2009). Predictive model assessment for count data. *Biometrics 65*(4), 1254–1261.

De Jong, P. (1988). The likelihood for a state space model. *Biometrika 75*(1), 165–169.

Donker, T., M. Boven, W. Ballegooijen, T. Klooster, C. Wielders, and J. Wallinga (2011). Nowcasting pandemic influenza A/H1N1 2009 hospitalizations in the Netherlands. *European Journal of Epidemiology 26*(3), 195–201.

Dowle, M., T. Short, and S. Lianoglou (2013). *`data.table`: Extension of `data.frame` for Fast Indexing, Fast Ordered Joins, Fast Assignment, Fast Grouping and List Columns.* `R` Package Version 1.8.8.

Drewe, J., L. Hoinville, A. Cook, T. Floyd, and K. Stärk (2012). Evaluation of animal and public health surveillance systems: a systematic review. *Epidemiology and infection 140*(04), 575–590.

Dunsmuir, W. and D. Scott (2015). The glarma Package for Observation-Driven Time Series Regression of Counts. *Journal of Statistical Software 67*(7), 1–36.

Elsaied, H. and R. Fried (2014). Robust fitting of INARCH models. *Journal of Time Series Analysis 35*(6), 517–535.

England, P. and R. Verrall (2002). Stochastic claims reserving in general insurance. *British Actuarial Journal 8*, 443–518.

Enki, D. G., A. Noufaily, P. Farrington, P. Garthwaite, N. Andrews, and A. Charlett (2016). Taylor's power law and the statistical modelling of infectious disease surveillance data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, n/a–n/a.

Enki, D. G., A. Noufaily, P. H. Garthwaite, N. J. Andrews, A. Charlett, C. Lane, and C. P. Farrington (2013). Automated biosurveillance data from England and Wales, 1991–2011. *Emerging infectious diseases 19*(1), 35.

Erbas, B. and R. J. Hyndman (2000). Seasonal adjustment methods for the analysis of respiratory disease in environmental epidemiology. Technical report, University of Melbourne, Monash University.

Faensen, D., H. Claus, J. Benzler, A. Ammon, T. Pfoch, T. Breuer, and G. Krause (2006). SurvNet@RKI – a multistate electronic reporting system for communicable diseases. *Eurosurveillance 11*(4), 100–103.

Fahrmeir, L., T. Kneib, S. Lang, and B. Marx (2013). *Regression: models, methods and applications*. Springer Science & Business Media.

Fahrmeir, L. and G. Tutz (2013). *Multivariate statistical modelling based on generalized linear models*. Springer Science & Business Media.

Farrington, C., N. Andrews, A. Beale, and M. Catchpole (1996). A statistical algorithm for the early detection of outbreaks of infectious disease. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 159*, 547–563.

Fricker, R. D., B. L. Hegler, and D. A. Dunfee (2008). Comparing syndromic surveillance detection methods: EARS' versus a CUSUM-based methodology. *Statistics in Medicine 27*(17), 3407–3429.

Fried, R., I. Agueusop, B. Bornkamp, K. Fokianos, J. Fruth, and K. Ickstadt (2015). Retrospective Bayesian outlier detection in INGARCH series. *Statistics and Computing 25*(2), 365–374.

Frisén, M. (2003). Statistical surveillance: Optimality and methods. *International Statistical Review 71*(2), 403–434.

Frisén, M. (2008). *Financial surveillance.* John Wiley & Sons.

Geraci, M. (2016). *Qtools: Utilities for Quantiles.* R package version 1.0.

Gergonne, B., A. Mazick, J. O'Donnell, A. Oza, B. Cox, F. Wuillaume, Z. Kaufman, M. Virtanen, H. Green, P. Hardelid, N. Andrews, R. Pebody, M. Holmberg, M. Detsis, C. Danis, H. Uphoff, L. Josseran, A. Fouillet, B. Nunes, P. Nogueira, C. Junker, L. van Asten, T. van Klooster, F. Simon, V. M. Flores, S. Tomsic, G. Spiteri, J. Nielsen, and K. Mølbak (2011). Work package 7 report: A European algorithm for a common monitoring of mortality across Europe. Technical report, Statens Serum Institut, Copenhaguen.

Gertler, M., M. Dürr, P. Renner, S. Poppert, M. Askar, J. Breidenbach, C. Frank, K. Preußel, A. Schielke, D. Werber, et al. (2015). Outbreak of Cryptosporidium hominis following river flooding in the city of Halle (Saale), Germany, August 2013. *BMC infectious diseases 15*(1), 88.

Guillou, A., M. Kratz, and Y. L. Strat (2014). An extreme value theory approach for the early detection of time clusters. A simulation-based assessment and an illustration to the surveillance of Salmonella. *Statistics in Medicine 33*(28), 5015–5027.

Hastie, T. (2015). *gam: Generalized Additive Models.* R package version 1.12.

Hawkings, D. M. and D. H. Olwell (1998). *Statistics for Engineering and Physical Science – Cumulative Sum Charts and Charting for Quality Improvement.* Springer-Verlag.

Heisterkamp, S. H., A. L. M. Dekkers, and J. C. M. Heijne (2006). Automated detection of infectious disease outbreaks: hierarchical time series models. *Statistics in Medcine 25*, 4179–4196.

Held, L., M. Höhle, and M. Hofmann (2005). A statistical framework for the analysis of multivariate infectious disease surveillance data. *Statistical Modelling 5*, 187–199.

Held, L. and D. Sabanés Bové (2014). *Applied Statistical Inference.* Springer-Verlag.

Held, L., B. Schrödle, and H. Rue (2010). *Posterior and Cross-validatory Predictive Checks: A Comparison of MCMC and INLA*, pp. 91–110. Statistical Modelling and Regression Structures. Physica-Verlag, Heidelberg.

Hess, K. T. and K. D. Schmidt (2002). A comparison of models for the chain-ladder method. *Insurance: Mathematics and Economics 31*(3), 351 – 364.

Heudorf, U., T. Eikmann, and M. Exner (2013). Rückblick auf 10 Jahre Infektionsschutzgesetz. *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz 56*(3), 455–465.

Hilbe, J. (2011). *Negative binomial regression.* Cambridge University Press.

Höhle, M. (2007). `surveillance`: An `R` package for the monitoring of infectious diseases. *Computational Statistics 22*(4), 571–582.

Höhle, M. and M. an der Heiden (2014). Bayesian nowcasting during the STEC O104:H4 outbreak in Germany, 2011. *Biometrics 70*(4), 993–1002.

Höhle, M. and A. Mazick (2010). Aberration detection in `R` illustrated by Danish mortality monitoring. In T. Kass-Hout and X. Zhang (Eds.), *Biosurveillance: A Health Protection Priority*, Chapter 12, pp. 215–238. CRC Press.

Höhle, M. and M. Paul (2008). Count data regression charts for the monitoring of surveillance time series. *Computational Statistics & Data Analysis 52*(9), 4357–4368.

Hulth, A., N. Andrews, S. Ethelberg, J. Dreesman, D. Faensen, W. van Pelt, and J. Schnitzler (2010). Practical usage of computer-supported outbreak detection in five European countries. *Eurosurveillance 15*(36).

Hutwagner, L., T. Browne, G. Seeman, and A. Fleischhauer (2005). Comparing Aberration Detection Methods with Simulated Data. *Emerging Infectious Diseases 11*, 314–316.

Hutwagner, L., W. Thompson, and G. M. Seeman (2003). The Bioterrorism Preparedness and Response Early Aberration Reporting System (EARS). *Journal of Urban Health: Bulletin of the New York Academy of Medicine 80*(2 Suppl 1), i89–96.

Jentsch, C. and A. Leucht (2015). Bootstrapping sample quantiles of discrete data. *Annals of the Institute of Statistical Mathematics*, 1–49.

Jones, G., S. Le Hello, N. Jourdan-da Silva, V. Vaillant, H. de Valk, F.-X. Weill, and Y. Le Strat (2014). The French human Salmonella surveillance system: evaluation of timeliness of laboratory reporting and factors associated with delays, 2007 to 2011. *Eurosurveillance 19*(1), 1–10.

Jung, R. C. and A. R. Tremayne (2011). Useful models for time series of counts or simply wrong ones? *AStA Advances in Statistical Analysis 95*(1), 59–91.

Kalbfleisch, J. D. and J. F. Lawless (1989). Inference based on retrospective ascertaintment: An analysis of the data on tranfusion related AIDS. *Journal of the American Statistical Association 84*(406), 360–372.

Kling, A., K. Hebing, M. Grünewald, and A. Hulth (2012). Two years of computer supported outbreak detection in Sweden: the user's perspective. *Journal of Health & Medical Informatics 3*(108), 2.

Koch, J., A. Schrauder, K. Alpers, D. Werber, C. Frank, R. Prager, W. Rabsch, S. Broll, F. Feil, P. Roggentin, et al. (2005). *Salmonella* agona outbreak from contaminated aniseed, Germany. *Emerging infectious diseases 11*(7), 1124–1127.

Krause, G., D. Altmann, D. Faensen, K. Porten, J. Benzler, T. Pfoch, A. Ammon, M. Kramer, and H. Claus (2007). SurvNet electronic surveillance system for infectious disease outbreaks, Germany. *Emerging Infectious Diseases 10*(13), 1548–1555.

Kulldorff, M. (1997). ***SaTScan**: Software for the Spatial, Temporal and Space-Time Scan Statistics*. Boston, MA, USA.

Läubrich, C., N. Bocter, H. Fickenscher, G. Selck, and P. Rautenberg (2011). Integriertes Bulletin zur automatisierten Surveillance meldepflichtiger Infektionserkrankungen in Schleswig-Holstein (IBISSH). *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz 54*(7), 875–885.

Lawless, J. (1994). Adjustments for reporting delays and the prediction of occurred but not reported events. *Canadian Journal of Statistics 22*(1), 15–31.

Lawless, J. F. (1987). Negative binomial and mixed Poisson regression. *Canadian Journal of Statistics 15*(3), 209–225.

Lee, H. C. and D. W. Apley (2011). Improved Design of Robust Exponentially Weighted Moving Average Control Charts for Autocorrelated Processes. *Quality and Reliability Engineering International 27*, 337–352.

Liboschik, T., K. Fokianos, and R. Fried (2015). Monitoring Count Time Series Following Generalized Linear Models. In preparation.

Liboschik, T., R. Fried, K. Fokianos, and P. Probst (2016). ***tscount**: Analysis of Count Time Series*. R package version 1.2.0.

Liboschik, T., P. Kerschke, K. Fokianos, and R. Fried (2014). Modelling interventions in INGARCH processes. *International Journal of Computer Mathematics* (ahead-of-print), 1–18.

Lin, H., P. S. F. Yip, and R. M. Huggins (2008). A double-nonparametric procedure for estimating the number of delay-reported cases. *Statistics in Medicine 27*, 3325–3339.

Ljung, G. M. and G. E. P. Box (1978). On a measure of lack of fit in time series models. *Biometrika 65*, 297–303.

Lloyd-Smith, J. O. (2007). Maximum likelihood estimation of the negative binomial dispersion parameter for highly overdispersed data, with applications to infectious diseases. *PloS ONE 2*(2), e180.

Lucas, J. M. and R. B. Crosier (1982). Fast initial response for CUSUM quality-control schemes: Give your CUSUM a head start. *Technometrics 24*(3), 199–205.

Luo, P., T. A. DeVol, and J. L. Sharp (2012). CUSUM analyses of time-interval data for online radiation monitoring. *Health physics 102*(6), 637–645.

Ma, Y., M. G. Genton, and E. Parzen (2011). Asymptotic properties of sample quantiles of discrete distributions. *Annals of the Institute of Statistical Mathematics 63*(2), 227–243.

Manitz, J. and M. Höhle (2013). Bayesian outbreak detection algorithm for monitoring reported cases of campylobacteriosis in Germany. *Biometrical Journal 55*(4), 509–526.

Martínez-Beneito, M. A., D. Conesa, A. López-Quílez, and A. López-Maside (2008). Bayesian Markov switching models for the early detection of influenza epidemics. *Statistics in medicine 27*(22), 4455–4468.

McCullagh, P. and J. Nelder (1983). *Generalized Linear Models.* London: Chapman & Hall.

Microsoft Corp. (2012). *Microsoft SQL Server Analysis Services, Version 2012.*

Midthune, D. N., M. P. Fay, L. X. Clegg, and E. J. Feuer (2005). Modeling reporting delays and reporting corrections in cancer registry data. *Journal of the American Statistical Association 100*(469), 61–70.

Murphy, S. P. and H. Burkom (2008). Recombinant temporal aberration detection algorithms for enhanced biosurveillance. *Journal of the American Medical Informatics Association 15*(1), 77–86.

Neill, D. B. (2012). Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 74*(2), 337–360.

Niemer, U. (2001). Das neue Infektionsschutzgesetz (IfSG). *Das Gesundheitswesen. Sonderheft 63*(2), S136–S138.

Noufaily, A., D. G. Enki, P. Farrington, P. Garthwaite, N. Andrews, and A. Charlett (2013). An improved algorithm for outbreak detection in multiple surveillance systems. *Statistics in Medicine 32*(7), 1206–1222.

Noufaily, A., P. Farrington, P. Garthwaite, D. G. Enki, N. Andrews, and A. Charlett (2015). Detection of infectious disease outbreaks from laboratory data with reporting delays. *Journal of the American Statistical Association 0*(ja), 1–32.

Noufaily, A., Y. Ghebremichael-Weldeselassie, D. G. Enki, P. Garthwaite, N. Andrews, A. Charlett, and P. Farrington (2014). Modelling reporting delays for outbreak detection in infectious disease data. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 178*(1), 205–222.

Parzen, E. (2004). Quantile probability and statistical data modeling. *Statistical Science 19*(4), 652–662.

Paul, M. and L. Held (2011). Predictive assessment of a non-linear random effects model for multivariate time series of infectious disease counts. *Statistics in Medicine 30*(10), 1118–1136.

Psarakis, S., A. K. Vynioua, and P. Castagliola (2014). Some Recent Developments on the Effects of Parameter Estimation on Control Charts. *Quality and Reliability Engineering International 30*, 1113–1129.

R Core Team (2015). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing.

Reis, B. Y., C. Kirby, L. E. Hadden, K. Olson, A. J. McMurry, J. B. Daniel, and K. D. Mandl (2007). AEGIS: a robust and scalable real-time public health surveillance system. *Journal of the American Medical Informatics Association 14*(5), 581–588.

Rigby, R. A. and D. M. Stasinopoulos (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society C (Applied Statistics) 54*(3), 507–554.

Ripley, B. and M. Lapsley (2012). ***RODBC**: ODBC Database Access.* R Package Version 1.3-6.

Rue, H., S. Martino, and N. Chopin (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(2), 319–392.

Rue, H., S. Martino, F. Lindgren, D. Simpson, A. Riebler, and E. T. Krainski (2015). *INLA: Functions which allow to perform full Bayesian analysis of latent Gaussian models using Integrated Nested Laplace Approximations.* R package version 0.0-1420281647.

Salmon, M., D. Schumacher, H. Burmann, C. Frank, H. Claus, and M. Höhle (2016). Automated outbreak detection system for notifiable diseases in Germany. *Eurosurveillance*. Accepted for publication.

Salmon, M., D. Schumacher, and M. Höhle (2016). Monitoring Count Time Series in R: Aberration Detection in Public Health Surveillance. *Journal of Statistical Software*. Accepted for publication.

Salmon, M., D. Schumacher, K. Stark, and M. Höhle (2015). Bayesian outbreak detection in the presence of reporting delays. *Biometrical Journal 57*(6), 1051–1067.

Sánchez-Vargas, F. M., M. A. Abu-El-Haija, and O. G. Gómez-Duarte (2011). *Salmonella* infections: An update on epidemiology, management, and prevention. *Travel Medicine and Infectious Disease 9*(6), 263 – 277.

Schmidt, K. D. (2006). *Versicherungsmathematik*. Springer-Verlag.

Schmidt, K. D. and A. Wünsche (1998). Chain ladder, marginal sum and maximum likelihood estimation. *Blätter der Deutschen Gesellschaft für Versicherungs und Finanz Mathematik 23*(3), 267–277.

Schuh, A., J. A. Camelio, and W. H. Woodall (2014). Control charts for accident frequency: a motivation for real-time occupational safety monitoring. *International Journal of Injury Control and Safety Promotion 21*(2), 154–162.

Schumacher, J., M. Diercke, M. Salmon, I. Czogiel, D. Schumacher, H. Claus, and A. Gilsdorf (2016). Timeliness in the German surveillance system for infectious diseases: Amendment of the Infection Protection Act in 2013 decreased transmission time to 1 day. In preparation.

Shewhart, W. (1931). *Economic Control of Quality of Manufactured Product*. Princeton: Van Nostrand Reinhold.

Shmueli, G. and H. Burkom (2010). Statistical challenges facing early outbreak detection in biosurveillance. *Technometrics 52*(1), 39–51.

Sonesson, C. and D. Bock (2003). A review and discussion of prospective statistical surveillance in public health. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 166*(1), 5–21.

Spiegelhalter, D., C. Sherlaw-Johnson, M. Bardsley, I. Blunt, C. Wood, and O. Grigg (2012). Statistical methods for healthcare regulation: rating, screening and surveillance. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 175*(1), 1–47.

Stasinopoulos, D. M. and R. A. Rigby (2007). Generalized Additive Models for Location Scale and Shape (GAMLSS) in R. *Journal of Statistical Software 23*(7), 1–46.

Stroup, D., G. Williamson, J. Herndon, and J. Karon (1989). Detection of aberrations in the occurrence of notifiable diseases surveillance data. *Statistics in Medicine 8*, 323–329.

Tango, T., K. Takahashi, and K. Kohriyama (2011). A space–time scan statistic for detecting emerging outbreaks. *Biometrics 67*(1), 106–115.

Unkel, S., C. P. Farrington, P. H. Garthwaite, C. Robertson, and N. Andrews (2012). Statistical methods for the prospective detection of infectious disease outbreaks: a review. *Journal of the Royal Statistical Society A 175*(1), 49–82.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (Fourth ed.). New York: Springer. ISBN 0-387-95457-0.

Wei, W. and L. Held (2014). Calibration tests for count data. *TEST 23*(4), 787–805.

Weiß, C. H. and Z. Lu (2015). SPC methods for time-dependent processes of counts – a literature review. *Cogent Mathematics 2*(1), 1111116.

Wickham, H. (2013). ***testthat**: Testthat Code. Tools to Make Testing Fun :)*. R Package Version 0.7.1.

Woodall, W. (2006). The Use of Control Charts in Health-Care and Public-Health Surveillance. *Journal of Quality Technology 38*(2), 89–104.

Zeger, S. L., L.-C. See, and P. J. Diggle (1989). Statistical methods for monitoring the AIDS epidemic. *Statistics in Medicine 8*(1), 3–21.

Zeileis, A., C. Kleiber, and S. Jackman (2008). Regression Models for Count Data in R. *Journal of Statistical Software 27*(1).

Danksagung

Danksagung

An dieser Stelle möchte ich mich erstens bei Michael Höhle herzlich bedanken. Det är jättekult att du skrev forskningförslaget som fick mig att jobba på detta tema. Tack för ditt goda råd – meningen som du skrev i marginalen, så kallade *IMHO* – och ditt tålamod. Jag har lärt jättemycket och är jättetacksam över att du var så krävande. Ja, det är jag! Tusen tack!

Diese Arbeit wurde durch das Robert Koch Institut finanziert, wofür ich sehr dankbar bin. Vielen Dank an Göran Kirchner, der mich auf die Stellenausschreibung hinwies.

Roland Fried und Volker Schmid danke ich dafür, dass Sie sich bereit erklärten meine Arbeit zu begutachten.

Natürlich bedanke ich mich bei Felix Weidemann, der als *Doktorbruder* ein sehr gutes Vorbild und immer hilfsbereit war, sei es für die wissenschaftlichen oder für die administrativen Aspekte dieser Arbeit. Ich bin für alle Deine Tipps sehr dankbar! Unserer jüngeren *Doktorschwester* Theresa Stocks wünsche übrigens alles Gute!

Ebenso bedanken möchte ich mich bei meinen Kollegen im Fachgebiet 35 für die sehr nette Atmosphäre. Insbesondere bedanke ich mich bei Susanne Behnke für Deine geduldige administrative Hilfe. Außer dem Fachgebiet hat es viel Spaß gemacht, mit Hendrik Burmann, Katrin Neubauer, Emilie Peron, Dirk Schumacher und Jakob Schumacher zu arbeiten. Bei den zwei Statistikern der Abteilung Matthias an der Heiden und Irina Czogiel möchte ich mich u.a. für Eures Feedback zu Präsentationen vor Konferenzen bedanken. Und außer dem RKI war Sebastian Meyer trotz meiner R-check-Errors immer nett und hilfsbereit, danke schön.

Ich will nicht vergessen, mich bei meinem *PhD writing buddy* Tobias Liboschik zu bedanken, der diese Dissertation sehr fleißig gelesen und kommentiert hat. Vielen Dank!

Ich will mich auch bei den Abteilungs- und Institutskollegen bedanken, die beim Yogakurs, beim Fitnesskurs, beim *déjeuner francophone* oder bei einfachen Kaffeepausen mitgemacht haben. Zwischenzeitlich habe ich natürlich auch gearbeitet aber die guten Pausen in einer freundlichen Atmosphäre fand ich hilfreich!

Danke an Florian Stallmann für Deine Hilfe in München und an Anna Loenenbach, die meinen Lebenslauf auf deutsche Rechtschreibung geprüft hat.

Bei Michael Kobr und Volker Klüpfel bedanke ich mich dafür, dass Ihre Arbeit mir stets wertvolle Kenntnisse über Bayern lieferte.

Letztens bedanke ich mich bei Natalie Müller, die diese Danksagung auf deutsche Rechtschreibung geprüft hat. Leider wurde dieser Satz von niemandem geprüft.

Acknowledgements

I would like to thank Doyo Enki, Paddy Farrington and Angela Noufaily for being so open about their work. It was a pleasure to get their very interesting emails and to get a glimpse into not already published manuscripts.

I also would like to thank Simon Le Hello who enthusiastically explained to me automatic aberration detection at the French Salmonella national reference centre, and also Gabrielle Jones and Yann Le Start for their inspiration regarding the importance of analysing reporting delays in surveillance systems.

I am moreover very thankful that Håvard Rue always took time to answer my questions about INLA in the users mailing list.

Lastly, I would like to thank all other people that published R packages, and asked and answered online questions about the R programming language.

Merci

Grácies als meus companys de feina a Barcelona i a l'Índia per ser tan simpatics. M'agrada molt treballar amb vosaltres!

Remerciements

Je tiens à remercier en premier lieu mon mari pour son indéfectible soutien et ses blagues (notre humour de vieux !). Je remercie également du fond du cœur ma famille pour son soutien à distance, ainsi que ma belle-famille. Quant à mes amis, ceux qui ont fait, font ou commencent une thèse ont été d'un grand soutien, ceux qui vivent à Berlin ou ailleurs en Allemagne ont rendu ces trois années allemandes fort agréables, et ceux qui ne rentrent dans aucune de ces catégories ont quand même le droit d'être remerciés pour leurs encouragements et les bons moments partagés ! Pour finir, Super Tom, travaille bien à l'école, surtout en maths !

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

SALMON Maëlle

Name, Vorname

Barcelona, 21.09.2016

Ort, Datum

Unterschrift Doktorand/in

Formular 3.2