

Use R!

Ottar N. Bjørnstad

Epidemics

Models and Data using R

Use R!

Series Editors

Robert Gentleman Kurt Hornik Giovanni Parmigiani

More information about this series at <http://www.springer.com/series/6991>

Ottar N. Bjørnstad

Epidemics

Models and Data using R



Springer

Ottar N. Bjørnstad
Center for Infectious Disease Dynamics
Pennsylvania State University
University Park, PA, USA

ISSN 2197-5736

Use R!

ISBN 978-3-319-97486-6

<https://doi.org/10.1007/978-3-319-97487-3>

ISSN 2197-5744 (electronic)

ISBN 978-3-319-97487-3 (eBook)

Library of Congress Control Number: 2018953687

© Springer Nature Switzerland AG 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

For Katriona, Esme, and Michael

Preface

Despite an undergraduate degree in Zoology and an MSc on the behavior of voles, I have long been fascinated by theoretical biology and the relationship between models and data, and the feedback between statistical analysis and conceptual developments in the area of infectious disease dynamics, in particular, and ecological dynamics in general. My perpetual frustration has been to read all the wonderfully clever books and journal articles exuding all sorts of nifty maths and stats, but not quite being able to *do* any of it myself when it came to infectious diseases that I care about. This frustration led me to attempt to make myself some worked examples of all this cleverness. Over the years the stack of how-tos has grown, and the following chapters are an attempt at organizing these so they may be useful for others. I have tried to organize the chapters and sections in a reasonably logical way: Chaps. 1–10 are a mix and match of models, data, and statistics pertaining to local disease dynamics; Chaps. 11–13 pertain to spatial and spatiotemporal dynamics; Chap. 14 highlights similarities between the dynamics of infectious disease and parasitoid-host dynamics; finally, Chaps. 15 and 16 overview additional statistical methodology I have found useful in studies of infectious disease dynamics. Some sections are marked as “advanced” for one of two reasons: (1) either the maths or stats is a bit more involved or (2) the topic in focus is a bit more esoteric. Although not marked as such, most of Chap. 10 is advanced in this respect. While less run-of-the-mill, I have thought it important to include these sections, because they cover topics that may be less easy to find code for online.

I have had invaluable help from students, colleagues, and collaborators in my quest. The preconference workshops of “Ecology and Evolution of Infectious Diseases” that I co-taught between 2005 and 2008 enhanced my motivation to annotate many worked examples; bare bones of several of the following sections were written during frantic 24-h stints prior to these workshops. Much of the other material arose from interactions with students and postdocs at Pennsylvania State University’s Center for Infectious Disease Dynamics (CIDD). Parts of the epidemics on networks and the R_0 removal estimator is from Matt Ferrari’s PhD research, and the age-structured SIR simulator and the SIRWS model are from Jennie Lavine’s PhD

work. Working with distributed-delay models has been a collaboration with Bill Nelson and my students Lindsay Beck-Johnson and Megan Greischar. Angie Luis and I cooked up the R code to do transfer functions as part of her PhD research. Much of the code on the catalytic model is from collaborations with Laura Pomeroy and then CIDD postdoctoral fellows Grainne Long and Jess Metcalf. The in-host TSIR was also a collaboration with Jess. The Gillespie code arose from collaborations with postdoctoral fellow Shouli Li and my honor student Reilly Mummah. Reilly also taught me how to write my first `Shiny` app. Away from Penn State, Aaron King and Ben Bolker have at various times been unbelievably patient in teaching me bits of maths I did not understand. Roger Nisbet painstakingly guided me through my first transfer functions during my postdoctoral fellowship at NCEAS. During the same period, Jordi Bascompte introduced me to coupled map lattice models. Finally, Bryan Grenfell showed me wavelets and introduced me to the field of infectious disease dynamics some 20 years ago.

The data used has been kindly shared by Janis Antonovics, Jeremy Burdon, Rebecca Grais, Sylvie Huygen, Jenn Keslow, Sandy Leibhold, Grainne Long, and Mary Poss. The first draft of the text was completed while I was on sabbatical at the University of Western Australia and University of Oslo/the Norwegian Veterinary Institute during 2017. My work leading up to this text has variously been funded by the National Science Foundation, the National Institutes of Health, the US Department of Agriculture, and the Bill and Melinda Gates Foundation.

University Park, PA, USA
May 2018

Ottar N. Bjørnstad

Contents

1	Introduction	1
1.1	Preamble	1
1.2	In-Host Persistence	2
1.3	Patterns of Endemicity	4
1.4	R	6
1.5	Other Resources	8
2	SIR	9
2.1	The SIR Model	9
2.2	Numerical Integration of the SIR Model	11
2.3	Final Epidemic Size	14
2.4	Open Epidemic	18
2.5	Phase Analyses	18
2.6	Stability and Periodicity	21
2.7	Advanced: More Realistic Infectious Periods	23
2.8	ShinyApp	27
3	R_0	31
3.1	Primacy of R_0	31
3.2	Preamble: Rates and Probabilities	32
3.3	Estimating R_0 from a Simple Epidemic	33
3.4	Maximum Likelihood: The Chain-Binomial Model	35
3.5	Stochastic Simulation	40
3.6	Further Examples	41
3.6.1	Influenza A/H1N1 1977	41
3.6.2	Ebola Sierra Leone 2014–2015	43
3.6.3	Ebola DRC 1995	46
3.7	R_0 from S(E)IR Flows	47
3.8	Other Rules of Thumb	49
3.8.1	Mean Age of Infection	49

3.8.2	Final Epidemic Size	49
3.8.3	Contact Tracing	49
3.9	Advanced: The Next-Generation Matrix	51
3.9.1	SEIR	51
3.9.2	SEIHFR	53
4	FoI and Age-Dependent Incidence	57
4.1	Burden of Disease	57
4.2	Force of Infection	58
4.3	Probability of Infection at Age: The Catalytic Model	59
4.4	More Flexible ϕ -Functions	62
4.5	A Log-Spline Model	66
4.6	Rubella	69
4.7	WAIFW	74
4.8	Advanced: RAS Model	76
5	Seasonality	81
5.1	Environmental Drivers	81
5.2	The Seasonally Forced SEIR Model	84
5.3	Seasonality in β	85
5.4	Bifurcation Analysis	89
5.5	Stroboscopic Section	90
5.6	Susceptible Recruitment	92
5.7	ShinyApp	94
6	Time-Series Analysis	95
6.1	Taxonomy of Methods	95
6.2	Time Domain: ACF and ARMA	95
6.2.1	ARMA	97
6.3	Frequency Domain	99
6.4	Wavelets	100
6.5	Measles in London	102
6.6	Project Tycho	107
6.7	Lomb Periodogram: Whooping Cough	107
6.8	Triennial Cycles: Philadelphia Measles	108
6.9	Wavelet Reconstruction and Wavelet Filter: Diphtheria	111
6.10	Advanced: FFT and Reconstruction	114
7	TSIR	117
7.1	Stochastic Variability	117
7.2	Estimating Parameters in Dynamic Models	120
7.3	Estimation Using the TSIR	121
7.3.1	Inference (Hypothetical)	121
7.4	Inference (for Real)	122
7.4.1	Susceptible Reconstruction	122
7.4.2	Estimation	124

7.5	Simulating the TSIR Model	127
7.6	Tycho Data	129
7.7	In-Host Malaria Dynamics	131
7.8	ShinyApp	135
8	Trajectory Matching	137
8.1	Preamble: Prevalence <i>Versus</i> Incidence	137
8.2	Event-Based Stochastic Simulation	137
8.3	Trajectory Matching	143
8.4	Likelihood Theory 101	144
8.5	SEIR with Error	147
8.6	Boarding School Flu Data	149
8.7	Measles	150
8.8	Outbreak-Response Vaccination	153
8.9	ShinyApp	156
9	Stability and Resonant Periodicity	159
9.1	Preamble: Rabies	159
9.2	Linear Stability Analysis	159
9.3	Finding Equilibria	161
9.4	Evaluating the Jacobian	163
9.5	Raccoon Rabies	164
9.6	Influenza	168
9.7	Advanced: Transfer Functions	171
9.7.1	SIR	171
9.7.2	The TSIR	173
9.8	(Even More) Advanced: Transfer Functions and ARMA Delay-Coordinates	176
9.9	ShinyApp	178
10	Exotica	179
10.1	Introduction	179
10.2	Chaos	180
10.3	Local Lyapunov Exponents	183
10.4	Coexisting Attractors	189
10.5	Repellors/Unstable Manifolds	192
10.6	Invasion Orbits	196
10.7	Stochastic Resonance	198
10.8	Predictability: Empirical Dynamic Modeling	203
	Appendix: Making a Pomp-Simulator	205
11	Spatial Dynamics	209
11.1	Dispersal Kernels	209
11.2	<i>Filipendula</i> Rust	210
11.3	Simulation	215
11.4	Gypsy Moth	217

11.5	Coupled Map Lattice SI Models	217
11.6	Making Movies	219
11.7	Nonparametric Covariance Functions for Spatiotemporal Data ..	220
11.8	Gravity Models	222
12	Transmission on Networks	227
12.1	S Preamble: Objects, Classes, and Functions	227
12.2	Networks	229
12.3	Models of Networks	229
12.3.1	Watts-Strogatz Networks	231
12.3.2	Barabasi-Albert Networks	233
12.4	Epidemics on Networks	234
13	Spatial and Spatiotemporal Patterns	241
13.1	Introduction	241
13.2	A Plant-Pathogen Case Study	241
13.3	Spatial Autocorrelation	242
13.4	Testing and Confidence Intervals	245
13.5	Mantel Test	246
13.6	Correlograms	246
13.7	Nonparametric Spatial Correlation Functions	247
13.8	LISA	249
13.9	Cross-Correlations	250
13.10	Gypsy Moth	252
14	Parasitoids	255
14.1	Parasitoid-Host Dynamics	255
14.2	Stability and Resonant Periodicity	260
14.3	Biological Control	262
14.4	Larch Bud Moth	263
14.5	Host-Parasitoid Metapopulation Dynamics	263
14.6	ShinyApp	265
15	Non-independent Data	267
15.1	Introduction	267
15.2	Spatial Dependence	267
15.2.1	Random Blocks	268
15.2.2	Spatial Regression	269
15.3	Repeated Measures of In-Host Mouse Malaria	273
15.4	<i>B. bronchiseptica</i> in Rabbits	279
16	Quantifying In-Host Patterns	283
16.1	The Experiments	283
16.2	Data	283
16.3	PCA of FIV Day 31 and 59 Data	284
16.4	LDA of FIV Day 31 and 59 Data	286

Contents	xiii
16.5 MANOVA of FIV Day 59 Data	290
16.6 PCA of Mouse Malaria	291
16.7 FDA of Mouse Malaria	293
References	297
Index	311

Chapter 1

Introduction



1.1 Preamble

The use of mathematical models to understand infectious disease dynamics has a very rich history in epidemiology. Kermack and McKendrick (1927) is the seminal paper that introduced the equations for the general Susceptible-Infected-Removed model and showed how a set of restrictive assumptions lead to the standard SIR model of ordinary differential equations. During the 1950s and early 1960s stochastic theories of disease dynamics were developed by Bailey (1957) and Bartlett (1960b). Bartlett (1956, 1960a) further pioneered the use of Monte Carlo simulations of epidemics with the aid of “electronic computers” (as opposed to regular human computers), while Muench (1959) proposed the “catalytic” framework for understanding age-incidence patterns.¹ The decades to follow saw broad expansions of theories as well as a surge in real-life application of mathematics to dynamics and control of infectious disease.

There are several excellent textbooks of mathematical epidemiology including Anderson and May (1991) and Keeling and Rohani (2008). The purpose of the current text is not to replicate these efforts but rather use these frameworks as a starting point to discuss practical implementation and analysis. The discussion will be centered around a somewhat haphazard collection of case studies selected to explore various conceptual, mathematical, and statistical issues. The text is designed to be more of a “practicum in infectious disease dynamics.”

The dynamics of infectious diseases shows a wide diversity of pattern. Some have locally persistent chains-of-transmission; others persist spatially in “consumer-resource metapopulations.” Some infections are prevalent among the young, some among the old, and some are age-invariant. Temporally, some diseases have little

¹ Though, as reviewed by Dietz and Heesterbeek (2002), the original calculations leading to the catalytic model was proposed by Daniel Bernoulli in the late eighteenth century.

variation in prevalence, some have predictable seasonal shifts, and others exhibit violent epidemics that may be regular or irregular in their timing. Models and “models-with-data” have proved invaluable for understanding and predicting this diversity, and thence help improve intervention and control. The following chapters are an attempt at providing some notes for a “field guide” for working with data, models, and “models-and-data” to understand epidemics and infectious disease dynamics in space and time.

1.2 In-Host Persistence

Infectious diseases can be classified according to their persistence *within* the host and attack rates with respect to age. Some infections result in life-long colonization of a host because the immune system does not clear them. Such “in-host persistence” may be because the immune system permits it—as for the many symbionts that are beneficial to the host (viz. commensals and mutualists)—or because detrimental symbionts (viz. pathogens) are able to evade clearance. Examples of “in-host persistent” pathogens are retroviruses such as HIV, latent viruses such as, herpes viruses, and a number of bacteria such as the causative agents of tuberculosis (*Mycobacterium tuberculosis*) and leprosy (*M. leprae*).

“Acute” infections, in contrast, result in transient colonization of the host—that in humans can last for days or months depending on the pathogen—followed by clearance. The clearance is usually immune-mediated, though some viruses like canine distemper virus may run out of target cells and some pathogens may have a programmed life cycle within the host. Some coccidian pathogens within the genus *Eimeria*, for example, go through an exact number of replication cycles in the host (as merozoites) before all pathogen cells are expelled into the environment (as oocysts). The more common example of transience is due to immune-mediated clearance. Examples are plentiful and include acute viruses like measles and influenza, bacteria such as many that causes respiratory disease like bacterial meningitis (e.g., *Neisseria meningitidis*) or whooping cough (*Bordetella pertussis* and *B. parapertussis*), and protozoans such as those that cause malaria (*Plasmodium* spp.).

Among the acute infections we further distinguish between those that leave sterilizing immunity following clearance versus those that leave no or short-lived immunity. This can happen via a number of mechanisms including variable gene expression, rapid evolution, co-circulating strain clouds, or other immune evasive maneuvers. *N. meningitidis* and its congener *N. gonorrhoeae* (which cause gonorrhea), for example, are thought to leave little effective immune memory because of the bacteria’s ability to express a very variable arsenal of surface proteins (e.g., Stern

et al. 1984; Tettelin et al. 2000). Many influenza subtypes, in contrast, render effective immune memory short-lived because of rapid evolution; high mutation rates lead to “antigenic drift” and viral recombination during coinfection leads to antigenic “shifts.” *Plasmodium falciparum* is thought to be comprised of a diverse set of strains with nonoverlapping “antigenic repertoires” (as well as variable antigen expression) that allows repeat reinfection (e.g., Gupta et al. 1998). A number of common viral afflictions of children have a somewhat more limited strain diversity that may allow several reinfection cycles, but the immune system is ultimately able to cover their antigenic space; Examples include rotavirus (Pitzer et al. 2011) and the enterovirus-complex that cause hand-foot-and-mouth disease (Takahashi et al. 2016). Finally, many pathogens have various “anti-immune devices.” Respiratory syncytial virus, for example, uses molecular decoys against neutralizing antibodies (Bukreyev et al. 2008) and *Bordetella pertussis* employs the pertussis toxin to, at least transiently, inhibit recruitment of immune effector cells to sites of infection (Kirimanjeswara et al. 2005).

Many of the remaining “acute, immunizing pathogens”—the ones that result in a transient infection followed by life-long sterilizing immunity—are the poster children of mathematical epidemiology. Notable examples are among the classic vaccine-preventable viruses like measles, rubella, and smallpox. From a biological point of view, the complete failure of immune escape of these pathogens is somewhat mysterious (Kennedy and Read 2017), but the resulting simple dynamical clockwork is a joy to anyone hoping to apply mathematics to understand the living world.

From an epidemiological point of view, it is important to make the *functional*—as opposed to taxonomical—classification of pathogens because it allows us to understand the differences in age-specific attack rates and contrasting disease dynamics. The acute, immunizing infections mainly circulate among the young and therefore comprise the many “childhood” infections because most or all older hosts are immune. From the point of view of the compartmental “SIR-like” formalism (Fig. 1.1), it is thus natural to divide the host population in S, I, and R compartments and assume a unidirectional flow from susceptible children through immune (“removed”) adults. In contrast, the prevalence of “in-host persistent” infections will tend to accumulate with age. With respect to the SIR formalism, it is thus natural to consider a model with a unidirectional flow from the S class to a terminal I class. The acute but imperfectly immunizing infections should lead to relatively age-invariant attack rates, and $S \rightarrow I \rightarrow S$ or $S \rightarrow I \rightarrow R \rightarrow S$ flows depending on the duration of immune protection.

The SIR-like framework predicts how the broad expectation for age-prevalence curves will be modulated by factors such as age-specific pattern of mixing and differential mortality between infected and noninfected individuals. Statistical epidemiology can thus be used to probe empirical patterns to discover subtleties in the dynamics of disease transmission that is hard to observe directly.

1.3 Patterns of Endemicity

We can classify the dynamics of infectious disease according to broad “patterns of endemicity.” First, there is the distinction between locally persistent *vs* locally non-persistent pathogens. Local persistence fails when a local chain-of-transmission breaks. This can happen for two very different reasons (Fig. 1.1): (i) The transmission bottleneck is when a pathogen is insufficiently transmissible to sustain a chain of transmission; (ii) at the opposite end of the spectrum is the susceptible bottleneck for acute pathogens that are so transmissible that they burn through susceptibles much faster than they are replenished. In measles, for example, prevaccination cities in the USA smaller than a *critical community size* (CCS) of 250k–500k people did not produce enough children to sustain a local chain-of-transmission (Bartlett 1960a) (Fig. 1.2). Recurrence of such pathogens typically involves spatial dynamics and persistence at the metapopulation scale through spread among asynchronous local host communities (Keeling et al. 2004) or core-satellite dynamics in which a few large cities above the CCS serve as persistent sources for spatial dissemination to communities below the CCS (Grenfell and Harwood 1997; Grenfell et al. 2001).

The 1988 and 2002 epidemics of a related morbilli virus, the phocine distemper virus, in European harbor seals is another illustrations of locally non-persistent infections due to high transmission relative to susceptible recruitment rates (e.g., Swinton 1998). Following introduction into each local population (“haul-out”), explosive local epidemics terminated after 1–4 months due to susceptible depletion. When such epidemics happens so fast that recruitment of susceptibles (through birth, immigration, or loss of immunity) is negligible during the course of the outbreak we call it a “closed epidemic.” The closed epidemic is the focus of the standard Susceptible-Infected-Recovered model which we will study in Chap. 2. At the opposite end of the transmissibility spectrum, pathogens may bottleneck because transmission is too ineffective. In particular, if the basic reproductive ratio (R_0 , the

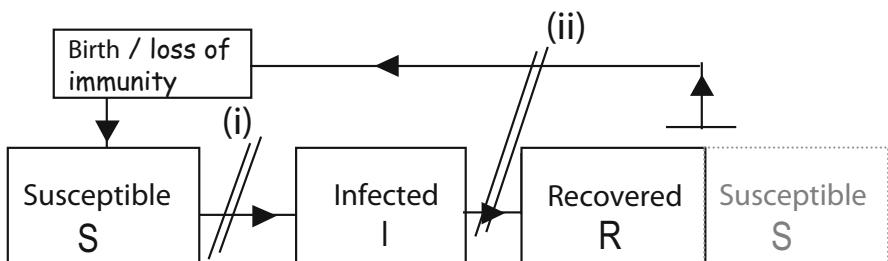


Fig. 1.1 The two bottlenecks for local persistence: (i) the transmission bottleneck for poorly transmitted infections and (ii) the susceptible bottleneck for highly transmissible, acute immunizing (or lethal) pathogens

expected number of secondary cases from a primary case in a completely susceptible population) is smaller than one, we see stuttering (“subcritical”) chains of transmission followed by pathogen fade-out. We see this in many zoonoses such as monkey pox and nipah (stage 3 zoonoses in the classification by Lloyd-Smith et al. 2009). Persistent recurrence of these typically involves reservoir host and intermittent zoonotic reintroduction. For example, in their study of Lassa fever in Sierra Leone, Iacono et al. (2015) concluded that about 20% of the human cases were caused by human-to-human transmission (with an average reproductive ratio below one) while the remaining majority was caused by transmission from the multimammate rat (*Mastomys natalensis*) reservoir.

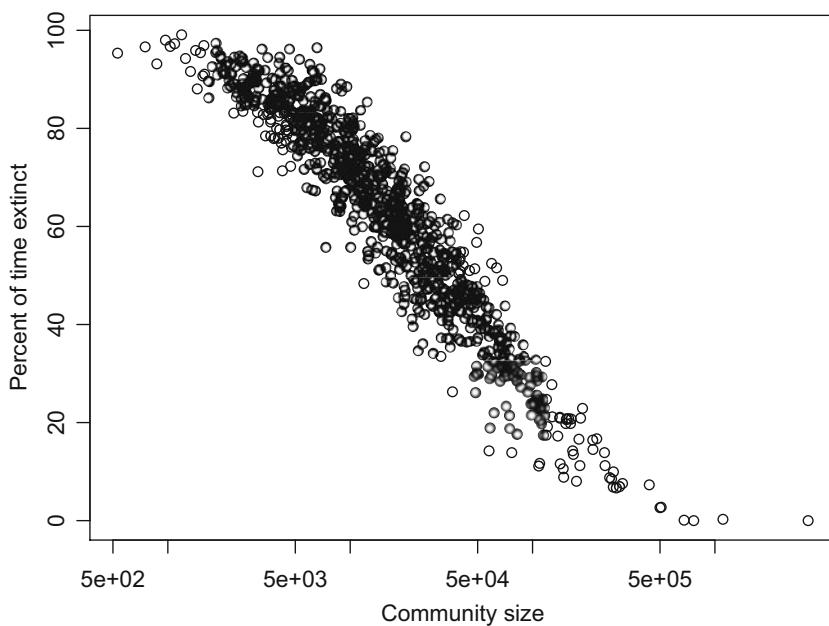


Fig. 1.2 Persistence of measles against population size for 954 cities and villages in pre-vaccination England and Wales (1944–1964). Communities below 500k exhibited occasional or frequent (depending on size) local extinction of the virus

The locally persistent infections can be classified as: (1) *Stable endemics* that show little variation in incidence through time. Many STDs with SI and SIS-like dynamics like gonorrhea (Fig. 1.3a) and HIV exhibit this pattern. (2) *Seasonal endemics* that show low’ish-level predictable seasonal variation around some mean. Many endemic vector-borne and water-borne infections exhibit this pattern. A classic example is the seasonal two-peaked mortality rate from Cholera in the province

of Dacca, East Bengal (King et al. 2008); The first peak at the beginning of the monsoon season and the second towards the end (Fig. 1.3b). Finally, (3) recurrent epidemics that may be regular or irregular are characterized by violent epidemic fluctuations over time. Many acute, immunizing highly contagious pathogens—measles being the poster-child—follow this pattern (Fig. 1.4).

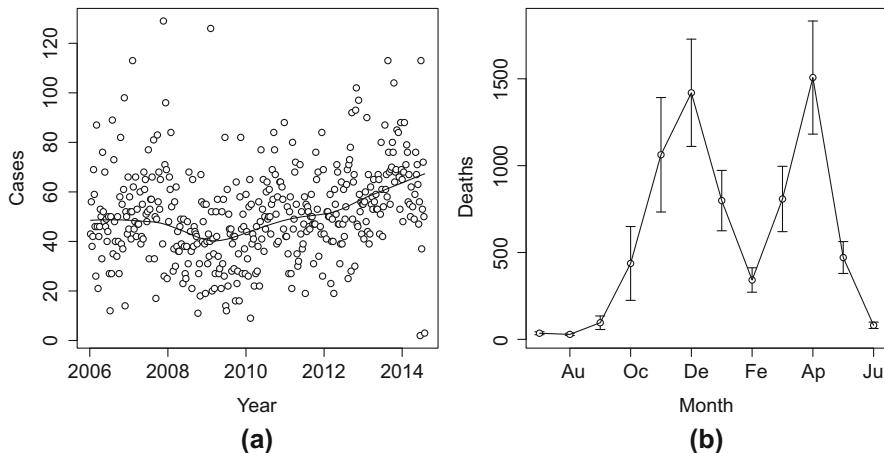


Fig. 1.3 Incidence of (a) weekly incidence of gonorrhea in Massachusetts (2006–2015) and (b) monthly average (\pm SE) mortality from cholera in the Dacca district (1891–1940)

1.4 R

To provide a cohesive framework for the practical calculations, all analyses are done in the open-source [R-program](#). The text is written assuming a basic knowledge of this platform. All functions, data, and ShinyApp's discussed in the text are contained in the `epimdr`-package. With the package everything contained herein should be reproducible. The above Figs. 1.2 and 1.4 were for example generated using the following code:

```
#Fig 1.2
data(ccs)
plot(ccs$size, ccs$ext*100, log="x", xlab=
  "Community size", ylab="Percent
  of time extinct")

#Fig 1.3a
plot(magono$time, magono$number, ylab="Cases",
  xlab="Year")
lines(lowess(x=magono$time, y=magono$number, f=.4))
```

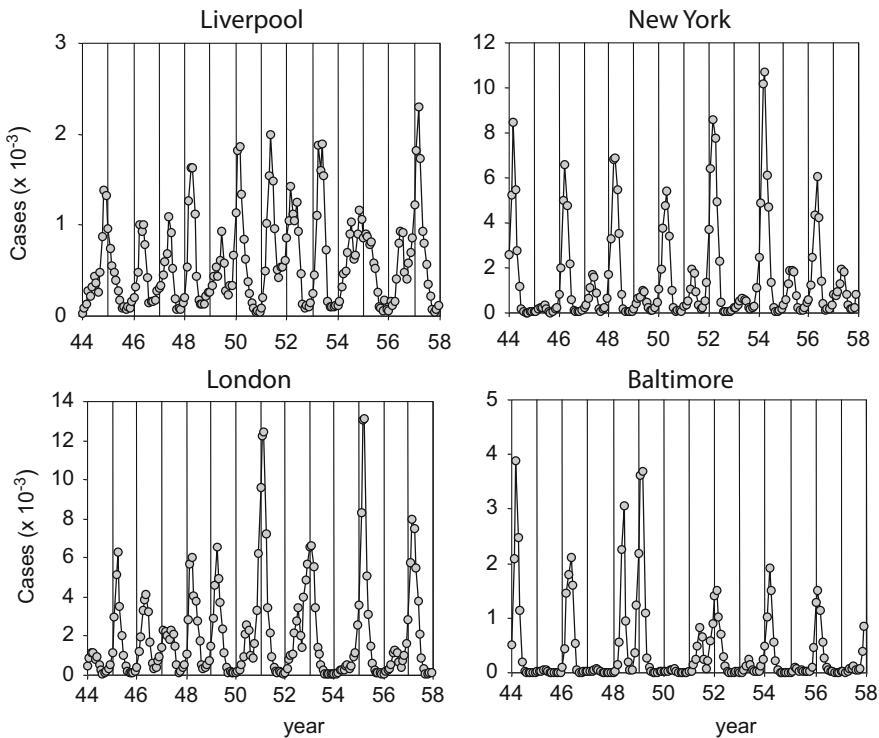


Fig. 1.4 Incidence of measles in various US and UK cities during the pre-vaccination era. The data represent fortnightly incidence (roughly corresponding to the virus' serial interval). The vertical bars mark annual intervals

```
#Fig 1.3b
data(cholera)
ses=sesdv=rep(NA, 12)
ses[c(7:12, 1:6)]=sapply(split(cholera$Dacca,
  cholera$Month), mean, na.rm=TRUE)
sesdv[c(7:12, 1:6)]=sapply(split(cholera$Dacca,
  cholera$Month), sd, na.rm=TRUE)/
  sqrt(length(split(cholera$Dacca, cholera$Month)))
require(plotrix)
plotCI(x=1:12, y=ses, ui=ses+sesdv, li=ses-
  sesdv, xlab="Month", ylab="Deaths")
lines(x=1:12, y=ses)
```

1.5 Other Resources

A 5 min overview of *Patterns of endemicity* can be watched from YouTube: https://www.youtube.com/watch?v=Mf_EZm5amxI. This video is part of the Pennsylvania State University-produced [epidemics-MOOC](#). The entire course is accessible free from <https://www.coursera.org/learn/epidemics>.

Chapter 2

SIR



2.1 The SIR Model

In 1927, Kermack and McKendrick (1927) published a set of general equations (Breda et al. 2012) to better understand the dynamics of an infectious disease spreading through a susceptible population. Their motivation was

“One of the most striking features in the study of epidemics is the difficulty of finding a causal factor which appears to be adequate to account for the magnitude of the frequent epidemics of disease which visit almost every population [...] The problem may be summarized as follows: One (or more) infected person is introduced into a community of individuals, more or less susceptible to the disease in question. The disease spreads from the affected to the unaffected by contact infection. Each infected person runs through the course of his sickness, and finally is removed from the number of those who are sick, by recovery or by death. The chances of recovery or death vary from day to day during the course of his illness. The chances that the affected may convey infection to the unaffected are likewise dependent upon the stage of the sickness. As the epidemic spreads, the number of unaffected members of the community becomes reduced [...] In the course of time the epidemic may come to an end. One of the most important problems in epidemiology is to ascertain whether this termination occurs only when no susceptible individuals are left, or whether the interplay of the various factors of infectivity, recovery and mortality, may result in termination, whilst many susceptible individuals are still present in the unaffected population.”

Following a general mathematical exposé, they suggested a set of pragmatic assumptions which lead to the standard SIR model of ordinary differential equations

This chapter uses the following R-packages: `deSolve`, `rootSolve`, `phaseR`, and `shiny`. A conceptual understanding of *reproductive ratios* and the *closed epidemic* is useful prior to this discussion. Five minute epidemics-MOOC introductions can be watched from YouTube:
Reproductive number <https://www.youtube.com/watch?v=ju26rvzfFg4>.
Closed epidemic <https://www.youtube.com/watch?v=sSLfrSSmJZM>.

for the flow of hosts between **Susceptible**, **Infectious**, and **Recovered** compartments. In modern notation, their simplest set of equations is (Fig. 2.1):

$$\frac{dS}{dt} = \mu(N - S) - \beta I \frac{S}{N} \quad (2.1)$$

$$\frac{dI}{dt} = \beta I \frac{S}{N} - (\mu + \gamma)I \quad (2.2)$$

$$\frac{dR}{dt} = \gamma I - \mu R. \quad (2.3)$$

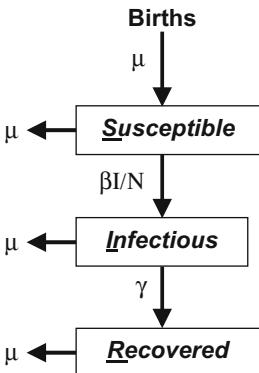


Fig. 2.1 The SIR flow diagram. Flows represent *per capita* flows from the donor compartments

The assumptions of Eqs. (2.1)–(2.3) are:

- The infection circulates in a population of size N , with a per capita “background” death rate, μ , which is balanced by a birth rate μN . From the sum of Eqs. (2.1)–(2.3), $dN/dt = 0$ and $N = S + I + R$ is thus constant.
- The infection causes acute morbidity (not mortality); That is, in this version of the SIR model we assume we can ignore disease-induced mortality. This is reasonable for certain infections like chickenpox, but certainly not for others like rabies, SARS, or ebola.
- Individuals are recruited directly into the susceptible class at birth (so we ignore perinatal maternal immunity).
- Transmission of infection from infectious to susceptible individuals is controlled by a bilinear contact term $\beta I \frac{S}{N}$. This stems from the assumption that the I infectious individuals are independently and randomly mixing with all other individuals, so the fraction S/N of the encounters is with susceptible individuals; β is the contact rate times the probability of transmission given a contact between a susceptible and an infectious individual.

- Chances of recovery or death is assumed not to change during the course of infection.
- Infectiousness is assumed not to change during the course of infection.
- Infected individuals move directly into the the infectious class (as opposed to the SEIR model; see Sect. 3.7) and remains there for an average infectious period of $1/\gamma$ (assuming $\mu \ll \gamma$).¹
- The model assumes that recovered individuals are immune from reinfection for life.

The basic reproductive ratio (R_0), defined as the expected number of secondary infections from a single index case in a completely susceptible population, is a very important quantity in epidemiology. Chapter 3 is entirely devoted to this quantity. For this simple SIR model $R_0 = \frac{\beta}{\gamma+\mu}$.

2.2 Numerical Integration of the SIR Model

If there are no (or negligible) births and deaths during the duration of an epidemic ($\mu \simeq 0$), it is commonly referred to as a *closed epidemic*. While it is occasionally possible to derive analytical solutions to systems of ODEs like Eqs. (2.1)–(2.3), we generally have to resort to numerical integration to predict the numbers over time. We use the `deSolve` R-package to numerically integrate the equations. We will numerically integrate a variety of different models. While the models differ, the basic recipe is generally the same: (1) define a R-function for the general system of equations, (2) specify the time points at which we want the integrator to save the state of the system, (3) provide values for the parameters, (4) give initial values for all state variables, and finally (5) invoke the R-function that does the integration. We use the `ode`-function in the `deSolve`-package.

¹ The implicit assumptions that stem from the use of deterministic, ordinary differential equation (ODE) are that the infectious periods (and resident times in all compartments) are exponentially distributed. This is a tractable approximation for exploring overall dynamics, but observed duration of infection periods is often much less variable—the *Eimeria*-gut parasite (a relative of *Plasmodium* that cause malaria) undergoes exactly 8 replication cycles before leaving a host; or much more variable—see superspread MOOC video: <https://www.youtube.com/watch?v=3H1tG4uz9uk>. Section 2.7 discusses a practical approach to model dynamics when the exponential assumption is deemed too simplistic.

```
require(deSolve)
```

Step 1: We define the function (often called the gradient-functions) for the equation systems. The deSolve-package requires the function to take the following parameters: time,² t, a vector with the values for the state variables (S, I, R), y, and parameter values (β, μ, γ , and N), parms:

```
sirmod = function(t, y, parms) {
  # Pull state variables from y vector
  S = y[1]
  I = y[2]
  R = y[3]
  # Pull parameter values from parms vector
  beta = parms["beta"]
  mu = parms["mu"]
  gamma = parms["gamma"]
  N = parms["N"]
  # Define equations
  dS = mu * (N - S) - beta * S * I/N
  dI = beta * S * I/N - (mu + gamma) * I
  dR = gamma * I - mu * R
  res = c(dS, dI, dR)
  # Return list of gradients
  list(res)
}
```

The `ode`-function solves differential equations numerically.

Steps 2–4: Specify the time points at which we want `ode` to record the states of the system (here we use 26 weeks with 10 time-increments per week as specified in the vector `times`), the parameter values (in this case as specified in the vector `parms`), and starting conditions (specified in `start`). In this case we model the *fraction* of individuals in each class, so we set $N = 1$, and consider a disease with an infectious period of 2 weeks ($\gamma = 1/2$), no births or deaths ($\mu = 0$) and a transmission rate of 2 ($\beta = 2$). For our starting conditions we assume that 0.1% of the initial population is infected and the remaining fraction is susceptible.

```
times = seq(0, 26, by = 1/10)
parms = c(mu = 0, N = 1, beta = 2, gamma = 1/2)
start = c(S = 0.999, I = 0.001, R = 0)
```

² Though, in the case of the simple SIR model there is no time-dependence in any of the parameters, so this parameter is not called within the gradient function; This will change when we consider seasonality (Chap. 5).

Step 5: Feed `start`, `times`, the gradient-function and parameter vector to the `ode`-function as suggested by `args(ode)`.³ For convenience we convert the output to a data frame (`ode` returns a list). The `head`-function shows the first 5 rows of `out`, and `round(, 3)` rounds the number to three decimals.

```
out=ode(y=start, times=times, func=sirmod, parms=
         parms)
out=as.data.frame(out)
head(round(out, 3))

##    time      S      I   R
## 1  0.0 0.999 0.001 0
## 2  0.1 0.999 0.001 0
## 3  0.2 0.999 0.001 0
## 4  0.3 0.998 0.002 0
## 5  0.4 0.998 0.002 0
## 6  0.5 0.998 0.002 0
```

We can plot the result (Fig. 2.2) to see that the model predicts an initial exponential growth of the epidemic that decelerates as susceptibles are depleted, and finally fade-out as susceptible numbers are too low to sustain the chain of transmission.

```
plot(x=out$time, y=out$S, ylab="Fraction", xlab=
      "Time", type="l")
lines(x=out$time, y=out$I, col="red")
lines(x=out$time, y=out$R, col="green")
```

R allows for a lot of customization of graphics—Rseek.org is a useful resource to find solutions to all things R... Fig. 2.2 has some added features such as a right-hand axis for the *effective* reproductive ratio (R_E)—the expected number of new cases per infected individuals in a *not* completely susceptible population—and a legend so that we can confirm that the turnover of the epidemic happens exactly when $R_E = R_0 s = 1$, where s is the fraction of remaining susceptibles. The threshold $R_0 s = 1 \Rightarrow s^* = 1/R_0$ results in the powerful rule of thumb for vaccine induced eradication and herd immunity: If we can—through vaccination —keep the susceptible population below a critical fraction, $p_c = 1 - 1/R_0$, then pathogen spread will dissipate and the pathogen will not be able to reinvoke the host population (e.g., Anderson and May 1982; Roberts and Heesterbeek 1993; Ferguson et al. 2003). This rule of thumb appeared to work well for smallpox, the only vaccine-eradicated human disease; Its R_0 was commonly around 5, and most countries saw elimination once vaccine cover exceeded 80% (Anderson and May 1982). The actual code used to produce Fig. 2.2 is:

³ For further details on usage, do `?function` on the R command-line, i.e., `?ode` in this instance.

```

#Calculate R0
R0=parms ["beta"] / (parms ["gamma"] +parms ["mu"])

#Adjust margins to accommodate a second right axis
par(mar = c(5,5,2,5))
#Plot state variables
plot(x=out$time, y=out$S, ylab="Fraction", xlab="Time",
      type="l")
lines(x=out$time, y=out$I, col="red")
lines(x=out$time, y=out$R, col="green")
#Add vertical line at turnover point
xx=out$time[which.max(out$I)]
lines(c(xx,xx), c(1/R0,max(out$I)), lty=3)

#prepare to superimpose 2nd plot
par(new=TRUE)
#plot effective reproductive ratio (w/o axes)
plot(x=out$time, y=R0*out$S, type="l", lty=2, lwd=2,
      col="black", axes=FALSE, xlab=NA, ylab=NA,
      ylim=c(-.5, 4.5))
lines(c(xx, 26), c(1,1), lty=3)
#Add right-hand axis for RE
axis(side = 4)
mtext(side = 4, line = 4, expression(R[E]))
#Add legend
legend("right", legend=c("S", "I", "R",
                         expression(R[E])), lty=c(1,1,1, 2),
                         col=c("black", "red", "green", "black"))

```

2.3 Final Epidemic Size

The closed epidemic model has two equilibria $\{S = 1, I = 0, R = 0\}$ which is unstable when $R_0 > 1$, and the $\{S^*, I^*, R^*\}$ -equilibrium which reflects the *final epidemic size*, for which $I^* = 0$ as the epidemic eventually self-extinguish in the absence of susceptible recruitment; S^* is the fraction of susceptibles that escape infection altogether; and R^* is the *final epidemic size*—the fraction of susceptibles that will be infected before the epidemic self-extinguish. For the closed epidemic, there is an exact mathematical solution to the final epidemic size (below). It is nevertheless useful to consider computational ways of finding equilibria in the absence of exact solutions.

The `rootSolve`-package will attempt to find equilibria of systems of differential equations through numerical integration. The function `runsteady` is really just a wrapper function around the `ode`-function that integrates until the system settles on some steady-state (if it exists). It takes the same arguments as `ode`. By varying initial conditions `rootSolve` should find multiple *stable* equilibria if there are more than one stable solution.⁴

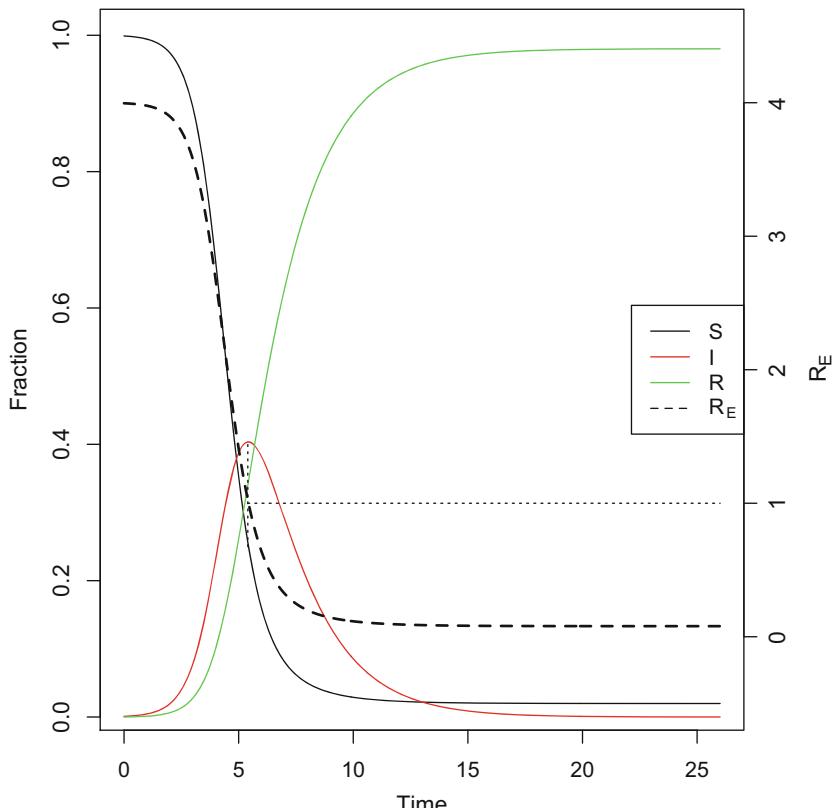


Fig. 2.2 The closed SIR epidemic with left and right axes and effective reproductive ratio, R_E . The epidemic turns over at $R_E = 1$

```
require(rootSolve)
equil=runsteady(y=c(S=1-1E-5, I=1E-5, R=0),
times=c(0,1E5), func=sirmod, parms=parms)
round(equil$y, 3)
```

⁴ It will not find unstable equilibria, for these we will need to use other strategies. We will consider finding all equilibria in more depth in Sect. 9.3.

```
##      S      I      R
## 0.02 0.00 0.98
```

So for these parameters, 2% of susceptibles are expected to escape infection altogether and 98%—the final epidemic size—are expected to be infected during the course of the epidemic.

Let us explore numerically how the final epidemic size depends on R_0 . Recall that for the specific SIR variant we are working with $R_0 = \beta/(\gamma + \mu)$, and since we are studying the closed epidemic $\mu = 0$. In the above example we assume an infectious period of 2 weeks (i.e., $\gamma = 1/2$), so we may vary β so R_0 goes from 0.1 to 5. For moderate to large R_0 this fraction has been shown to be approximately $1 - \exp(-R_0)$ (e.g., Anderson and May 1982). We can check how well this approximation holds (Fig. 2.3).⁵

```
#Candidate values for R0 and beta
R0 = seq(0.1, 5, length=50)
betas= R0 * 1/2
#Vector of NAs to be filled with numbers
f = rep(NA, 50)
#Loop over i from 1, 2, ..., 50
for(i in seq(from=1, to=50, by=1)){
  equil=runsteady(y=c(S=1-1E-5, I=1E-5,
    R=0), times=c(0,1E5), func=sirmod,
    parms=c(mu=0, N=1, beta=betas[i], gamma=1/2))
  f[i]=equil$y["R"]
}
plot(R0, f, type="l", xlab=expression(R[0]))
curve(1-exp(-x), from=1, to=5, add=TRUE, col="red")
```

We see that the approximation is good for $R_0 > 2.5$ but overestimates the final epidemic size for smaller R_0 (and is terrible for $R_0 < 1$).

For the closed epidemic SIR model, there is an exact mathematical solution to the fraction of susceptibles that escapes infection ($1 - f$) given by the implicit equation $f = \exp(-R_0(1 - f))$ or equivalently $\exp(-R_0(1 - f)) - f = 0$ (Swinton 1998). So we can also find the final size by applying the `uniroot`-function to the equation. The `uniroot`-function finds numerical solutions to equations with one unknown variable (which has to be named `x`).

```
#Define function
fn=function(x, R0){
  exp(-(R0*(1-x))) - x
```

⁵ We use a `for`-loop here to calculate the final epidemic size for a range of values of R_0 ; A loop works by repeating calculations (in this case 50 times), after each repeat the value of the looping variable (in this case `i`) is changed to the next value in the looping vector. So in this example `i` will be 1 first, then 2, then ... until the loop ends after `i=50`.

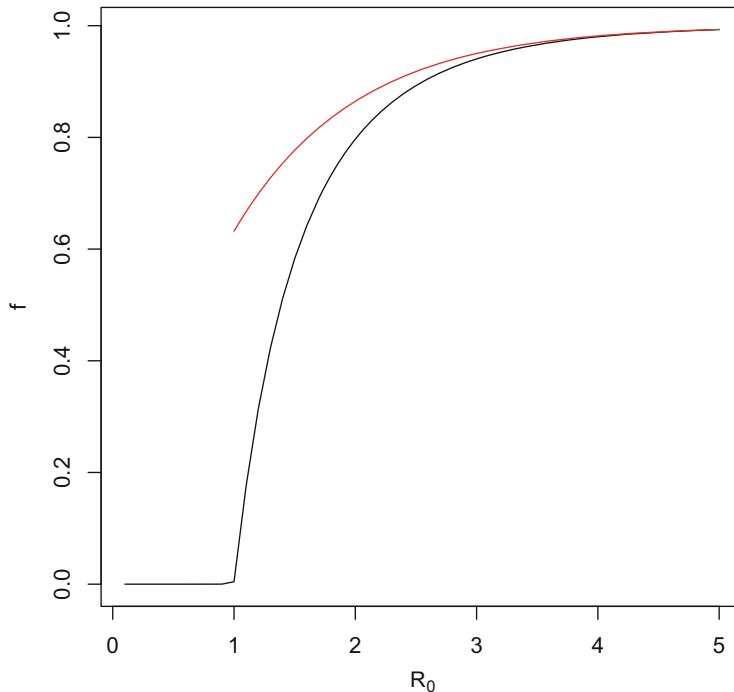


Fig. 2.3 The final epidemic size as a function of R_0 . The black line is the solution based on numerically integrating the closed epidemic, and the red line is the approximation $f \simeq 1 - \exp(-R_0)$

```

}
1-uniroot(fn, lower = 0, upper = 1-1E-9,
           tol = 1e-9, R0=2)$root
## [1] 0.7968121

#check accuracy of approximation:
exp(-2)-uniroot(fn, lower = 0, upper = 1-1E-9,
                  tol = 1e-9, R0=2)$root
## [1] -0.06785259

```

So for $R_0 = 2$ the final epidemic size is 79.6% and the approximation is off by around 6.7% points.

2.4 Open Epidemic

The *open epidemic* has recruitment of new susceptibles (i.e., $\mu > 0$). As long as $R_0 > 1$, the open epidemic has an “endemic equilibrium” where the pathogen and host coexist. If we use the SIR equations to model fractions (i.e., set $N = 1$), Eq. (2.2) of the SIR model implies that $S^* = (\gamma + \mu)/\beta = 1/R_0$ is the endemic S -equilibrium, which when substituted into Eq. (2.1) gives $I^* = \mu(R_0 - 1)/\beta$, and finally, $R^* = N - I^* - S^*$ as the I and R endemic equilibria. We can study the predicted dynamics of the open epidemic using the `sirmod`-function. Let us assume a life expectancy of 50 years, a stable population size, and thus a weekly birth rate of $\mu = 1/(50 * 52)$. Let’s assume that 19% of the initial population is susceptible and 1% is infected and numerically integrate the model for 50 years (Fig. 2.4).

```
times  = seq(0, 52*50, by=.1)
parms  = c(mu = 1/(50*52), N = 1, beta = 2,
          gamma = 1/2)
start = c(S=0.19, I=0.01, R = 0.8)
out   = as.data.frame(ode(y=start, times=times,
                           func=sirmod, parms=parms))
par(mfrow=c(1,2)) #Make room for side-by-side plots
plot(times, out$I, ylab="Fraction", xlab="Time",
     type="l")
plot(out$S, out$I, type="l", xlab="Susceptible",
     ylab="Infected")
```

2.5 Phase Analyses

When working with dynamical systems we are often interested in studying the dynamics in the phase plane and derive the isolines that divide this plane in regions of increase and decrease of the various state variables. The `phaseR` package is a wrapper around `ode` that makes it easy to analyze 1D and 2D ode’s.⁶ The R-state in the SIR model does not influence the dynamics, so we can rewrite the SIR model as a 2D system.

```
simod = function(t, y, parameters) {
  S = y[1]
  I = y[2]
```

⁶ The `phaseR` package requires the `gradient` function to take the arguments `t`, `y`, and `parameters`.

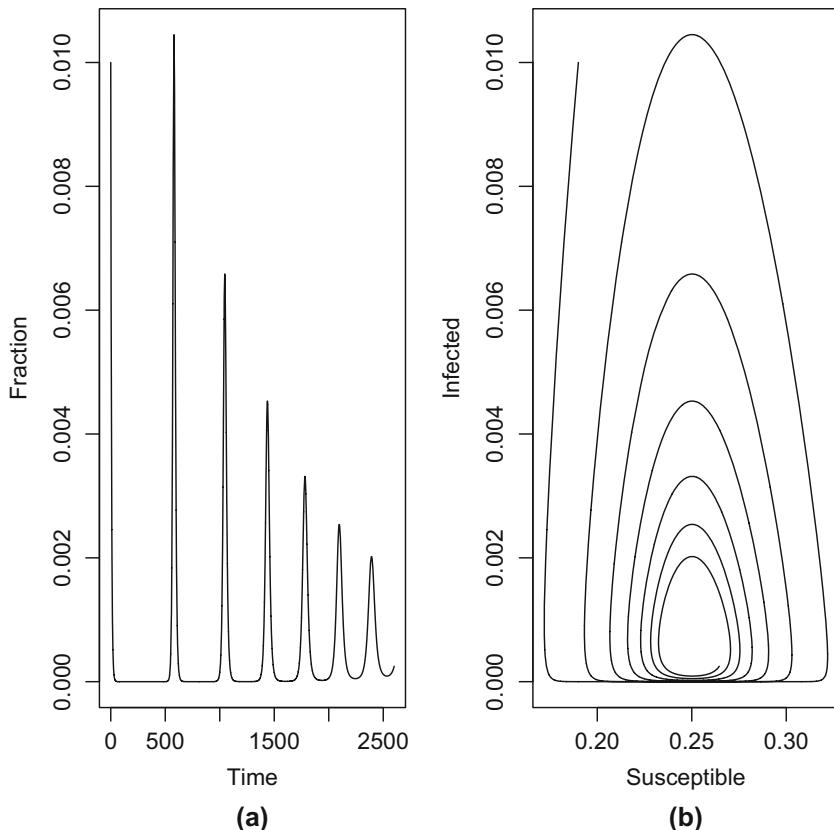


Fig. 2.4 The open SIR epidemic. **(a)** The fraction infected over time. **(b)** The joint time series of infecteds and susceptibles in the S-I phase plane. The trajectory forms a counter-clockwise inwards spiral in the S-I plane (note that the 50-year simulation is not long enough for the system to reach the steady-state endemic equilibrium at the center of the spiral)

```

beta = parameters["beta"]
mu = parameters["mu"]
gamma = parameters["gamma"]
N = parameters["N"]

dS = mu * (N - S) - beta * S * I/N
dI = beta * S * I/N - (mu + gamma) * I
res = c(dS, dI)
list(res)
}

```

The isoclines (sometimes called the nullclines) in this system are given by the solution to the equations $dS/dt = 0$ and $dI/dt = 0$ and partitions the phase plane into regions where S and I are increasing and decreasing. For $N = 1$, the I -isocline is $S = (\gamma + \mu)/\beta = 1/R_0$ and the S -isocline is $I = \mu(1/S - 1)/\beta$. We can draw these in the phase plane and add a simulated trajectory to the plot (Fig. 2.5). The trajectory cycles in a counter-clockwise damped fashion towards the endemic equilibrium (Fig. 2.5). To visualize the expected change to the system at arbitrary points in the phase plane, we can further use the function `flowField` in the `phaseR`-package to superimpose predicted arrows of change (vectors).

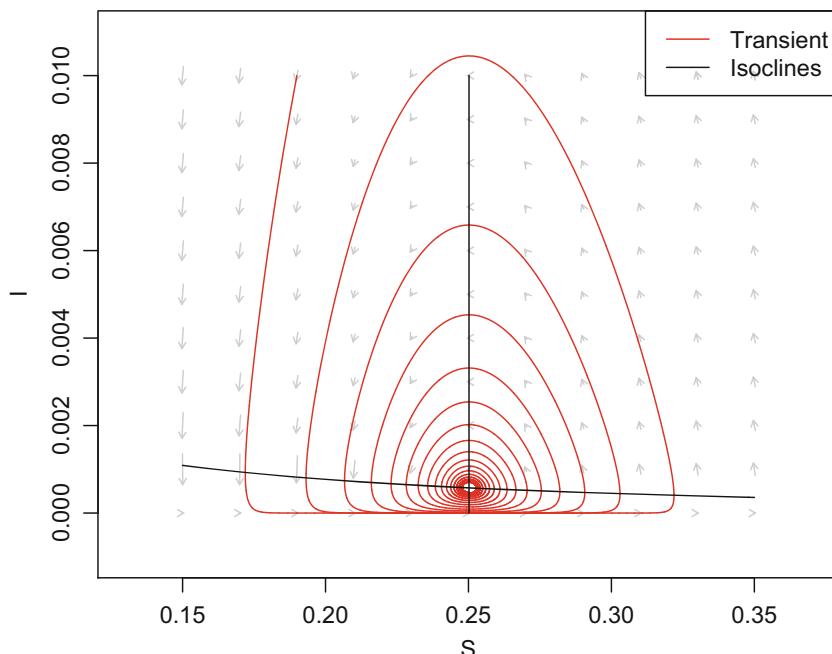


Fig. 2.5 The S-I phase plane with isoclines and the predicted anti-clockwise trajectory towards the equilibrium

```
require(phaseR)
#Plot vector field
fld=flowField(simod, x.lim=c(0.15,0.35), y.lim=c(0,.01),
  parameters=parms, system="two.dim", add=FALSE,
  ylab="I", xlab="S")
#Add trajectory
out = as.data.frame(ode(y = c(S=0.19, I=0.01), times=
  seq(0, 52*100, by=.1), func=simod, parms=parms))
lines(out$S, out$I, col="red")
```

```
#Add S-isocline
curve(parms["mu"]*(1/x-1)/parms["beta"], 0.15, 0.35,
      xlab="S", ylab="I", add=TRUE)
#Add I-isocline
shat=(parms["gamma"]+parms["mu"])/parms["beta"]
lines(rep(shat, 2), c(0,0.01))

legend("topright", legend=c("Transient", "Isoclines"),
       lty=c(1, 1), col=c("red", "black"))
```

2.6 Stability and Periodicity

If we work with continuous-time ODE models like the SIR, equilibria are locally stable if (and only if) all the real part of the eigenvalues of the [Jacobian matrix](#)—when evaluated at the equilibrium—are smaller than zero. We will discuss stability and resonant periodicity in detail in Chap. 9, so this section is just a teaser... An equilibrium is (1) a node (i.e., all trajectories moves monotonically towards/away from the equilibrium) if the largest eigenvalue has only real parts, or (2) a focus (trajectories spiral towards or away from the equilibrium) if the largest eigenvalues are a conjugate pair of complex numbers ($a \pm bi$).⁷ For a focus the imaginary part determines the dampening period of the cycle according to $2\pi/b$. We can thus use the Jacobian matrix to study the SIR model's equilibria. If we let $F = dS/dt = \mu(N - S) - \beta SI/N$ and $G = dI/dt = \beta SI/N - (\mu + \gamma)I$, the Jacobian of the SIR system is

$$J = \begin{pmatrix} \frac{\partial F}{\partial S} & \frac{\partial F}{\partial I} \\ \frac{\partial G}{\partial S} & \frac{\partial G}{\partial I} \end{pmatrix}, \quad (2.4)$$

and the two equilibria are the disease-free equilibrium and the endemic equilibrium as defined above.

R can help with all of this. We first calculate the equilibria:

```
# Pull values from parms vector
gamma = parms["gamma"]
beta = parms["beta"]
mu = parms["mu"]
N = parms["N"]

# Endemic equilibrium
```

⁷ And a center—like the [Lotka-Volterra predator-prey](#) model—if conjugate pair only has imaginary parts.

```
Sstar = (gamma + mu) /beta
Istar = mu * (beta/(gamma + mu) - 1) /beta
eq1 = list(S = Sstar, I = Istar)
```

We then calculate the elements of the Jacobian using R's D-function:

```
# Define equations
dS = expression(mu * (N - S) - beta * S * I/N)
dI = expression(beta * S * I/N - (mu + gamma) * I)
# Differentiate w.r.t. S and I
j11 = D(dS, "S")
j12 = D(dS, "I")
j21 = D(dI, "S")
j22 = D(dI, "I")
```

We pass the values for S^* and I^* in the eq1-list to the Jacobian,⁸ and use eigen-function to calculate the eigenvalues:

```
#Evaluate Jacobian at equilibrium
J=with(data=eq1, expr=matrix(c(eval(j11),eval(j12),
    eval(j21),eval(j22)), nrow=2, byrow=TRUE))
#Calculate eigenvalues
eigen(J)$values
```

```
## [1] -0.00076864+0.02400384i -0.00076864-0.02400384i
```

For the endemic equilibrium, the eigenvalues are a pair of complex conjugates which real parts are negative, so it is a stable focus. The period of the inwards spiral is:

```
2 * pi/(Im(eigen(J)$values[1]))
```

```
## [1] 261.7575
```

So with these parameters the dampening period is predicted to be 261 weeks (just over 5 years). Thus, during disease invasion we expect this system to exhibit initial outbreaks every 5 years. A further significance of this number is that if the system is stochastically perturbed by, say, environmental variability affecting transmission, we expect the system to exhibit low amplitude “phase-forgetting” cycles (Nisbet

⁸ In previous coding like for the sirmod-function, we “pulled” parameter values from the input arguments inside the function to make the code as transparent as possible; while it makes the code easy to read, it makes for extra coding, and can clutter up the workspace with variables that are defined in multiple locations. The with-function allows the evaluation of an expression using variables defined in a data list.

and Gurney 1982) with approximately this period in the long run (see Chap. 9). We can make more accurate calculations of the stochastic system using *transfer functions* (Nisbet and Gurney 1982; Priestley 1981). We will visit on this slightly more advanced topic in Sect. 9.7.

The same protocol can be used on the disease-free equilibrium $\{S^* = 1, I^* = 0\}$.

```
eq2=list(S=1, I=0)
J=with(eq2,
       matrix(c(eval(j11), eval(j12), eval(j21),
                 eval(j22)), nrow=2, byrow=TRUE))
eigen(J)$values

## [1] 1.4996153846 -0.0003846154
```

The eigenvalues are strictly real and the largest value is greater than zero, so it is an unstable node (a “saddle”); The epidemic trajectory is predicted to move monotonically away from this disease free equilibrium if infection is introduced into the system. This makes sense because with the parameter values used, $R_0 = 3.99$ which is greater than the invasion-threshold value of 1.

2.7 Advanced: More Realistic Infectious Periods

The S(E)IR-type differential equation models assumes that rate of exit from the infectious classes are constant, the implicit assumption is thus that the infectious period is exponentially distributed among infected individuals; The average infectious period is $1/(\gamma + \mu)$, but an exponential fraction is infectious much shorter/longer than this. The chain-binomial model (see Sect. 3.4), in contrast, assumes that everybody is infectious for a fixed period and then all instantaneously recover (or die). These assumptions are mathematically convenient, but in reality neither are particularly realistic. Hope-Simpson (1952) traced the chains of transmission of measles in multi-sibling household. The timing of secondary and tertiary cases was analyzed in detail by Bailey (1956) and Bailey and Alff-Steinberger (1970). The average latent and infectious periods were calculated to be 8.58 and 6.57 days, respectively. While the distribution around each of these averages were not estimated separately (the latent period was assumed to be distributed and the infectious period assumed fixed), the variance around the roughly fortnight period of infection was estimated to be 3.13. The mean duration of infection is thus 15.15 days with a standard deviation of 1.77 (Fig. 2.6). So neither a fixed nor an exponential distribution is very accurate (Keeling and Grenfell 1997; Lloyd 2001).

Kermack and McKendrick’s (1927) original model allows for arbitrary infectious-period distributions. We can write Kermack and McKendrick’s origi-

nal equations as “renewal equations” (Breda et al. 2012), introducing the additional notation of $k(t)$ being the (instantaneous) *incidence* at time t (i.e., flux into the I -class at time t).

$$\frac{dS}{dt} = \mu(N - S) - k(t) \quad (2.5)$$

$$k(t) = \beta I \frac{S}{N} \quad (2.6)$$

$$\frac{dI}{dt} = k(t) - \mu I - \int_0^\infty \frac{h(\tau)}{1 - H(\tau)} k(t - \tau) d\tau \quad (2.7)$$

$$\frac{dR}{dt} = \int_0^\infty \frac{h(\tau)}{1 - H(\tau)} k(t - \tau) d\tau - \mu R, \quad (2.8)$$

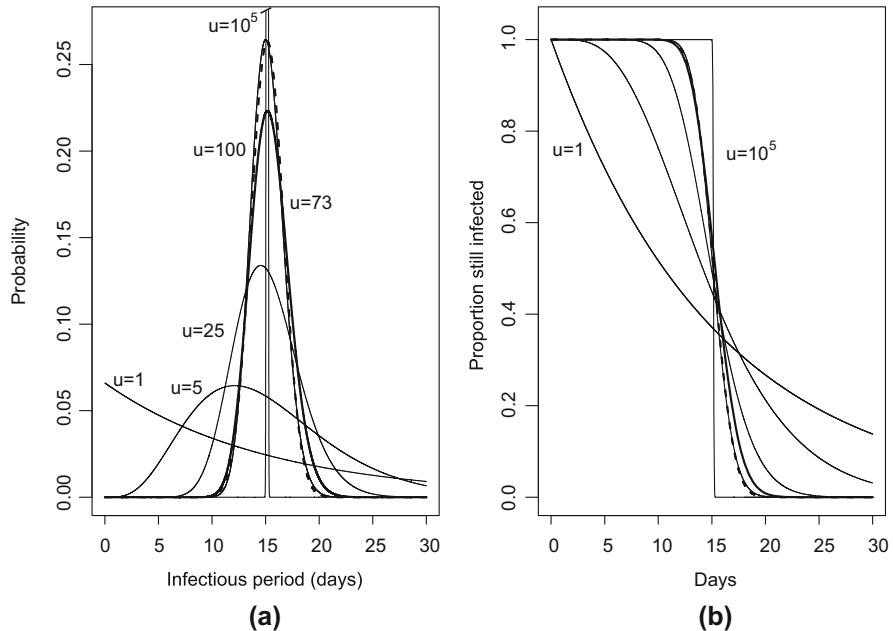


Fig. 2.6 Gamma distributed infectious periods: (a) The predicted infectious period distribution based on a Gamma distribution with shape $u = 1, 5, 25, 100$, and $100,000$; $u = 1$ corresponds to the exponential distribution implicit in the standard SIR model; the bold line ($u = 73$) is the one corresponding to the variance observed in Hope-Simpson’s (1952) study of measles. The dotted line (virtually indistinguishable from the $u = 100$) is a Gaussian distribution intended to show that when u is large the Gamma distribution converges on the Gaussian. (b) The probability of still being infectious as a function of time for the different distributions; as u becomes large, the distribution converges on a fixed infectious period. Note that the empirical distribution (bold) is quite different from the exponential

where $k(t - \tau)$ is the number of individuals that was infected τ time units ago, $h(\tau)$ is the probability of recovering on infection-day τ , and $H(\tau)$ is the cumulative probability of having recovered by infection-day τ ; $k(t - \tau)/(1 - H(\tau))$ is thus the fraction of individuals infected at time $t - \tau$ that still remains in the infected class on day t and the integral is over all previous infections so as to quantify the total flux into the removed class at time t . Though intuitive, these general integro-differential equations (Eqs. (2.5)–(2.8)) are not easy to work with in general. For a restricted set of distributions for the $h()$ -function, however—the [Erlang distribution](#) (the Gamma distribution with an integer shape parameter)—the model can be numerically integrated using a “Gamma-chain” model (referred to as “linear chain trickery” by Metz and Diekmann 1991) of coupled ordinary differential equations (e.g., Blythe et al. 1984; de Valpine et al. 2014; Bjørnstad et al. 2016). The trick is to separate any distributed-delay compartment into u sub-compartments through which individuals pass through at a rate overallrate $* u$. The resultant infectious period will have a mean of 1/overallrate and a coefficient-of-variation of $1/\sqrt{u}$.

We can write a chain-SIR model to simulate $S \rightarrow I \rightarrow R$ flows with more realistic infectious period distributions⁹:

```
chainSIR=function(t, logx, params) {
  x=exp(logx)
  u=params["u"]
  S=x[1]
  I=x[2:(u+1)]
  R=x[u+2]
  with(as.list(params), {
    dS = mu * (N - S) - sum(beta * S * I) / N
    dI = rep(0, u)
    dI[1] = sum(beta * S * I) / N - (mu + u*gamma) * I[1]
    if(u>1) {
      for(i in 2:u) {
        dI[i] = u*gamma * I[i-1] - (mu+u*gamma) * I[i]
      }
    }
    dR = u*gamma * I[u] - mu * R
    res=c(dS/S, dI/I, dR/R)
    list(res)
  })
}
```

⁹ With high number of compartments this system of equations can become “stiff” with the computer potentially making rounding errors leading to erroneous negative numbers. We use a “log-trick” (Ellner and Guckenheimer 2011) available for systems where all state variables are strictly positive: we solve the system in log-coordinates to smooth abrupt changes and force all number to be greater than zero. To employ this technique we log-transform all initial values in start, change the first line in the function to `x = exp(logx)` and the last line to return `dS/S`, etc. in place of `dS` which comes from the [chain-rule](#) of differentiation and the fact that $D(\log x) = 1/x$.

We can compare the predicted dynamics of the simple SIR model with the $u = 2$ chain model, the $u = 500$ chain model (which is effectively the fixed-period delayed-differential model) and the “measles-realistic” $u = 73$ model.

```

times = seq(0, 10, by=1/52)
paras2 = c(mu = 1/75, N = 1, beta = 625,
           gamma = 365/14, u=1)
xstart2 = log(c(S=.06, I=c(0.001, rep(0.0001,
           paras2["u"]-1)), R = 0.0001))
out = as.data.frame(ode(xstart2, times, chainSIR,
           paras2))
plot(times, exp(out[,3]), ylab="Infected", xlab=
      "Time", ylim=c(0, 0.01), type='l')

paras2["u"] = 2
xstart2 = log(c(S=.06, I=c(0.001, rep(0.0001/
           paras2["u"]), paras2["u"]-1)), R = 0.0001)
out2 = as.data.frame(ode(xstart2, times, chainSIR,
           paras2))
lines(times, apply(exp(out2[,-c(1:2,length(out2))]),
      1, sum), col='blue')

paras2["u"] = 73
xstart2 = log(c(S=.06, I=c(0.001, rep(0.0001/
           paras2["u"]), paras2["u"]-1)), R = 0.0001)
out3 = as.data.frame(ode(xstart2, times, chainSIR,
           paras2))
lines(times, apply(exp(out3[,-c(1:2,length(out3))]),
      1, sum), col='red', lwd=2, lty=2)

paras2["u"] = 500
xstart2 = log(c(S=.06, I=c(0.001, rep(0.0001/
           paras2["u"]), paras2["u"]-1)), R = 0.0001)
out4 = as.data.frame(ode(xstart2, times, chainSIR,
           paras2))
lines(times, apply(exp(out4[,-c(1:2,length(out4))]),
      1, sum, na.rm=TRUE), col='green')

legend("topright", legend=c("SIR", "u=2", "u=500",
    "u=73 (H-S)", lty=c(1,1,1,2), lwd=c(1,1,1, 2),
    col=c("black", "blue", "green", "red")))

```

The more narrow the infectious-period distribution, the more punctuated the predicted epidemics. However, infectious-period narrowing—alone—cannot sustain recurrent epidemics; In the absence of stochastic or seasonal forcing epidemics will

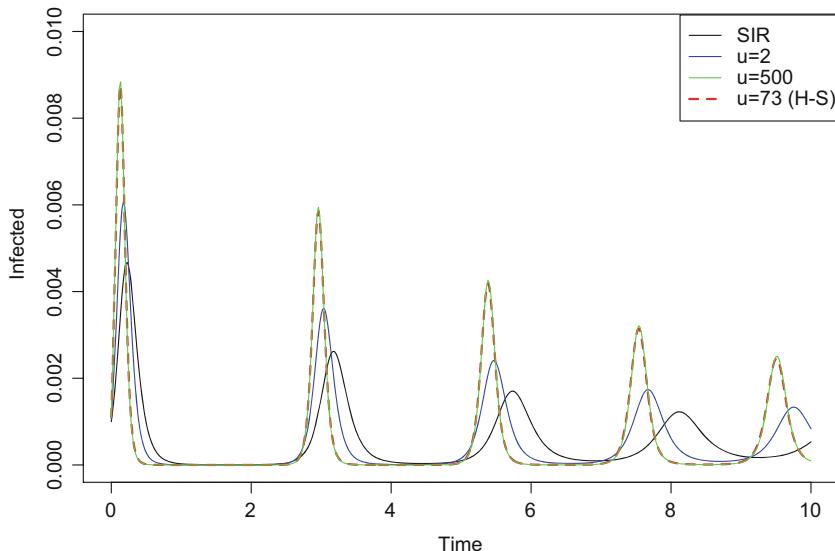


Fig. 2.7 Chain-SIR models with different infectious period distributions

dampen to the endemic equilibrium (though the damping period is slightly accelerated and the convergence on the equilibrium is slightly slower with narrowing infectious period distributions) (Fig. 2.7).

In the above we considered non-exponential infectious-period distributions. However, the general ODE chain method can be used for any compartment. Lavine et al. (2011), for example, used it to model non-exponential waning of natural and vaccine-induced immunity to whooping cough.

2.8 ShinyApp

The following code will launch a local shinyApp of the SIR model in your local browser. This App can also be launched by calling `SIR.app` in the `epimdr`-package. Several of the subsequent chapters also have associated shinyApps. Those will only be accessible from the package (because the code is long and a bit tedious). We quote an annotated version of the `SIR.app` in full.

```
require(shiny)
require(deSolve)
require(phaseR)

#This creates the User Interface (UI)
ui = pageWithSidebar(
```

```

#The title
headerPanel("The SIR model"),
#The sidebar for parameter input
sidebarPanel(
  #Sliders:
  sliderInput("beta", "Transmission (yr^-1):", 300,
              min = 0, max = 1000),
  sliderInput("infper", "Infectious period (days)", 5,
              min = 1, max = 100),
  sliderInput("mu", "birth rate:", 5,
              min = 0, max = 100),
  sliderInput("T", "Time range:",
              min = 0, max = 1, value = c(0,1))
),
#Main panel for figures and equations
mainPanel(
  #Multiple tabs in main panel
  tabsetPanel(
    #Tab 1: Time plot (plot1 from server)
    tabPanel("Time", plotOutput("plot1")),
    #Tab 2: Phase plot (plot2 from server)
    tabPanel("Phase plane", plotOutput("plot2",
                                         height = 500)),
    #Tab 3: MathJax typeset equations
    tabPanel("Equations",
      withMathJax(
        helpText("Susceptible $$\\frac{ds}{dt} = "
                "\\mu (N - S) - \\frac{\\beta I S}{N}$$"),
        helpText("Infecitous $$\\frac{dI}{dt} = "
                "\\frac{\\beta I S}{N} - (\\mu + \\sigma) I$$"),
        helpText("Removed $$\\frac{dR}{dt} = "
                "\\gamma I - \\mu R$$"),
        helpText("Reproductive ratio $$R_0 = "
                "\\frac{\\gamma + \\mu}{\\beta N}$$")
      )
    )))
  #End of ui()

# This creates the 'behind the scenes' code (Server)
server = function(input, output) {
  #Gradient function for SIR model
  sirmod=function(t, x, parms){
    S=x[1]
    I=x[2]
    R=x[3]

```

```

beta=parms ["beta"]
mu=parms ["mu"]
gamma=parms ["gamma"]
N=parms ["N"]
dS = mu * (N - S) - beta * S * I / N
dI = beta * S * I / N - (mu + gamma) * I
dR = gamma * I - mu * R
res=c(dS, dI, dR)
list(res)
}

#Gradient function used for phaseR phase-plot
simod=function(t, y, parameters){
  S=y[1]
  I=y[2]
  beta=parameters["beta"]
  mu=parameters["mu"]
  gamma=parameters["gamma"]
  N=parameters["N"]
  dS = mu * (N - S) - beta * S * I / N
  dI = beta * S * I / N - (mu + gamma) * I
  res=c(dS, dI)
  list(res)
}

#Plot1: renderPlot to be passed to UI tab 1
output$plot1 = renderPlot({
  #input\xxx's are pulled from UI
  times = seq(0, input$T[2], by=1/1000)
  parms = c(mu = input$mu, N = 1, beta = input$beta,
            gamma = 365/input$infper)
  start = c(S=0.999, I=0.001, R = 0)
  R0 = round(with(as.list(parms), beta/(gamma+mu)), 1)

  #Integrate ode with parameters pulled from UI
  out=ode(y=start, times=times, func=sirmod,
           parms=parms)
  out=as.data.frame(out)

  #Plot1
  sel=out$time>input$T[1] & out$time<input$T[2]
  plot(x=out$time[sel], y=out$S[sel], ylab="fraction",
        xlab="time", type="l", ylim=range(out[sel,-c(1,4)]))
  title(paste("R0=", R0))
})

```

```

lines(x=out$time[sel], y=out$I[sel], col="red")
lines(x=out$time[sel], y=out$R[sel], col="green")
legend("right", legend=c("S", "I", "R"),
       lty=c(1,1,1), col=c("black", "red", "green"))
})

#Plot2: renderPlot to be passed to UI tab 2
output$plot2 = renderPlot({
  times = seq(0, input$T[2], by=1/1000)
  parms = c(mu = input$mu, N = 1, beta = input$beta,
            gamma = 365/input$infper)
  start = c(S=0.999, I=0.001, R = 0)
  R0 = round(with(as.list(parms), beta/(gamma+mu)), 1)

  #Integrate simod
  out=ode(y=start[-3], times=times, func=simod,
           parms=parms)
  out=as.data.frame(out)

  #Plot2
  plot(x=out$S, y=out$I, xlab="Fraction suceptible",
        ylab="Fraction infected", type="l")
  title(paste("R0=", R0))
  #Add vector field
  fld=flowField(simod, x.lim=range(out$S), y.lim=
    range(out$I), parameters=parms, system="two.dim",
    add=TRUE, ylab="I", xlab="S")
  #Add isoclines
  abline(v=1/R0, col="green")
  curve(parms["mu"]*(1-x)/(parms["beta"]*x), min(out$S),
         max(out$S), add=TRUE, col="red")
  legend("topright", legend=c("I-socline",
    "S-isocline"), lty=c(1,1), col=c("red", "green"))
})
} #End of server()

shinyApp(ui, server)

```

Chapter 3

R_0



3.1 Primacy of R_0

For directly transmitted pathogens, R_0 is, per definition, the expected number of secondary cases that arise from a typical infectious index-case in a completely susceptible host population. R_0 plays a critical role for a number of aspects of disease dynamics and is therefore the focus of much study in historical and contemporary infectious disease dynamics (Heesterbeek and Dietz 1996). For perfectly immunizing infections in homogeneously mixing populations these include (e.g., Anderson and May 1991):

- The threshold for pathogen establishment. When R_0 is greater than one, a pathogen can invade. When it is smaller than one, the chain of transmission will stutter and break (Lloyd-Smith et al. 2009). For directly transmitted wildlife diseases there is often an associated *critical host density* for disease invasion. This has for example been estimated to be 1 red fox per km² for rabies in Europe (Anderson et al. 1981) and 17 mice/ha for *Sin nombre* hantavirus in Montana (Luis et al. 2015).
- The threshold for vaccine-induced herd immunity: If a sufficient number of individuals are vaccinated, the effective reproductive ratio will be below one, and the population will be resistant to pathogen invasion. The threshold is $p_c = 1 - 1/R_0$. Thus, measles with a R_0 of up to 20 requires around 95% vaccine cover for elimination and smallpox ($R_0 \simeq 5$) 80%.

This chapter uses the following R-packages: `bbmle` and `statnet`.

A conceptual understanding of *reproductive ratios* and the *simple epidemic* is useful prior to this discussion. Five minute epidemics-MOOC intros can be watched from YouTube:

Reproductive number <https://www.youtube.com/watch?v=ju26rvzfFg4>.

Simple epidemic <https://www.youtube.com/watch?v=sSLfrSSmJZM>.

- As discussed in Sect. 2.3, the final epidemic size is given by R_0 according to the approximate relationship $f \simeq \exp(-R_0)$.
- In a stable host population, the mean age of infection is approximately $\bar{a} \simeq L/(R_0 - 1)$, where L is host life-expectancy (Dietz and Schenzle 1985). In a changing population a more accurate calculation is $\bar{a} \simeq 1/(\mu(R_0 - 1))$, where μ is the host birth rate.
- As derived in Sect. 2.5, the susceptible fraction at equilibrium is $S^* = 1/R_0$. A consequence of this is that for competing strains that elicit cross-protecting immunity, R_0 will determine competitive dominance and strain replacement (Shrestha et al. 2014).¹

A lot of attention has been given to measuring R_0 for various infectious diseases.

3.2 Preamble: Rates and Probabilities

When working with data, models, and “models-and-data” for infectious disease dynamics, it is important to keep a cool head in terms of keeping track of which quantities are **probabilities** and which quantities are **rates**, and how to move between these two mathematical currencies.² Confusion arises because the nomenclature of *epidemiology* and *mathematical epidemiology* is related but not always identical. In epidemiology the “case-fatality rate” is used to denote the fraction of infections that ends in death, which from a mathematical/statistical point of view is **not a rate** but a **probability**: the probability that an infection will lead to death (Dietz and Heesterbeek 2002). Likewise, in epidemiology, the seasonal influenza “attack rate” denotes the fraction of people that contracts the flu in a given influenza season. Again, from a mathematical/statistical/dynamical-systems point of view this quantity is **not a rate** but a **probability** representing the chance of any randomly chosen individual of unknown previous influenza infection-history getting infected during the season.

When considering events in modeling terms, a rate x per time unit is defined on $[0, \infty]$ and $1/x$ is the average time to an event (if the rate remains constant). If events are random and independent, the probability of no events in a time interval Δt is $1 - \exp(-x\Delta t)$ and the number of events in Δt follows a Poisson-distribution with mean $x\Delta t$ (if the rate remains constant). A probability, in contrast, is defined on $[0, 1]$. If we observe a probability p of something happening in a time interval, we can back-calculate the associated (constant) rate as $x = -\log(1 - p)/\Delta t$.

¹ This result is parallel to Tilman (1976)’s R^* -theory of resource-based competition of free-living organisms: whichever species can sustain positive growth at the lowest concentration of the limited resource will be competitively dominant.

² The disease dynamics literature has many examples of how easy it is to confuse the two; cf some of the mathematical models of ebola dynamics published during the 2014 West Africa outbreak.

If we have two competing rates, x at which event one (e.g., recovery) happens and y at which event two (e.g., death) happens, the probability of ending up with outcome one is $x/(x+y)$ and the probability of ending up with outcome two is $y/(x+y)$. This scales such that with three competing rates the probability of outcome one is $x/(x+y+z)$.

3.3 Estimating R_0 from a Simple Epidemic

A variety of methods have been proposed to estimate R_0 (or the effective reproductive ratio, R_E ³) in an epidemic setting (such as the 2014–15 West African ebola outbreak; Althaus 2014). Some are purely model based, others involve very elaborate model fitting exercises, and some use fairly simple ideas based on the closed epidemic and analogies to the ecology of free-living organisms (Dietz 1993).

The simplest idea is that during the initial spread phase susceptible depletion may be sufficiently negligible that the epidemic may be assumed to grow in a density-independent, exponential fashion. Basic ecology of free-living organisms tells us that the rate of exponential growth is $r = \log(R_0)/G$, where G is the generation time; thus $R_0 = \exp(rG)$.⁴ Moreover, since an exponentially growing population grows according to $N(t) = N(0)\exp(rt)$, the time for a population to double is $\log(2)/r$. We can apply these ideas to the early phase of an epidemic to get a rough value for R_0 .

For pathogens, the N s above would represent the *prevalence*. The G represents the *serial interval* (V) which is the average time between infection and reinfection. This interval will normally be a little shorter than the latent plus infectious period. Disease data, however, most often represents *incidence*—i.e., the number of new infections, not the number of current infections. However, incidence also grows at the same exponential rate. The simplest way to estimate R_0 is thus to regress $\log(\text{cumulative incidence})$ on time to estimate the rate of exponential increase (r) and then calculate $R_0 = Vr + 1$ (e.g., Anderson and May 1991). The logic comes from the fact that in one serial interval each infected is expected to give rise to R_0 secondary cases and one removal (thus the total change is $R_0 - 1$).

Let us explore using weekly measles-data from the 2003 outbreak in Niamey, Niger (Grais et al. 2008). The data is available as `niamey` in the `epimdr`-package. The `tot_cases`-column represents the total incidence across the city for each week of the outbreak.⁵

³ The effective reproductive ratio is the expected number of secondary cases in a partially immune population $R_E = sR_0$, where s is the fraction of the population that is susceptible.

⁴ Unless explicitly stated otherwise, `log` will always be taken to mean the natural logarithm in this text.

⁵ All data sets analyzed in this text are included in the `epimdr`-package. To get a more detailed description of each data set consult the R help-pages.

```
data(niamey)
head(niamey[, 1:5])

##   absweek week tot_cases tot_mort lethality
## 1       1    45      11       0 0.000000
## 2       2    46      12       1 8.333333
## 3       3    47      15       0 0.000000
## 4       4    48      14       1 7.142857
## 5       5    49      30       0 0.000000
## 6       6    50      41       1 2.439024
```

We can do a visual inspection to identify the initial period of exponential growth:

```
par(mar = c(5,5,2,5))
plot(niamey$absweek, niamey$tot_cases, type="b",
     xlab="Week", ylab="Incidence")
par(new=T)
plot(niamey$absweek, niamey$cum_cases, type="l",
     col="red", axes=FALSE, xlab=NA, ylab=NA, log="y")
axis(side = 4)
mtext(side = 4, line = 4, "Cumulative incidence")
legend("topleft", legend=c("Cases", "Cumulative"),
       lty=c(1,1), pch=c(1,NA), col=c("black", "red"))
```

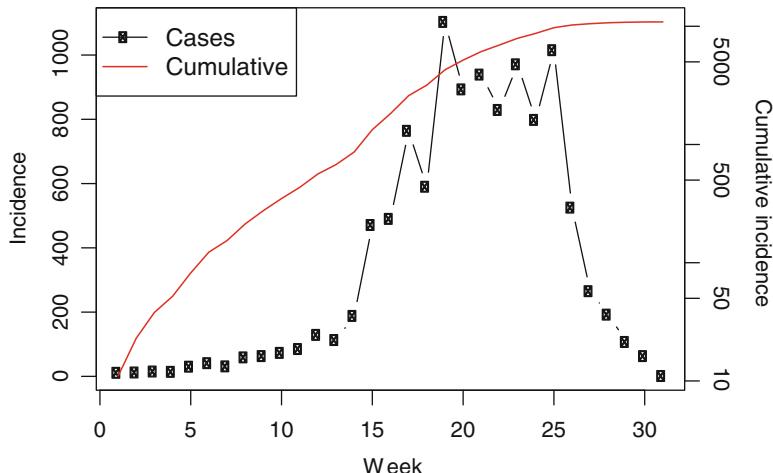


Fig. 3.1 Weekly incidence of measles in Niamey, Niger during the 2003–2004 outbreak

The cumulative incidence looks pretty log-linear for the first 6 weeks or so (Fig. 3.1). The data is weekly and the serial interval for measles is around 10–12 days, thus V is around 1.5–1.8 weeks; We calculate R_0 assuming either 1.5 or 1.8:

```
fit=lm(log(cum_cases) ~ absweek, subset=absweek<7,
       data=niamey)
r=fit$coef["absweek"]
V=c(1.5, 1.8)
V*r+1

## [1] 1.694233 1.833080
```

So a fast-and-furious estimate of the reproductive ratio for this outbreak places it in the 1.5–2 range. Measles exhibits recurrent epidemics in the presence of various vaccination campaigns in Niger, so this number represents an estimate of the *effective* reproductive ratio, R_E , at the beginning of this epidemic.

In their analysis of the SARS epidemics, Lipsitch et al. (2003) showed that for an infection with distinct latent and infectious periods a more refined estimate is given by $R = Vr + 1 + f(1 - f)(Vr)^2$, where f is the ratio of infectious period to serial interval. For measles the infectious period is around 5 days:

```
V = c(1.5, 1.8)
f = (5/7)/V
V * r + 1 + f * (1 - f) * (V * r)^2

## [1] 1.814450 1.999198
```

Lipsitch et al.'s (2003) refined calculations thus produce slightly higher estimates of R_E in the range of 1.8–2. These simple methods based on initial growth are very handy because they are simple. However, they only use a portion of the data, and as pointed out by King et al. (2015a) it may be desirable to carry out more rigorous estimation.

3.4 Maximum Likelihood: The Chain-Binomial Model

Ferrari et al. (2005) proposed a maximum likelihood “removal” method for estimating R_0 for the simple epidemic based on the so-called “chain-binomial” model of infectious disease dynamics. The chain-binomial model, originally proposed by Bailey (1957), is a discrete-time, stochastic alternative⁶ to the continuous-time, deterministic SIR model introduced in Chap. 2.

⁶ This model also forms the foundation for the TSIR model (Bjørnstad et al. 2002a; Grenfell et al. 2002) which is the focus of Chap. 7.

In contrast to the S(E)IR models, the chain-binomial assumes that an epidemic is formed from a succession of discrete generations of infectious individuals in a coin-flip fashion. Just like in the SIR we assume that infectious individuals exert a force of infection on susceptibles of $\beta I/N$. In a generation, t , of duration given by the serial interval (which we use as the basic time unit). The probability that any given susceptible will escape an infectious contact will be $\exp(-\beta I/N)$. This comes from the basic result that if some event—such as contacts between a susceptible and the population of infectious individuals—is happening at rate, x , the number of events in Δt will be distributed according to a Poisson($x\Delta t$) distribution, so the probability of no events—no contacts—will be $e^{-x\Delta t}$. The converse outcome will happen with a probability $1 - \exp(-\beta I/N)$, thus if there are S_t susceptibles we expect $S_t(1 - \exp(-\beta I_t/N))$ new infecteds in generation $t + 1$. Since we assume that contacts happen at random, the stochastic chain-binomial model is:

$$I_{t+1} \sim \text{Binomial}(S_t, 1 - \exp(-\beta I_t/N)). \quad (3.1)$$

$$S_{t+1} = S_t - I_{t+1} = S_0 - \sum_{i=1}^t I_i$$

If we ignore observational error, we thus have two unknown parameters: the initial number of susceptibles, S_0 , and the transmission rate. The reproductive ratio is a composite of these two $R = S_0(1 - \exp(-\beta/N))$, which for large populations is approximately $\beta S_0/N$ because $1 - \exp(-x) \simeq x$ for $x \ll 1$. Thus β is approximately the reproductive ratio at the beginning of the epidemic, which makes sense, since infectious individuals are expected to transmit for exactly a time unit before recovering.

If we make the assumption that each epidemic generation depends only on the state of the system in the previous time step (“conditional independence”), the removal method estimates β and S_0 from a sequence of binomial likelihoods. The advantage of this method relative to the earlier methods is that we can use all the data and not just a few observations from the beginning of an epidemic.

We employ a standard recipe, for doing a “nonstandard” maximum likelihood analysis (see Bolker 2008, for an excellent discussion of this). The first step is to write a function for the likelihood. Conditional on some parameters, the function returns the negative log-likelihood of observing the data given the model. The likelihood, which is the probability of observing data given a model and some parameter values, is the working-horse of a large part of statistics. R has inbuilt `dxxxxx`-functions to calculate the likelihood for any conceivable probability distribution. The function to calculate a binomial likelihood is `dbinom`. We can thus define a likelihood-function for the chain-binomial model ⁷:

⁷ Note that the `[-x]` subsetting in R means “drop the x’th observation”; thus the `[-n]` and `[-1]` make sure that adjacent pairs of observations are aligned correctly. We use the `floor`-function for the vector of S ’s because `dbinom` requires the denominator and numerator to be integers.

```
llik.cb = function(S0, beta, I) {
  n = length(I)
  S = floor(S0 - cumsum(I[-n]))
  p = 1 - exp(-beta * (I[-n])/S0)
  L = -sum(dbinom(I[-1], S, p, log = TRUE))
  return(L)
}
```

For the real statistical analysis (below), the two parameters will be estimated simultaneously. However, in order to ease into the idea of likelihood estimation we will consider the two sequentially and visualize the likelihood by plotting it over a grid of potential values. We illustrate with the data on measles from one of the three different reporting centers in Niamey, Niger from 2003 (Grais et al. 2008). We first need to aggregate the data into 2-week intervals which is roughly the serial interval for measles. The epidemic in district 1 lasted for 30 weeks (the 31st week is a zero)⁸:

```
twoweek = rep(1:15, each = 2)
y = sapply(split(niamey$cases_1[1:30], twoweek), sum)
sum(y)
## [1] 5920
```

In district 1 there were 5920 cases during the epidemics, so S_0 needs to be at least that number. In the above parameterization $R_E \simeq \beta$, lets initially assume a candidate value of 6500 for S_0 and calculate the likelihood for each candidate value of β between 1 and 10 by 0.1 (Fig. 3.2):

```
S0cand=6500
betacand=seq(0,10, by=.1)
ll=rep(NA, length(betacand))
for(i in 1:length(betacand)) {
  ll[i]=llik.cb(S0=S0cand, beta=betacand[i], I=y)
}
plot(ll~betacand, ylab="Neg log-lik", xlab=
  expression(beta))
betacand[which.min(ll)]
## [1] 2.3
```

We follow the convention of using the negative log-likelihood in the profile. Intuitively, one may think that it would be more natural to consider the *likelihood* itself (i.e., the probability of observing the data, given particular parameter values). How-

⁸ The function `split` splits a vector into a list based on some grouping variable, and `sapply` applies a function—in this case `sum`—to the list.

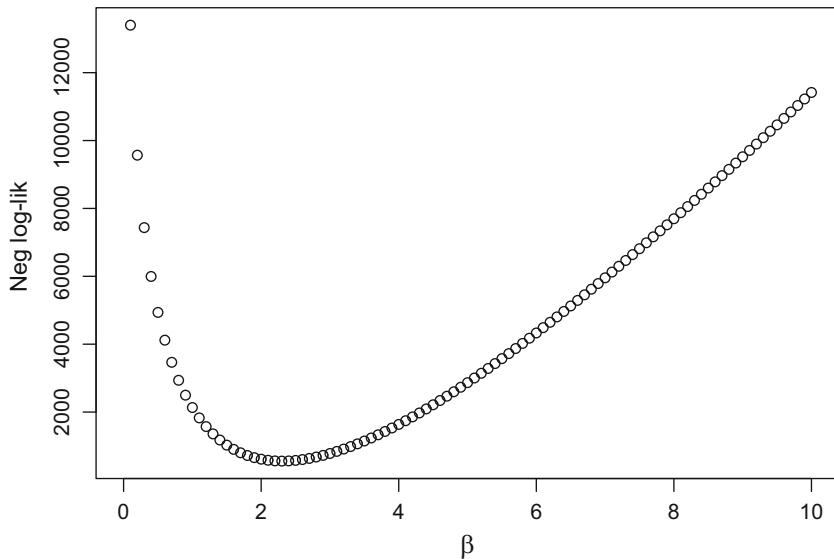


Fig. 3.2 The conditional profile log-likelihood of β for Niamey's district 1 assuming $S_0 = 6500$

ever, since this would be a product of small numbers (one for each observation), computers are not precise enough to distinguish the joint probability from zero if the data set is large.

If our S_0 guess is right, then β should be around 2.3. We can do a similar check for S_0 (assuming β is 2.3). The grid-value associated with the highest likelihood value is 7084.8 (Fig. 3.3), so our original S_0 guess was good but not perfect.

```
betacand=2.3
S0cand=seq(5920,8000, length=101)
ll=rep(NA, length=101)
for(i in 1:101){
  ll[i]=llik.cb(S0=S0cand[i], beta=betacand, I=y)
}
plot(ll~S0cand, ylab="Neg log-lik", xlab=
  expression(S[0]))
S0cand[which.min(ll)]
## [1] 7084.8
```

For a proper analysis we minimize the negative log-likelihood by varying both parameters simultaneously. We can do this using the generic `optim`-function or the `mle2`-function in the `bbmle`-package. The `mle2`-function uses `optim` to find

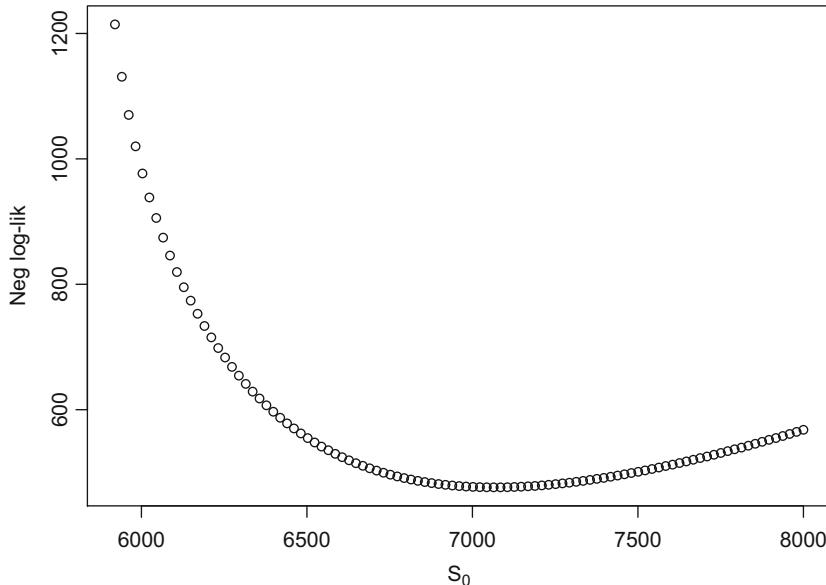


Fig. 3.3 The conditional profile log-likelihood of S_0 for Niamey's district 1 assuming $\beta = 2.3$

maximum likelihood estimates, but also provides confidence intervals, profile likelihoods, and a variety of other useful measures (Bolker 2008). We summarize the basic pertinent likelihood theory for these other measures in Sect. 8.4.

```
require(bbmle)
fit=mle2(llik.cb, start=list(S0=7085, beta=2.3),
           method="Nelder-Mead", data = list(I = y))
summary(fit)

## Maximum likelihood estimation
##
## Call:
## mle2(minuslogl = llik.cb, start = list(S0 = 7085,
##       beta = 2.3), beta = 2), data = list(I = y))
##
## Coefficients:
##             Estimate Std. Error z value     Pr(z)
## S0    7.8158e+03 1.3022e+02 60.019 < 2.2e-16 ***
## beta  1.8931e+00 3.6968e-02 51.209 < 2.2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -2 log L: 841.831
```

```
confint(fit)

## Profiling...
##          2.5 %      97.5 %
## S0    7577.967212 8088.641095
## beta   1.820943   1.966336
```

So the joint MLE estimates are $S_0 = 7816$ (CI: 7578, 8088) and $\beta = 1.89$ (CI: 1.82, 1.7).

Applying statistical tools to biological models—like the chain-binomial—can usefully highlight uncertainties due to parametric interdependencies. In the case of a “simple epidemic” like the measles outbreak considered here, for example, it is conceivable that similar epidemic trajectories can arise from having a large number of initial susceptibles and a low transmission rate, or a more moderate number of susceptibles and a higher transmission rate. We can quantify this through considering the correlation matrix among the parameters of our likelihood analysis; `vcov` calculates their variance-covariance matrix from which we can calculate standard errors according to `sqrt(diag(vcov(fit)))` and `cov2cor` converts this to a correlation matrix. As intuition suggested there is a strong negative correlation between the estimates of the β and S_0 parameters.

```
cov2cor(vcov(fit))

##           S0        beta
## S0    1.0000000 -0.7444261
## beta -0.7444261  1.0000000
```

3.5 Stochastic Simulation

The chain-binomial is both a statistical model for estimation and a stochastic model for dynamics. We can thus write a function to simulate dynamics using the estimated parameters.⁹

```
sim.cb=function(S0, beta, I0){
  I=I0
  S=S0
  i=1
  while (!any(I==0)) {
```

⁹ In contrast to the loop introduced in Sect. 2.3, where the number of iterations is constant and known, the number of epidemic generations may vary among realizations because disease extinction is a stochastic process. We therefore use `while` instead of `for` when looping; `!` means “not” in R.

```

    i=i+1
    I[i]=rbinom(1, size=S[i-1], prob=1-
                 exp(-beta*I[i-1]/S0))
    S[i]=S[i-1]-I[i]
}
out=data.frame(S=S, I=I)
return(out)
}

```

We superimpose 100 stochastic simulations on the observed epidemic. The simulations from the chain-binomial model brackets the observed epidemic nicely (Fig. 3.4), suggesting that the model is a reasonable first approximation to the underlying dynamics. We will revisit on this case study in the context of outbreak-response vaccination in Sect. 8.8.

```

plot(y, type="n", xlim=c(1,18),
      ylab="Predicted/observed", xlab="Week")
for(i in 1:100){
  sim=sim.cb(S0=floor(coef(fit)[ "S0"]),
              beta=coef(fit)[ "beta"], I0=11)
  lines(sim$I, col=grey(.5))
}
points(y, type="b", col=2)

```

3.6 Further Examples

3.6.1 Influenza A/HINI 1977

The `flu` data set in the `epimdr`-package represents the number of children confined to bed each day during a 1978 outbreak of the reemerging influenza A/H1N1 strain in a boarding school in North England (Fig. 3.5). This subtype of influenza had been absent from human circulation after the A/H2N2 pandemic of 1957 but reemerge (presumably from some laboratory freezer) in 1977. The school had 763 boys of which 512 boys were confined to bed sometime during the outbreak. None of the boys would have had previous exposure to A/H1N1.

The typical time of illness was 5–7 days. Since the data is number confined to bed each day, the data is not incidence but (a proxy for) *prevalence*. The data looks pretty log-linear for the first 5 days. Family studies have been used to estimate the serial interval for flu between 2 and 4 days (most between 2 and 3; Cowling et al.

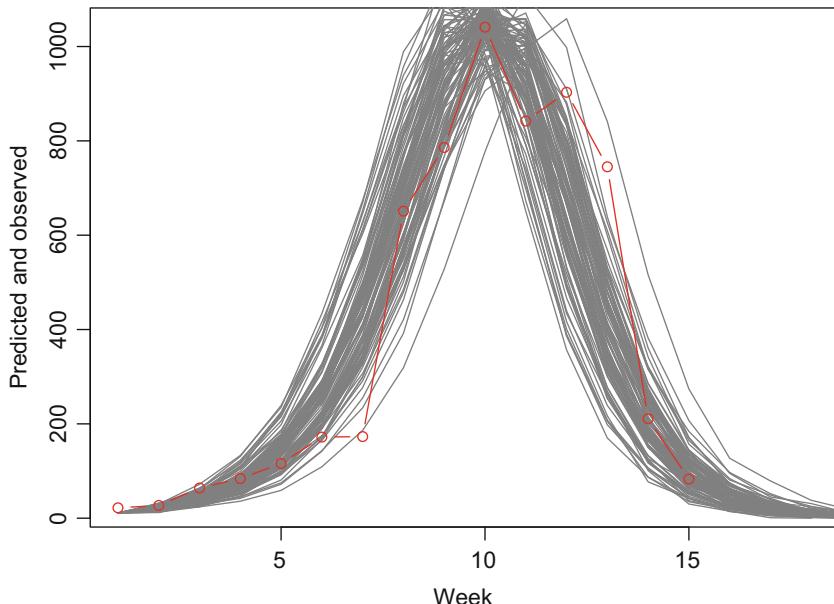


Fig. 3.4 Observed (red) and 100 simulated (gray) epidemics using the chain-binomial model and ML parameters for S_0 and β from Niamey's district 1 data

2009; Vink et al. 2014). Volunteer studies show the mean infectious period around 5 days (Carrat et al. 2008).

```
data(flu)
plot(flu$day, flu$cases, type="b", xlab="Day",
      ylab="In bed", log="y")
tail(flu)

##      day cases
## 9       9   192
## 10     10   126
## 11     11    70
## 12     12    28
## 13     13    12
## 14     14     5
```

The “fast-and-furious” estimate of R_0 is thus:

```
fit=lm(log(cases) ~ day, subset=day<=5,
       data=flu)
lambda=fit$coef ["day"]
V=c(2,3)
V*lambda+1
```

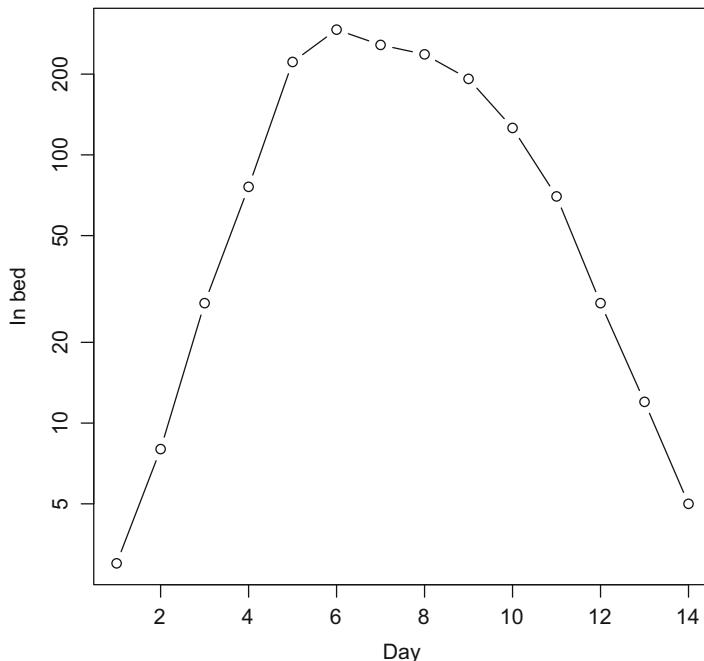


Fig. 3.5 Daily number of children confined to bed in a boarding school in North England during an outbreak in 1978 of the reemerging A/H1N1 strain

```
## [1] 3.171884 4.257827
```

This is higher than most estimates of R_0 of pandemic flu (which typically lies in the 1.5–2.5 interval). However, contact rates within a boarding school is likely to be higher than average across human populations as a whole.

3.6.2 Ebola Sierra Leone 2014–2015

The CDC's record for the 2014–2015 ebola outbreak in Sierra Leone is in the `ebola-data` set. The serial interval for ebola is estimated at around 15 days with an incubation period of 11 days. The mean time to hospitalization is 5 days and mean time to death or dismissal was 5 and 11 days, respectively (WHO Ebola Response Team 2014; White and Pagano 2008). The data is the back-calculated incidence as the difference of the cumulative cases reported by the [CDC](#). Because of the complexities of reporting and revisions of case-load through time, this leads to some negative numbers for certain dates. These were set to zero as a crude fix (Fig. 3.6).

```

data(ebola)
par(mar = c(5,5,2,5))
plot(ebola$day, ebola$cases, type="b", xlab="Week",
      ylab="Incidence")
par(new=T)
plot(ebola$day, ebola$cum_cases, type="l", col="red",
      axes=FALSE, xlab=NA, ylab=NA, log="Y")
axis(side = 4)
mtext(side = 4, line = 4, "Cumulative incidence")
legend("right", legend=c("Cases", "Cumulative"),
       lty=c(1,1), pch=c(1,NA), col=c("black", "red"))
tail(ebola)

##           date day cum_cases cases
## 98    7/8/15 468     13945    34
## 99    7/15/15 475     13982    37
## 100   7/22/15 482     14001    19
## 101   7/29/15 489     14061    60
## 102   8/5/15 496     14089    28
## 103   8/12/15 503     14122    33

```

We first use the regression method with Lipsitch's correction:

```

V = 15
f = 0.5
V * lambda + 1 + f * (1 - f) * (V * lambda)^2

##      day
## 1.6988

```

We next aggregate the data in 2-week increments roughly corresponding to the serial interval, so we can apply the removal method.¹⁰

```

#Data aggregation
cases=sapply(split(ebola$cases,
                   floor((ebola$day-.1)/14)), sum)
sum(cases)

## [1] 14721

#Removal MLE
fit=mle2(llik.cb, start=list(S0=20000, beta=2),
          method="Nelder-Mead", data = list(I = cases))

```

¹⁰ Because of the difference in magnitude of the estimates of S_0 (in the ten thousands) and R_0 (around 1.4), the numerical method used to calculate confidence intervals struggles, so we suggest starting standard errors for the confint-function.

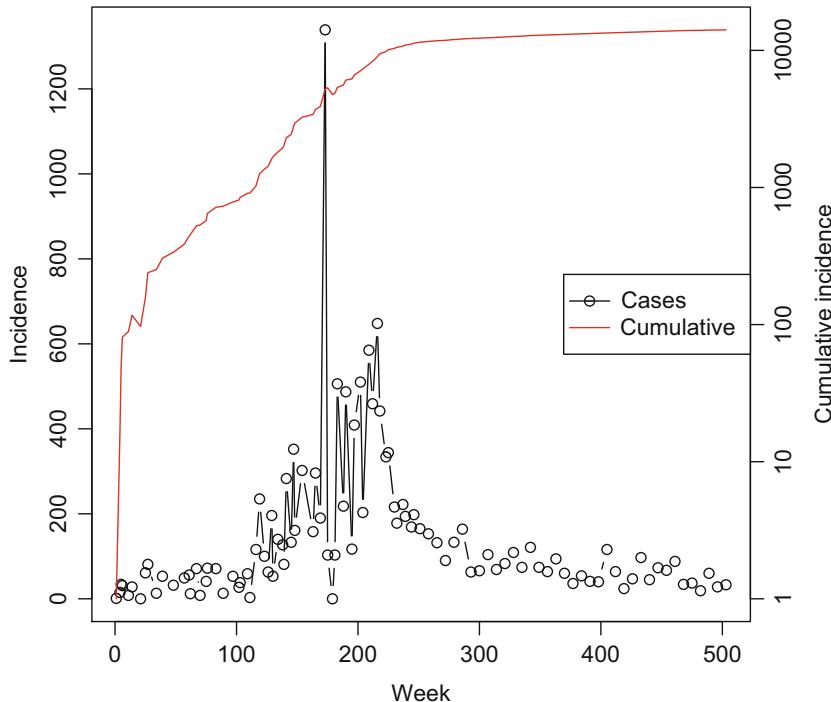


Fig. 3.6 Incidence and cumulative incidence of ebola during the 2014–2015 outbreak in Sierra Leone

```
summary(fit)

## Maximum likelihood estimation
##
## Call:
## mle2(minuslogl = llik.cb, start = list(S0 = 20000,
##       beta = 2), data = list(I = cases))
##
## Coefficients:
##             Estimate Std. Error     z value    Pr(z)
## S0  2.7731e+04 2.5949e-07 1.0687e+11 < 2.2e-16 ***
## beta 1.4237e+00 1.1783e-02 1.2083e+02 < 2.2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -2 log L: 5546.683

confint(fit, std.err=c(100,0.1))
```

```
## Profiling...
##           2.5 %      97.5 %
## S0    26393.579452 29287.725327
## beta   1.384683    1.463184
```

The removal and Lipsitch methods provide comparable estimates that are somewhat lower than those concluded by more elaborate analyses by the WHO team for the Sierra Leone outbreak (WHO Ebola Response Team 2014).

3.6.3 Ebola DRC 1995

The `ferrari`-data set holds the incidence data for a number of outbreaks—Ebola DRC '95, Ebola Uganda '00, SARS Hong Kong '03, SARS Singapore '03, Hog Cholera Netherlands '97, and Foot-and-mouth UK '00—aggregated by disease-specific serial intervals (Table 3.1; Ferrari et al. 2005).

Table 3.1 Serial intervals for each outbreak in the `ferrari` data set

Disease	Serial interval	Location	Year
Ebola	14d	DRC	1995
		Uganda	2000
SARS	5d	Hong Kong	2003
		Singapore	
Hog cholera	7d	Netherlands	1997
FMD	21d	UK	2000

```
names(ferrari)

## [1] "Eboladeaths00" "Ebolacases00"   "Ebolacases95"
## [4] "FMDfarms"       "HogCholera"     "SarsHk"
## [7] "SarsSing"

ferrari$Ebolacases95

## [1] 4 6 5 18 36 99 40 17 4 1 NA NA NA NA NA NA

sum(ferrari$Ebolacases95, na.rm = TRUE)

## [1] 230

Y = c(na.omit(ferrari$Ebolacases95))
```

The number of initial susceptibles must be larger than the summed incidence, so we make an initial guess of 300.

```

fit=mle2(llik.cb, method="Nelder-Mead",
          start=list(S0=300, beta=2), data = list(I = y))
fit

## 
## Call:
## mle2(minuslogl = llik.cb, start = list(S0 = 300,
##       beta = 2), data = list(I = y))
##
## Coefficients:
##             S0          beta
## 241.118108   3.181465
##
## Log-likelihood: -48.3

confint(fit, std.err=2)

## Profiling...
##           2.5 %      97.5 %
## S0    233.973778 254.051292
## beta   2.692505   3.718357

```

The estimated R_0 is 3.2. It thus appears that the Ebola outbreak in DRC in 1995 was more explosive than in Sierra Leone in 2014. This could be due to aggregation across a larger geographic area of the latter and/or the more intensive public health interventions. We will revisit on the DRC outbreak using the “next-generation matrix” method in Sect. 3.9.2.

3.7 R_0 from S(E)IR Flows

As discussed in Sect. 2.1, $R_0 = \beta/(\gamma + \mu)$ for the simple SIR model. This is the correct quantity assuming that the force-of-infection (the rate at which susceptibles are infected) is $\beta I/N$, there is no latent period and no disease-induced mortality, so the index case is expected to be infectious for a period of $1/(\gamma + \mu)$ time units during which it will transmit at a rate of $\beta * N/N$. The numerator comes about because all the N individuals in the population is by definition susceptible when we consider the basic reproductive ratio.

Different SIR-like flows will produce different definitions of R_0 but we can use the same logic for all linear SIR-like flows. Consider, for example, the SEIR model (Fig. 3.7) of the flow of hosts between Susceptible, Exposed (but not yet infectious), Infectious, and Recovered compartments in a randomly mixing population:

$$\frac{dS}{dt} = \mu(N[1-p] - S) - \frac{\beta IS}{N} \quad (3.2)$$

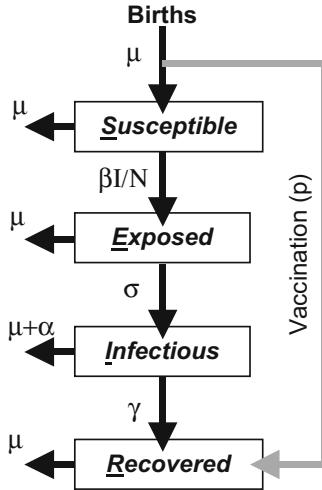


Fig. 3.7 The SEIR flow diagram. Apart from vaccination, flows represent *per capita* rates of flow from the donor compartment. Vaccination is assumed to be a fraction of children vaccinated at birth

$$\frac{dE}{dt} = \frac{\beta IS}{N} - (\mu + \sigma)E \quad (3.3)$$

$$\frac{dI}{dt} = \sigma E - (\mu + \gamma + \alpha)I \quad (3.4)$$

$$\frac{dR}{dt} = \gamma I - \mu R + \mu N p, \quad (3.5)$$

where susceptibles are either vaccinated at birth (fraction p) or infected at a rate $\beta I/N$. Infected individuals will remain in the latent class for an average period of $1/(\sigma + \mu)$ and subsequently (if they escape natural mortality at a rate μ) enter the infectious class for an average time of $1/(\gamma + \mu + \alpha)$; α is the *rate* of disease induced mortality (*not* case fatality rate). By the rules of competing rates (Sect. 3.2), the case fatality rate is $\alpha/(\gamma + \mu + \alpha)$ because during the time an individual is expected to remain in the infectious class the disease is killing them at a rate α . By a similar logic, the probability of recovering with immunity (for life in the case of the SEIR model) is $\gamma/(\gamma + \mu + \alpha)$. Putting all these pieces together, the expected number of secondary cases in a completely susceptible population is thus: probability of making it through latent stage without dying * expected infectious period * transmission rate while infectious. Thus, $R_0 = \frac{\sigma}{\sigma + \mu} \frac{1}{\gamma + \mu + \alpha} \frac{\beta N}{N} = \frac{\sigma}{\sigma + \mu} \frac{\beta}{\gamma + \mu + \alpha}$.

3.8 Other Rules of Thumb

3.8.1 Mean Age of Infection

For endemic, fully immunizing infections, in a constant-sized host population R_0 is related to mean age of infection, \bar{a} , according to $R_0 \simeq 1 + L/\bar{a}$ where L is the life expectancy of the host (e.g., Dietz and Schenzle 1985). This rule of thumb is often used in conjunction with seroprevalence-by-age profiles to get ballpark estimates of R_0 . Chapter 4 discusses age-incidence patterns in more detail.

3.8.2 Final Epidemic Size

In principle, the reproductive ratio can be estimated from the final epidemic size according to the equations discussed in Sect. 2.3. If there is some preexisting immunity and there is homogeneous mixing, then R_0 can be quantified according to $\frac{\log(s_0) - \log(s_\infty)}{s_0 - s_\infty}$, where s_0 and s_∞ are the fractions of the population that is susceptible at the beginning and end of the epidemic, respectively (Heesterbeek and Dietz 1996). However, this is unlikely to be very reliable because the final epidemic size calculations assume that the epidemic is progressing according to the deterministic model (and all its assumptions) including no changes in host behavior in the face of the epidemic. For example, ebola is thought to have an R_0 in the 2–3.5 range, which is what lead CDC to warn that the West-African outbreak could result in millions of cases. In the end the total number of cases in Guinea, Liberia, and Sierra Leone was a far lower number, around 25,000, because of extensive public health interventions and changes to dangerous funeral practices.

For certain common infections like seasonal influenza the rule of thumb may hold; The annual *attack rate* for the flu is around 10–15% which is probably close to that expected from its R_0 (around 1.5–2) and the typical fraction of susceptible of around a quarter (pre-vaccination; assuming immunity following infection lasted around 4 years).

3.8.3 Contact Tracing

Contact tracing can provide direct estimates of R_0 . Blumberg and Lloyd-Smith (2013) showed that this together with size-distributions of subcritical transmission-chains can provide estimates in important low R settings, such as human monkey pox in the face of eroding smallpox herd-immunity. They estimated the human-to-human reproductive ratio to be 0.32. Given that the smallpox vaccine is likely to

be cross-protective against monkey pox, the worry is that this effective reproductive ratio will increase over time since smallpox vaccination is no longer carried out. Contact tracing was also used to estimate R_0 during the early spread of SARS during the 2003 outbreak (Riley et al. 2003).

De et al. (2004) did a contact-tracing study of the spread of gonorrhea across a sexual network in Alberta, Canada. The directional transmission graph among the 89 individuals is in the gonnet-data set. The initial cluster of 17 cases all frequented the same bar, each infected between 0 and 7 other partners with 2.17 as the average. We can use the statnet-package to visualize the chains of transmission (Fig. 3.8):

```
require(statnet)
data(gonnet)
nwt = network(gonnet, directed = TRUE)
plot(nwt, vertex.col = c(0, rep(1, 17), rep(2, 71)))
```

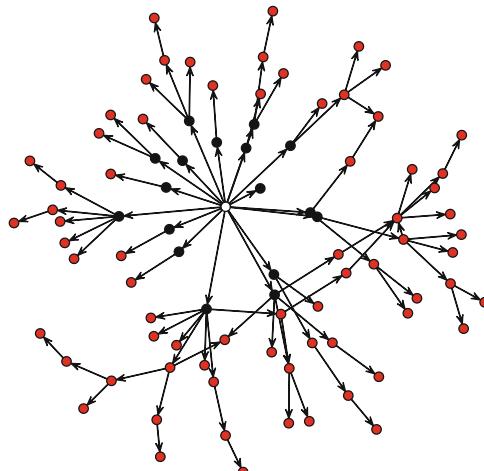


Fig. 3.8 Network of spread of gonorrhea as studied by De et al. (2004). The initial 17 cases (in black) frequented the same bar (white) were ultimately responsible for a cluster of 89 cases identified through contact tracing

The subsequent infections, in turn, infected between 0 and 6 partners with an average of 0.62. The drop is (1) due to the sexual network being depleted of susceptibles, and (2) because infection across heterogenous networks will differentially infect individuals according to their number of contacts (Ferrari et al. 2006a). Epidemics across social networks is the topic of Chap. 12 and we will revisit on this network therein.

3.9 Advanced: The Next-Generation Matrix

For models that are not simple linear chains, it is less straightforward to calculate R_0 from parameterized models using the “logical method.” The [next-generation matrix](#) is the general approach that work for all compartmental models of any complexity (Diekmann et al. 1990). It is done in a sequence of steps:

1. Identify all n infected compartments,
2. Construct a $n \times 1$ matrix, \mathbf{F} , that contains expressions for all *completely new* infections entering each infected compartment,
3. Construct a $n \times 1$ matrix, \mathbf{V}^- , that contains expressions for all losses out of each infected compartment,
4. Construct a $n \times 1$ matrix, \mathbf{V}^+ , that contains expressions for all gains into each infected compartment that does *not* represent *new* infections but transfers among infectious classes,
5. Construct a $n \times 1$ matrix, $\mathbf{V} = \mathbf{V}^- - \mathbf{V}^+$,
6. Generate two $n \times n$ Jacobian matrices \mathbf{f} and \mathbf{v} that are the partial derivatives of \mathbf{F} and \mathbf{V} with respect to the n infectious state variables,
7. Evaluate the matrices at the disease free equilibrium (dfe), and finally
8. R_0 is the greatest eigenvalue of $\mathbf{f}\mathbf{v}^{-1}|_{dfe}$.

3.9.1 SEIR

This is quite an elaborate scheme, so we will try it out first for the SEIR model for which we already know the answer. Unfortunately, R cannot do vectorized *symbolic* calculations, so we need to do this, one matrix element at a time.¹¹ In Chap. 2, we discussed how to use `expression` to do symbolic calculations in R. The `quote`-function is an alternative way to define mathematical expressions; `substitute` allows some simple additional manipulations.

Step 1: Infected classes are E and I , let us label them 1 and 2.

Step 2: All new infections: $dE/dt = \beta SI/N$, $dI/dt = 0$

```
F1 = quote(beta * S * I/N)
F2 = 0
```

Step 3: All losses $dE/dt = (\mu + \sigma)E$, $dI/dt = (\mu + \alpha + \gamma)I$

```
Vm1 = quote(mu * E + sigma * E)
Vm2 = quote(mu * I + alpha * I + gamma * I)
```

Step 4: All gained transfers $dE/dt = 0$, $dI/dt = (\sigma)E$

¹¹ Though it is possible to do calculations more compactly using a `list` of equations.

```
Vp1 = 0
Vp2 = quote(sigma * E)
```

Step 5: Subtract Vp from Vm

```
V1 = substitute(a - b, list(a = Vm1, b = Vp1))
V2 = substitute(a - b, list(a = Vm2, b = Vp2))
```

Step 6: Generate the partial derivatives for the two Jacobians

```
f11 = D(F1, "E"); f12 = D(F1, "I")
f21 = D(F2, "E"); f22 = D(F2, "I")

v11 = D(V1, "E"); v12 = D(V1, "I")
v21 = D(V2, "E"); v22 = D(V2, "I")
```

Step 7: Assuming $N=1$, the disease free equilibrium (dfe) is $S = 1, E = 0, I = 0, R = 0$. We also need values for other parameters. Assuming a weekly time-step and something chickenpox-like we may use $\mu = 0, \alpha = 0, \beta = 5, \gamma = .8, \sigma = 1.2$, and $N = 1$.

```
paras=list(S=1, E=0, I=0, R=0, mu=0, alpha=0,
           beta=5, gamma=.8, sigma=1.2, N=1)

f=with(paras,
       matrix(c(eval(f11), eval(f12), eval(f21),
                 eval(f22)), nrow=2, byrow=TRUE))

v=with(paras,
       matrix(c(eval(v11), eval(v12), eval(v21),
                 eval(v22)), nrow=2, byrow=TRUE))
```

Step8: Calculate the largest eigenvalue of $f \times \text{inverse}(v)$. Note that the function for inverting matrices in R is `solve`.

```
max(eigen(f %*% solve(v))$values)
## [1] 6.25
```

Let us check that the next-generation method and the “flow” method are in agreement recalling that for the SEIR-flow $R_0 = \frac{\sigma}{\sigma+\mu} \frac{\beta}{\gamma+\mu+\alpha}$.

```
with(paras, sigma/(sigma + mu) * beta/(gamma + mu + alpha))
## [1] 6.25
```

3.9.2 SEIHFR

Legrand et al. (2007) forms the foundation for many of the recent Ebola models. The model has five compartments corresponding to Susceptible, Exposed, Infectious in community, Infectious in hospital, Dead but not yet buried, and removed (either buried or immune). The model is more complex than previous compartmental models and cannot be represented by a simple linear chain (Fig. 3.9). The parameterization used here is motivated by the original formulation of Legrand et al. (2007), but the notation conforms to the other sections of this book; Each infectious compartment contributes to the force of infection through their individual β s. There are two branching-points in the flow: The hospitalization of a fraction Θ of the infectious cases after an average time of $1/\gamma_h$ days following onset of symptoms, and the death of a fraction Λ of the I - and H -class after an average time of $1/\gamma_f$ days and $1/\eta_f$ days, respectively. For the 1995 DRC outbreak, Legrand et al. (2007) assumed that hospitalization affected transmission rates but not duration of infection or probability of dying. Model parameters are given in Table 3.2, and the model equations are:

$$\frac{dS}{dt} = -(\beta_i I + \beta_h H + \beta_f F)S/N \quad (3.6)$$

$$\frac{dE}{dt} = (\beta_i I + \beta_h H + \beta_f F)S/N - \sigma E \quad (3.7)$$

$$\frac{dI}{dt} = \sigma E - \Theta \gamma_h I - (1 - \Theta)(1 - \Lambda) \gamma_I I - (1 - \Theta)\Lambda \gamma_f I \quad (3.8)$$

$$\frac{dH}{dt} = \Theta \gamma_h I - \Lambda \eta_f H - (1 - \Lambda) \eta_r H \quad (3.9)$$

$$\frac{dF}{dt} = (1 - \Theta)(1 - \Lambda) \gamma_I I + \Lambda \eta_f H - \chi F \quad (3.10)$$

$$\frac{dR}{dt} = (1 - \Theta)(1 - \Lambda) \gamma_I I + (1 - \Lambda) \eta_r H + \chi F \quad (3.11)$$

There are four infected compartments (E , I , H , and F), thus \mathbf{F} , \mathbf{V}^- , and \mathbf{V}^+ will be 4×1 matrices, and \mathbf{f} and \mathbf{v} will be 4×4 matrices.

Step 1: Infected classes are E , I , H , and F , and let us label them 1–4.

Step 2: All new infections $dE/dt = \beta S I / N$, $dI/dt = 0$

```
F1=expression(betai * S * I / N + betah* S * H / N +
    betaf * S * F / N)
F2=0
F3=0
F4=0
```

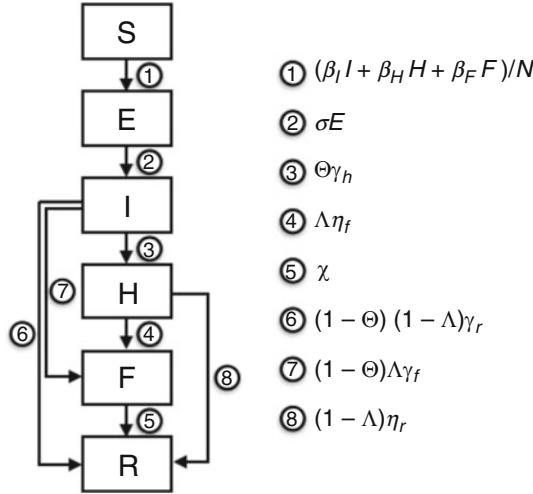


Fig. 3.9 The SEIHFR flow diagram for ebola dynamics

Table 3.2 Parameters for Legrand et al. (2007)'s Ebola model using the data from the 1995 DRC epidemic

Parameter	Meaning	Value
N	Population size	
$1/\sigma$	Incubation period	7d
$1/\gamma_h$	Onset to hospitalization	5d
$1/\gamma_f$	Onset to death	9.6d
$1/\gamma_r$	Onset to recovery	10d
$1/\eta_f$	Hospitalization to death	4.6d
$1/\eta_r$	Hospitalization to recovery	5d
$1/\chi$	Death to burial	2d
Θ	Proportion hospitalized	80%
Λ	Case fatality ratio	81%
β_i	Transmission rate in community	0.588
β_h	Transmission rate in hospital	0.794
β_f	Transmission rate at funeral	7.653

To avoid confusion, we use lowercase Greek for rates and uppercase for probabilities

Step 3: All losses

```
Vm1=quote(sigma * E)
Vm2=quote(Theta * gammah * I + (1 - Theta) * (1 -
Lambda) * gammar * I + (1 - Theta) * Lambda *
gammaf * I)
Vm3=quote(Lambda * etaf * H + (1 - Lambda) * etar * H)
Vm4=quote(chi * F)
```

Step 4: All gained transfers

```
Vp1=0
Vp2=quote(sigma * E)
Vp3=quote(Theta * gammah * I)
Vp4=quote((1 - Theta) * (1 - Lambda) * gammar * I +
Lambda * etaf * H)
```

Step 5: Subtract Vp fromVm

```
V1 = substitute(a - b, list(a = Vm1, b = Vp1))
V2 = substitute(a - b, list(a = Vm2, b = Vp2))
V3 = substitute(a - b, list(a = Vm3, b = Vp3))
V4 = substitute(a - b, list(a = Vm4, b = Vp4))
```

Step 6: Generate the partial derivatives for the two Jacobians

```
f11 = D(F1, "E"); f12 = D(F1, "I"); f13 = D(F1, "H")
      f14 = D(F1, "F")
f21 = D(F2, "E"); f22 = D(F2, "I"); f23 = D(F2, "H")
      f24 = D(F2, "F")
f31 = D(F3, "E"); f32 = D(F3, "I"); f33 = D(F3, "H")
      f34 = D(F3, "F")
f41 = D(F4, "E"); f42 = D(F4, "I"); f43 = D(F4, "H")
      f44 = D(F4, "F")

v11 = D(V1, "E"); v12 = D(V1, "I"); v13 = D(V1, "H")
      v14 = D(V1, "F")
v21 = D(V2, "E"); v22 = D(V2, "I"); v23 = D(V2, "H")
      v24 = D(V2, "F")
v31 = D(V3, "E"); v32 = D(V3, "I"); v33 = D(V3, "H")
      v34 = D(V3, "F")
v41 = D(V4, "E"); v42 = D(V4, "I"); v43 = D(V4, "H")
      v44 = D(V4, "F")
```

Step 7: Disease free equilibrium: the dfe is $S = 1, E = 0, I = 0, H = 0, F = 0, R = 0$. We also need values for other parameters. We use the estimates from the DRC 1995 outbreak scaled as weekly rates from tables and appendices of Legrand et al. (2007).

```
gammah = 1/5 * 7
gammaf = 1/9.6 * 7
gammar = 1/10 * 7
chi = 1/2 * 7
etaf = 1/4.6 * 7
etar = 1/5 * 7
```

```

paras=list(S=1,E=0, I=0, H=0, F=0,R=0,
           sigma=1/7*7, Theta=0.81, Lambda=0.81, betai=0.588,
           betah=0.794, betaf=7.653,N=1, gammah=gammah,
           gammaf=gammaf, gamma=gamma, etaf=etaf,
           etar=etar, chi=chi)

f=with(paras,
matrix(c(eval(f11),eval(f12),eval(f13),eval(f14),
        eval(f21),eval(f22),eval(f23),eval(f24),
        eval(f31),eval(f32),eval(f33),eval(f34),
        eval(f41),eval(f42),eval(f43),eval(f44)),
       nrow=4, byrow=T))

v=with(paras,
matrix(c(eval(v11),eval(v12),eval(v13),eval(v14),
        eval(v21),eval(v22),eval(v23),eval(v24),
        eval(v31),eval(v32),eval(v33),eval(v34),
        eval(v41),eval(v42),eval(v43),eval(v44)),
       nrow=4, byrow=T))

```

Step 8: Calculate the largest eigenvalue of $f \times \text{inverse}(v)$

```

max(eigen(f %*% solve(v))$values)
## [1] 2.582429

```

Chapter 4

FoI and Age-Dependent Incidence



4.1 Burden of Disease

In everyday conversation about contagious maladies, “disease” and “infection” are sometimes used interchangeably. Often this imprecision does not matter. It is however useful to keep in mind that disease strictly speaking refers to symptomology and infection to pathogen/parasite colonization-status. The latent period —the time between a pathogen colonizes a host and the host can pass the infection on—is different from the incubation period—the time from colonization to onset of symptoms (“disease”). Such distinctions are obvious for certain infections; We all recognize the distinction between *AIDS* and *HIV positive*. The former refers to disease status, the latter to infection status. For “the flu,” the virus is typically cleared in less than a week, but noncontagious cough and discomfort can last for another week or more. Thus clinical relevance is not always the same as dynamic relevance.

The severity of disease of many infections depends on age. The very young are often prone to more severe disease. Both measles and whooping cough, for example, cause highest morbidity and mortality in children under one (e.g., Miller and Fletcher 1976; Grais et al. 2007). Other diseases are more severe in the elderly. Mortality from influenza-like illness is a common example. “Teratogenic” diseases are those that cause complications during pregnancies. Rubella, chicken pox, and

This chapter uses the following R-packages: `splines` and `fields`.

A conceptual understanding of *Force of Infection* is useful prior to this discussion. A 5-min epidemics-MOOC intro can be watched from YouTube:

Force of Infection <https://www.youtube.com/watch?v=dj1DiqA4Lvg>.

Pathogens and Extinction <https://www.youtube.com/watch?v=v67gtiACBTY>.

Zika are important examples (Metcalf and Barrett 2016). For these, infections of reproductive-age women are the most pressing public health concern. It is important to understand determinants of age-prevalence curves for two reasons: First, because of such age-specificity in burden of disease, and second—as we shall see—because age-structure can mold infectious disease dynamics in important ways.

4.2 Force of Infection

The Force of Infection (FoI) is the *per capita* rate at which susceptibles are exposed to infection. The FoI in the S(E)IR compartmental model (Eqs. (2.1)–(2.3) and (3.1)–(3.3)) is $\phi = \beta I/N$ because each susceptible is assumed to contact other individuals in the population at some rate, the fraction of those contacts that are with infected individuals is I/N and β is by definition the contact rate times the probability of infection upon contact.¹

An important basic and applied question is how the FoI scales with population density/size (de Jong et al. 1995). The literature suggests two extreme situations termed: “density-dependent” transmission for which the FoI scales linearly with density and “frequency-dependent” transmission for which the FoI is independent of density. Roberts and Heesterbeek (1993) points out that there is some significant confusion in the literature about the meaning of these terms, as the denominator N in the SEIR formulations is by some wrongly interpreted as Eqs. (3.3)–(3.5) being a “frequency-dependent” model. Roberts and Heesterbeek (1993) clarify that this is a mistaken interpretation; the I/N simply stems from the idea that only this fraction of random contacts are with infectious individuals (as opposed to the complementary fraction which is with noninfectious individuals). The issue of density- *versus* frequency-dependence should be thought of in terms of how β (= contact rate * transmission probability) scales with density (Roberts and Heesterbeek 1993; Ferrari et al. 2011). For the strictly density-dependent model, numbers of contacts are proportional to density, so $\beta(N) \propto N$ and thus transmission and R_0 scales linearly with density. In contrast the strictly frequency-dependent model assumes that contact rates are independent of N and, therefore, so is R_0 . The frequency-dependent model is often used for sexually transmitted diseases (STDs) and vector-borne infections with the logic that the number of sexual partner does not scale with density and neither does the feeding requirements of mosquitos.

An interesting ecological implication is that in the absence of an alternative host, a deadly density-dependently transmitted pathogen is less likely to drive a host extinct because as the pathogen decimates the host, the reproductive ratio is expected

¹ The theoretical FoI is model specific, so more complicated models may have more complicated FoIs. The FoI for the SEIHFR model of Sect. 3.9.2, for example, is given by rate ① in Fig. 3.9.

to eventually decrease below one, at which time the chain-of-transmission will falter and break. Frequency-dependent pathogens, in contrast, may be able to sustain the chain-of-transmission to a bitter end as the reproductive ratio may remain supercritical (De Castro and Bolker 2005).

4.3 Probability of Infection at Age: The Catalytic Model

The FoI is a *rate*, thus if age-invariant, in a randomly mixing population the expected waiting-time to first infection is $1/\phi$. For endemic, fully immunizing infections in a constant-sized host population, R_0 determines the mean age of infection, \bar{a} , according to $R_0 \simeq 1 + L/\bar{a}$ where L is the life expectancy of the host. Thus the mean age of infection will be² $\bar{a} \simeq L/(R_0 - 1)$.

The general rate, $\phi(a,t)$, at which any susceptible will be infected may depend on age (a) and time (t). Ignoring time-dependence (but see Ferrari et al. 2010), the integrated rate of infection to age a is $\int_0^a \phi(a)da$, thus the probability of not being infected by age a is $1 - p(a) = \exp(-\int_0^a \phi(a)da)$ and the probability of being infected on or before age a is (by the logic laid out in Sect. 3.2):

$$p(a) = 1 - e^{-\int_0^a \phi(a)da}. \quad (4.1)$$

This is called the catalytic model (Muench 1959; Hens et al. 2010).³ Age-intensity curves and age-seroprevalence curves are important data-sources for estimating the FoI. For nonlethal, persistent infections and nonlethal, fully immunizing infections the former/latter provides excellent data for estimating ϕ . In the simplest case we assume that the FoI is independent of both age and time, in this case the probability of being infected by age a is $1 - \exp(-\phi a)$. If we have data on number of infected individuals by age, we can use the standard generalized linear model (glm) framework to estimate the FoI for this *simpler* model.

Generalized linear models have two components: an error distribution (such as binomial, Poisson, negative binomial, normal, etc.) and a “link” function which specifies how the expected (predicted) values \hat{y} are linked to the “linear predictors” $x = a + b_1x_1 + c_1x_2 \dots$. Common link functions are (depending on error distributions): “identity,” “log,” “logit” (= “log-odds” = $\log(\hat{y}/(1 - \hat{y}))$), and “complementary log-log” (= $\log(-\log(1 - \hat{y}))$) (McCullagh and Nelder 1989). The link functions are associated with inverse link functions which for the aforementioned are: “identity,” e^x , $\frac{e^x}{1+e^x}$, and $1 - e^{-x}$, respectively.

² In populations of changing size a more accurate calculation is $\bar{a} \simeq 1/(\mu(R_0 - 1))$, where μ is the host birth rate (Dietz and Schenzle 1985).

³ If immunity wanes at a rate ω , the reversible catalytic model is $p(a) = \frac{\phi(a)}{\phi(a)+\omega}(1 - e^{-\int_0^a \phi(a)+\omega da})$ (see e.g., Pomeroy et al. 2015, for an example). Heisey et al. (2006) discuss corrections needed if infection causes significant disease-induced mortality.

Let us assume we test some n_a individuals of each age a and find from serology that i_a individuals have been previously infected. Inferring ϕ from this data is a standard(ish) binomial regression problem: $p(a) = 1 - \exp(-\phi a)$ is the expected fraction infected (or seropositive) by age a . Thus $\log(-\log(1 - p(a))) = \log(\phi) + \log(a)$, so we can estimate a constant log-FoI as the intercept from a `glm` with binomial error, a complimentary log-log link and log-age as a regression “offset.”⁴ The R call will be of the form⁵:

```
glm(cbind(inf, notinf) ~ offset(log(a)),
    family=binomial(link="cloglog"))
```

We can illustrate the approach using the pre-vaccination Measles antibody data of Black (1959). The data contain seroprevalence-by-age-bracket of some 300 people from around New Haven, Connecticut from blood drawn in the summer of 1957:

```
data(black)
black

##      age     mid   n pos neg         f
## 1     <1 0.75 10  8  2 0.8000000
## 2  1-4 2.50 21  4 17 0.1904762
## 3  5-9 7.00 41 31 10 0.7560976
## 4 10-14 12.00 52 50  2 0.9615385
## 5 15-19 17.00 30 28  2 0.9333333
## 6 20-29 25.00 38 37  1 0.9736842
## 7 30-39 35.00 51 49  2 0.9607843
## 8 40-49 45.00 35 31  4 0.8857143
## 9  >50 60.00 30 26  4 0.8666667
```

The age-profile of seroprevalence takes the characteristic shape of many pre-vaccination childhood diseases: High seroprevalence of the very young (<1 year) due to the presence of maternal antibodies that wanes with age, followed by rapid build-up of immunity to almost 100% seroprevalence by age 20 (Fig. 4.1). There is perhaps some evidence of loss of immunity in the elderly. We use the binomial regression scheme to estimate the log-FoI based on the data for people in the 1–40 year groups, and compare predicted and observed seroprevalence by age (Fig. 4.1):

⁴ An offset is a covariate that has a fixed coefficient of unity in a regression.

⁵ Binomial regression either takes a binary 0/1 variable as the response or a matrix with two columns representing number of successes and failures for each covariate level.

```

b2=black[-c(1,8,9),] #subsetting age brackets
#Estimate log-FoI
fit=glm(cbind(pos,neg) ~ offset(log(mid)),
        family=binomial(link="cloglog"), data=b2)
#Plot predicted and observed
phi=exp(coef(fit))
curve(1-exp(-phi*x), from=0, to=60,
      ylab='Seroprevalence', xlab='Age')
points(black$mid, black$f, pch='*', col='red')
points(b2$mid, b2$f, pch=8)
exp(fit$coef)

## (Intercept)
##     0.1653329

```

The estimated FoI is 0.16/year, giving a predicted mean age of infection of 6 years.

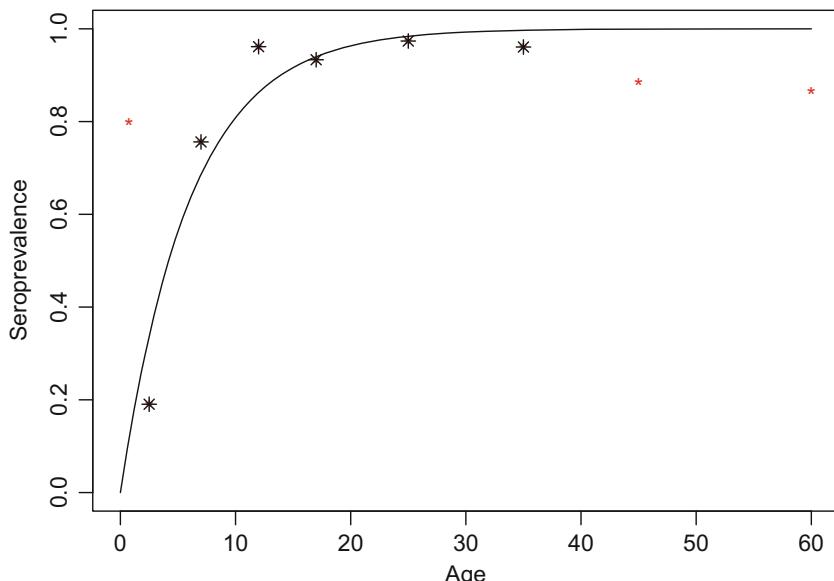


Fig. 4.1 Seroprevalence-by-age from the measles antibody study of Black (1959) from pre-vaccination Connecticut. The solid line is the predicted age-prevalence curve for the subset of the data used for estimation (black stars). The smaller red stars are data excluded from estimates due to maternal antibodies or possibly waning titers. Data are centered on the midpoints of each age-bracket

4.4 More Flexible ϕ -Functions

The assumption of a constant, age-invariant FoI is usually too simplistic because of age- or time-varying patterns of mixing. We can use Long et al.'s (2010) data on prevalence of the bacterium *Bordetella bronchiseptica* in a rabbit breeding facility to illustrate. *B. bronchiseptica* is a non-immunizing, largely avirulent (though it can cause snuffles), persistent infection of rabbits. Two-hundred-and-fourteen rabbits of known age were swabbed nasally and tested for the bacterium.

```
data(rabbit)
head(rabbit)

##      a  n inf
## 1 1.0 59   3
## 2 2.0  8   7
## 3 2.5  4   4
## 4 3.0  2   1
## 5 3.5  5   1
## 6 4.0  2   0
```

We first calculate the average FoI from the binomial regression scheme introduced above. In the breeding facility the older breeding animals are kept separate from the younger animals, so we restrict ourselves to rabbits <1 year old. We superimpose our fit on the plot of prevalence by age. In Fig. 4.2 the size of the circles is proportional to the sample size:

```
rabbit$notinf=rabbit$n-rabbit$inf
#Binomial regression
fit=glm(cbind(inf, notinf)~offset(log(a)),
        family=binomial(link="cloglog"),
        data=rabbit, subset=a<12)
#Plot data
symbols(rabbit$inf/rabbit$n~rabbit$a, circles=rabbit$n,
        inches=.5, xlab="Age", ylab="Prevalence")
#Predicted curves for <1 and all
phi=exp(coef(fit))
curve(1-exp(-phi*x), from=0, to=12, add=TRUE)
curve(1-exp(-phi*x), from=0, to=30, add=TRUE, lty=2)
1/phi

## (Intercept)
##      5.918273
```

The predicted median age of infection is just under 6 months. The constant-FoI model seems to do well for up to about 15 months of age, but the model overpredicts

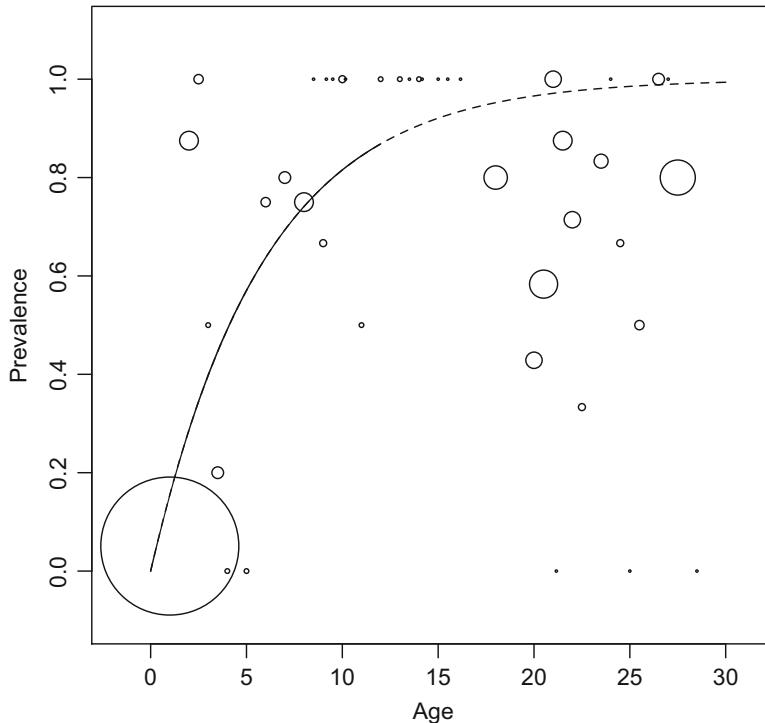


Fig. 4.2 Age-prevalence of *B. bronchiseptica* in a rabbit breeding facility. Circle size is proportionate to the number of animals tested in each age group. The solid line is the predicted age-prevalence curve for the subset of the data used for estimation (up to 1-year animals). The dotted line is the extrapolation to older individuals

the prevalence in older individuals. To allow for the scenario that the FoI varies with age, we need to implement our own framework (as opposed to using `glm`) using the maximum likelihood ideas introduced in Sect. 3.4. A simple model for age-specific FoI assumes a piecewise constant model (Grenfell and Anderson 1985), where individuals are classified into discrete age classes. For a piecewise constant model the integrand in Eq. (4.1) integrates to $\phi_a(a - c_a) + \sum_{k < a} \phi_k d_k$, where ϕ_a is the FoI of individuals in the a 'th age bracket, and c_a and d_a are the lower cut-off age and duration of that bracket, respectively. We define a function for the integrand which takes the argument a for age, up is a vector of the upper cut-offs for each age bracket, and foi is the vector of age-specific FoIs:

```
integrandpc=function(a, up, foi){
  #Find which interval a belongs to
  wh=findInterval(a, sort(c(0,up)))
  #Calculate duration of each interval
```

```

dur=diff(sort(c(0,up)))
#Evaluate integrand
inte=ifelse(wh==1, foi[1]*a,
            sum(foi[1:(wh-1)]*dur[1:(wh-1)]) +
            foi[wh] * (a-up[wh-1]))
return(inte)
}

```

The negative log-likelihood function for the piecewise constant model takes arguments corresponding to log-FoI (par), age (age), number of positives (num), number tested in each age group (denom), and age-class cut-offs (up). Estimating the FoI on a log-scale (foi=exp (par)) ensures that all rates will be positive.

```

llik.pc = function(par, age, num, denom, up) {
  ll = 0
  for (i in 1:length(age)) {
    p = 1 - exp(-integrandpc(a=age[i], up = up,
                               foi = exp(par)))
    ll = ll + dbinom(num[i], denom[i], p, log = T)
  }
  return(-ll)
}

```

We use 1, 4, 8, 12, 18, 24, and 30 months as cut-off points for the age categories and assign arbitrary initial values of 0.1 for each piece of the FoI-function:

```

x = c(1, 4, 8, 12, 18, 24, 30)
para = rep(0.1, length(x))

```

For the analysis we use the optim-function to find maximum likelihood estimates:

```

est = optim(par=log(para), fn=llik.pc, age=rabbit$a,
            num=rabbit$inf, denom=rabbit$n, up=x,
            method="Nelder-Mead", control=list(trace=2))

```

The maximum likelihood estimates for the log-FoI is given in est\$par. The associated age-specific FoIs are:

```
round(exp(est$par), 6)
```

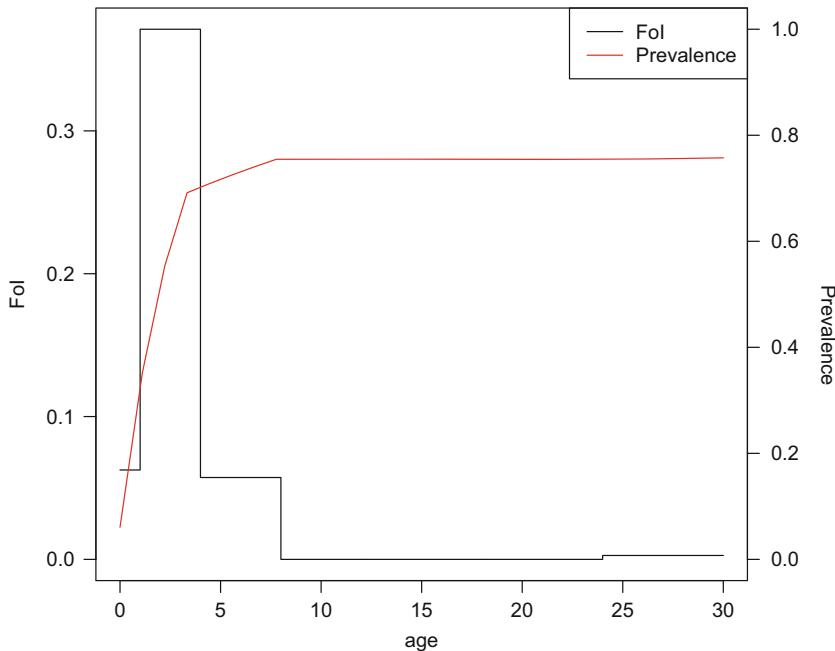


Fig. 4.3 The piecewise constant age-specific FoI of *B. bronchiseptica* in a rabbit breeding facility and the associated predicted age-prevalence curve

We can predict the age-prevalence curve and plot it as a step function (Fig. 4.3).

```
#Make space for left and right axes
par(mar = c(5,5,2,5))
#Add beginning and ends to x and y for step plot
xvals=c(0,x)
yvals=exp(c(est$par, est$par[7]))
plot(xvals, yvals, type="s", xlab="age", ylab="FoI")

#Superimpose predicted curve
par(new=T)
p = rep(0, 28)
for (i in 1:28) {
  p[i] = 1 - exp(-integrandpc(a=i, up = x,
    foi = exp(est$par)))
}
plot(p~c(1:28), ylim=c(0,1), type="l", col="red",
  axes=FALSE, xlab=NA, ylab=NA)

#Add right axis and legend
```

```
axis(side = 4)
mtext(side = 4, line = 4, "Prevalence")
legend("right", legend=c("FoI", "Prevalence"),
       lty=c(1,1), col=c("black", "red"))
```

The FoI peaks perinatally and then falls to zero after the 8-month age class. This is likely due to the older breeder females being housed separately and only having contact with their kittens. Long et al. (2010) used this (in combination with some other analyses; see Sect. 15.4) to conclude that most infections happen at a young age from infected mothers to their offspring and then among litter mates.

4.5 A Log-Spline Model

An alternative nonparametric approach to the piecewise constant model is to use smoothing splines. A [spline](#) is a smooth curve that can take an arbitrary shape except that it is constrained to be continuous and with continuous first and second derivatives (Härdle 1990; Hastie and Tibshirani 1990). The popularity of splines in nonparametric regression stems from its computational tractability; A spline can be fit by multiple regression on a set of “basis function”-decompositions of a covariate. The `gam` and `mgcv` packages offer automated ways to fit a variety of spline-variants to binomial data (and any other error distribution within the exponential family). Unfortunately, as with the case of the piecewise constant model, fitting the log-spline model is a bit more involved because of the integration step in Eq. (4.1). The `splines` package has functions to create various spline-bases that can be used with `lm`; `predict.lm` can predict values for the spline given regression coefficients.

The approach taken here is a bit cheeky in that it “hi-jacks” a spline-regression object created using the `bs`-spline basis functions in combination with `lm` and use `optim` to update/override the regression coefficients in the `lm`-object until a maximum likelihood solution is found. First we set the number of degrees-of-freedom for the spline. The `d1`-object will end up as the hi-jacked object for the age-specific FoI (Long et al. 2010).

```
require(splines)
# Degrees-of-freedom
df = 7
# Construct dummy lm-object
d1 = lm(inf ~ bs(a, df), data = rabbit)
```

We write a `tmpfn`-function to predict the spline on a log-transformed scale to ensure that the force-of-infection (FoI) is strictly positive:

```
tmpfn = function(x, dl) {
  x = predict(dl, newdata = data.frame(a = x))
  exp(x)
}
```

The tmpfn2-function calculates the negative log-likelihood of the FoI as we did in the foipc-function above. In contrast to the piecewise constant model, the integrated splines do not have a closed form solution, so we use R's inbuilt numerical integrator, `integrate`:

```
tmpfn2=function(par,data, df){
  #Dummy lm-object
  dl=lm(inf~bs(a,df), data=data)
  #Overwrite spline coefficients with new values
  dl$coefficients=par
  #Calculate log-likelihood
  ll=0
  for(i in 1:length(data$a)){
    p = 1 - exp(-integrate(tmpfn, 0, i, dl = dl)$value)
    ll=ll+dbinom(data$inf[i],data$n[i],p,log=T)
  }
  return(-ll)
}
```

We use arbitrary initial values and minimize the negative log-likelihood using `optim`.

```
para=rep(-1, df+1)
dspline = optim(par=para, fn=tmpfn2, data=rabbit,
  df=df, method="Nelder-Mead", control=
  list(trace=2, maxit=2000))
```

We can plot the resultant maximum likelihood fits (Fig. 4.4).

```
par(mar = c(5,5,2,5)) #Room for two axes
#Overwrite dummy-objects coefficients with MLEs
dl$coefficients=dspline$par
#Age-prevalce plot
plot(tmpfn(rabbit$a,dl)~rabbit$a, type="l", ylab="FoI",
  xlab="Age (mos)", las=1)
#Overlay FoI
par(new=T)
p = rep(0, 28)
for (i in 1:28) {
  p[i] = 1 - exp(-integrate(tmpfn, 0, i,
    dl = dl)$value)
```

```

}

plot(p~c(1:28), ylim=c(0,1), type="l", col="red",
      axes=FALSE, xlab=NA, ylab=NA)
axis(side = 4, las=1)
mtext(side = 4, line = 4, "Prevalence")
legend("topright", legend=c("FoI", "Prevalence"),
      lty=c(1,1), col=c("black", "red"))

```

Both the piecewise and spline models show strong evidence of age-specificity in the FoI with a peak in transmission somewhere between 1 and 5 months of age, suggesting that circulation is mainly among the young and among littermates (Long et al. 2010). We revisit on this case study in Sect. 15.4.

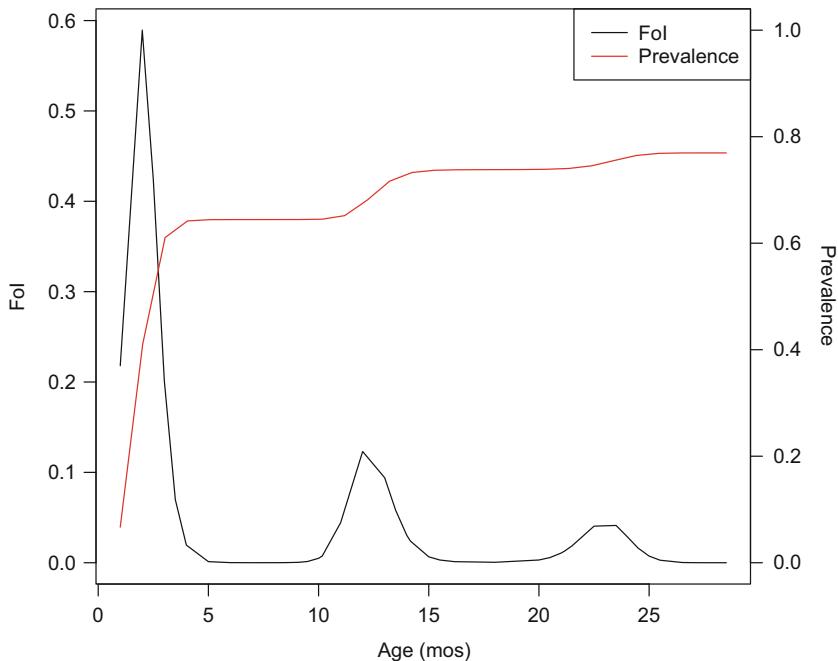


Fig. 4.4 The spline-estimate of the age-specific FoI of *B. bronchiseptica* in a rabbit breeding facility

4.6 Rubella

Rubella is a relatively mild, vaccine-preventable infection except that infection during pregnancy leads to stillbirths or [congenital rubella syndrome](#). The main public health objective is therefore to minimize the FoI in women of childbearing age. The issue was made clear because of a surprising surge in CRS cases in Greece in the mid-90s following a low-intensity vaccination campaign (Panagiotopoulos et al. 1999).

Age-intensity data is less ideal than seroprevalence data for catalytic analysis; however, it is more common and therefore worth considering. Metcalf et al. (2011c) studied age-intensity curves for rubella across the provinces of Peru between 1997 and 2009. There were 24,116 reported cases during the period. The data are 1/2-monthly to age 1 and yearly thereafter (Fig. 4.5). With age-incidence data on immunizing infections, we can use the catalytic framework to estimate the relative age-specific FoI using the cumulative incidence by age (in place of age-seroprevalence or age-prevalence). For the analysis we use the total number of cases as our denominator because the actual number of susceptibles in each age group is not monitored. Hence, the estimate is a *relative* FoI because of the unknown baseline. Using the total cases as a denominator, further leads to severe biases of the FoI at old age classes (because exactly all of the assumed susceptibles in the final age class will be presumed to be infected at the time), so it should only be applied to the younger portion of the data. Its application also assumes a uniform age-distribution, so a correction for the age-pyramid may be necessary for a more refined analysis (Ferrari et al. 2010).

```
data(peru)
head(peru)

##           age  incidence  cumulative      n
## 2  0.01095890        1       56  24116
## 3  0.01369863        1       57  24116
## 4  0.01643836        1       58  24116
## 5  0.01917808        2       60  24116
## 6  0.03561644        1       61  24116
## 7  0.03835616        2       63  24116

#Calculate cumulative incidence
peru$cumulative=cumsum(peru$incidence)
#Define denominator
peru$n=sum(peru$incidence)
par(mar = c(5,5,2,5)) #Make room for two axes and plot
#Plot incidence with cumulative overlaid
plot(peru$incidence~peru$age, type="b", xlab="Age",
      ylab="Incidence")
par(new=T)
plot(peru$cumulative~peru$age, type="l", col="red",
```

```

axes=FALSE, xlab=NA, ylab=NA)
axis(side = 4)
mtext(side = 4, line = 4, "Cumulative")
legend("right", legend=c("Incidence", "Cumulative"),
lty=c(1,1), col=c("black", "red"))

```

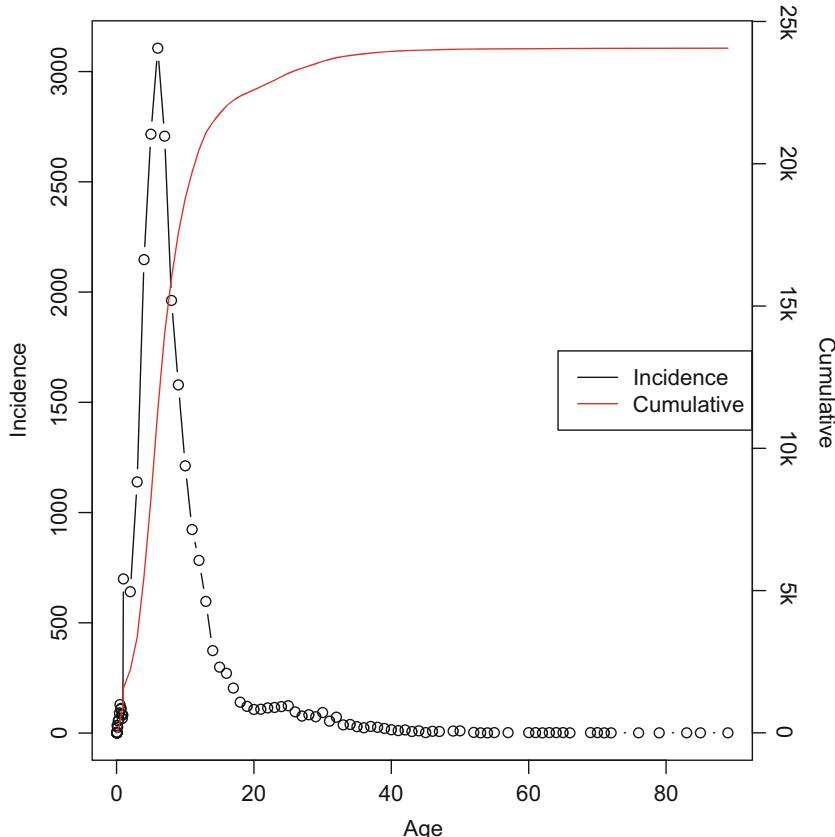


Fig. 4.5 Age-specific incidence and cumulative incidence of rubella in Peru 1997–2009

We first apply the piecewise model assuming a separate FoI for each year up to age 20 and 10 year classes thereafter. Convergence of the piecewise model with this many segments is very slow, so the actual figure (Fig. 4.6) was produced by doing repeat calls to `optim` using different optimization methods (Nelder-Mead, BFGS, and SANN), feeding the estimates from each call as starting values for the next. However, the basic analysis is:

```
#Upper age cut-offs
up=c(1:20,30, 40, 50, 60, 70,100)
para=rep(.1,length(up)) #Initial values
#Minimize log-likelihood
est2 = optim(par=log(para),fn=llik.pc, age=peru$age,
             num=peru$cumulative, denom=peru$n, up=up,
             method="Nelder-Mead", control=
               list(trace=2, maxit=2000))
#Step plot
x=c(0, up)
y=exp(c(est2$par, est2$par[26]))
plot(x, y, ylab="Relative FoI", xlab="Age", type="l",
      ylim=c(0,0.25), xlim=c(0,80))
```

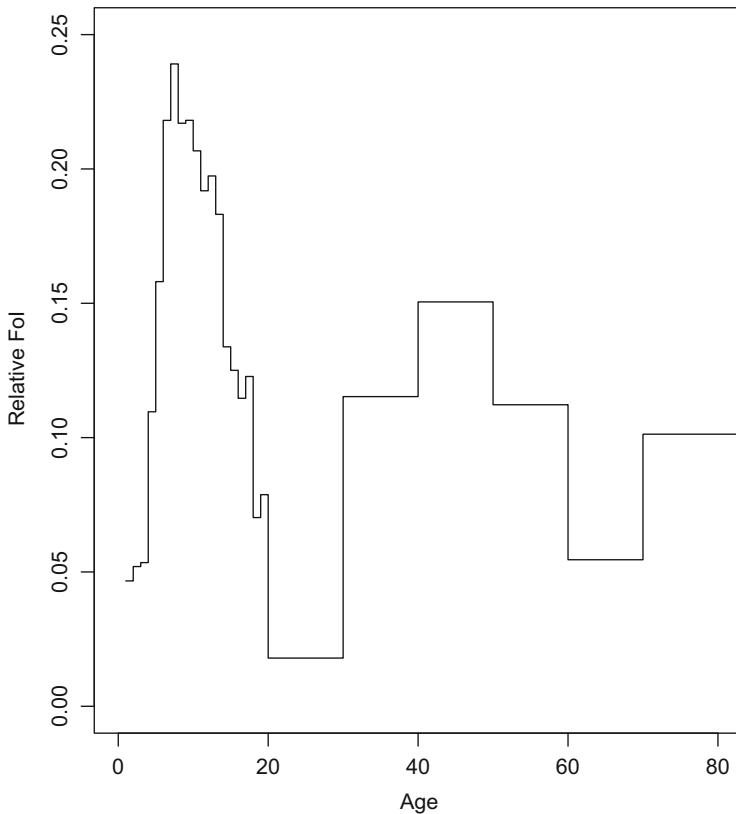


Fig. 4.6 The relative age-specific FoI of rubella in Peru as estimated using the piecewise-constant model

We see a clear peak in FoI in the 8–10 age group. The pattern makes sense given the biology of rubella and the assortative mixing commonly seen in the human host with most contacts being among same-aged individuals (see Sect. 4.7). Peru has a life-expectancy of around 75 years, and the R_0 of rubella is typically quoted in the 4–10 range, so according to $\bar{a} \simeq L/(R_0 - 1)$ the peak in circulation is predicted to be in an interval around 10 years of age.

We can do a more refined scenario-analyses regarding consequences of vaccination using the spline model. We focus on the 0–45-year age-range as this spans the pre to post child-bearing age:

```
data3 = peru[peru$age < 45, ]
df = 5
para = rep(0.1, df + 1)
```

We use a log-transformation to constrain the FoI to be positive, create the “dummy” `lm`-object, and define the function to evaluate the negative log-likelihood of the FoI curve given the data:

```
#Prediction function
tmpfn=function(x,d1){
  x=predict(d1, newdata=data.frame(age=x))
  exp(x)}
#Dummy lm-object
d1=lm(cumulative~bs(age,df), data=data3)
#Log-likelihood function
tmpfn2=function(par,data, df){
  d1=lm(cumulative~bs(age,df), data=data)
  d1$coefficients=par
  ll=0
  for(a in 1:length(data$age)){
    p=((1-exp(-integrate(tmpfn, 0,data$age[a],
      d1=d1)$value)))
    ll=ll+dbinom(data$cumulative[a], data$n[a], p, log=T)
  }
  return(-ll)
}
```

Getting a good fit is, again, computationally expensive, but reveals an interesting two-peaked force-of-infection (Fig. 4.7): A dominant peak just under 10 years and a subdominant peak around 35. A plausible scenario is that most people get infected in school but the fraction that escapes this dominant mode of infection are most likely to contract the virus from their children when they reach school age.

```
#Fit model
dspline.a45=df5=optim(par=log(para), fn=tmpfn2,
  data=data3, df=df, method="Nelder-Mead",
```

```

control=list(trace=4, maxit=5000)
#Overwrite dummy-objects coefficients with MLEs
dl$coefficients=dspline.a45.df5$par
plot(exp(predict(dl))~data3$age, xlab="Age",
      ylab="Relative FoI", type="l")

```

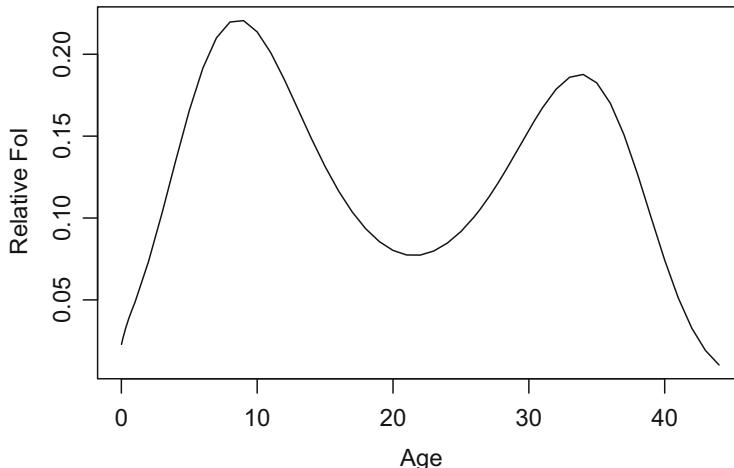


Fig. 4.7 The relative age-specific FoI of Rubella in Peru as estimated using the spline model

The fraction of cases that is predicted to occur in the child-bearing age-bracket (say, 15–40 years of age) is the joint probability of not being infected by age 15 and the probability of being infected in the 15–40 age range.

$$\exp\left(-\int_0^{15} \phi(a) da\right)\left(1 - \exp\left(-\int_{15}^{40} \phi(a) da\right)\right) \quad (4.2)$$

We can predict this fraction from the spline model.

```

(exp(-integrate(tmpfn, 0, 15, dl=dl)$value))*(1-
exp(-integrate(tmpfn, 15, 40, dl=dl)$value))

## [1] 0.08815273

```

Thus, with the current pattern of circulation just over 9% of the cases are predicted to occur in the at-risk age group. Let us ask how this fraction will change with a flat 50% reduction in FoI.

```
redn=0.5
(exp(-redn*integrate(tmpfn,0,15,d1=
d1)$value)) * (1-exp(-redn*integrate(tmpfn,
15,40,d1=d1)$value))

## [1] 0.2376147
```

The reduction in FoI results, as predicted by theory, in an increase in the mean age of infection (in reality this will also likely lead to a change in the age-specific FoI curve), so that almost 24% of cases is predicted to fall in the at-risk group. Assuming an associated 50% reduction in cases, the total number in the age-bracket of concern would thus *increase* given this intervention—predicting an intervention-induced enhancement of the public health problem as was seen in Greece during the 1990s (Panagiotopoulos et al. 1999).

Metcalf and Barrett (2016) discuss public health issues related to the possible introduction of vaccines against Zika virus, which can cause microcephaly in children of mothers infected during pregnancy, in light of the lessons learnt from rubella. Whooping cough is another vaccine preventable disease that causes significant morbidity and mortality in perinatal children. Lavine et al. (2011) discuss how an imperfect (waning) vaccine could increase circulation among people of child-bearing age and thus increase the risk of parent-newborn transmission. They recommended that cocoon-vaccination of expecting parents should be considered if the current acellular vaccine is as leaky as is feared (Warfel et al. 2014). Althouse and Scarpino (2015) provide further discussion of the utility of cocoon-vaccination and other interventions.

4.7 WAIFW

Age-structured FoIs result from non-assortative mixing among different age groups. The Who-Acquires-Infection-From-Whom (WAIFW) matrix is used to describe the patterns of nonhomogenous mixing among different age groups (Grenfell and Anderson 1989). Mossong et al. (2008) conducted a diary-based social study to map age-stratified contact rates for various countries in Europe as part of the POLY-MOD project. The contact rates by contactor and contactee are provided in the `mossong`-data set. We can visualize the diary data using an image plot with contours superimposed (Fig. 4.8)

```
data(mossong)
head(mossong)

##   contactor contactee contact.rate
## 1          1          1     120.37234
## 2          2          1      33.45833
## 3          3          1      23.13380
```

```

## 4          4          1      24.33333
## 5          5          1      29.00662
## 6          6          1      14.50331

x=y=mossong$contactor[1:30]
z=matrix(mossong$contact.rate, ncol=30, nrow=30)
image(x=x, y=y, z=z, xlab="Contactor",
      ylab="Contactee", col=gray((12:32)/32))
contour(x=x, y=y, z=z, add=TRUE)

```

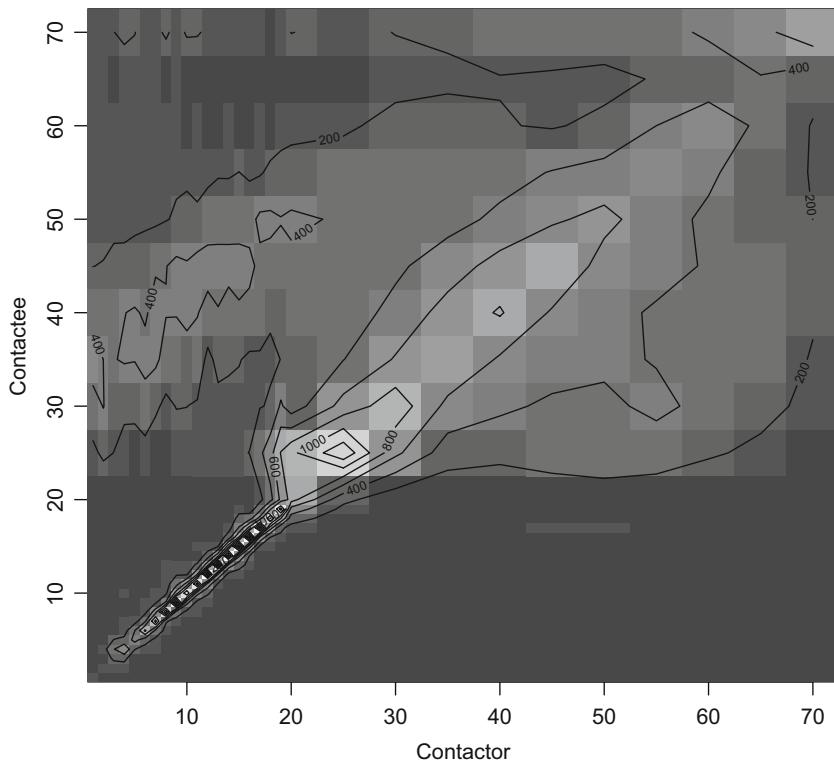


Fig. 4.8 The contact rates reported in the diary study of Mossong et al. (2008)

The reported contact rates are not symmetrical—which a WAIFW matrix will be—because of age-specific biases in diary entry rates as well as the age-profile of the contactors *versus* contactees. Before we “symmetrize” the matrix, we look at the reported marginal contact rate for each age group. Most contacts are among same-aged individuals and school-age children have the greatest number of contacts (Fig. 4.9). We do, however, also see off-diagonal ridges resulting from parent-offspring or children teacher interactions.

```
plot(apply(z, 1, mean) ~ x, ylab="Total contact rate",
      xlab="Age")
```

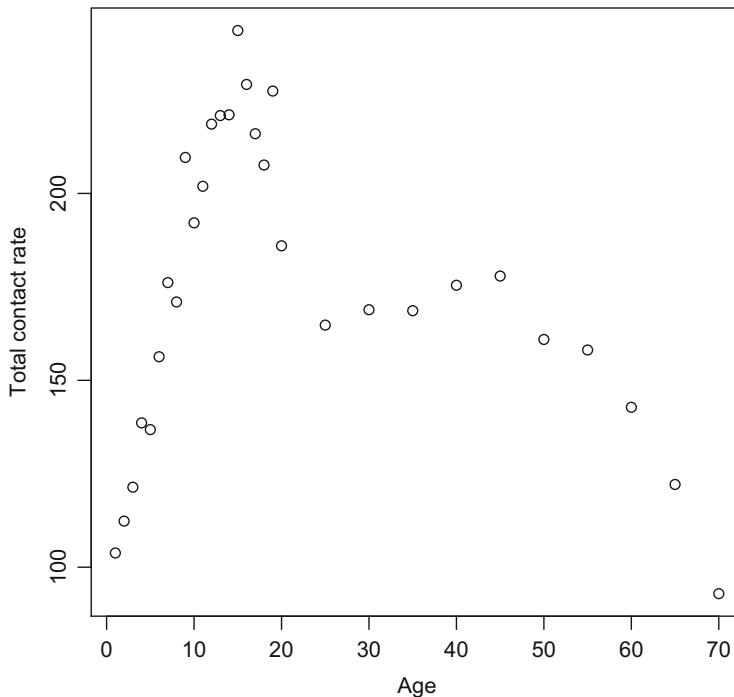


Fig. 4.9 The age-specific contact rates reported by the diary study of Mossong et al. (2008)

The symmetrized contact rate matrix (Fig. 4.10) is an estimate of the “WAIFW”-matrix.

4.8 Advanced: RAS Model

Schenzle (1984) discussed the importance of age-structured mixing when modeling infectious disease dynamics. Bolker and Grenfell (1993) extended this model to the “realistic age-structured (RAS) model” which in its full elaboration is an age-structured compartmental model with discrete aging of each birth cohort (at the beginning of each school year) and seasonality in transmission. Seasonality is the topic of Chap. 5. We can incorporate the POLYMOD contact matrix in a simpler age-structured model. We will make the simplifying assumptions that individuals age exponentially with rates set such that they will on average spend the right amount in each age-bracket. This allows us to formulate the model using chains of

ordinary differential equations. The upper-age cut-offs and age-progression rates for the $n = 30$ age categories are \mathbf{x} and

```
a = c(1/diff(x), 0)
```

We can in principle use the raw symmetrized WAIFW matrix in our model, but we will use a thin-plate spline smoothed matrix using the `Tps`-function in the `fields`-package. The smoothing protocol also allows interpolation to use different age-brackets for the model than used in the contact survey whenever necessary (Fig. 4.10).

```
require("fields")
n=length(x)
z2=(z+t(z))/2
z3=as.vector(z2)
xy=data.frame(x=rep(x[1:n], n), y=rep(y[1:n], each=n))
polysmooth=Tps(xy, z3, df=100)
surface(polysmooth, xlab="", ylab="",
        col=gray((12:32)/32))

## [1] 6400      2
```

For our age-structured SIR model we first normalize the WAIFW matrix:

```
W=matrix(polysmooth$fitted.values[,1]/mean(polysmooth$fitted.values), nrow=n)
```

The age-specific force-of infection is $\phi = \beta \mathbf{WI}/N$. The age-structured SIR model is thus (in log-coordinates)⁶:

```
siragemod = function(t, logx, params){
  n=length(params$a)
  x = exp(logx)
  S = x[1:n]
  I = x[(n+1):(2*n)]
  R = x[(2*n+1):(3*n)]
  with(as.list(params), {
    phi = (beta*W%*%I)/N
    dS = c(mu, rep(0,n-1)) - (phi+a)*S +
      c(0,a[1:n-1]*S[1:n-1])*(1-p) - mu*S
    dI = phi*S + c(0,a[1:n-1]*I[1:n-1]) -
      (gamma+a)*I - mu*I
    dR = c(0,a[1:n-1]*S[1:n-1])*p +
      c(0,a[1:n-1]*R[1:n-1]) + gamma*I -
```

⁶ Recall that the `with(as.list(...))` allows us to evaluate the equations using the definitions in the `params`-vector.

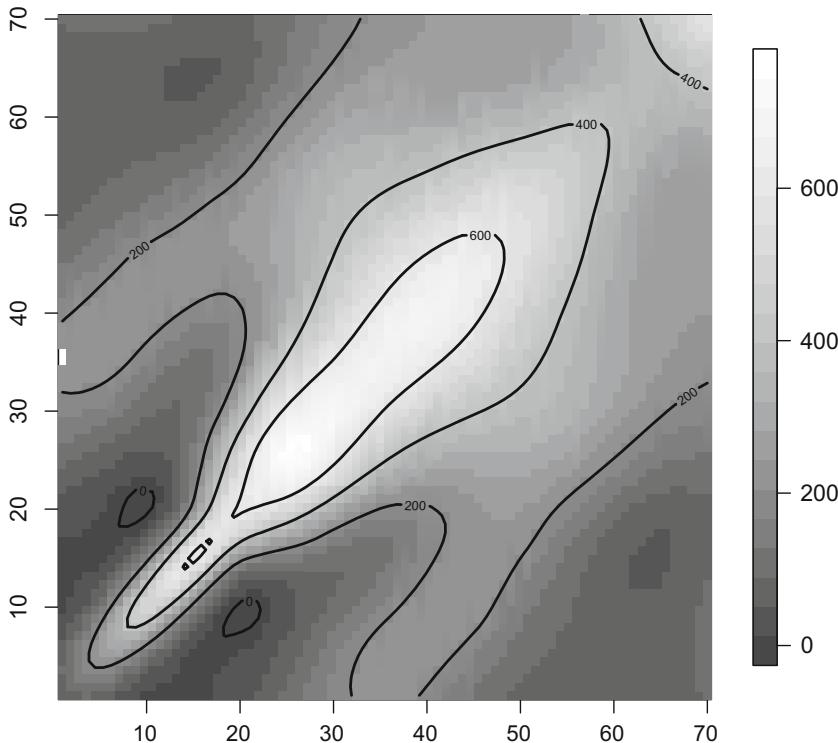


Fig. 4.10 The thin-plate spline smooth estimate of the WAIFW

```

    a*R - mu*R
    res = c(dS/S, dI/I, dR/R)
    list((res))
  })
}
}
```

where S , I , and R are vectors of length n , ϕ is the age-specific force of infection predicted by the WAIFW matrix, and p is a vector of length n that allows for age-specific vaccination rates (we will assume no vaccination). The a -vector sets appropriate aging rates when age groups vary in duration. We use the following parameters and initial conditions:

```

p.pre=rep(0,n)
pars.pre =list(N=1, gamma=365/14, mu=0.02, sigma=0.2,
  beta=100, W=W,p=p.pre, a=a)
ystart=log(c(S=rep(0.099/n,n), I=rep(0.001/n,n),
  R=rep(0.9/n,n)))
}
```

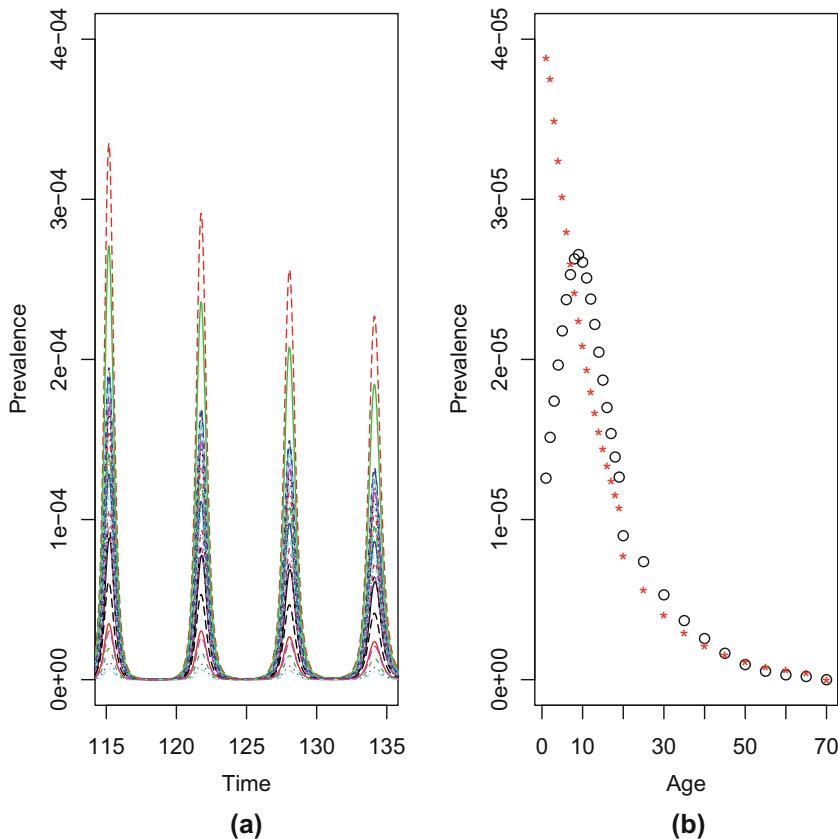


Fig. 4.11 The age-specific prevalences from the age-structured SIR model. (a) Trajectory through time. (b) Equilibrium age-incidence curves for the polymod matrix (o) vs homogenous mixing (*)

and integrate to plot the age-specific I-dynamics (Fig. 4.11a) and equilibrium age-specific prevalence (Fig. 4.11b) for the polymod matrix. Figure 4.11b also shows the predicted age-prevalence curve for the age-structured model with homogenous mixing.

```
times=seq(0,500,by=14/365)
#Polymod mixing
out=as.data.frame(ode(ystart, times,
    func=siragemod, parms=pars.pre))
par(mfrow=c(1,2)) #Room for side-by-side plots
#Time series
matplot(times, exp(out[,32:61]), type="l", xlab="Time",
    ylab="Prevalence", xlim=c(50,90), ylim=c(0, 0.0005))
#Final age-prevalence curve
plot(x, t(exp(out[13036,32:61])*a), ylab="Prevalence",
```

```
    xlab="Age", ylim=c(0, 4E-5))  
#Homogenous mixing:  
pars.pre$W=matrix(1, ncol=30, nrow=30)  
out2=as.data.frame(ode(ystart, times=times,  
    func=siragemod, parms=pars.pre))  
points(x, t(exp(out2[13036,32:61])*a), col=2, pch="*")
```

In contrast to the model with homogenous mixing which predicts that age-intensity curves decay exponentially with age, the RAS model can lead to a variety of age-incidence curves including the hump-shaped curve with a mode at around 10 years seen in Fig. 4.11b.

Chapter 5

Seasonality



5.1 Environmental Drivers

Host behavior and environmental factors influence disease dynamics in a variety of ways through affecting the parasite/pathogen—the survival of infective stages outside the host, speed of development of free-living stages, etc.; and the host population—changing birth-rates, carrying capacity, social organization, etc. Sometimes such influences have relatively subtle consequences (e.g., slight changes in R_0) as is likely the effect of absolute humidity on influenza transmission (Lowen et al. 2007). Other times the consequences are substantial by changing the dynamics qualitatively such as inducing multiannual or chaotic epidemics (Dalziel et al. 2016) or initiating ecological cascades (Jones et al. 1998; Glass et al. 2000). It is useful to distinguish between trends, predictable variability (such as seasonality), and non-predictable variability due to “environmental” and “demographic” stochasticity (see Chap. 7).

Some level of seasonality in transmission is very common in infectious disease dynamics and is usually reflected in seasonal cycles in incidence (Altizer et al. 2006); Seasonality in *incidence* is the norm even for persistent infections for which prevalence may remain relatively stable. Influenza is the poster-child for seasonality in infection risk in the public eye (e.g., Bjørnstad and Viboud 2016). Figure 5.1a shows the mean weekly influenza-related deaths in Pennsylvania between 1972 and

This chapter uses the following R-packages: `deSolve` and `plotrix`.

A conceptual understanding of *seasonality* is useful prior to this discussion. A 5-min epidemics-MOOC can be seen on YouTube: <https://www.youtube.com/watch?v=TDuuM-wm6nw>.

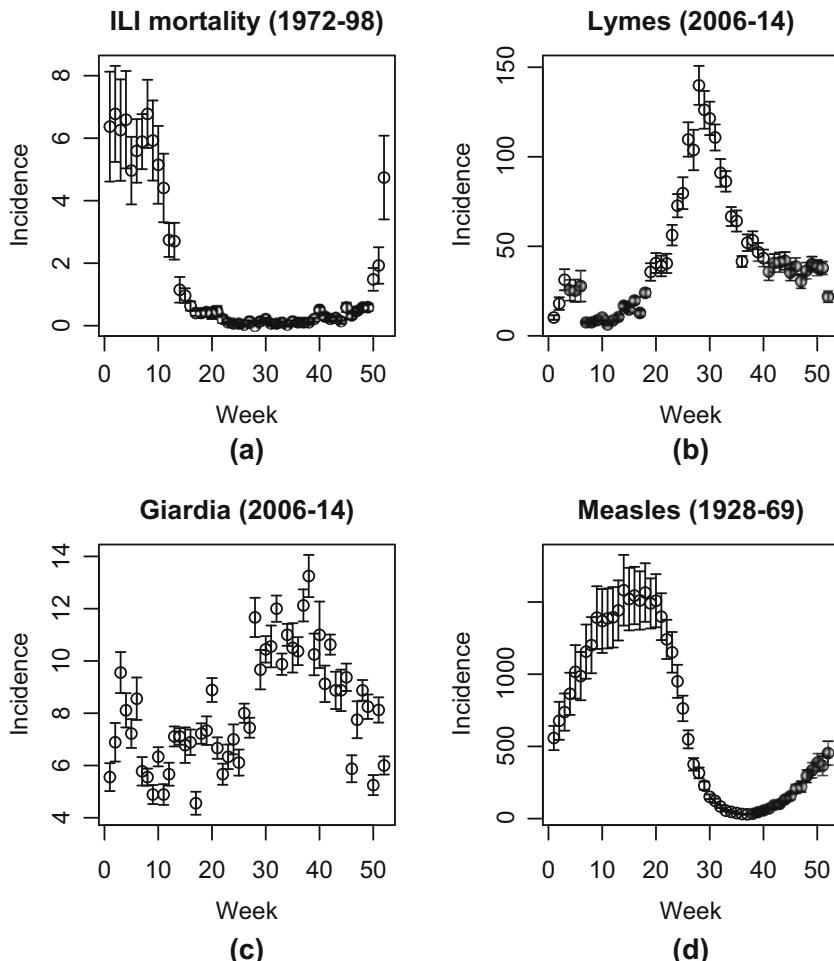


Fig. 5.1 Mean (± 1 SD) weekly incidence of (a) deaths due to influenza like illness, (b) Lyme's disease, (c) giardiosis, and (d) pre-vaccination measles in Pennsylvania

1998. The pronounced winter-peaked seasonality is not fully understood, but are thought to be linked to how climate conditions—notably absolute humidity (Shaman and Kohn 2009)—affect rates of viral degradation outside the host.

We can illustrate various types of seasonality using four diseases in Pennsylvania contained in the `paili`, `palymes`, `pagiard`, and `pameasle` data sets. We first write a simple function to extract and plot weekly average incidence (and standard errors) through the year from time series. Weekly incidence data occasionally has 53 reporting weeks (because years are 52.14 weeks, and leap years are 52.28 weeks). The function omits these extras.

```
ppp=function(wk, x){
  require(plotrix)
  x=x[wk<53]
  wk=wk[wk<53]
  ses=sapply(split(x, wk), mean, na.rm=TRUE)
  sesdv=sapply(split(x, wk), sd, na.rm=
    TRUE)/sqrt(length(split(x, wk)))
  plotCI(x=c(1:52), y=ses, ui=ses+sesdv,
    li=ses-sesdv, xlab="Week", ylab="Incidence")
}
```

We apply the function to influenza-like illness, Lyme's disease, giardiosis, and measles (Fig. 5.1):

```
par(mfrow = c(2, 2)) #A four panel plot
ppp(paili[, "WEEK"], paili[, "PENNSYLVANIA"])
title("ILI mortality (1972-98)")
ppp(palymes[, "WEEK"], palymes[, "PENNSYLVANIA"])
title("Lyme (2006-14)")
ppp(pagiard[, "WEEK"], pagiard[, "PENNSYLVANIA"])
title("Giardia (2006-14)")
ppp(pameasle[, "WEEK"], pameasle[, "PENNSYLVANIA"])
title("Measles (1928-69)")
```

Seasonality arises from a variety of causes depending on the mode of transmission of the pathogen: air-borne (like influenza), vector-borne, or water/food-borne. Lyme's disease, for example, is caused by tick-vectored bacteria in the genus *Borrelia*. Figure 5.1b shows the sharply seasonal incidence of human cases of Lyme's in Pennsylvania between 2006 and 2014. The seasonality is the combined effect of seasonality in tick activity levels and human use of wilderness. Most mosquito-vectored pathogens also show strong seasonality because of the temperature- and precipitation-dependence of the vector life cycle. The seasonality of cholera infections, caused by the *Vibrio cholerae* bacterium, is among the most studied water-borne pathogens. The seasonality in southeast Asia is caused by rainfall variation associated with the monsoon season (Codeço 2001; Ruiz-Moreno et al. 2007) (Fig. 1.3b). However, other water-borne diseases like giardiasis also show marked seasonality (Fig. 5.1c). Host behavior can further cause seasonality in contact rates. Childhood disease dynamics, for example, are often shaped by “term-time” forcing: increased transmission when schools are open (e.g., Fine and Clarkson 1982; Kucharski et al. 2015). Weekly average pre-vaccination incidence of measles in Pennsylvania, for instance, collapses as school closes for the summer only to resume robust circulation after the vacation end (Fig. 5.1d). Additionally, seasonal urban-rural migration in Niger has been shown to generate strong seasonality in measles transmission (Ferrari et al. 2008). Seasonally varying birth rates can induce seasonality in susceptible recruitment in wildlife (Peel et al. 2014) and humans (Martinez-Bakker et al. 2014).

5.2 The Seasonally Forced SEIR Model

To study the effect of seasonality in transmission we will modify the SEIR model (Eqs. (3.3)–(3.5)). We first define the gradient functions for the “undriven” system:

```
seirmod = function(t, y, parms) {
  S = y[1]
  E = y[2]
  I = y[3]
  R = y[4]

  mu = parms["mu"]
  N = parms["N"]
  beta = parms["beta"]
  sigma = parms["sigma"]
  gamma = parms["gamma"]

  dS = mu * (N - S) - beta * S * I/N
  dE = beta * S * I/N - (mu + sigma) * E
  dI = sigma * E - (mu + gamma) * I
  dR = gamma * I - mu * R
  res = c(dS, dE, dI, dR)
  list(res)
}
```

We simulate 10 years of dynamics using the basic recipe introduced in Sect. 2.2. The seasonally forced SEIR model has been very successfully applied to understand the dynamics of measles (and other immunizing childhood infections). To simulate measles-like dynamics we assume a latent period of 8 days and an infectious period of 5 days. We assume the initial host population to be 0.1% infectious, 6% susceptibles, and the rest immune; The R_0 of measles is typically quoted in the 13–20 range, which means that the equilibrium fraction of susceptibles is somewhere around 5%. For simplicity we assume a host life span of 50 years and set $N = 1$ to model the fraction in each compartment.

```
require(deSolve)
times = seq(0, 10, by=1/120)
paras = c(mu = 1/50, N = 1, beta = 1000,
          sigma = 365/8, gamma = 365/5)
start = c(S=0.06, E=0, I=0.001, R = 0.939)
```

As discussed in Sect. 3.7, the R_0 for this system—assuming disease induced mortality is negligible—is $\frac{\sigma}{\sigma+\mu} \frac{\beta}{\gamma+\mu}$. We can verify that our choice of β places R_0 in the “measles-like” range. We use expression to define the equation for R_0 . As in

Sect. 4.8, we use `with(as.list(...))` to evaluate the expression using the definitions in the `paras`-vector.

```
R0 = expression(sigma/(sigma + mu) * beta/(gamma + mu))
with(as.list(paras), eval(R0))

## [1] 13.68888
```

We integrate the ODEs and plot the time series and the phase plane (Fig. 5.2). As is the case with the SIR model, the unforced SEIR model predicts damped oscillations toward the endemic equilibrium.

```
out = as.data.frame(ode(start, times, seirmod, paras))
par(mfrow = c(1,2)) #Two plots side by side
plot(times, out$I, ylab = "Prevalence",
      xlab = "Time", type = "l")
plot(out$S, out$I, ylab = "Prevalence",
      xlab = "Susceptible", type = "l")
```

5.3 Seasonality in β

The predicted damped oscillations toward an equilibrium is at odds with the recurrent outbreaks seen in many immunizing infections (e.g., Fig. 1.4). Sustained oscillations require either additional predictable seasonal drivers—the topic of this chapter—or stochasticity (Sect. 7.1). An important driver in human childhood infections is seasonality in contact rates because of aggregation of children during the school term (Fine and Clarkson 1982; Kucharski et al. 2015). For simplicity we can analyze the consequences of seasonality by assuming sinusoidal forcing on the transmission rate¹ according to $\beta(t) = \beta_0(1 + \beta_1 \cos(2\pi t))$. The mean transmission rate is β_0 but the realized transmission varies cyclically with a period of one time unit, and the magnitude of the seasonal variation is controlled by the parameter β_1 . The modified gradient function is:

```
seirmod2=function(t, y, parms){
  S=y[1]
  E=y[2]
  I=y[3]
  R=y[4]
  with(as.list(parms), {
```

¹ It is possible to analyze more realistic patterns of seasonality, such as a more explicit “term-time” forcing; see Keeling et al. (2001) and Chap. 7. The qualitative (but not detailed) results appear to be robust to the exact shape of the forcing function.

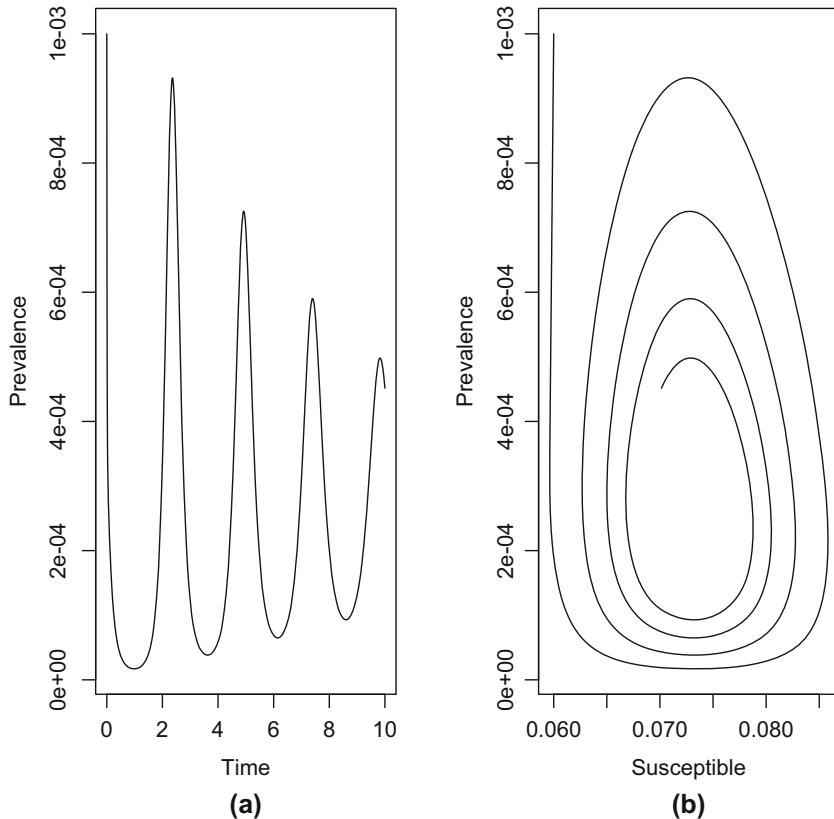


Fig. 5.2 Predicted prevalence from the SEIR model **(a)** in time, and **(b)** in the phase plane with $\mu = 1/50$, $N = 1$ (to model fractions), $\beta = 1000$, $\sigma = 365/8$, and $\gamma = 365/5$. Ten years are not long enough for the simulation to settle on the endemic equilibrium

```

dS = mu * (N - S) - beta0 * (1+beta1 *
      cos(2 * pi * t)) * S * I / N
dE = beta0 * (1 + beta1 * cos(2*pi * t)) *
      S * I / N - (mu + sigma) * E
dI = sigma * E - (mu + gamma) * I
dR = gamma * I - mu * R
res=c(dS, dE, dI, dR)
list(res)
})
}
}
```

With no seasonality the model predicts damped oscillation, with moderate seasonality the prediction is low-amplitude annual outbreaks. However, as seasonality increases (to $\beta_1 = 0.2$, say) we start seeing some surprising consequences of the sea-

sonal forcing (Fig. 5.3): the appearance of harmonic resonance between the internal cyclic dynamics of the SEIR clockwork and the annual seasonal forcing function.

```
times = seq(0, 100, by=1/120)
paras = c(mu = 1/50, N = 1, beta0 = 1000, beta1 = 0.2,
          sigma = 365/8, gamma = 365/5)
start = c(S=0.06, E=0, I=0.001, R = 0.939)
out = as.data.frame(ode(start, times, seirmod2, paras))
par(mfrow=c(1,2)) #Side-by-side plot
plot(times, out$I, ylab="Infected", xlab="Time",
      xlim=c(90, 100), ylim=c(0,
      max(out$I[11001:12000])), type="l")
plot(out$S[11001:12000], out$I[11001:12000],
      ylab="Infected", xlab="Susceptible", type="l")
```

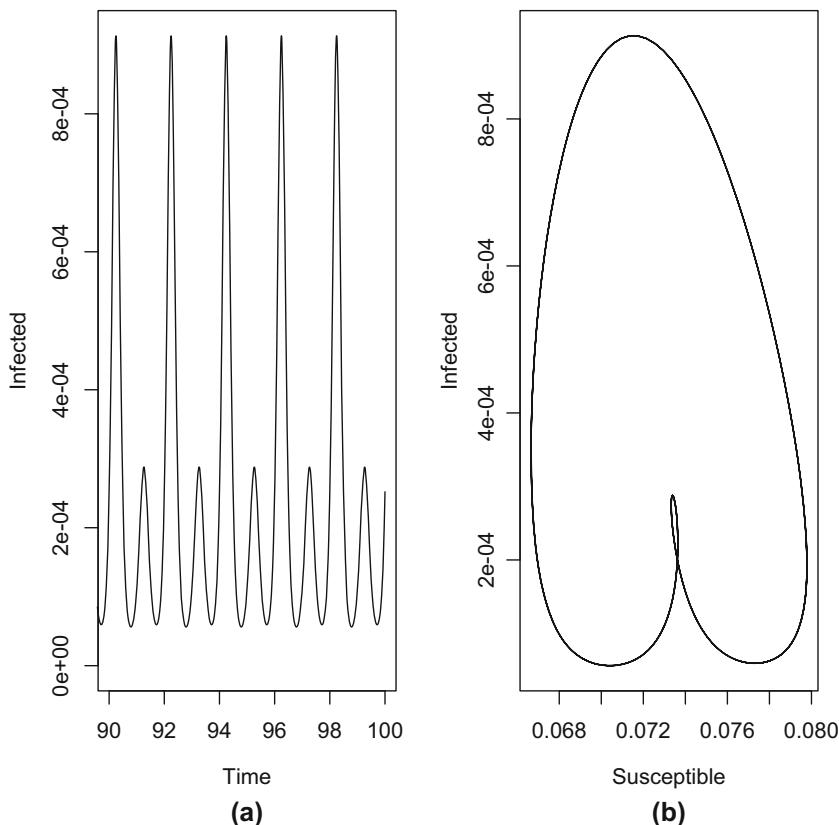


Fig. 5.3 The 10 last years of the forced SEIR model for $\beta_1 = 0.2$. (a) Predicted prevalence and (b) the S-I phase plane

The emergent pattern of recurrence in the forced SEIR is the result of an interaction between the internal periodic clockwork (the “damping period”) of the SEIR flow and the externally imposed periodic forcing. The damping period is the focus of Chap. 9; however, we can use the results previewed in Sect. 2.6: When working with a continuous-time ODE model which results in cyclic behavior like the SEIR model, the dominant eigenvalues of the [Jacobian matrix](#)—when evaluated at the equilibrium—are a conjugate pair of complex numbers ($a \pm bi$) that determines the period of the cycle according to $2\pi/b$.

The endemic equilibrium of the SEIR model is (ignoring the absorbing R compartment): $S^* = 1/R_0$, $I^* = \mu(1 - 1/R_0)R_0/\beta$ and $E^* = (\mu + \gamma)I^*/\sigma$. We first calculate the endemic equilibrium:

```
mu = paras["mu"]
N = paras["N"]
beta0 = paras["beta0"]
beta1 = paras["beta1"]
sigma = paras["sigma"]
gamma = paras["gamma"]
R0 = sigma/(sigma + mu) * beta0/(gamma + mu)

Sstar = 1/R0
Istar = mu * (1 - 1/R0) * R0/beta0
Estar = (mu + gamma) * Istar/sigma

eq = list(S = Sstar, E = Estar, I = Istar)
```

We next use R's inbuilt D-function to carry out symbolic differentiation, and to generate and evaluate the Jacobian matrix (ignoring the absorbing R compartment):

```
ds = expression(mu * (N - S) - beta0 * S * I / N)
dE= expression(beta0 * S * I / N - (mu + sigma) * E)
dI = expression(sigma*E - (mu + gamma) * I)

j11 = D(ds, "S"); j12 = D(ds, "E"); j13 = D(ds, "I")
j21 = D(dE, "S"); j22 = D(dE, "E"); j23 = D(dE, "I")
j31 = D(dI, "S"); j32 = D(dI, "E"); j33 = D(dI, "I")

J=with(eq,
matrix(c(eval(j11),eval(j12),eval(j13),
eval(j21),eval(j22),eval(j23),
eval(j31),eval(j32),eval(j33)),
nrow=3, byrow=TRUE))
```

We finally calculate the eigenvalues. The dominant pair of complex conjugates is at the second and third place in the vector of eigenvalues. The associated resonant period is:

```
round(eigen(J)$values, 3)
## [1] -118.725+0.000i   -0.107+2.667i   -0.107-2.667i
2 * pi/(Im(eigen(J)$values[2]))
## [1] 2.355891
```

So the recurrent biennial epidemics are sustained because the internal epidemic clock-work cycles with a period of 2.3 years, but it is forced at an annual time scale, so as a “compromise” the epidemics are locked on to the annual clock, but with alternating major and minor epidemics such as seen, for example, in pre-vaccination measles in New York 1944–1958 (Fig. 1.4b) and London 1950–1965 (Fig. 1.4c).

5.4 Bifurcation Analysis

We can make a more comprehensive summary of the consequences of seasonality on the SEIR-flow using a bifurcation analysis: a systematic search across a range of β_1 values. For annually forced models we study the dynamics by “strobing” the system once each year. To study the long-term (asymptotic) dynamics we discard the initial transient part of the simulation. In the below we hence use one data point per year for the last 42 years of simulation—which the `sel` variable flags—so that an annual cycle produces a single value (so will a fixed-point equilibrium), biannual cycles two values, etc. The resultant bifurcation plot shows when annual epidemics gives way to biannual cycles and finally chaotic dynamics as seasonality increases (Fig. 5.4). The irregular dynamics with strong seasonality comes about because there is no simple resonant compromise between the internal clock and the external forcing function. We may think of it as “resonance” giving place to “dissonance” in the dynamical system. That stronger seasonality pushes measles from regular to irregular epidemics has been predicted by the theoretical literature (e.g., Aron and Schwartz 1984) and is supported by an empirical comparison of measles in pre-vaccination UK *versus* USA by Dalziel et al. (2016) (see Sect. 10.2).

We define initial conditions and the sequence of parameter values to be considered for β_1 and then do the numerical integration for each parameter set:

```
times = seq(0, 100, by = 1/120)
start = c(S = 0.06, E = 0, I = 0.001, R = 0.939)
beta1 = seq(0, 0.25, length=101)
```

```
#Matrix to store infecteds
Imat = matrix(NA, ncol = 12001, nrow = 101)
#Loop over beta1's
for(i in 1:101){
  paras = c(mu = 1/50, N = 1, beta0 = 1000,
            beta1=beta1[i], sigma = 365/8, gamma = 365/5)
  out = as.data.frame(ode(start, times,
                           seirmod2, paras))
  Imat[i,] = out$I
}
}
```

For the visualization we select one observation per year for the last 42 years of simulation and plot the values against the associated β_1 values (Fig. 5.4).

```
sel = seq(7001, 12000, by = 120)
plot(NA, xlim = range(beta1), ylim = c(1E-7,
                                         max(Imat[,sel])), log="y", xlab="beta1",
                                         ylab="prevalence")
for(i in 1:101){
  points(rep(beta1[i], length(sel)),
         Imat[i, sel], pch=20)
}
```

5.5 Stroboscopic Section

Rand and Wilson (1991) studied the seasonally forced SEIR model with a particular set of parameters resulting in chaotic dynamics. It is interesting to integrate the model with these parameters for a very long time (in this case for 10,000 years) to better understand/visualize the meaning of quasi-periodic chaos. Figure 5.5 shows a time series of prevalence and the dynamics in the S-I phase plane strobbed at the annual time scale—the annual “stroboscopic section” of the S-I plane. The time series is erratic, but the paired S-I series trace out a very intricate pattern (Fig. 5.5b): The four-armed shape corresponds to the propensity of the chaotic pattern to adhere to a wobbly (“quasiperiodic”) 4-year recurrence. We will revisit on this attractor and its role in facilitating “chaotic stochasticity” and “stochastic resonance” in disease dynamics in Chap. 10.

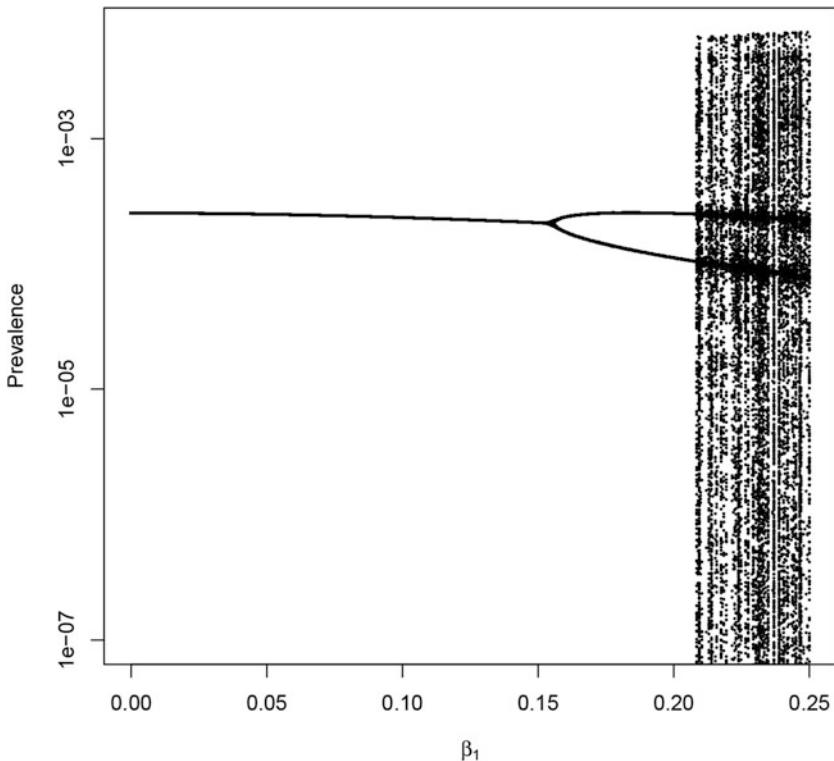


Fig. 5.4 The bifurcation plot of prevalence against seasonality for the forced SEIR model

```

times = seq(0, 100000, by = 1/120)
start = c(S = 0.06, E = 0, I = 0.001, R = 0.939)
paras = c(mu = 1/50, N = 1, beta0 = 1800, beta1=0.28,
          sigma = 35.84, gamma = 100)
out = as.data.frame(ode(start, times, seirmod2, paras))
sel=seq(7001, 12000000, by=120)
par(mfrow=c(1,2))
plot(out$time[7001:13001], out$I[7001:13001],
     type="l", xlab="Year", ylab="Prevalence")
plot(out$S[sel], out$I[sel], type="p", xlab="S",
     ylab="I", log="y", pch=20, cex=0.25)

```

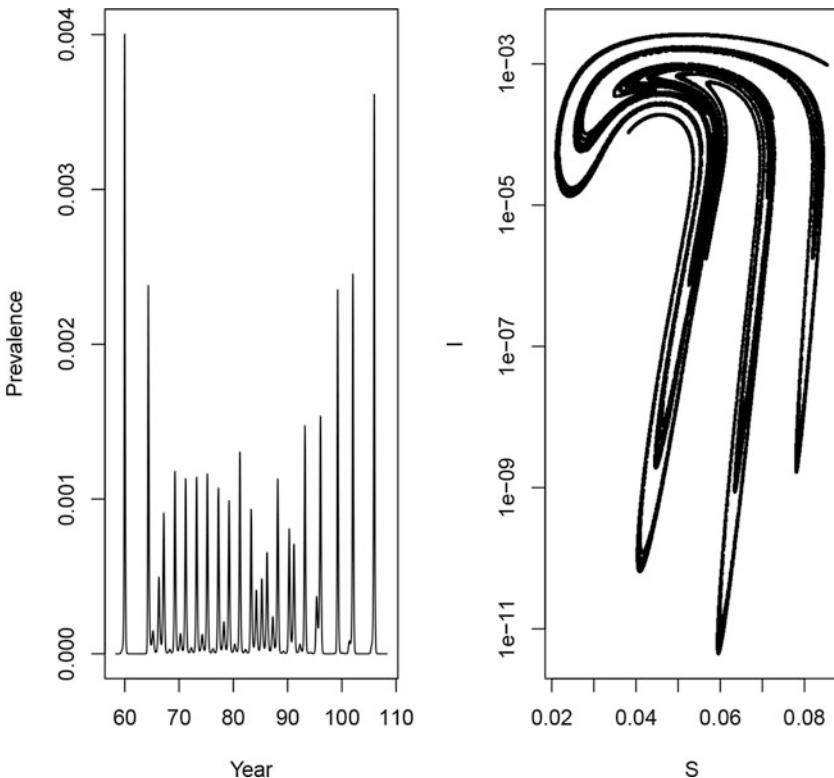


Fig. 5.5 Stroboscopic section in the S-I phase plane of the quasiperiodic chaotic prevalence of the seasonally forced SEIR model ($\mu = 0.02$, $\beta_0 = 1800$, $\beta_1 = 0.28$, $\sigma = 35.84$, $\gamma = 100$) (Rand and Wilson 1991)

5.6 Susceptible Recruitment

The patterns of recurrent epidemics are also shaped by other characteristics of the host and pathogen. Earn et al. (2000b) studied how susceptible recruitment affects dynamics of the seasonally forced SEIR model by doing a bifurcation analysis over μ (Fig. 5.6). As concluded by Earn et al. (2000b), reduced susceptible recruitment in the seasonally forced SEIR model leads to a cascade from annual to biennial to coexisting annual and complex attractors. To trace out the coexisting attractors it is necessary to use multiple starting conditions because each attractor will have its own “basin of attraction.” We do this by looping forwards and backwards over μ using the final values of the previous simulation as initial conditions for the next.

```
times = seq(0, 100, by = 1/120)
start = c(S = 0.06, E = 0, I = 0.001, R = 0.939)
mu=seq(from = 0.005, to = 0.02, length = 101)
ImatF=ImatB=matrix(NA, ncol = 12001, nrow = 101)
```

```

for(i in 1:101){
  paras = c(mu = mu[i], N = 1, beta0 = 2500,
            beta1=0.12, sigma = 365/8, gamma = 365/5)
  out = as.data.frame(ode(start, times, seirmod2,
                           paras))
  ImatF[i,] = out$I
  start = c(S = out$S[12001], E = out$E[12001],
            I = out$I[12001], R = out$R[12001])
}
start = c(S = 0.06, E = 0, I = 0.001, R = 0.939)
for(i in 101:1){
  paras = c(mu = mu[i], N = 1, beta0 = 2500,
            beta1=0.12, sigma = 365/8, gamma = 365/5)
  out = as.data.frame(ode(start, times, seirmod2,
                           paras))
  ImatB[i,] = out$I
  start = c(S = out$S[12001], E = out$E[12001],
            I = out$I[12001], R = out$R[12001])
}
sel=seq(7001, 12000, by=120)
par(mfrow=c(1,1))
  plot(NA, xlim=range(mu), ylim=range(ImatF[,sel]),
        log="y", xlab="mu", ylab="prevalence")
for(i in 1:101){
  points(rep(mu[i], dim(ImatF)[2]), ImatF[i, ],
         pch=20, cex=0.25)
  points(rep(mu[i], dim(ImatB)[2]), ImatB[i, ],
         pch=20, cex=0.25, col=2)
}

```

In Fig. 5.6 the attractor from the “forward” analysis is shown in black and “backward” analysis in red. This color coding clearly reveals the coexisting attractors for a range of parameter values. The transition from annual to biennial epidemics predicted as μ varies between 15 per thousand per year and 25 per thousand per year is clearly seen following the baby boom post-World War II in the dynamics of measles in the UK (Sect. 6.5). The complex dynamics at lower susceptible recruitment rates, again, comes about because of dissonance between the external annual forcing and the internal periodic clock. With a *per capita* susceptible recruitment rate of 0.002/year which corresponds to 90% vaccination rate in our model, the dampening period is predicted to be 7.4 years. Earn et al. (2000b) predicted that vaccination may—depending on seasonality—lead to chaotic epidemics. A complication that makes this less likely in real populations is that the troughs following major epidemics are so deep that the chain of transmission will almost always break, leading to disease fade-out (Ferrari et al. 2008) (though see Sect. 10.2 for a counter example). In his mathematical study of rabies spread, Mollison (1991) dubbed this

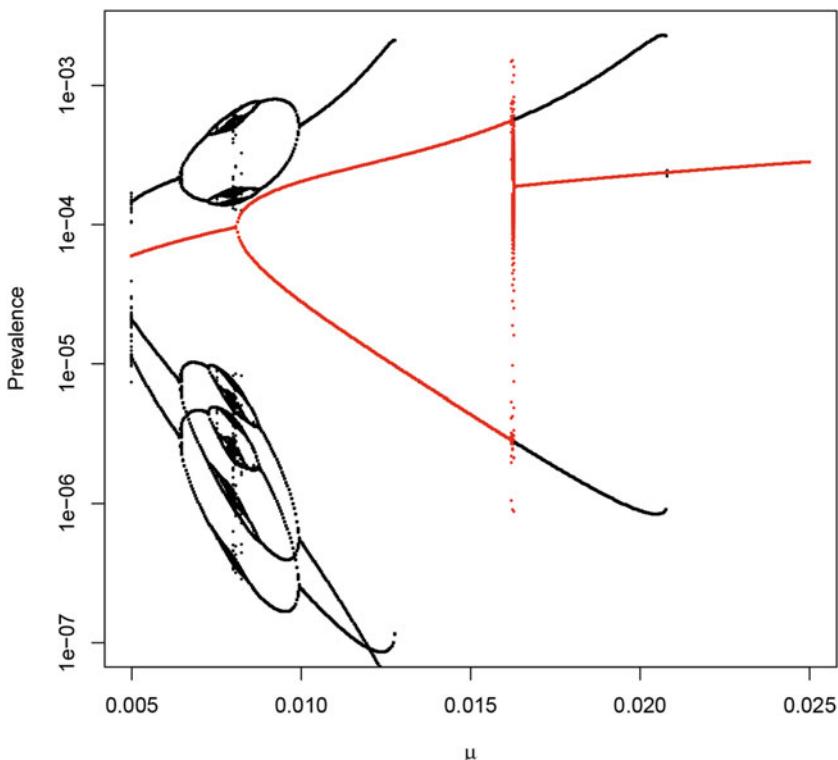


Fig. 5.6 The bifurcation plot of prevalence against birth rate μ for the forced SEIR model with intermediate seasonality. Black represents values from the “forward” analysis and red the “backward” analysis

the “atto-fox” of deterministic models; If the models predict that there is a 10^{-18} th of a rabid fox running around, deterministic predictions may not be very relevant to real-life epidemics.

5.7 ShinyApp

The seasonally forced SEIR model can be further studied using the `SEIR.app` in the `epimdr`-package. The app can be launched from R through:

```
require(shiny)
SEIR.app
```

Chapter 6

Time-Series Analysis



6.1 Taxonomy of Methods

Analysis of epidemic time series is a large endeavor because of the richness of dynamical patterns and plentitude of historical data (Rohani and King 2010). A wide range of tools are used, some of which are borrowed from mainstream statistics other of which are “custom made.” The classic “mainstream” methods belong to two categories: the so-called time-domain and frequency-domain methods. The autocorrelation function and ARIMA models belong to the former class and spectral analysis and the periodogram belong to the latter. Hybrid time/frequency methods have become increasingly prominent in the form of wavelet analysis because it allows the study of changes in disease dynamics through time (Grenfell et al. 2001). This chapter discusses a variety of “mainstream” methods using a variety of time-series data. Examples of “custom made” methods are mechanistic models such as the time-series SIR (TSIR) which is the focus of Chap. 7, semi-parametric models (Ellner et al. 1998) and nonparametric (“empirical dynamic”) models. An example of the latter is discussed in Sect. 10.8.

6.2 Time Domain: ACF and ARMA

The autocorrelation function (ACF) and the autoregressive-moving-average (ARMA) model are classic tools for describing serial dependence in time series in the time-domain. We first apply the ACF to the (weekly) time series of prevalence

This chapter uses the following R-packages: `forecast`, `Rwave`, `imputeTS`, `nlts`, and `plotrix`.

from the seasonally forced SEIR model. The ACF quantifies serial correlations at different time lags. Figure 6.1 shows the ACF for lags up to 3 years (=156 weeks):

```
times = seq(0, 100, by=1/52)
paras = c(mu = 1/50, N = 1, beta0 = 1000, beta1 = 0.2,
          sigma = 365/8, gamma = 365/5)
xstart = c(S=0.06, E=0, I=0.001, R = 0.939)
out = as.data.frame(ode(xstart, times, seirmod2, paras))

par(mfrow=c(1, 2))
plot(times, out$I, ylab="Infected", xlab="Time",
      xlim=c(90,100), type="l")
acf(out$I, lag.max = 156, main="")
```

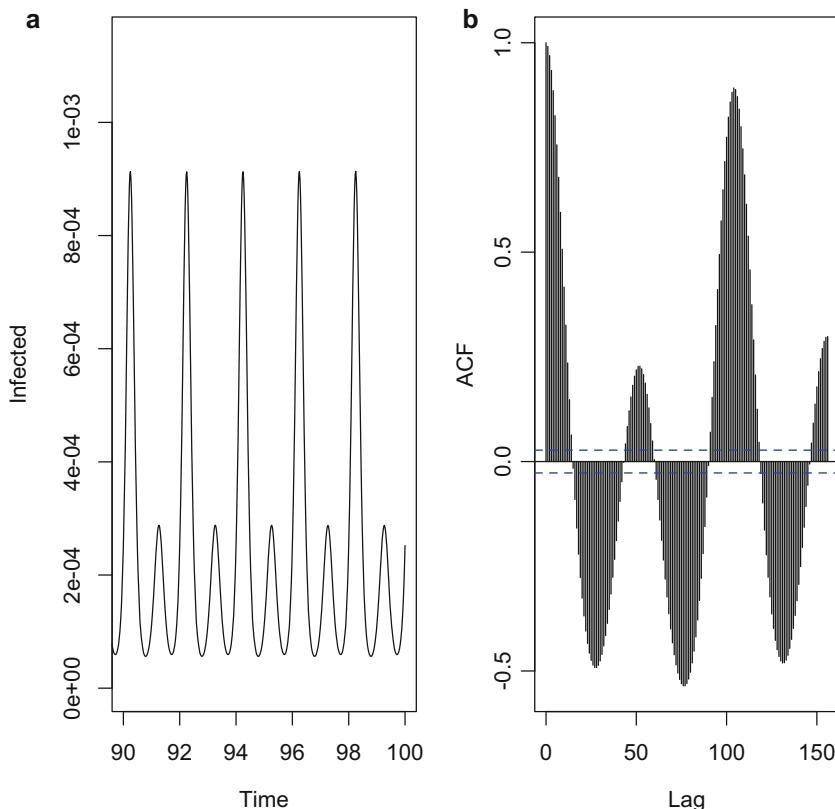


Fig. 6.1 The ACF of prevalence from the seasonally forced SEIR model. (a) time series, (b) ACF

The major peak in autocorrelation at 104 weeks reflects the dominant 2-year periodicity; the minor peak at 52 weeks reflects the subdominant annual periodicity.

6.2.1 ARMA

Autoregressive moving-average models have been used to forecast disease dynamics (e.g., influenza-like illness; Choi and Thacker 1981). The ARMA(p,q)-model assumes that the future incidence (Y_t) can be predicted according to $Y_t = a_1 Y_{t-1} + \dots + a_p Y_{t-p} + \varepsilon_t - b_1 e_{t-1} - \dots - b_q e_{t-q}$, where the ε 's represent stochasticity and the echo of past stochasticity.¹ We apply the ARMA model to monthly ILI incidence in Iceland using the `forecast`-package:

```
require(forecast)
data(Icelandflu)
```

We convert the data frame to a time-series `ts`-object and do a seasonal decomposition (Fig. 6.2). There is a slight trend through the data, but as expected the winter seasonality is the dominant feature of the time series. Because the epidemics are very peaky we consider square-root transformed numbers:

```
ilits=ts(sqrt(Icelandflu$ili), start=c(1980, 1),
         end=c(2009, 12), frequency=12)
plot(decompose(ilits))
```

We train a seasonal ARMA(2,1) model on for example 1990 through 2000 epidemics and do a 24-month forecast (Fig. 6.3):

```
wts=window(ilits, start=c(1990,6), end=c(2000,5))
fit = arima(sqrt(wts), order=c(2, 0, 1),
            list(order=c(1, 0, 0), period = 12))
coef(fit)

##           ar1          ar2          ma1          sar1
## 1.4460827 -0.7323795 -0.7819940  0.2026528
## intercept
## 2.4823415

fore = predict(fit, n.ahead=24)
```

¹ The ARMA model is usually considered a purely statistical model (i.e., not containing biological mechanism), though it can be shown that the linearized discrete-time SIR model with stochastic transmission can be rewritten as a ARMA(2,1) model (see Sect. 9.8).

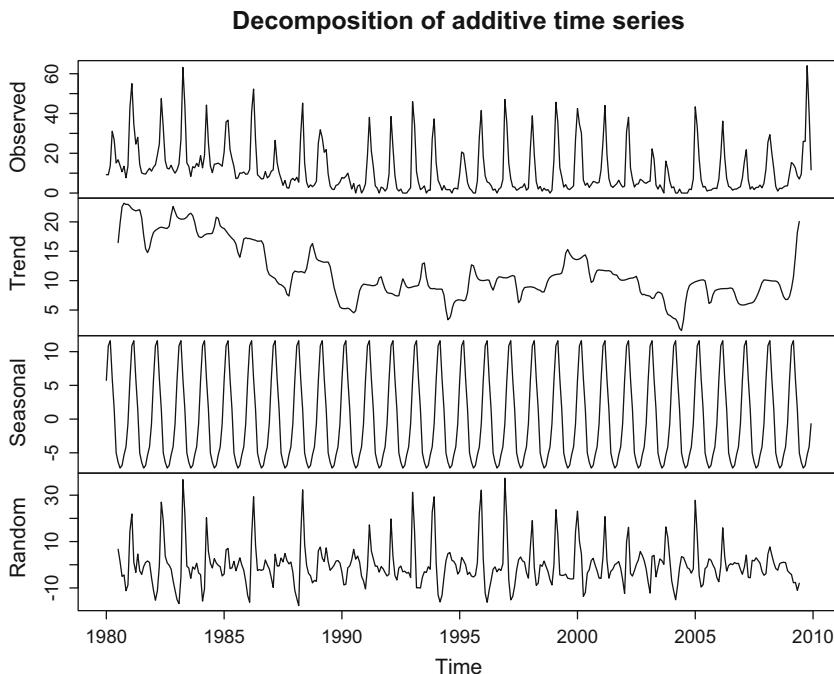


Fig. 6.2 A decomposition of the Iceland ILI time series

```
#Calculate approximate upper (U) and
#lower (L) prediction intervals
U = fore$pred + 2*fore$se
L = fore$pred - 2*fore$se
# plot observed and predicted values
ts.plot(sqrt(wts), fore$pred, U, L, col=c(1, 2, 4, 4),
        lty=c(1, 1, 2, 2), ylab="Sqrt(cases)")
legend("bottomleft", c("ILI", "Forecast",
                      "95% Error Bounds"), col=c(1, 2, 4), lty=c(1, 1, 2))
```

While ARMA forecasting is useful in many disciplines and is an important part of the broad statistical toolbox, it suffers from lacking mechanism and can therefore not answer questions like “how are dynamics likely to change if we vaccinate 50% of susceptible children?” It furthermore assumes that time series are *stationary*, essentially meaning that dynamical patterns do not change radically over time. As we frequently see in infectious diseases, this is not a good assumption. In Chap. 7 we discuss how time-series methods that incorporate more biological mechanisms (like the “time-series SIR” model) are better able to capture/predict dynamic transitions.

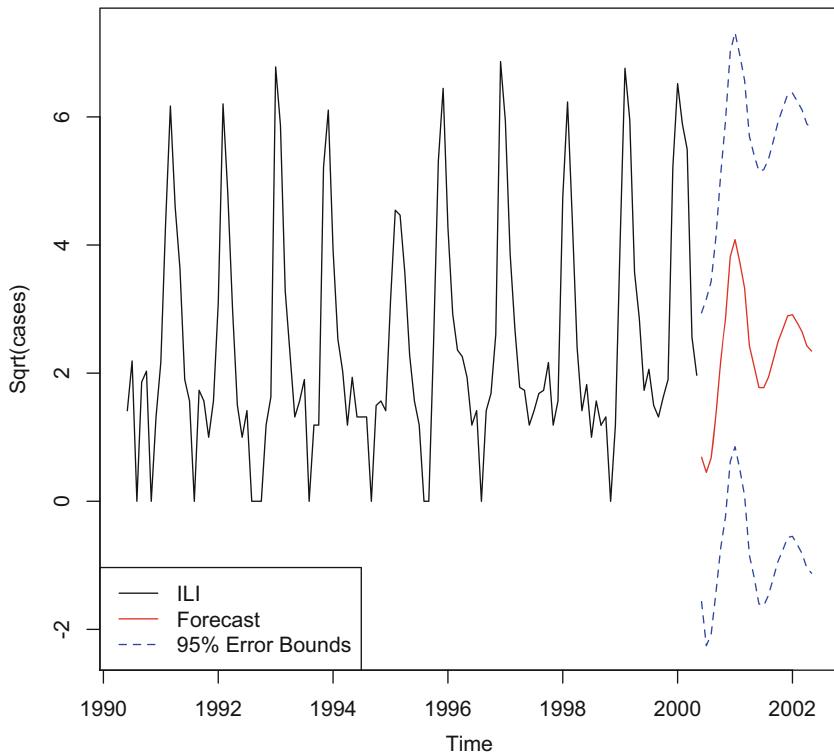


Fig. 6.3 Forecast of square-root transformed ILI incidence in Iceland for the 2001 and 2002 seasons using a seasonal ARMA(2,1) model

6.3 Frequency Domain

The Schuster periodogram is a direct way of estimating and testing for significant periodicity. The periodogram decomposes a time series into cycles of different frequencies (frequency = 1/period). The importance of each frequency is measured by the spectral amplitude. We use the spectrum-function to calculate the periodogram for the time series from the seasonally forced SEIR model. The analysis clearly identifies the two superimposed periods (Fig. 6.4).

```
my.spec = spectrum(out$I, plot=FALSE)
par(mfrow=c(1,2))
#default plot (less default labels)
plot(my.spec, xlab="Frequency", ylab="Log-amplitude",
     main="", sub="")
#plot with period (rather than frequency)
plot(1/my.spec$freq/52, my.spec$spec, type="b", xlab=
  "Period (year)", ylab="Amplitude", xlim=c(0,5))
```

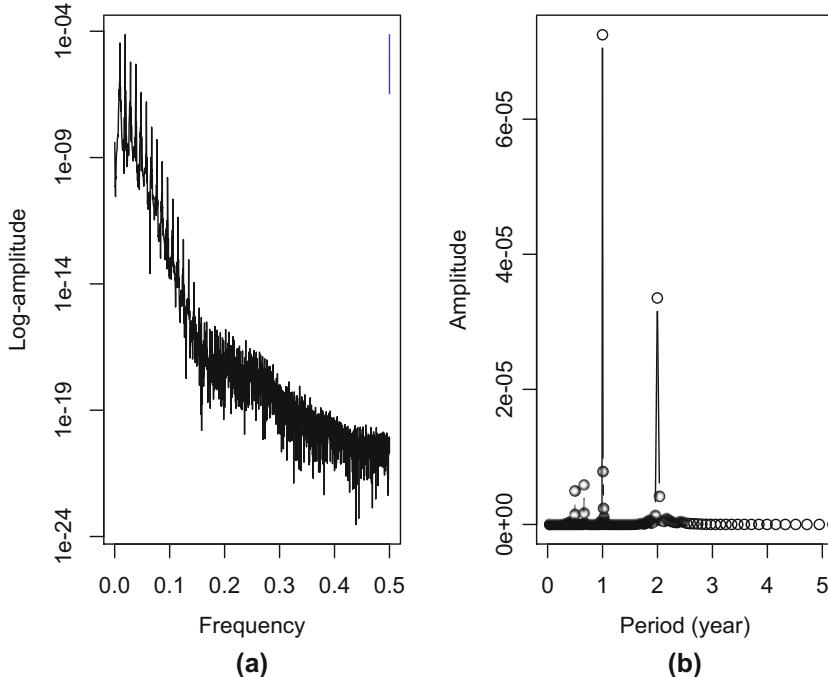


Fig. 6.4 The power spectrum of prevalence for the seasonally forced SEIR model. (a) Default plot of log-amplitude against frequency and (b) amplitude against period (in years)

Using the *fast Fourier transform* (FFT), the Schuster periodogram will automatically estimate the spectrum of a time series (of length T) at the following $T/2$ frequencies: $f = \{\frac{1}{T}, \frac{2}{T}, \dots, \frac{T/2}{T}\}$ (or equivalent periods: $\{T, \frac{T}{2}, \dots, 2\}$). An upside of using FFT is that it is fast. A downside is that the Schuster periodogram is not a *consistent* method, meaning that the estimated periodogram does not converge on the true power spectrum as the time series gets longer because the number of frequencies considered (and thus the number of parameters) increases linearly with time-series length. Numerous fixes of this have been developed, the most common is to smooth the periodogram (Priestley 1981), but nonparametric density estimation has also been proposed. We use Kooperberg et al.'s (1995) log-spline method in Sect. 9.7.

6.4 Wavelets

The wavelet spectrum is an extension of spectral analysis that allows an additional time axis and therefore to allow the study of changes in dynamics over time (Tor-

rence and Compo 1998). Unlike the periodogram, wavelets do not have “canonical” periods for decomposition. If we use the Morlet wavelet (which is provided by the `cwt`-function in the `Rwave`-package), we need to specify the periods we wish to consider through the number of *octaves*, `no`, and *voices*, `nv`. With 8 octaves the main periods will be $\{2^1, 2^2, \dots, 2^8\} = \{2, 4, \dots, 256\}$. The number of voices specifies how many subdivisions to estimate within each octave. With four voices the resultant periods will be $\{2^1, 2^{1.25}, 2^{1.5}, 2^{1.75}, 2^2, 2^{2.25}, \dots\}$. We first consider the simulated time series of prevalence for the unforced SEIR model (Fig. 6.5).

```
#Simulate and plot time series
times = seq(0, 25, by=1/52)
paras = c(mu = 1/50, N = 1, beta = 1000,
          sigma = 365/8, gamma = 365/5)
xstart = c(S=0.06, E=0, I=0.001, R = 0.939)
out2 = as.data.frame(ode(xstart, times, seirmod, paras))
par(mfrow = c(1, 2)) #Side-by-side plots
plot(times, out2$I, type="l", xlab="Time",
      ylab = "Infected")

#Wavelet analysis
require(Rwave)
#Set the number of "octaves" and "voices"
no = 8; nv = 32
#Calculate periods
a = 2^seq(1,no+1-1/nv, by = 1/nv)
#Do the continuous wavelet decomposition
wfit = cwt(out2$I, no, nv, plot=FALSE)
#Calculate the wavelet spectrum
wspec = Mod(wfit)

#Wavelet plot with contours
image(x=times, wspec, col=gray((12:32)/32), y=a/52,
      ylim=c(0,4), xlab="Time", ylab="Period")
contour(x=times, wspec, y=a/52, ylim=c(0,4),
      zlim=c(mean(wspec), max(wspec)), add=TRUE)
```

The initial inter-epidemic period at around 2.5 years is strong (recall that the dampening period of the SEIR with these parameters is 2.3 years with these parameters; Sect. 5.3), but then wanes as the system converges towards the stable endemic equilibrium. We see this clearly illustrated if we compare the wavelet spectrum at, for example, the beginning of year 2 and the beginning of year 10 (Fig. 6.6).

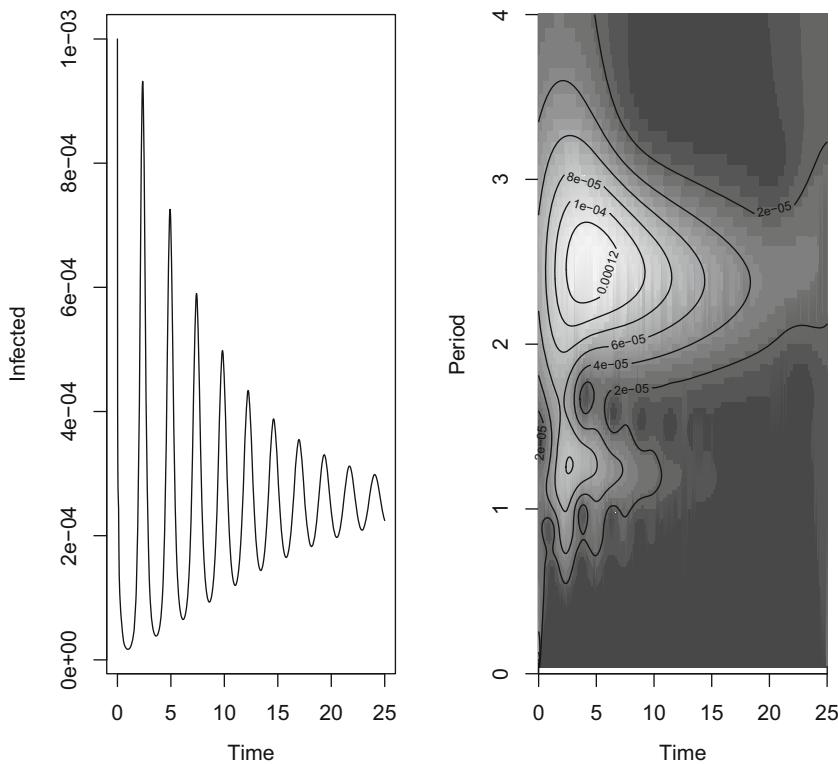


Fig. 6.5 Prevalence against time for the unforced SEIR model ($\mu = 1/50$, $N = 1$, $\beta = 1000$, $\sigma = 365/8$, $\gamma = 365/5$) with associated wavelet spectrum

```
plot(a/52, wspec[104,], type="l", ylab="Amplitude",
     xlab="Period")
lines(a/52, wspec[1040,], type="l",
      lty=2, col="red")
legend("topright", legend=c("Year 2", "Year 10"),
      lty=c(1,2), col=c("black", "red"))
```

6.5 Measles in London

The pre-vaccination incidence of measles shows interesting non-stationarities that have been traced back to changing susceptible recruitment due to the post-World War II baby boom (Fig. 6.7). The `meas` data set contains the biweekly incidence and births from 1944 and 1965.

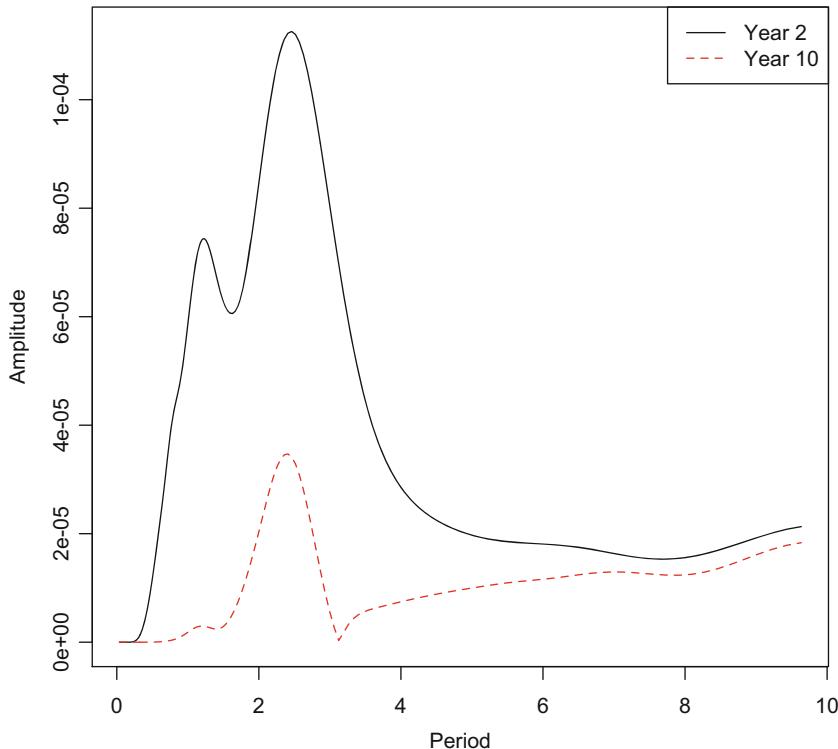


Fig. 6.6 The estimated wavelet spectrum at the first week of year 2 and year 10 for the unforced SEIR model

```

data(meas)
head(meas)

##      year week      time London     B
## 1    44     2 44.00000   180 1725
## 2    44     4 44.03846   271 1725
## 3    44     6 44.07692   423 1725
## 4    44     8 44.11538   465 1725
## 5    44    10 44.15385   523 1725
## 6    44    12 44.19231   649 1725

par(mar = c(5,5,2,5)) #Make room for two axes
plot(meas$time, meas$London, type="b", xlab="Week",
     ylab="Incidence", ylim=c(0,8000))
par(new=T) #Superimposed births plot
plot(meas$time, meas$B, type="l", col="red",
     axes=FALSE, xlab=NA, ylab=NA, ylim=c(1000, 2700))
axis(side = 4)

```

```
mtext(side = 4, line = 3, "Births")
legend("topright", legend=c("Cases", "Births"),
lty=c(1,1), col=c("black", "red"))
```

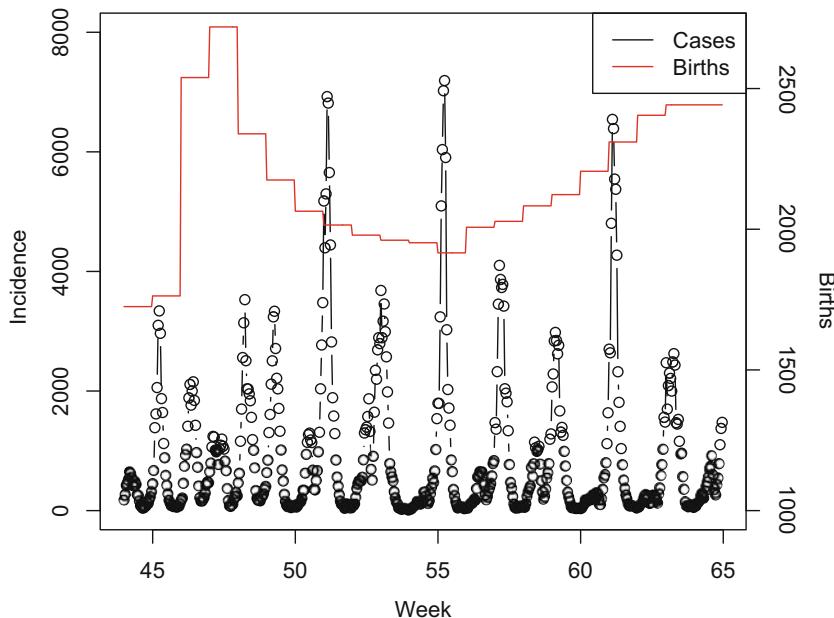


Fig. 6.7 Biweekly incidence of measles in London between 1944 and 1965 with susceptible recruitment (births) superimposed

We apply the wavelet analysis to the historical measles dynamics from London (Grenfell et al. 2001). In addition to providing a continuous wavelet transform, the Rwave-package has a “crazy climber” algorithm to highlight ridges in the wavelet spectrum (implemented with the `crc` and `cfamily`-functions). When applied to the London measles data, the crazy climber reveals the background annual rhythm and the punctuated appearance of the biennial cycle in the early 1950s (Fig. 6.8).

```
#Set octaves, voices and associated periods
no = 8; nv = 32
a = 2^seq(1,no+1-1/nv, by = 1/nv)
#Continuous wavelet decomposition
wfit = cwt(meas$London, no, nv, plot=FALSE)
wspec = Mod(wfit)
#Crazy climber
crcinc<-crc(wspec, nbclimb=10, bstep=100)
```

```

fcrcinc<-cfamily(crcinc, ptile=0.5, nbchain=1000,
  bstep=10)

## There are 2 chains.

ridges<-fcrcinc[[1]]
ridges[which(ridges==0)]<-NA
#Wavelet plot with crazy-climber and contours
image(x=meas$time, wspec, col=gray((12:32)/32), y=a/26,
  ylim=c(0.1,3), ylab="Period", xlab="Year")
contour(x=meas$time, wspec, y=a/26, ylim=c(0,3),
  nlevels = 6, zlim=c(mean(wspec), max(wspec)), add=TRUE)
image(x=meas$time, y=a/26, z=ridges, add=TRUE,
  col=gray(0))

```

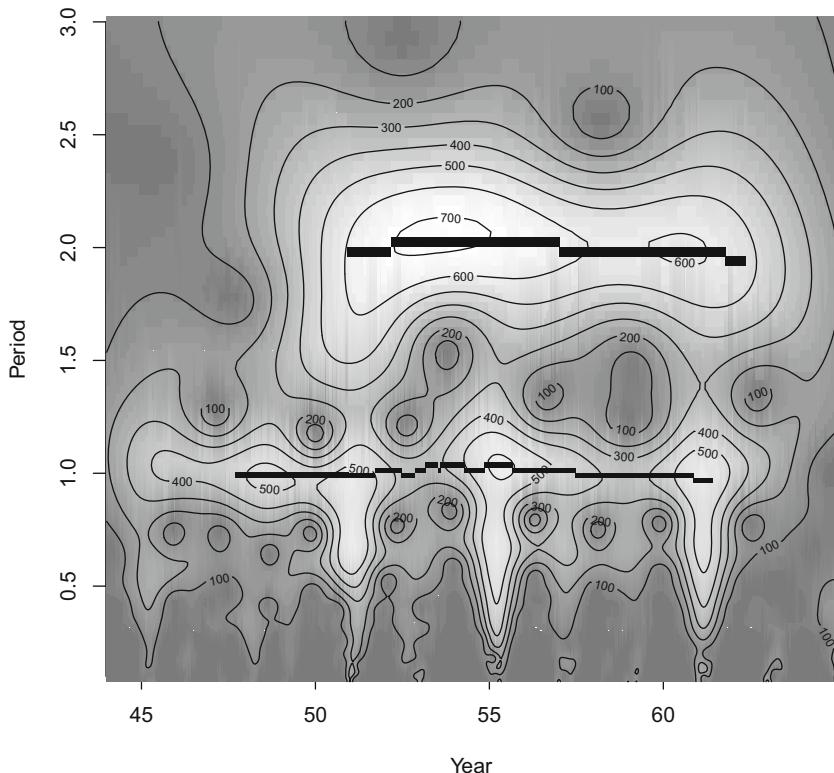


Fig. 6.8 The wavelet spectrum of the London measles incidence with “crazy-climber” ridges. The appearance of a significant biennial rhythm in the 1950s is conspicuous

We can contrast the spectrum of the first biweek of January 1945 and the first biweek of January 1954 (Fig. 6.9). The transition from a dominance of annual to biennial epidemics is conspicuous. Two-year cycles are pronounced when birth rates are around 20 per thousand per year; Annual epidemics are associated with higher birth rates. This transition, due to the post-World War II baby boom, is as predicted by the seasonally forced SEIR model with dropping birth rates (Earn et al. 2000b, Fig. 5.6).

```
plot(a/26, wspec[261,], type="l", xlim=c(0, 3),
      xlab="period (years)", ylab="amplitude")
lines(a/26, wspec[27,], type="l", lty=2, col="red")
legend("topleft", legend=c("1945", "1954"),
       lty=c(2,1), col=c("red", "black"))
```

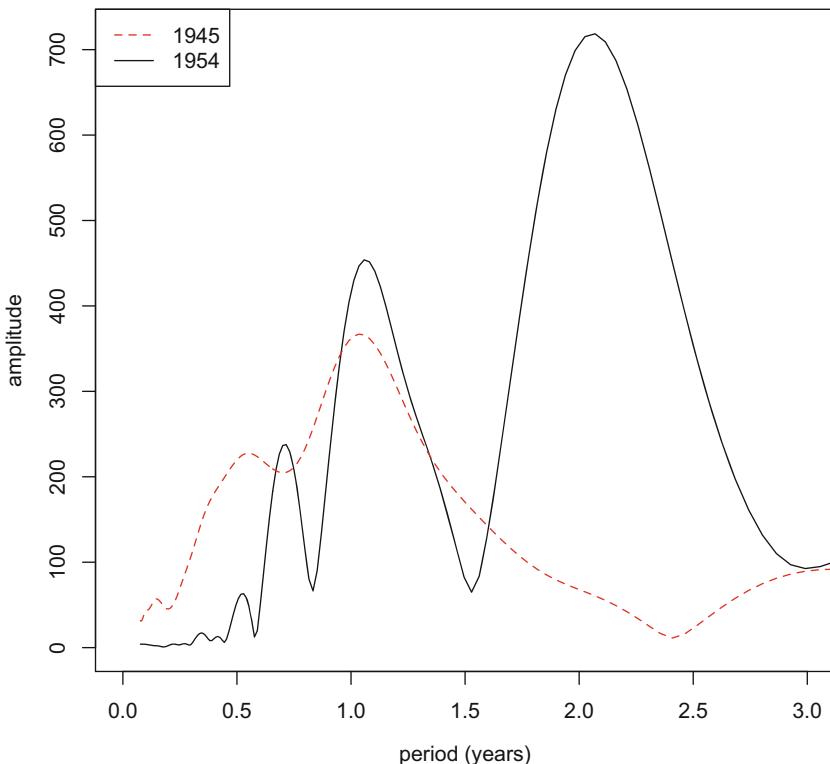


Fig. 6.9 The wavelet spectrum of the London measles in Jan 1945 vs Jan 1954

The above methods of time-series analysis require regularly spaced time series without any missing values. Lomb (1976) developed the Lomb periodogram for unequally spaced data. Furthermore, the classic spectral methods

cannot quantify rhythms in cruder “nonmetric” data such as presence/absence of infection. Legendre et al. (1981) developed the “contingency periodogram” for such situations. The `nlts`-package has the functions `spec.lomb` and `contingency.periodogram` to carry out such analyses. The `mvcwt`-package can do wavelet analyses of time series with missing data.

6.6 Project Tycho

Project Tycho (<http://www.tycho.pitt.edu>) is a great resource for time series on historical disease incidence. The data used in Sect. 5.1 were downloaded from this database. Weekly data of whooping cough (1925–1947), diphtheria (1914–1947), and measles (1914–1947) in the city of Philadelphia are from Project Tycho and are saved in the `tywhooping`, `tydiphtheria`, and `tymeasles` data sets. These were all important causes of childhood mortality in the early twentieth century and were therefore “reportable infections” in the USA. Whooping cough is caused by bacterial colonization of the lower respiratory tract by congeneric species in the genus *Bordetella*, most notably *B. pertussis*, and cause violent coughing, vomiting, and pneumonia. Diphtheria is caused by infection by *Corynebacterium diphtheriae* which toxins cause a range of health complications. Measles is a severely immunocompromising paramyxovirus. We will use these time series to illustrate some additional aspects of disease dynamics/time-series analysis.

```
data(tywhooping)
tywhooping$TIME=tywhooping$YEAR+tywhooping$WEEK/52
tywhooping$TM=1:length(tywhooping$YEAR)
data(tydiphtheria)
data(tymeasles)
tydiphtheria$TIME=tymeasles$TIME=tymeasles$YEAR+
tymeasles$WEEK/52
```

These time series have occasional weeks of missing data which we interpolate. We will use the `imputeTS`-package. But first we use the whooping cough data to illustrate the use of the Lomb periodogram for spectral analysis of unevenly spaced data.

6.7 Lomb Periodogram: Whooping Cough

There are 14 missing weeks in the `tywhooping` data set. For frequency-domain analyses of this data we either have to interpolate the missing weeks or use the Lomb periodogram. We compare the two approaches:

```

data(tywhooping)
whp=na.omit(tywhooping)

#data with missing values interpolated
require(imputeTS)
sum(is.na(tywhooping$PHILADELPHIA))

## [1] 14

tywhooping$PHILADELPHIA=
    na.interpolation(ts(tywhooping$PHILADELPHIA))

#Classic periodogram
my.spec = spectrum(sqrt(tywhooping$PHILADELPHIA))
#Lomb periodogram
require(nlts)
my.lomb=spec.lomb(x=whp$TM, y=sqrt(whp$PHILADELPHIA))

plot(1/my.spec$freq/52, my.spec$spec, type="b",
      xlab="Period (year)", ylab="Amplitude")
par(new=TRUE)
plot(1/my.lomb$freq/52, my.lomb$spec, axes=FALSE,
      type="b", col=2, xlab="", ylab="")
legend("topright", legend=c("Classic", "Lomb"),
      lty=c(1,1), pch=c(1,1), col=c("black", "red"))

```

With only 14 missing values in a 1000+ week long time series the shape of the Schuster periodogram (on interpolated data) and the Lomb periodogram are almost identical (Fig. 6.10).

6.8 Triennial Cycles: Philadelphia Measles

Like in London, pre-vaccination measles dynamics in Philadelphia exhibit interesting nonstationarities we can highlight with the wavelet analysis. There are 24 missing weeks we interpolate:

```

data(tymeasles)
sum(is.na(tymeasles$PHILADELPHIA))

## [1] 24

tymeasles$PHILADELPHIA=
    na.interpolation(ts(tymeasles$PHILADELPHIA))

```

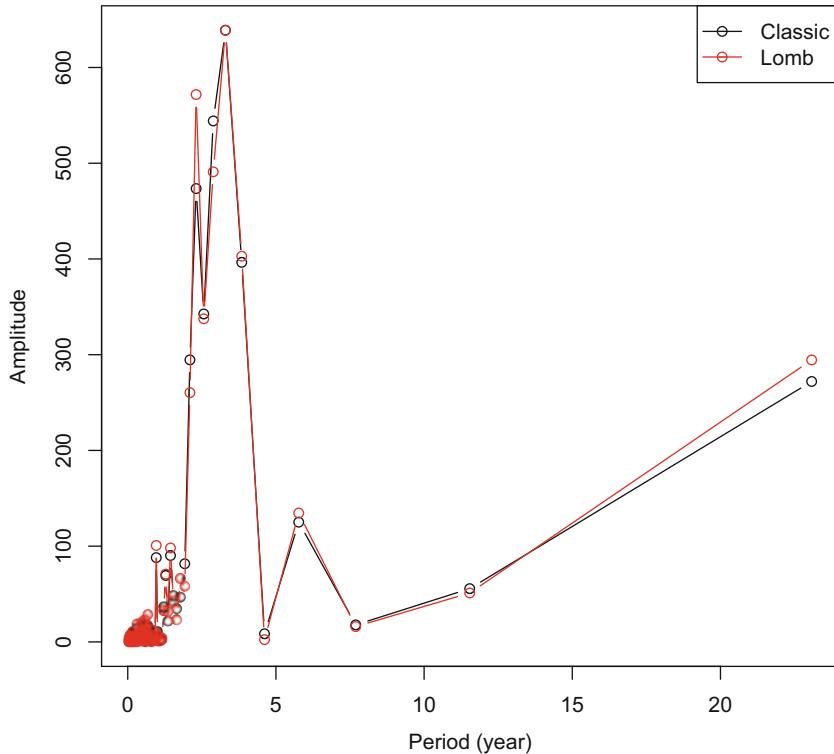


Fig. 6.10 The Lomb periodogram and the classic periodogram (on interpolated data) of the Philadelphia whooping cough time series

We twiddle with the graphics margins and layout using the `par` and `layout` functions to make a prettier compound graphic (Fig. 6.11).

```

par(mfrow=c(2,1), mar=c(2,4,2,1))
layout(matrix(c(1,1,2,2,2), ncol=1))
plot(tymeasles$TIME, sqrt(tymeasles$PHILADELPHIA),
      type="b", ylab="Sqrt (incidence)")
title("Measles 1914-47")

no = 8; nv = 16; a = 2^seq(1,no+1-1/nv, by = 1/nv)
wfit = cwt(sqrt(tymeasles$PHILADELPHIA),
            no, nv, plot=FALSE)
wspec = Mod(wfit)

par(mar=c(1,4,0.25,1))
image(z=wspec, y=a/52, ylim=c(0,4), ylab="Period(year)",
      col=gray((12:32)/32), xaxt='n')
contour(z=wspec, y=a/52, ylim=c(0,4), nlevels = 6,

```

```
zlim=c(mean(wspec), max(wspec)), add=TRUE)
```

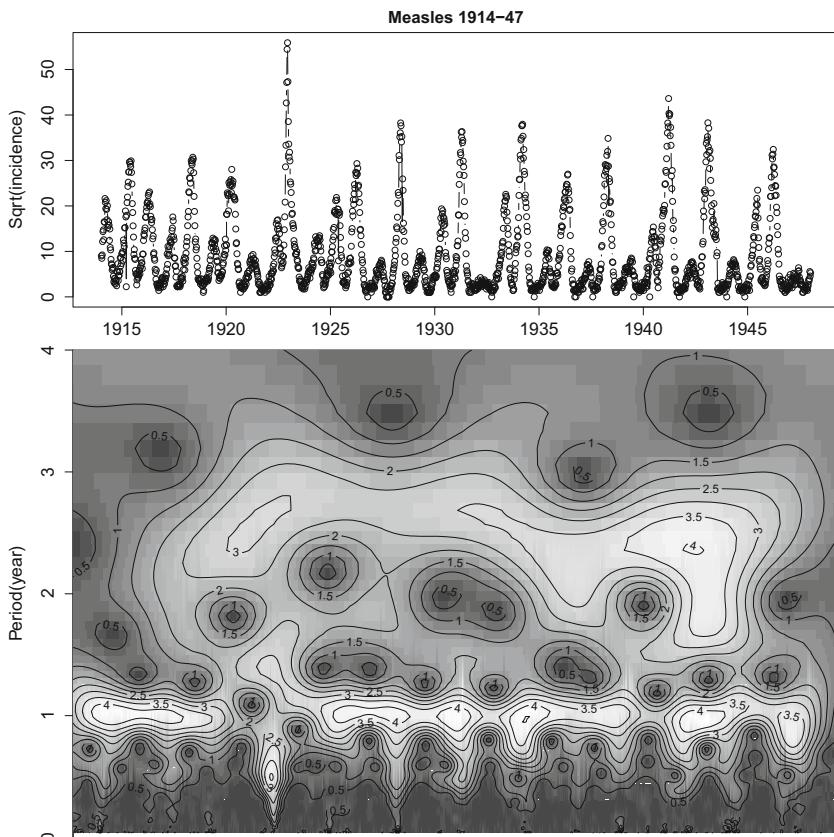


Fig. 6.11 Wavelet spectrum of measles in Philadelphia

The early annual epidemics give way to triennial epidemic cycles from 1920 onwards (Fig. 6.12). The tri-annual cycles are the hallmarks of chaotic epidemics (Dalziel et al. 2016) we discuss further in Sect. 10.2.

```
plot(a/52, wspec[54,], type="l", xlim=c(0,4),
      xlab="Period (years)", ylab="Amplitude",
      col="red", lty=2)
lines(a/52, wspec[1357,], type="l", xlim=c(0,4))
legend("topleft", legend=c("1915", "1940"),
       lty=c(2,1), col=c("red", "black"))
```

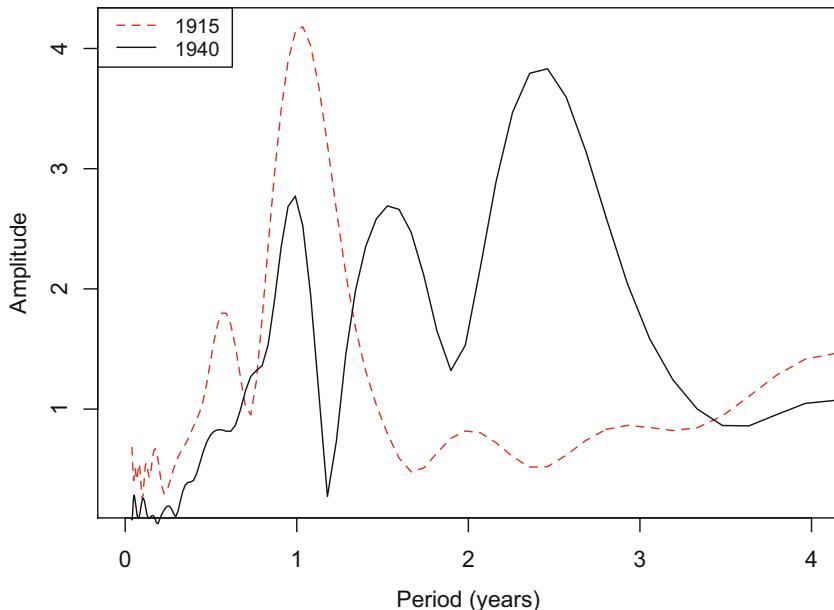


Fig. 6.12 The Jan 1915 versus Jan 1940 measles wavelet spectrum; Annual epidemics give way to triennial cycles

6.9 Wavelet Reconstruction and Wavelet Filter: Diphtheria

Diphtheria exhibited conspicuous annual cycles during the beginning of the twentieth century until the addition of an [adjuvant to the toxoid vaccine in 1926](#) led to a strong secular downward trend and effectively the elimination of the disease (Fig. 6.13). The wavelet lets us study how adjuvant-induced reduction in incidence is associated with a loss of periodicity and increase in high-frequency variability (“noise”) (Fig. 6.13). There are 18 missing values we interpolate prior to the analysis.

```
data(tydiphtheria)
sum(is.na(tydiphtheria$PHILADELPHIA))

## [1] 18

tydiphtheria$PHILADELPHIA=
  na.interpolation(ts(tydiphtheria$PHILADELPHIA))

par(mfrow=c(2,1), mar=c(2,4,2,1))
layout(matrix(c(1,1,2,2,2), ncol=1))
plot(tydiphtheria$TIME, sqrt(tydiphtheria$PHILADELPHIA),
```

```

type="b", ylab="Sqrt(incidence)")
title("Diphtheria 1914-47")

no = 8; nv = 16; a = 2^seq(1,no+1-1/nv, by = 1/nv)
wfit = cwt(sqrt(tyldiphtheria$PHILADELPHIA),
            no, nv, plot=FALSE)
wspec = Mod(wfit)
par(mar=c(1,4,0.25,1))
image(z=wspec, y=a/52, ylim=c(0,3), ylab="Period(year)",
      col=gray((12:32)/32), xaxt='n')
contour(z=wspec, y=a/52, ylim=c(0,3), nlevels = 6,
        zlim=c(mean(wspec), max(wspec)), add=TRUE)

```

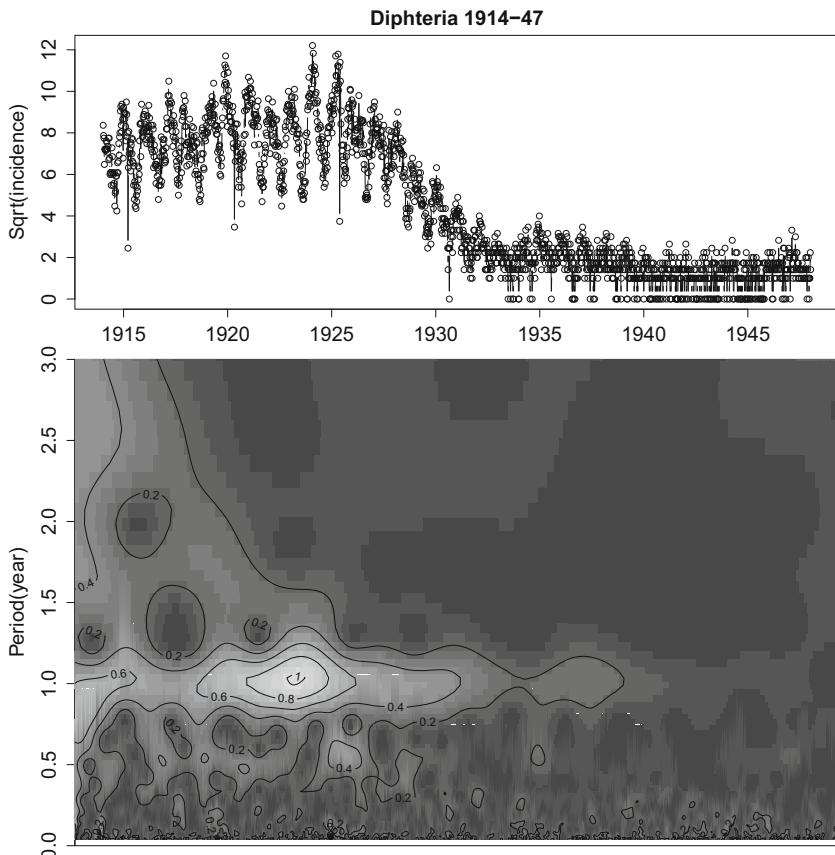


Fig. 6.13 Wavelet spectrum of diphtheria in Philadelphia

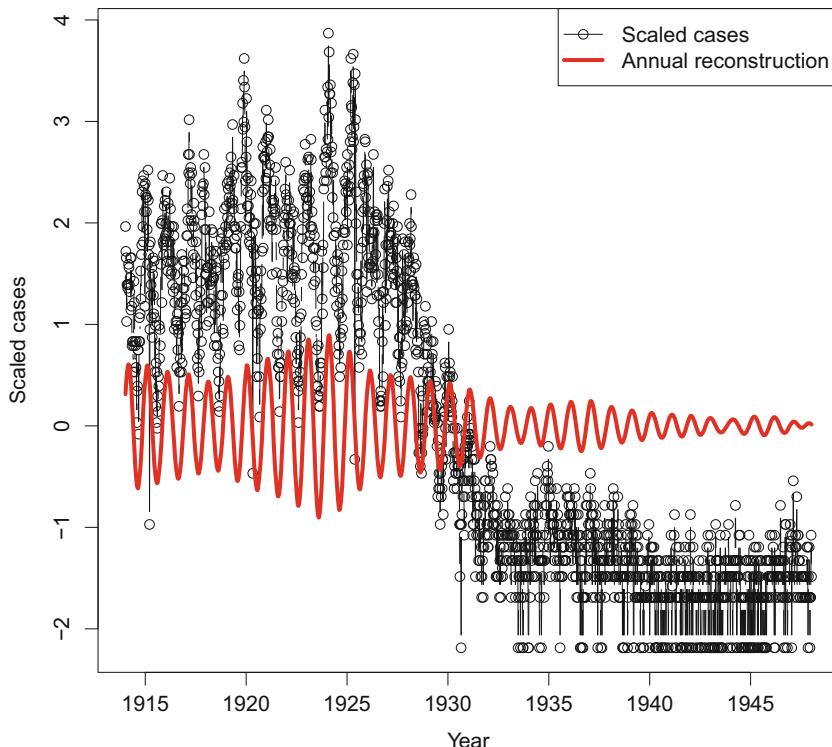


Fig. 6.14 Wavelet reconstructed variability in the 45–60 week range of diphtheria in Philadelphia

We are sometimes interested in using the wavelet as a “filter.” We may for example want to quantitate how the strength of the annual cycle of diphtheria (in the 45–60 week range, say) changes over time. To do this we use wavelet reconstruction around the relevant time scales (Fig. 6.14). For the Morlet wavelet the formula for reconstruction using the j 'th though $j+s$ th scales is provided by Torrence and Compo (1998). The mid-pass filter clearly illustrates the loss of annual signal over time (Fig. 6.14).

```
#midpass filter
sel=a>45 & a<60
rec=0.6*apply(Re(wfit[,sel])/sqrt(a[sel]), 1,
  sum)/(0.776*(pi^(-1/4)))
data=pi*scale(sqrt(tydiphtheria$PHILADELPHIA))/2
plot(tydiphtheria$TIME, data, type="b", xlab="Year",
  ylab="Scaled cases")
lines(tydiphtheria$TIME, rec, type="l", col=2, lwd=3)
legend("topright", legend=c("Scaled cases",
  "Annual reconstruction"), pch=c(1, NA), lty=c(1, 1),
  lwd=c(1, 3), col=c("black", "red"))
```

6.10 Advanced: FFT and Reconstruction

One-hundred-and-twenty years ago, Arthur Schuster proposed the bold idea that any discrete time series can be decomposed and exactly reconstructed from a sum of trigonometric functions. Given its nonstationary transition from annual to biennial epidemics, the pre-vaccination 1944–1964 London measles time series (in the `meas` data set) offers a nice test-bed for this assertion.

The below code generates an animated visualization of the reconstruction. Section 11.6 discusses making in-line and permanent animations in more detail. A web-optimized animated gif can be found in <https://github.com/objornstad/epimdr/blob/master/mov/fftreccon.gif>.

If z is the fast Fourier transform of the time series, then the trigonometric “signal” of the k ’th observation is $\frac{1}{T}(\sum_f(Re(z)\cos(2\pi(k-1)f)) - Im(z)\sin(2\pi(k-1)f))$, where $Re()$ and $Im()$ represent real and imaginary parts. We first piece together relevant bits for the formula; we then do the reconstruction in the `rec2`-object where the contribution of each frequency is weighed:

```
# fft
x <- meas$London
p <- length(x)
z <- fft(x)
f <- seq(from = 0, length = p, by = 1/p)
a <- Re(z)
b <- Im(z)
# reconstruction
rec2 = matrix(NA, ncol = p, nrow = p)
for (k in 1:p) {
  rec2[, k] <- (a * cos(2 * pi * (k - 1) * f) - b *
    sin(2 * pi * (k - 1) * f)) / p
}
```

Finally we can visualize the convergence on the original signal using the sequence of frequencies order by amplitude (highest to lowest importance):

```
sim=rep(0, p)
n=0
samp=order(a^2+b^2, decreasing=TRUE)
for(g in samp){
n=n+1
par(mfrow=c(1,2))
plot(x, ylim=c(0,11000), ylab="Incidence",
      xlab="Biweek")
title(paste("nfreq = ", n))
sim=sim+rec2[g,]
lines(sim, col=2)
```

```
par(new=TRUE)
sc=scale((cos(2*pi*(0:(p-1))*f[g]) -
  sin(2*pi*(0:(p-1))*f[g]))/p)
plot(sc*(a^2+b^2)[g]/max(a^2+b^2), type="l", col=
  gray(.5), ylim=c(-8, 2), axes=FALSE, xlab="", ylab="")
plot(x, sim, ylab="Reconstructed", xlab="Observed",
  ylim=c(0,8000))
#Sys.sleep makes R wait a bit
Sys.sleep(.2)
}
```

Chapter 7

TSIR



7.1 Stochastic Variability

Much environmental forcing is non-predictable “environmental stochasticity”. In such cases stochastic simulation can be very useful.¹ One can use stochastic analogues of the continuous-time deterministic compartmental models using event-based simulations (see Sect. 8.2), or consider extensions of the chain-binomial model we introduced in Sects. 3.4 and 3.5. We will consider a variant of the chain-binomial model: the time-series SIR (TSIR) model (Bjørnstad et al. 2002a; Finkenstädt et al. 2002; Grenfell et al. 2002).

The TSIR model is as follows: If we use a discrete time step equal to the generation time of the pathogen (about 2 weeks in the case of infections like measles, diphtheria, scarlet fever, and chickenpox), we can write the model (subsuming a latent period) as:

$$S_{t+1} = S_t + B_t - I_t, \quad (7.1)$$

$$\lambda_{t+1} = \beta \frac{S_t}{N_t} I_t^\alpha \quad (7.2)$$

where S_t and I_t are the numbers of susceptibles and infecteds in pathogen generation t , B_t is the number of susceptible recruits (births) during the time interval, N is the population size, and β is the transmission rate. The α is an exponent (normally just

This chapter uses the following R-packages: `imputeTS` and `plotrix`.

¹ It is also possible to use more powerful mathematical approaches in such cases with signal theory and transfer functions; see Sect. 9.7.

under 1) that accounts for discretizing the underlying continuous process (Glass et al. 2003).

The final variable λ_{t+1} represents the *expectation* for the new number of infecteds in generation $t + 1$. The actual number of infecteds that will appear in generation $t + 1$ will follow some stochastic distribution around λ_{t+1} . For example $I_{t+1} \sim Po(\lambda_{t+1})$ (Miramontes and Rohani 1998) or $I_{t+1} \sim NegBin(\lambda, I_t)$ (Bjørnstad et al. 2002a) depending on the exact assumptions regarding the variability in the underlying process.² The link to the chain-binomial comes about because $1 - exp(-\phi) \approx \phi$, when ϕ is small, and the binomial process—for which we need to know the susceptible denominator (which is usually unknown)—can be approximated with a Poisson or negative binomial distribution (depending on assumptions; Bjørnstad et al. 2002a) neither of which require known denominators.

Stochasticity may further enter through variable numbers of births or random variation in the transmission rates. The `SimTSIR`-function does stochastic simulation akin to the chain-binomial simulator (Sect. 3.5) but with the possibility of having stochastic variation in β (controlled by the `sdbeta` argument):

```
SimTsir=function(alpha=0.97, B=2300, beta=25,
  sdbeta=0, S0 = 0.06, I0=180, IT=520,
  N=3.3E6){
  #Set up simulation
  lambda = rep(NA, IT)
  I = rep(NA, IT)
  S = rep(NA, IT)
  #Add initial conditions
  I[1] = I0
  lambda[1] = I0
  S[1] = S0*N

  #Run simulation
  for(i in 2:IT) {
    lambda[i] = rnorm(1, mean=beta, sd=sdbeta) *
      I[i - 1]^alpha * S[i - 1] /N
    if(lambda[i]<0) {lambda[i]=0}
    I[i] = rpois(1, lambda[i])
    S[i] = S[i - 1] + B - I[i]
  }
  #Return result
  list(I = I, S = S)
}
```

² The negative binomial arises from assuming an epidemic birth-and-death process, in which case the offspring distribution from each infected is a geometric distribution and the sum of I_t geometrics is a negative binomial with “clumping parameter” I_t .

In the function `IT` is the length of the time series to be simulated. `S0` and `I0` are initial conditions, and `B` is the susceptible recruitment. The parameters in the model are provided with default values. These values correspond roughly to estimates from the measles time series for London for the period 1944–1965 (see Sect. 7.4), a city with 3.3 million inhabitants at the time. The trajectories in time and in the phase plane are (Fig. 7.1):

```
out = SimTsir()
par(mfrow=c(1, 2))
plot(out$I, ylab="Infected", xlab="Time", type="l")
plot(out$S, out$I, ylab="Infected", xlab="Susceptible",
     type="l")
```

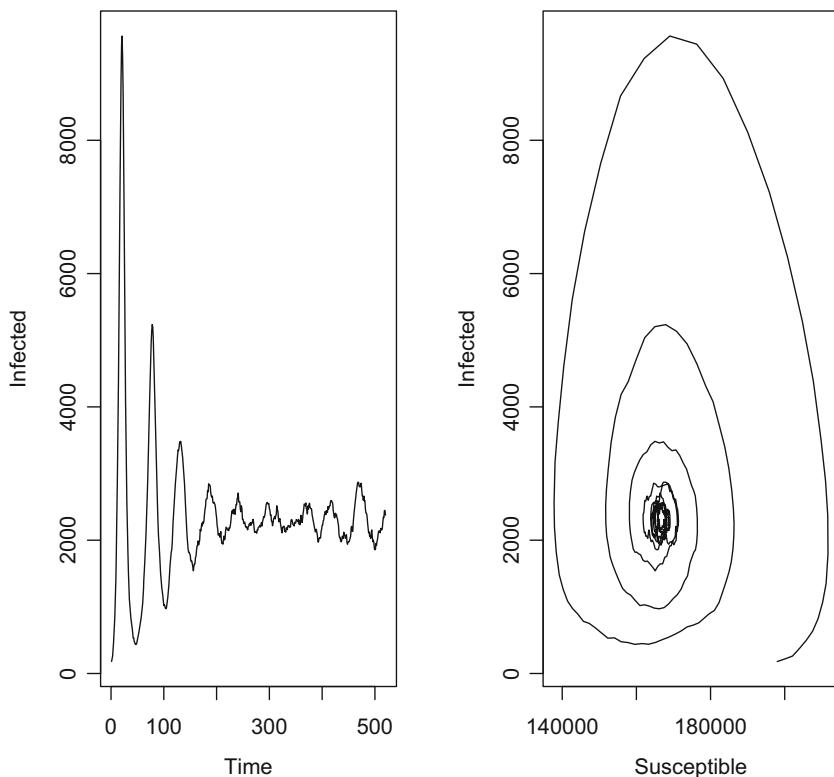


Fig. 7.1 A stochastic realization from the TSIR model with demographic stochasticity

7.2 Estimating Parameters in Dynamic Models

There are many strategies for estimating the parameters of dynamic models from time-series data. They differ conceptually in the way they handle demographic and environmental stochasticity (sometimes referred to jointly as “process error”), observation error, and partial (missing) observation. The strategies also often vary by whether the underlying dynamics is thought to be best approximated in continuous time (differential models) or discrete time (difference models).

In reality, disease dynamics is always affected by some level of demographic and environmental stochasticity, and observation error comes in the form of both inaccuracies in observation and missing information—the exposed (latent) class of an SEIR-like system, for example, is very rarely monitored (tuberculosis perhaps being an exception). Furthermore, while disease dynamics very rarely play out in discrete generation (in-host dynamics of *Eimeria* and sometimes *Plasmodium* being exceptions; Mideo et al. 2013), they also never follow the exponential waiting-time distributions implicit in ODE-models (Fig. 2.6). As discussed in Sect. 2.7 various types of distributed-delay models or renewal equations can cover the continuum between fixed (discrete) and exponential distributions (the original formulations of Kermack and McKendrick (1927) is an early example), but these come at a prize of mathematical/computational overhead.

Our previous implementation of Ferrari et al.’s (2005) removal method (Sect. 3.4) for fitting the chain-binomial model for a simple epidemic is an example of a model that assumes that all process error is due to demographic stochasticity (according to the chain-binomial) and that observation error is sufficiently insignificant to be ignored.³ The method finds parameters (S_0 and β) that predict an epidemic curve that most closely (in a likelihood-sense) resembles the data. This is an example of the strategy of *trajectory matching*; Chose parameters that produces a predicted trajectory that comes most close to the observed. It is also common to use trajectory matching for a continuous-time SIR model of simple epidemics, but instead assume that all process error is sufficiently insignificant to be ignored. In this case we find parameters that predict prevalence curves that most closely resembles the data assuming an underlying deterministic epidemic clockwork cloaked by observation-error only (Chap. 8).

A variety of other approaches has been proposed to fit dynamical models to ecological and epidemiological time-series models including:

- Gradient matching: Fitting ODEs in the presence of significant process noise (Ellner et al. 2002). The idea is to estimate derivatives dx/dt along the time series (for example, by fitting a spline and calculate its derivatives) and then relate them to relevant state variables.
- Probe matching was introduced by Kendall et al. (1999) and its statistical properties were later formalized in Wood’s (2010) “synthetic likelihood.” The idea is to

³ And sacrificed on the “bias-variance altar” of hierarchical uncertainty, as discussed by Polacheck et al. (1993).

choose parameters that make the model most closely reproduce what are deemed to be the critical dynamical features of the system.

- Hierarchical models using MCMC (e.g., Clark and Bjørnstad 2004) which has been much refined as Partially observed Markov Processes in the `pomp`-package (King et al. 2015b).
- One of the simpler methods, which this chapter is devoted to, is the time-series SIR model (Finkenstädt and Grenfell 2000; Bjørnstad et al. 2002a; Grenfell et al. 2002).

7.3 Estimation Using the TSIR

Estimation using the TSIR came out of a pragmatic attempt at dealing with the complexities of disease dynamics and incidence data using basic statistical tools. In its original form it assumes that process noise is due to demographic stochasticity and observation error is in the form of time-invariant-ish under-reporting that in large populations can be adequately corrected for in a deterministic fashion.

We will consider the biweekly incidence (number of cases for each 2-week period) of measles from London between 1944 and 1965 (Fig. 6.8) introduced in Sect. 6.5.

```
data (meas)
```

The incidence are accessed as `meas$London`. In addition, the data set contains columns reporting `meas$year`, `meas$week`, the two combined into `meas$time`, and biweekly number of births (`meas$B`). Birth numbers are annual, so in the data set, this number is evenly distributed across the 26 biweeks of each year. We should be able to use this data to estimate key epidemiological parameters. However, we somehow have to reconstruct the susceptible time series (and correct for under-reporting)...

7.3.1 Inference (Hypothetical)

Given Eq. (7.2) and time series of I and S , the candidate for estimation is obvious:

$$\log(I_{t+1}) = \log(\beta) + \alpha \log(I_t) + \log(S_t) - \log(N). \quad (7.3)$$

We can estimate the unknown parameters β and α by a regression of $\log(I_{t+1})$ on $\log(I_t)$ with $\log(S_t)$ and $-\log(N)$ as offsets (i.e., the slope for the $\log(S)$ and the $-\log(N)$ variables are fixed at unity) or the equivalent generalized linear model with a \log -link. The intercept of this regression would be the estimate of $\log(\beta)$ and the slope against $\log(I_t)$ would be the estimate of α .

In R, this would go something like:

```
# Align time series
IT = length(meas$London)
Inow = log(meas$London[2:IT])
Ilag = log(meas$London[1:(IT - 1)])
Slag = log(S[1:(IT - 1)]) #This does not yet exist
# now the regression
glm(Inow ~ Ilag + offset(Slag) + offset(-N))
```

7.4 Inference (for Real)

The challenge is that most real data sets do not contain perfect records on the state variables. For example, the `meas` data does not contain information on S , and I is generally under-reported. Another challenge is the strong seasonality in transmission rates that result from aggregation of children during school term but not during school holidays.

7.4.1 Susceptible Reconstruction

While we do not have observation on susceptibles, we do have information on the number of births. The idea of susceptible reconstruction was laid out by Bobashev et al. (2000). Consider how the recursive susceptible equation (Eq. 7.1) can be rewritten as:

$$S_t = \bar{S} + D_0 + \sum_{k=0}^t B_k - \sum_{k=0}^t I_k / \rho, \quad (7.4)$$

where \bar{S} is the mean number of susceptibles, D_0 is the unknown deviations around the mean at time 0, and ρ is the (known or unknown) reporting rate. We can reconstruct the time series D_t of how the susceptible numbers deviate from the mean value, $D_t = S_t - \bar{S}$, by rewriting (7.4) as,

$$\sum_{k=0}^t B_k = \bar{S} + D_0 + 1/\rho \sum_{k=0}^t I_k + D_t, \quad (7.5)$$

from which it is clear that D_t is the residual from the regression of cumulative number of births on the cumulative number of cases. Note, that this reconstruction still works when the reporting rate ρ is unknown because under-reporting can be accounted for by the slope of the cumulative-cumulative regression.

As it turns out, reporting rates sometimes vary subtly through time, so it is good to use a slightly more flexible model than linear regression (Finkenstädt et al. 2002), for example, a smoothing spline (with 5 degrees-of-freedom) (Fig. 7.2).

```
cum.reg = smooth.spline(cumsum(meas$B),
  cumsum(meas$London), df=5)
D = - resid(cum.reg) #The residuals

plot(cumsum(meas$B), cumsum(meas$London), type="l",
  xlab="Cumulative births", ylab="Cumulative incidence")
lines(cum.reg)
abline(a=0, b=1)
```

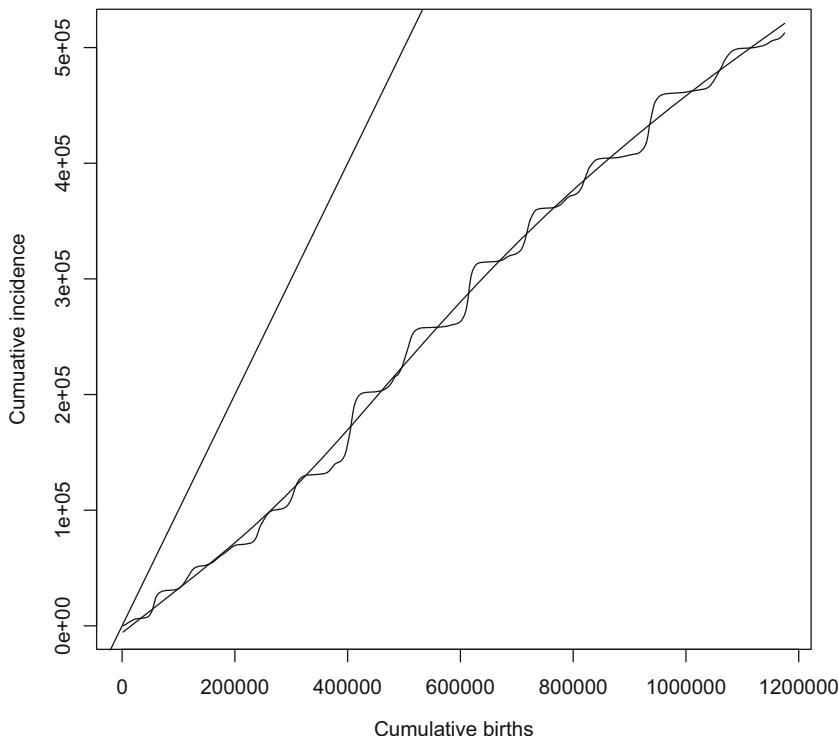


Fig. 7.2 Cumulative incidence versus cumulative births. The straight line is the 1-to-1 reference line

The 1-to-1 line generated by the abline-command shows that the cumulative number of cases are less than the cumulative number of births (Fig. 7.2). The discrepancy is informative because we know from serology that almost all children

were infected with the common childhood infections before the age of 20 in the pre-vaccination era; Black's (1959) data, for example, has seroprevalence >95% by age 15 in pre-vaccination Connecticut (Sect. 4.3). The slope of the cumulative regression, therefore, is an estimate of the reporting rate. We can get the estimated reporting rates for each time step from the slope of the fitted spline:

```
rr = predict(cum.reg, deriv = 1)$y
summary(rr)

##      Min. 1st Qu. Median     Mean 3rd Qu.      Max.
## 0.3485  0.3841  0.4424  0.4522  0.5214  0.5635
```

The reporting rate is fairly steady across the 20 years at around 45%. We can create time series corrected for reporting of both incidence, I_c , and susceptible deviation, D_c :

```
Ic = meas$London/rr
Dc = D/rr
```

7.4.2 Estimation

To estimate parameters we rewrite the model (Eq. 7.3) in terms of the data and unknown parameters on a log-scale (recall that λ_{t+1} is the expected number of cases in generation $t + 1$):

$$\log(\lambda_{t+1}) = \log(\beta_u) - \log(N) + \log(D_t + \bar{S}) + \alpha \log(I_t).$$

This is almost (but not quite) a linear regression with unknown parameters β_u , α , and \bar{S} . Before we are ready to estimate the parameters, however, we need to consider the fact that β_u varies seasonally (because of the school year); Thus the subscript u . The most flexible model is to assume that each of the 26 biweeks of the year has its own transmission rate. Under that assumption we have 28 parameters to estimate (26 β s, α , and \bar{S}). Let us define a vector that flags the periodic β s across the 21 years, and create the three vectors of current (`lInew`) and lagged log-infecteds (`lIold`) and lagged “residual susceptibles” (`Dold`):

```
seas = rep(1:26, 21)[1:545]
lInew = log(Ic[2:546])
lIold = log(Ic[1:545])
Dold = Dc[1:545]
```

Given a value for \bar{S} , the models falls neatly within the linear regression framework.⁴ We can therefore use `glm` to find a profile likelihood estimate of \bar{S} . We know from serology that the average proportion of susceptibles in measles is somewhere in the 2–20% range and—given the size of London at the time (3.3M)—we can postulate a reasonable range of candidate values:

```
N = 3300000
Smean = seq(0.02, 0.2, by = 0.001) * N
offsetN = rep(-log(N), 545)
```

We set up a vector to store the log-likelihood values corresponding to each candidate, and loop over the candidate values to generate a likelihood profile for \bar{S} (Fig. 7.3).

```
llik = rep(NA, length(Smean))
for(i in 1:length(Smean)){
  lSold = log(Smean[i] + Dold)
  glmfit = glm(lInew ~ -1 +as.factor(seas) + lIold +
    offset(lSold+offsetN))
  llik[i] = glmfit$deviance / 2
}
par(mfrow=c(1,1))
plot(Smean/3.3E6, llik, ylim=c(min(llik), 25),
  xlab="Sbar", ylab="Neg log-lik")
```

The `-1` in the regression formula removes the intercept, so that `as.factor(seas)` becomes the estimates of the $\log\beta$ s. Note, further that `glmfit$deviance` holds $-2\log$ likelihood, so the negative log-likelihood is $1/2$ the deviance.⁵

Our best estimates for the TSIR model is:

```
lSold = log(Smean[which(llik==min(llik))] + Dold)
glmfit = glm(lInew ~ -1 +as.factor(seas) + lIold +
  offset(lSold+offsetN))
```

That is, \bar{S} is

```
Smean[which.min(llik)]/3300000
## [1] 0.045
```

⁴ Though had it not, we could write out the likelihood, and use `optim` or `mle2` to find MLEs as in Chap. 3.

⁵ An alternative to the Gaussian likelihood is to use a “counting”-likelihood such as the Poisson quasi-likelihood with a `log`-link. For this we would replace the code with:

```
lnew = Ic[2:546]
glmfit = glm(lnew ~ -1 + as.factor(seas) + lIold +
  offset(lSold+offsetN), family=quasipoisson(link="log")).
```

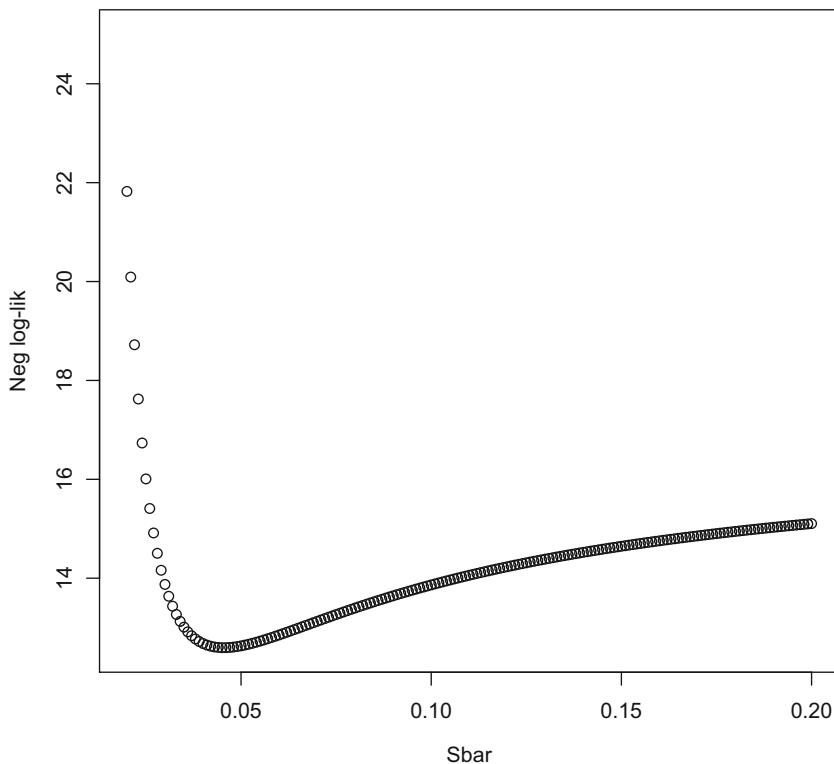


Fig. 7.3 The likelihood profile for \bar{S} from the TSIR applied to the London measles time series

α is

```
glmfit$coef[27]
##      lIold
## 0.9636908
```

and the log- β s are in `glmfit$coef[1:26]`. The seasonal β s with SEs are plotted in Fig. 7.4. The β s vary significantly through the year and are lowest during the summer holidays.

```
require(plotrix)
beta=exp(glmfit$coef[1:26])
ubeta=exp(glmfit$coef[1:26] +
           summary(glmfit)$coef[1:26, 2])
lbeta=exp(glmfit$coef[1:26] -
           summary(glmfit)$coef[1:26, 2])
plotCI(x=c(1:26), y=beta, ui=ubeta, li=lbeta,
```

```
xlab="Biweek", ylab=expression(beta) )
```

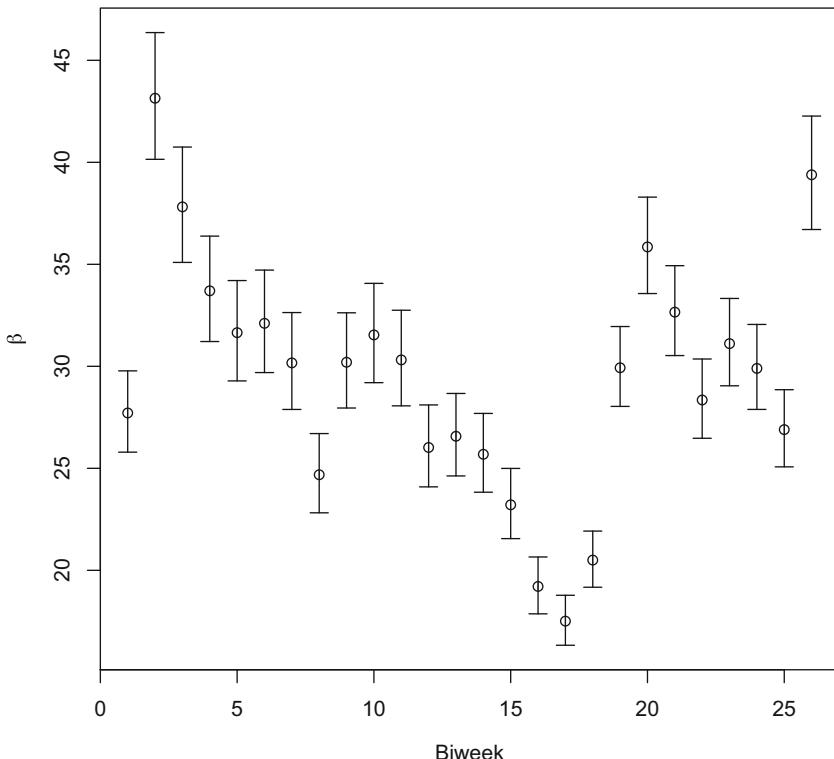


Fig. 7.4 The estimated seasonal β s with SEs from the London measles time series

7.5 Simulating the TSIR Model

We can write a general function `SimTsir2` to simulate the seasonally forced TSIR using the estimated parameters. We write it to perform either a deterministic (`type = "det"`) or a stochastic (assuming demographic stochasticity `type = "stoc"`) simulation.

```
SimTsir2=function(beta, alpha, B, N,  inits =
  list(Snull = 0, Inull = 0), type = "det"){
  type = charmatch(type, c("det", "stoc"),
    nomatch = NA)
  if(is.na(type))
    stop("method should be \"det\" , \"stoc\"")
```

```

IT = length(B)
s = length(beta)
lambda = rep(NA, IT)
I = rep(NA, IT)
S = rep(NA, IT)

I[1] = initss$Inull
lambda[1] = initss$Inull
S[1] = initss$Snull

for(i in 2:IT) {
    lambda[i] = beta[((i - 2) %% s) + 1] *
        S[i - 1] * (I[i - 1]^alpha)/N
    if(type == 2) {
        I[i] = rpois(1, lambda[i])
    }
    if(type == 1) {
        I[i] = lambda[i]
    }
    S[i] = S[i - 1] + B[i] - I[i]
}
return(list(I = I, S = S))
}

```

Simulated dynamics is sensitive to the value of α and there is evidence that the TSIR regression is biased with respect to this parameter.⁶ A simulation using the estimated parameters (with a small correction in α) is shown in Fig. 7.5. While not perfect, the model nicely captures the transition from annual to biennial epidemics as birth rates fall following the post-World War II baby boom.

```

sim=SimTsir2(beta=exp(glmfit$coef[1:26]), alpha=0.966,
               B=meas$B, N=N, initss=list(Snull=Dc[1]+
               Smean[which(llik==min(llik))], Inull=Ic[1]))
plot(sim$I, type="b", ylim=c(0, max(Ic)),
      ylab="Incidence", xlab="Biweek")
lines(exp(lInew), col="red")
legend("topleft", legend=c("sim", "Ic"), lty=c(1,1),
       pch=c(1,NA), col=c("black", "red"))

```

Bjørnstad et al. (2002a) and Grenfell et al. (2002) used the TSIR on measles time series from 60 conurbations in pre-vaccination England and Wales to better understand the dynamics and how it scales with community size. They found

⁶ Metcalf et al. (2011a) proposed to use a “Whittle estimator” of α to reestimate this variable by matching simulated and observed power spectra.

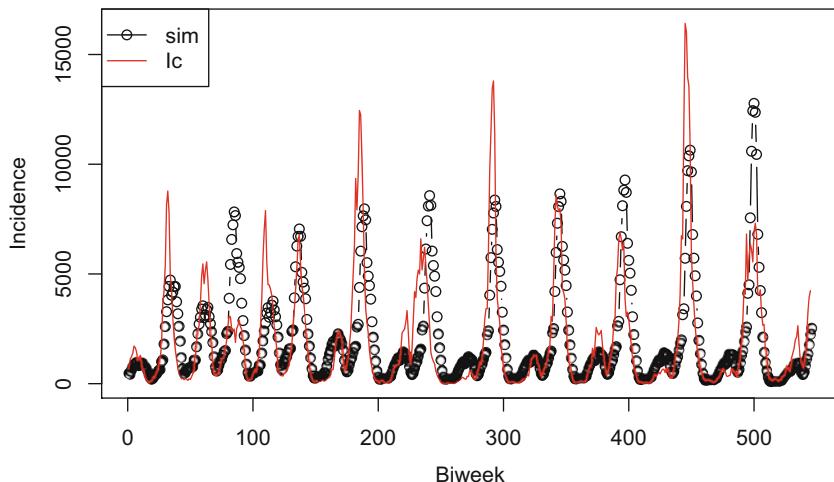


Fig. 7.5 Observed and TSIR-simulated dynamics for measles in London 1944–1965

that R_0 was independent of population size, suggesting that even if transmission is density-dependent, the social clique-size does not differ between large cities and small towns. They speculated that this is because school classes are similar in size across the conurbations (see also Ferrari et al. 2011).

The TSIR has also been used to study the dynamics of rubella (Metcalf et al. 2011a), hand-foot-and-mouth (Takahashi et al. 2016), a variety of other childhood diseases (e.g., Metcalf et al. 2009; Mahmud et al. 2017) and the in-host dynamics of malaria (see Sect. 7.7).

7.6 Tycho Data

Dalziel et al. (2016) compiled demographic data consistent with the level-2 measles data in Project Tycho for 40 cities in the USA (1906–1948) and 40 cities in the UK (1944–1964). The full data set is available from datadryad.org. The US portion of the data are in the `dalziel` data set of the `epimdr`-package:

```
data(dalziel)
```

`pop` contains interpolated population sizes from the 10-year census surveys, and `rec` contains reconstructed number of births. The data are biweekly (to match the generation time of measles). We can use this data to fit the TSIR to other diseases with a two-week’ish serial interval. The Philadelphia scarlet fever data is weekly from Jan 1915 to Dec 1947. To prepare it for TSIR modeling we delete the occasional 53rd week and aggregate in 2-week intervals:

```
data(tyscarlet)
tyscarlet=tyscarlet[tyscarlet$WEEK<53,]
tyscarlet=tyscarlet[tyscarlet$YEAR>1914,]
ag=rep(1:(dim(tyscarlet)[1]/2), each=2)
scarlet2=sapply(split(tyscarlet$PHILADELPHIA, ag), sum)
```

Then merge it with the appropriate part of the `dalziel` data set (and impute a dozen missing values):

```
require(imputeTS)
philly=dalziel[dalziel$loc=="PHILADELPHIA", ]
philly=philly[philly$year > 1914 & philly$year < 1948, ]
philly$cases=na.interpolation(ts(scarlet2))
```

```
cum.reg = smooth.spline(cumsum(philly$rec),
                        cumsum(philly$cases), df=10)
D = - resid(cum.reg) #The residuals
rr = predict(cum.reg, deriv=1)$y
summary(rr)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 0.03348 0.09533 0.10970 0.10990 0.13720 0.16680
```

The reporting rate is around 10%. We create time series corrected for under-reporting of both incidence, I_c , and susceptible deviation, D_c :

```
Ic = philly$cases/rr
Dc = D/rr

seas = rep(1:26, 21)[1:597]
lInew = log(Ic[2:598])
lIold = log(Ic[1:597])
Dold = Dc[1:597]
N = median(philly$pop)
offsetN = rep(-log(N), 597)
```

We set up the vectors for profile likelihood on \bar{S} and loop over all candidates to find the MLE.

```
Smean = seq(0.02, 0.6, by=0.001)*N
llik = rep(NA, length(Smean))
for(i in 1:length(Smean)){
  lSold = log(Smean[i] + Dold)
```

```

    glmfit = glm(lInew ~ -1 +as.factor(seas) + lIold +
                 offset(lSold+offsetN))
    llik[i] = glmfit$deviance
}
Smean[which(llik==min(llik))]/N

## [1] 0.206

```

The estimated mean fraction of susceptibles suggests an R_0 of around 5. Our best estimates for alpha are:

```

lSold = log(Smean[which.min(llik)] + Dold)
glmfit = glm(lInew ~ -1 +as.factor(seas) + lIold +
              offset(lSold+offsetN))
#alpha
glmfit$coef[27]

##      lIold
## 0.8336856

```

and the $\log\beta$ s with SEs are shown in Fig. 7.6. The lower transmission during the summer holiday is conspicuous.

```

beta=exp(glmfit$coef[1:26])
ubeta=exp(glmfit$coef[1:26] +
           summary(glmfit)$coef[1:26, 2])
lbeta=exp(glmfit$coef[1:26] -
           summary(glmfit)$coef[1:26, 2])
plotCI(x=c(1:26), y=beta, ui=ubeta, li=lbeta,
       xlab="Biweek", ylab=expression(beta))

```

7.7 In-Host Malaria Dynamics

Metcalf et al. (2011b) noted the analogies between the TSIR-like dynamics of immunizing human pathogens and the within-host dynamics of malaria-causing parasites in their mammalian hosts. During the blood-stage of infection, infected red blood cells (RBC) burst open in synchrony every 24, 48, or 72 h depending on species to release 6–30 merozoites (depending on species). Merozoites then have a narrow time window to find and invade susceptible cells to start the next replication cycle. The malaria TSIR model assumes that the number of infected cells at generation $t+1$ is captured by $I_{t+1} = P_{E,t} I_t S_t$, where—in this *in-host* model— I_t and S_t are the number of infected and uninfected RBCs and $P_{E,t}$ is the time-varying effective propagation number. This quantity can be thought of as the product of

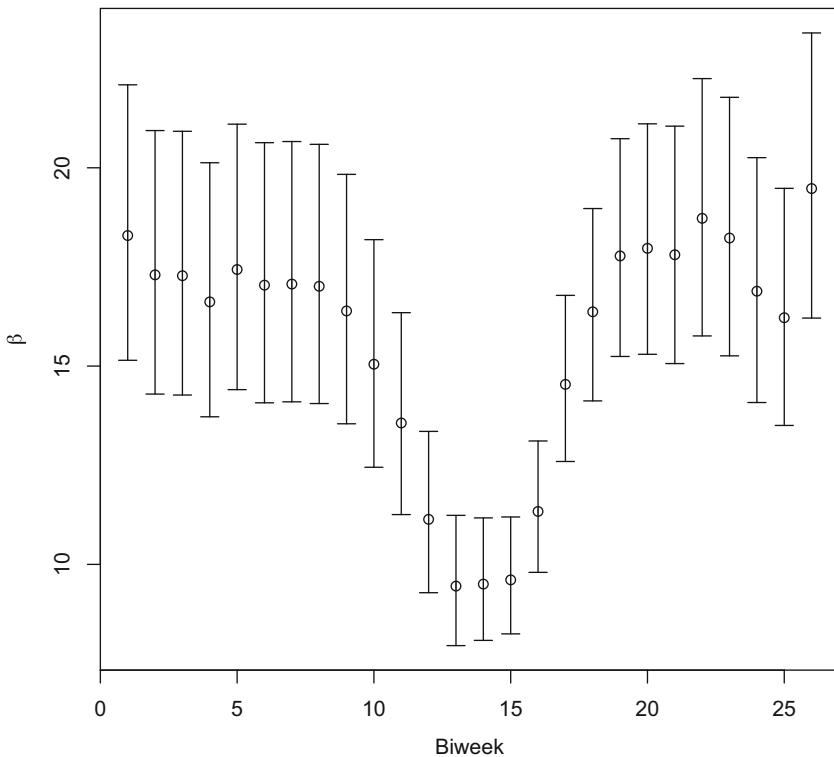


Fig. 7.6 The estimated seasonal β s with SEs for scarlet fever in Philadelphia between 1914 and 1948

merozoite burst size, evasion of host immunity, contact rates between merozoites and uninfected RBCs, and invasion probability given that a contact has occurred.

We will consider data for the mouse parasite, *Plasmodium chabaudii*, which has a 24-h replication cycle, and use daily data from day 3 to 21 of 10 laboratory mice infected with the AQ-strain of *P. chabaudii* as collected by Sylvie Huijben (we will revisit on these data in Sects. 15.3 and 16.6). The SH9 data is in long-format. We use reshape to make matrices with the time series of infected (paras) and uninfected (RBC) red blood cells, and do some basic data formatting:

```

data(SH9)
#subset RBC data
SH9rbc=SH9[,-c(1,3,4,7,8,10,11)]
#Bump up RBC to microliter
SH9rbc[,4]=SH9rbc[,4]*10^6
#subset parasitemia data
SH9para=SH9[,-c(1,3,4,7,8,9, 10)]

```

```
#reshape to wide
SH9rbcw = reshape(SH9rbc, idvar = "Ind2", direction =
    "wide", timevar="Day")
SH9pw = reshape(SH9para, idvar = "Ind2", direction =
    "wide", timevar="Day")
#delete duplicate columns
SH9rbcw=SH9rbcw[,-seq(4,50,by=2)]
names(SH9rbcw) [2] ="Treatment"
SH9pw=SH9pw[,-seq(4,50,by=2)]
names(SH9pw) [2] ="Treatment"

#drop last columns of data not counted every day
SH9pw=SH9pw[, -c(22:27)]
SH9rbcw=SH9rbcw[, -c(22:27)]
#Pull out AQ mice
paras=SH9pw[1:10, -c(1:2)]
SH9rbcw=as.matrix(SH9rbcw[1:10, -c(1:2)])
#Uninfected are total RBCs less infected
RBCs=as.matrix(SH9rbcw-paras)
```

The time series of infected and susceptible RBCs is shown in Fig. 7.7.

```
par(mfrow=c(1,2),bty="l")
matplot(t(log(RBCs)),type="l",xlab="Day",
        ylab="Uninfected log-RBCs")
matplot(t(log(paras)),type="l",xlab="Days",
        ylab="Infected log-RBCs")
```

We fit the model after log-transforming and lagging as needed:

```
Tmax = length(paras[,]) ##max number of days
Nind = length(paras[,1]) ##number of individuals
day = matrix(rep(1:(Tmax-1),each=Nind),Nind,Tmax-1)
day = c(day)

#Log infected cells
log.para = log(paras[,2:Tmax])
log.para.lag = log(paras[,1:(Tmax-1)])
log.para = unlist(c(log.para))
log.para.lag = unlist(c(log.para.lag))

#Log uninfected cells
log.rbcslag = log(RBCs[,1:(Tmax-1)])
log.rbcslag = unlist(c(log.rbcslag))
```

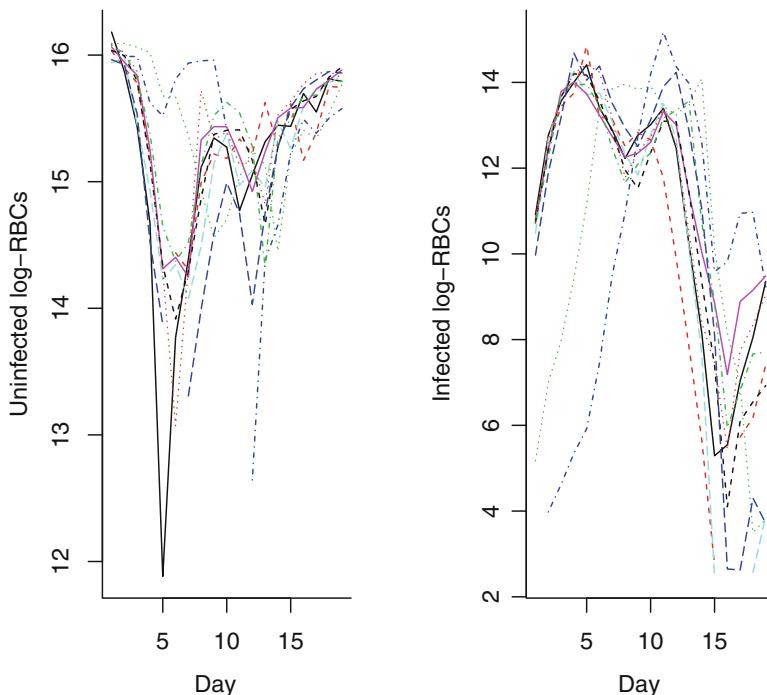


Fig. 7.7 Numbers of infected and susceptible red blood cells in mice infected by the *P. chaudaudi* AQ-strain in 10 different mice

Occasionally, the parasite count is below the detection limit. We replace these zeros with minimum observed values:

```
log.para[!is.finite(log.para)] =
  min(log.para[is.finite(log.para)] , na.rm=T)
log.para.lag[!is.finite(log.para.lag)] =
  min(log.para[is.finite(log.para)] , na.rm=T)
```

The model fitting is similar to that done for measles, except that *Plasmodium* replication occurs in discrete, synchronous cycles (Mideo et al. 2013); so the model does not need the α exponent.

```
data = data.frame(log.para=log.para, day=day,
  log.para.lag=log.para.lag, log.rbcslag=log.rbcslag)
fit = glm(log.para ~ -1+as.factor(day) +
  offset(log.para.lag+log.rbcslag), data=data)
```

Figure 7.8 shows the estimated daily propagation numbers and associated in-host effective reproductive numbers ($R_{E,t} = P_{E,t}S_t$). The R_E s are initially around 6 which is close to (but a little smaller than) the burst-size of *P. chaubodi*. This drops to around one after a week. Metcalf et al. (2011b) discuss how the drop in R_E reflects susceptible depletion and the action of innate and acquired immunity.

```
par(mfrow=c(1, 2))
require(plotrix)
ses = summary(fit)$coeff[, 2]
beta=exp(fit$coef)
ubeta=exp(fit$coef+ses)
lbeta=exp(fit$coef-ses)
plotCI(x=c(3:20), y=beta, ui=ubeta, li=lbeta,
       xlab="Day", ylab=expression(P[E]))
points(x=c(3:20), exp(fit$coeff), type="b", pch=19)
plotCI(x=c(3:20), y=beta*colMeans(RBCs)[-19], ui=ubeta*
       colMeans(RBCs)[-19], li=lbeta*colMeans(RBCs)[-19],
       xlab="Day", ylab=expression(R[E]))
points(x=c(3:20), beta*colMeans(RBCs)[-19],
       type="b", pch=19)
abline(h=1, lty=3)
```

7.8 ShinyApp

The epimdr-package contains the TSIR.app to simulate the nonseasonal TSIR. The app can be launched from R through:

```
require(shiny)
TSIR.app
```

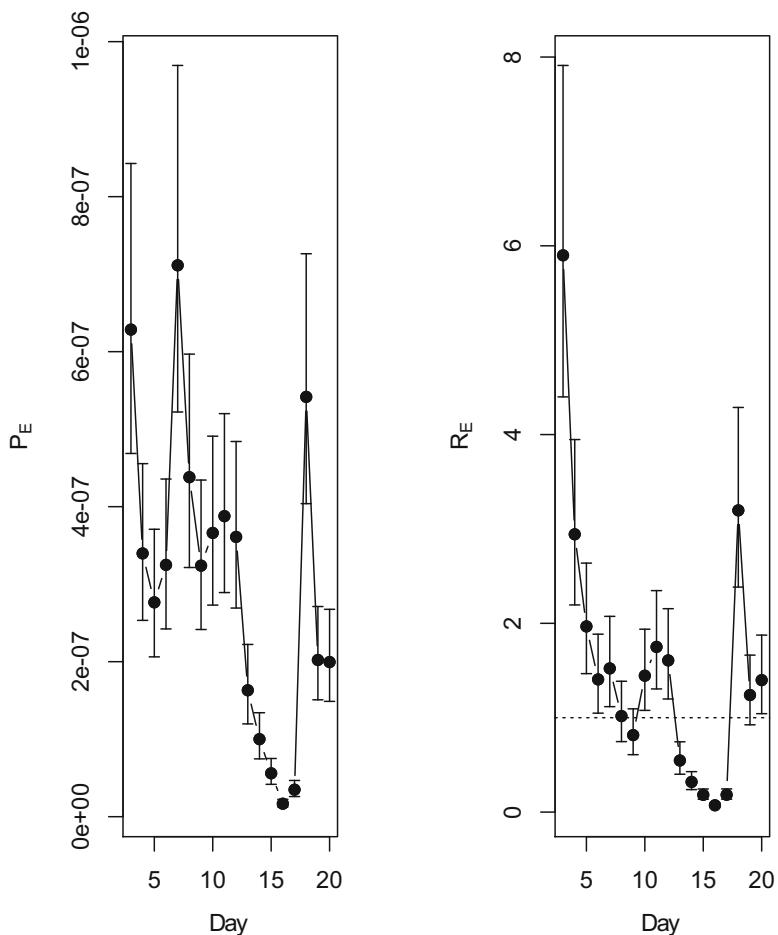


Fig. 7.8 Estimated propagation numbers and effective in-host reproductive numbers of *P. chau-daudi* in mice infected with the AQ strain

Chapter 8

Trajectory Matching



8.1 Preamble: Prevalence Versus Incidence

When we fit mechanistic models to data, we have to consider carefully the relationship between the nature of the data *versus* the nature of the model state variables. For example, when we work with continuous-time S(E)IR models it is important to keep in mind that *incidence is not prevalence*; so if our data is incidence we will need to do something more than trying to match simulated prevalence with observed incidence. We therefore start with a toy example using simulated data.

When/if we can assume that dynamics is unaffected by *process noise* (demographic and environmental stochasticity), we can fit models to data using trajectory matching. The assumption is that discrepancies between the observations and the predictions from the dynamic model are due to observational errors. The upside of trajectory matching is that we can easily fit continuous-time models to variably spaced observations on any/all state-variable, the downside is that these assumptions are usually restrictive.

8.2 Event-Based Stochastic Simulation

To begin, we will consider how to stochastically simulate the continuous-time SIR model (Eqs. (2.1)–(2.3)). Previously we consider stochastic simulation using discrete-time models. An alternative is to do continuous-time stochastic simulation using an event-based approach: The Gillespie exact algorithm (Gillespie 1977) and

This chapter uses the following R-package: deSolve.

the τ -leap approximation (Gillespie 2001). As discussed in Sect. 2.7, the S(E)IR-model (and all simple ODEs) implies exponentially distributed waiting times between events. The Gillespie algorithms take advantage of this idea. If we for example consider how the SIR-states of the SIR flows ((2.1)–(2.3)) should change over time, we expect the following six possible changes:

- $S \rightarrow S + 1$ at rate μN from births
- $S \rightarrow S - 1$ at rate μS from deaths
- $S \rightarrow S - 1$ and $I \rightarrow I + 1$ at rate $\beta SI/N$ from infection
- $I \rightarrow I - 1$ at rate μI from deaths
- $I \rightarrow I - 1$ and $R \rightarrow R + 1$ at rate γI from recovery
- $R \rightarrow R - 1$ at rate μR from deaths

Thus, the system is expected to change by an overall summed rate of $r = \mu N + \mu S + \beta SI/N + \mu I + \gamma I + \mu R$. We can therefore draw a random exponential waiting-time with mean r to update a continuous-time clock, then draw a random event from a multinomial distribution with probabilities given by the relative rates, update the state variables accordingly, and repeat...

Because of the many versions of compartmental models used in studying disease dynamics, it is useful to write a general purpose stochastic simulator that can be applied to any set of rate equations. To this end we first define a `rlist`-list of equations corresponding to the rates for the six transitions of the SIR flows. The `quote`-formalism allows us to set up the list such that all equations can be evaluated in a single `sapply`-call as the simulation progress.

```
rlist=c(quote(mu * (S+I+R)), #Births
quote(mu * S), #Susceptible deaths
quote(beta * S * I / (S+I+R)), #Infection
quote(mu * I), #Infected death
quote(gamma * I), #Recovery
quote(mu*R) ) #Recovered death
```

We next define a transition matrix associated with each SIR event. The three columns correspond to changes in S, I, and R, respectively; The rows correspond to the six possible events.

```
emat=matrix(c(1, 0, 0,
-1, 0, 0,
-1, 1, 0,
0, -1, 0,
0, -1, 1,
0, 0, -1),
ncol=3, byrow=TRUE)
```

We finally write a general-purpose function to simulate a dynamical systems using the Gillespie algorithm. The idea is to write a function that is sufficiently robust

and general that it can be applied to event-based stochastic simulation of any model that fits within a compartmental framework. The function takes five arguments to accomplish this:

- `rateqs`—a list of E rate equations corresponding to each of the E possible events using the quote-formalism
- `eventmatrix`—a E-by-S matrix of changes to each of the S state variables associated with each event
- `parameters`—a vector of parameter values
- `initialvals`—a vector of initial values for the S states
- `numevents`—number of events to be simulated

```
gillespie=function(rateqs, eventmatrix, parameters,
    initialvals, numevents){
  res=data.frame(matrix(NA, ncol=length(initialvals)+1,
    nrow=numevents+1))
  names(res)=c("time", names(initialvals))
  res[1,]=c(0, initialvals)
  for(i in 1:numevents){
    #evaluate rates
    rat=sapply(rateqs, eval,
      as.list(c(parameters, res[i,])))
    #update clock
    res[i+1,1]=res[i,1]+rexp(1, sum(rat))
    #draw event
    whichevent=sample(1:nrow(eventmatrix), 1, prob=rat)
    #update states
    res[i+1,-1]=res[i,-1]+eventmatrix[whichevent,]
  }
  return(res)
}
```

We provide parameters and initial conditions for a stochastic simulation assuming an infectious period of 20 days (Fig. 8.1):

```
paras=c(mu=1, beta=500, gamma=365/20)
inits=c(S=100, I=2, R=0)
sim=gillespie(rlist, emat, paras, inits, 1000)
matplot(sim[,1],sim[,2:4], type="l", ylab="Numbers",
  xlab="Time", log="y")
legend("topright", c("S", "I", "R"), lty=c(1,1,1),
  col=c(1,2,3))
```

The Gillespie algorithm provides an “exact” stochastic simulation in the sense that the time-evolution of the system is changing exactly according to the expo-

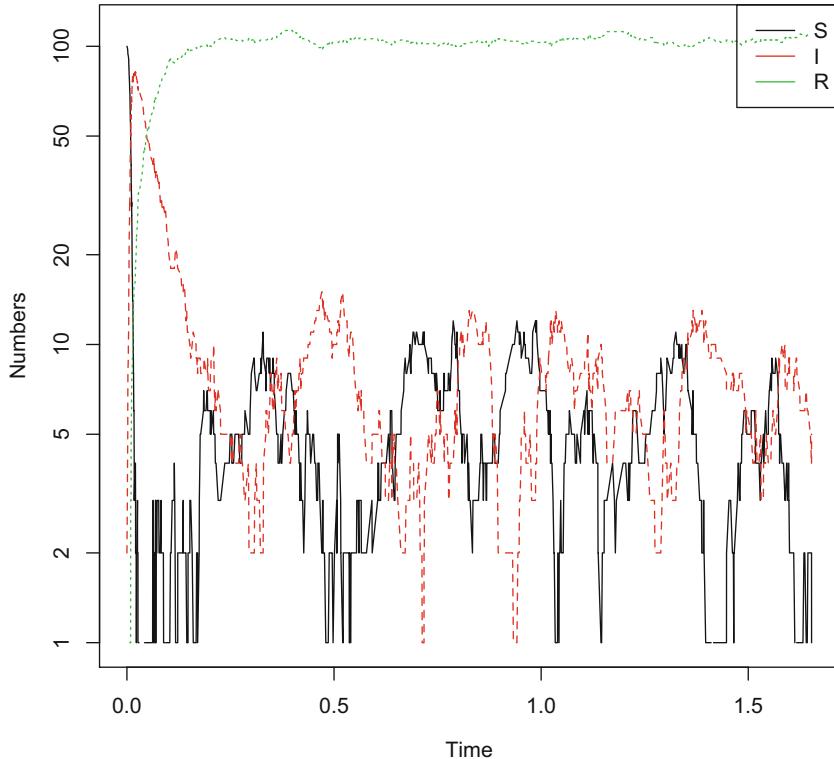


Fig. 8.1 A Gillespie exact simulation of the stochastic SIR model with $\mu = 1$, $\beta = 500$, and $\gamma = 365/20$

ential waiting-time distributions of the stochastic differential system. It is, however, computationally expensive as every event is recorded separately. Gillespie's τ -leap method uses the Poisson approximation corresponding to the discussion of Sect. 7.1; If we assume that the interval, Δt , is sufficiently short that any change in the rates are negligible, the number of events should be Poisson-distributed with mean *overallrate* * Δt and multinomially divided among the events according to their relative rates.

We write a general τ -leap simulator and then apply it to the SEIR model. The SEIR model has eight possible events:

- $S \rightarrow S + 1$ at rate μN from births
- $S \rightarrow S - 1$ at rate μS from deaths
- $S \rightarrow S - 1$ and $E \rightarrow E + 1$ at rate $\beta SI/N$ from infection
- $E \rightarrow E - 1$ at rate μE from deaths
- $E \rightarrow E - 1$ and $I \rightarrow I + 1$ at rate σE from becoming infectious
- $I \rightarrow I - 1$ at rate μI from deaths
- $I \rightarrow I - 1$ and $R \rightarrow R + 1$ at rate γI from recovery
- $R \rightarrow R - 1$ at rate μR from deaths

We thus have the following event matrix:

```
emat2=matrix(c(1,0,0,0,
-1,0,0,0,
-1,1,0,0,
0,-1,0,0,
0,-1,1,0,
0,0,-1,0,
0,0,-1,1,
0,0,0,-1),
ncol=4, byrow=TRUE)
```

The SEIR equations associated with each event are:

```
rlist2=c(quote(mu * (S+E+I+R)), quote(mu * S),
        quote(beta * S * I/(S+E+I+R)), quote(mu*E),
        quote(sigma * E), quote(mu * I),
        quote(gamma * I), quote(mu*R))
```

A general-purpose τ -leap simulator is:

```
tau=function(rateqs, eventmatrix, parameters,
            initialvals, deltaT, endT){
  time=seq(0, endT, by=deltaT)
  res=data.frame(matrix(NA, ncol=length(initialvals)+1,
                        nrow=length(time)))
  res[,1]=time
  names(res)=c("time", names(initialvals))
  res[1,]=c(0, initialvals)
  for(i in 1:(length(time)-1)){
    #calculate overall rates
    rat=sapply(rateqs, eval, as.list(c(parameters,
                                         res[i,])))
    evts=rpois(1, sum(rat)*deltaT)
    if(evts>0){
      #draw events
      whichevent=sample(1:nrow(eventmatrix), evts,
                        prob=rat, replace=TRUE)
      mt=rbind(eventmatrix[whichevent,],
                t(matrix(res[i,-1])))
      mt=matrix(as.numeric(mt), ncol=ncol(mt))
      #update states
      res[i+1,-1]=apply(mt, 2, sum)
      res[i+1, ][res[i+1,<0]]=0
    }
    else{ #if no events in delat
```

```

    res[i+1, -1] = res[i, -1]
  }
return(res)
}

```

We assume an initial population comprised of 1000 individuals and 1 initial infected and simulate daily incidence for 2 years and assume measles-like parameters:

```

paras  = c(mu = 1, beta = 1000,
           sigma = 365/8, gamma = 365/5)
inits = c(S=999, E=0, I=1, R = 0)
sim2=tau(rlist2, emat2, paras, inits, 1/365, 2)
matplot(sim2[,1],sim2[,2:5], type="l", log="y",
        ylab="Numbers", xlab="Time")
legend("bottomright", c("S", "E", "I", "R"),
       lty=c(1,1,1,1), col=c(1,2,3,4))

```

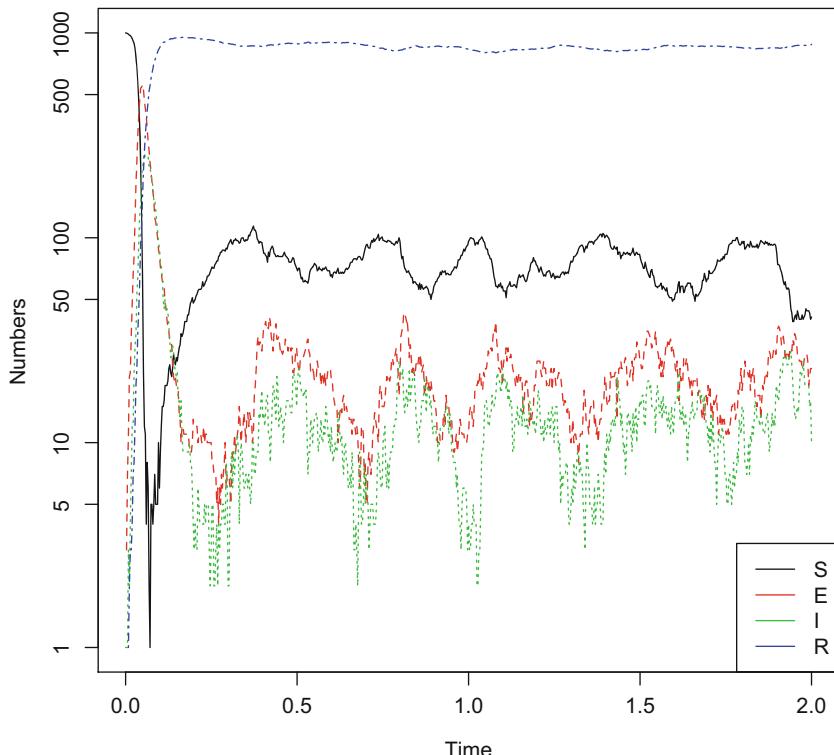


Fig. 8.2 A τ -leap simulation of the SEIR model using a daily time step for 2 years assuming $\mu = 1$, $\beta = 1000$, an infectious period of 8 days, a latent period of 5 days, and an initial population comprised of 1000 individuals one of which is infected

Following the virgin epidemic, the inherent birth/death stochasticity leads to low-amplitude oscillations (Fig. 8.2) according to the resonant periodicity of the SEIR model (see Chap. 9).

8.3 Trajectory Matching

Trajectory matching assumes that the discrepancies between models and data are due to error of observation. The event-based, stochastic simulation breaks with this assumption as model discrepancies are due to demographic stochasticity; Let us nevertheless see if we can fit the SEIR model to the event-based simulation. We first recall the gradient-function for the system:

```
require(deSolve)
seirmod = function(t, y, parms) {
  S = y[1]
  E = y[2]
  I = y[3]
  R = y[4]

  with(as.list(parms), {
    dS = mu * (N - S) - beta * S * I/N
    dE = beta * S * I/N - (mu + sigma) * E
    dI = sigma * E - (mu + gamma) * I
    dR = gamma * I - mu * R
    res = c(dS, dE, dI, dR)
    list(res)
  })
}
```

Following the ideas introduced in Sect. 3.4, we define a likelihood function to estimate parameters. The Gaussian log-likelihood is $= \text{const} - \frac{n}{2} \log(\text{RSS})$, where n is the length of the time series, RSS is the residual sum-of-squares, and the constant is $n(\log(n) - \log(2\pi) - 1)/2$ (Aitkin et al. 2005).¹

```
lfn=function(p) {
  times = seq(0, 2, by=1/365)
  start = c(S=999, E=0, I=1, R = 0)
  paras=exp(c(mu=p[1], N=p[2], beta=p[3],
```

¹ If in a hurry we can ignore the constant and minimize $\frac{n}{2} \log(\text{RSS})$ because it is the relative likelihood that matters.

```

        sigma=p[4], gamma=p[5]))
out = as.data.frame(ode(start, times,
    seirmod, paras))
n=length(sim2$I)
rss=sum((sim2$I-out$I)^2)
return(log(rss)*(n/2)-n*(log(n)-log(2*pi)-1)/2)
}

```

We next estimate parameters:

```

# initial values for mu, N, beta, sigma, gamma
paras0 = log(c(2, 500, 500, 365/7, 365/7))
fit = optim(paras0, lfn, hessian = TRUE)

```

and plot the deterministic prediction:

```

times = seq(0, 2, by=1/365)
paras = exp(c(mu = fit$par[1], N = fit$par[2],
    beta = fit$par[3], sigma = fit$par[4],
    gamma = fit$par[5]))
start = c(S=999, E=0, I=1, R = 0)
out = as.data.frame(ode(start, times, seirmod, paras))
plot(out$time, out$I, xlab="Time", ylab="Prevalence",
    type="l")
lines(sim2$time, sim2$I, col=2, type="l")
legend("topright", c("Gillespie simulation",
    "SEIR fit"), lty=c(1,1), col=c(2,1))

```

The trajectory-match'ed fit predicts the virgin epidemic and the next damped epidemic well, but not—as expected—the subsequent stochastically excited low-amplitude cycles (Fig. 8.3). In addition to finding parameter estimates we are usually interested in uncertainty and trade-offs among parameters in producing a fit to the data.

8.4 Likelihood Theory 101

We have used maximum likelihood principles in several of our previous analysis of, for example, the chain-binomial, the catalytic and the TSIR models. We have, however, not discussed likelihood theory in a formal fashion.² For our purposes it

² Bolker (2008) is an excellent broad discussion on estimation for ecologically realistic models using a variety of methods.

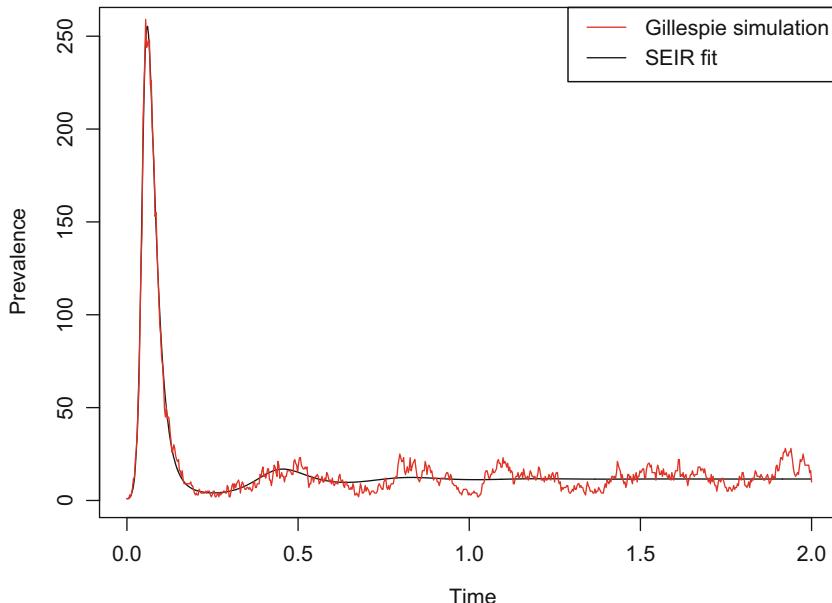


Fig. 8.3 SEIR fitted predicted trajectory superimposed on the τ -leap simulation of the SEIR model

is useful to summarize the key results with respect to inference from ‘‘elementary’’ likelihood theory with maximum brevity (see, for example, appendix A of McCullagh and Nelder 1989):

- Let $L(D|\theta)$ be the function that calculates the likelihood for a set of data, D ; i.e., the probability of observing the data given some values for the parameters, θ . The values that maximize this probability are the maximum likelihood estimates (MLEs) of the parameters, $\hat{\theta}$.
- If $\ell(\theta)$ is the negative log-likelihood (i.e., $-\log L$), then $\hat{\theta}$ are the values that minimizes ℓ . If data points are independent, then the joint log-likelihood is simply the sum of the log-likelihoods of the data points.
- The MLE is a minimum of ℓ , so the *score function* $U(\theta) = \partial\ell/\partial\theta$ is zero at the MLE.
- The likelihood profile graphs how $\ell(\theta)$ changes with θ . The 95% confidence interval is the set of values of θ for which $\ell(\theta)$ is within $\chi^2(0.95, p)/2$ of the minimum, where p is the number of parameters. The quantity $2\ell(\theta)$ is referred to as the *deviance*, so if we work with the deviance we would use $\chi^2(0.95, p)$ as the cut-off.
- The second derivative of $\ell(\theta)$ with respect to θ is called the *Fisher information*, $i(\theta) = \partial^2\ell/\partial\theta^2$. The inverted information matrix is an approximation to the variance-covariance matrix of the parameters, so we can obtain approximate standard errors as the square-root of the diagonal of the inverted information ma-

trix. The approximate correlation matrix is the standardized inverted information matrix.

- A matrix of second derivatives is generally referred to as a [Hessian matrix](#). If we call `optim(..., hessian=TRUE)`, R will numerically estimate the Hessian at the minimum, so if the function to be minimized is the negative log-likelihood, we can obtain approximate SEs and the approximate correlation matrix from this Hessian.
- If we have two alternative models that are *nested*—meaning that the more complex model contains all the parameters of the simpler—then we can test for significant model improvement; the difference in the log-likelihood is $\chi^2(df = \Delta p)/2$ -distributed, where Δp is the number of extra parameters in the complex model.³

We apply these ideas to our model fit:

```
# MLEs:
round(exp(fit$par), 4)
## [1] 1.5165 642.3009 563.3407 50.3001 69.6786

# Approximate SES:
round(sqrt(diag(solve(fit$hessian))), 4)
## [1] 1.2744 1.2441 1.2561 1.0224 1.0240

# Correlation matrix:
round(cov2cor(solve(fit$hessian)), 4)
##          [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 1.0000 -0.9976 -0.9972  0.6107 -0.9546
## [2,] -0.9976  1.0000  0.9974 -0.5872  0.9649
## [3,] -0.9972  0.9974  1.0000 -0.6421  0.9543
## [4,]  0.6107 -0.5872 -0.6421  1.0000 -0.4653
## [5,] -0.9546  0.9649  0.9543 -0.4653  1.0000
```

The true parameter values used in the simulation were $\mu = 1$, $N = 1000$, $\beta = 1000$, $\sigma = 45.6$, and $\gamma = 73$. So while the model prediction gives a good fit, the parameter estimates are not particularly accurate. This is where it is useful to use likelihood theory more extensively. From the normalized inverted Hessian we see that several of the parameters are highly (positively or negatively) correlated, and several with correlations more extreme than ± 0.9 . That means that different parameter combination may provide a very similar fit to the data. This is an illustration of *identifiability problems*; With observations only on the infectious stage, for instance, a relatively short infectious period and high transmission rate will predict a

³ If the models are *non-nested*, formal tests are not available but information theoretical rankings of models using AIC, BIC, AIC-weights, etc. are useful.

similar trajectory to a relatively short latent period and a lower transmission rate. Furthermore, a smaller population size and higher birth rate can result in identical susceptible recruitment rate than a larger population with lower birth rate. For inference it is therefore normally best to inform the analysis with any known biological quantities; For example if the latent and infectious periods are known from household or clinical studies, it may be best not to attempt to infer these from the time series alone (though, as King et al. (2008) point out for Cholera dynamics, conventional wisdom may not always be consistent with dynamical patterns). Moreover, if there are strong correlations, the individual SEs (and CIs derived there from) may be a poor representation of parametric uncertainty. It may then be better to look at pairwise confidence ellipses (e.g., Bolker 2008).

8.5 SEIR with Error

We can use `ode` to integrate the SEIR model and add noise, to generate a data set that exactly adheres to the assumption that the dynamics is only affected by observational noise. Let us simulate 10 years of weekly data assuming measles'ish parameters and that 6% of the initial population is susceptible:

```

times  = seq(0, 10, by=1/52)
paras = c(mu = 1/50, N = 1, beta = 1000,
          sigma = 365/8, gamma = 365/5)
          start = c(S=0.06, E=0, I=0.001, R = 0.939)
out = as.data.frame(ode(start, times, seirmod, paras))

```

We add noise to the data using the `jitter`-function (Fig. 8.4),

```
datay = jitter(out$1, amount = 1e-04)
```

```
plot(times, datay, ylab = "Infected", xlab = "Time")
lines(times, outSI, col = 2)
```

define a Gaussian likelihood function.

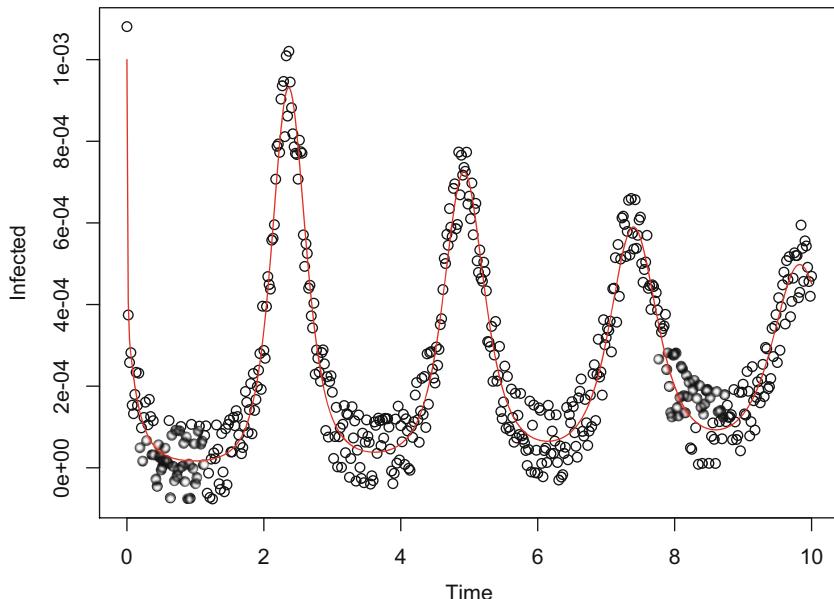


Fig. 8.4 Fraction infectious and jittered data from the SEIR model assuming $\mu = 0.02$, $\beta = 1000$, $\sigma = 45.6$, and $\gamma = 73/\text{year}$

```

rss=sum((data-out$I)^2)
return(log(rss)*(n/2)-n*(log(n)-log(2*pi)-1)/2)
}

```

and estimate parameters using the jittered observations.

```

# mu, N, beta, sigma, gamma
paras0 = c(1/30, 1, 1500, 365/4, 365/10)
fit = optim(paras0, lfn, data = datay, hessian = TRUE)

```

The estimates are

```

# MLEs:
round(fit$par, 3)

## [1] 0.036 1.031 2179.946 71.197 138.851

# Approximate SES:
round(sqrt(diag(solve(fit$hessian))), 3)

## [1] 0.003 0.066 200.261 7.584 8.718

# Correlation matrix:
round(cov2cor(solve(fit$hessian)), 3)

```

```
##      [,1]   [,2]   [,3]   [,4]   [,5]
## [1,] 1.000 -0.743 -0.330 -0.407  0.374
## [2,] -0.743 1.000  0.820 -0.258  0.204
## [3,] -0.330  0.820  1.000 -0.658  0.714
## [4,] -0.407 -0.258 -0.658  1.000 -0.821
## [5,]  0.374  0.204  0.714 -0.821  1.000
```

8.6 Boarding School Flu Data

The boarding school flu data set introduced in Sect. 3.6.1 has an approximate match between *observation* and *prevalence* because the data represents the number of children confined to bed each day, and while the average stay in bed (3–7 days) is maybe a bit different than the infectious period, the durations are comparable.

```
data(flu)
```

We define the gradient functions for a closed SIR epidemic:

```
sirmod = function(t, y, params) {
  S = y[1]
  I = y[2]
  R = y[3]
  with(as.list(params), {
    dS = -beta * S * I/N
    dI = beta * S * I/N - gamma * I
    dR = gamma * I
    res = c(dS, dI, dR)
    list(res)
  })
}
```

and define the likelihood function assuming normally distributed errors

```
lfn2 = function(p, I, N) {
  times = seq(1, 14, by = 1)
  start = c(S = N, I = 1, R = 0)
  paras = c(beta = p[1], gamma = p[2], N = N)
  out = as.data.frame(ode(start, times = times,
    sirmod, paras))
  n = length(I)
  rss = sum((I - out$I)^2)
```

```

    return(log(rss) * (n/2) - n * (log(n) -
        log(2 * pi) - 1)/2)
}

```

There are two parameters to estimate: β and γ . The time-scale is daily so we set reasonable initial conditions and maximize the likelihood:

```

#beta, gamma
paras0 = c(1.5, 1/2)
flufit = optim(paras0, lfn2, I = flu$cases, N = 763,
               hessian = TRUE)

```

The estimated parameters and basic reproductive ratio, R_0 , are:

```

# parameters
flufit$par

## [1] 1.9566375 0.4738335

# R0:
flufit$par[1]/flufit$par[2]

## [1] 4.129377

```

The R_0 estimate is comparable to the estimate we made in Chap. 3. The observed and predicted outbreaks are seemingly a good match (Fig. 8.5):

```

times = seq(1, 20, by=.1)
start = c(S=762, I=1, R = 0)
paras=c(beta=flufit$par[1], gamma=flufit$par[2], N=763)
out = as.data.frame(ode(start, times,
                        sirmod, paras))
plot(out$time, out$I, ylab="Prevalence",
      xlab="Day", type="l")
points(flu$day, flu$cases)

```

8.7 Measles

We consider, again, the measles incidence data collected by Doctors Without Borders (MSF) during the 03/04 outbreak in Niamey, Niger (Fig. 8.6; Sect. 3.4) but using data at a daily resolution. We can compile the daily incidence in a vector y .

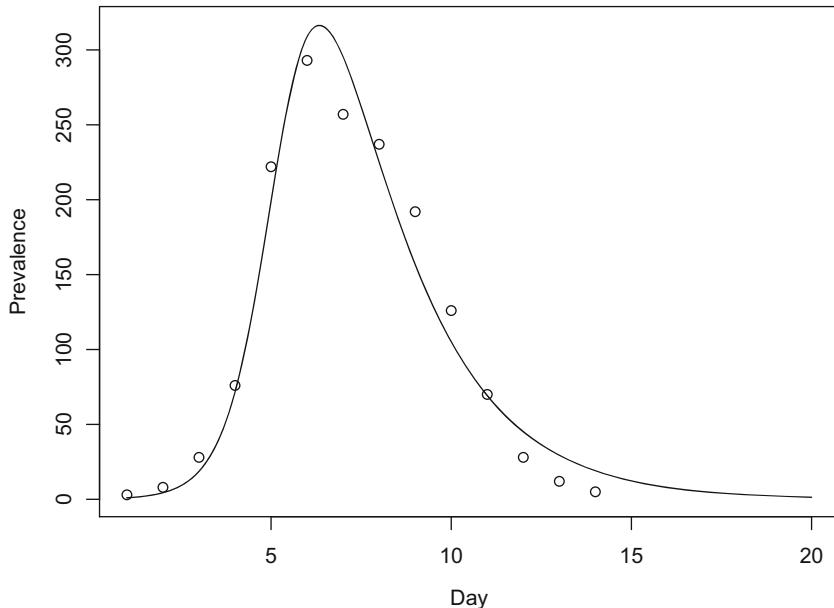


Fig. 8.5 Predicted and observed influenza prevalence for the 1978 boarding school data

```
y = as.vector(table(niamey_daily))
```

The challenge with this data is that we need to make the SEIR-formulation relevant to the data on incidence. The complication is that I represents *prevalence* (i.e., current number of infected individuals), while incidence, y , represents appearance of new cases (i.e., *flux*) into the infected class. If we recast the SEIR model to also keep track of cumulative incidence, K , we can *difference* the K time series at time-steps corresponding to that of the observations to predict incidence (y). We define the SEIRK-model assuming known latent and infectious periods of 8 and 5 days, respectively.

```
times= unique(niamey_daily$day)
paras = c(mu = 0, N = 1, beta = 5, sigma = 1/8,
          gamma = 1/5)
start = c(S=0.999, E=0, I=0.001, R = 0, K = 0)
```

The resultant gradient function is:

```
seirkmod = function(t, x, params) {
  S = x[1]
  E = x[2]
  I = x[3]
  R = x[4]
```

```

K = x[5]

with(as.list(params), {
  dS = mu * (N - S) - beta * S * I/N
  dE = beta * S * I/N - (mu + sigma) * E
  dI = sigma * E - (mu + gamma) * I
  dR = gamma * I - mu * R
  dK = sigma * E
  res = c(dS, dE, dI, dR, dK)
  list(res)
})
}
}

```

We next define the likelihood function (assuming Poisson distributed errors) for the unknown transmission rate, β , and initial susceptible number, N . According to the MSF outbreak response protocol, an outbreak is declared once five cases have been confirmed. The unknown infectious fraction is thus $5/N$.

```

lfn4=function(p, I){
times=unique(niamey_daily$day)
xstart=c(S=(p[1]-5)/p[1], E=0, I=5/p[1], R = 0,
         K=0)
paras=c(mu=0, N=p[1], beta=p[2], sigma=1/8,
        gamma=1/5)
out=as.data.frame(ode(xstart, times=times, seirkmod,
                      paras))
predinci=c(xstart["I"], diff(out$K))*p[1]
ll=-sum(dpois(I, predinci, log=TRUE))
return(ll)
}

```

For starting values we assume initial susceptible numbers $N = 11,000$ and $\beta = 5$ and optimize:

```

#N, beta
paras0 = c(11000, 5)
measfit=optim(paras0,lfn4,I=y, hessian=TRUE)
day = 1:230
xstart = c(S=(measfit$par[1]-5)/measfit$par[1], E=0,
           I=5/measfit$par[1], R = 0, K = 0)
paras=c(mu=0, N=measfit$par[1], beta=measfit$par[2],
        sigma=1/8, gamma=1/5)
out = as.data.frame(ode(xstart, times=day,
                        seirkmod, paras))

```

```
plot(table(niamey_daily), xlab="Day", ylab="Incidence")
lines(out$time, c(xstart["I"], diff(out$K)) *
    measfit$par[1], col=2, lwd=2)
```

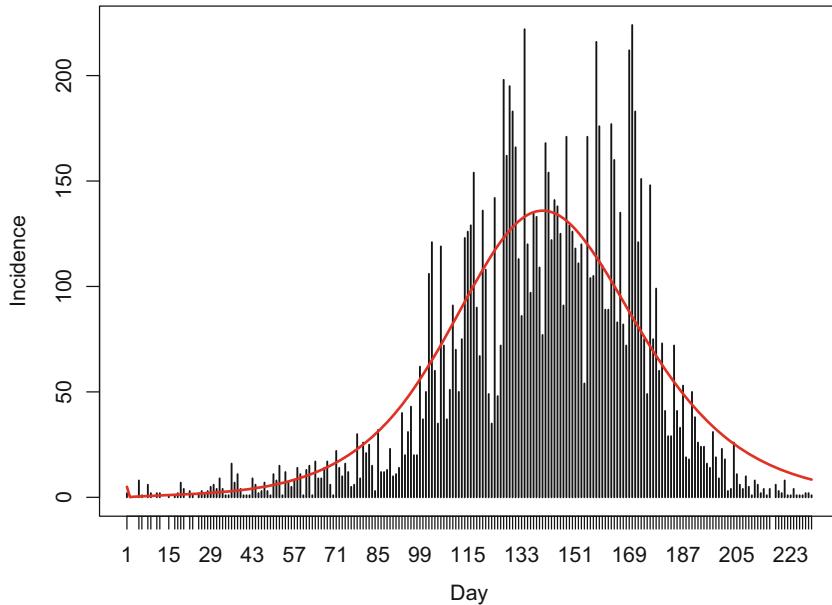


Fig. 8.6 Predicted and observed measles incidence using the MLEs from the Poisson-likelihood

The estimated effective reproductive ratio, R_E , is comparable to the estimates obtained in Chap. 3:

```
with(as.list(paras),
  sigma/(sigma+mu)*1/(gamma+mu)*beta /N)
## [1] 1.761133
```

8.8 Outbreak-Response Vaccination

Grais et al.'s (2008) objective in fitting a model to the Niamey outbreak data was to evaluate the effectiveness of outbreak-response vaccination (ORV) in reducing the burden of disease during an on-going outbreak. The ORV campaign began on

day 161 after the beginning of the epidemic with a goal of vaccinating 50% of all children of ages between 9 months and 5 years. After 10 days, almost 85,000 (57%) of this at-risk group was vaccinated (without knowledge of previous disease or vaccination status). Assuming vaccination was at random with respect to immune status, we can write a modified SEIR function to study the problem. The vaccine cover is a fraction—effectively a *probability*—so we need to translate it to a rate using the relation discussed in Sect. 3.2: $r = -\log(1-p)/D$, where D is the length of the campaign. We define two functions to carry out the efficacy calculations. The `sivmod`-function integrates the SI-model with outbreak-response vaccination and the `retrospec`-function compares predicted epidemic trajectories with and without the ORV.

```
sivmod=function(t,x,parms){
  S=x[1]
  E=x[2]
  I=x[3]
  R=x[4]
  K=x[5]
  with(as.list(parms),{
    Q= ifelse(t<T | t>T+Dt,0,(-log(1-P)/Dt))
    dS= -B*S*I-q*Q*S
    dE= B*S*I-r*E
    dI= r*E - g*I
    dR= g*I+q*Q*S
    dK=r*E
    res=c(dS,dE,dI,dR,dK)
    list(res)
  })
}

retrospec=function(R, day, vaccine_efficiency,
  target_vaccination,intervention_length,
  mtime, LP=7, IP=7, N=10000){
  steps=1:mtime
  out=matrix(NA,nrow=mtime, ncol=3)
  #starting values
  xstrt=c(S=1-1/N,E=0,I=1/N,R=0,K=0)
  beta= R/IP           #transmission rate
  #Without ORV
  par=c(B=beta, r=1/LP, g = 1/IP, q = vaccine_efficiency,
    P = 0, Dt = 0, T = Inf, R=R)
```

```

outv=as.data.frame(ode(xstrt,steps,sivmod,par) )
fsv=max(outv$K)
#With ORV
par=c(B=beta, r=1/LP, g = 1/IP, q = vaccine_efficiency,
       P = target_vaccination, Dt =
       intervention_length, T = day)
outi=as.data.frame(ode(xstrt,steps,sivmod,par) )
fsi=max(outi$K)
res=list(redn=fsi/fsv, out=outv, orv=outi, B=par["B"] ,
         r=par["r"], g=par["g"], q=par["q"], P=par["P"] ,
         Dt=par["Dt"], T=par["T"], R=R)
class(res)="retro"
return(res)
}

```

We will discuss S3-class programming more formally in Sect. 12.1. However, as a preview we define a `plot.retro`-function for objects of class `retro` as the list returned by the `retrospec`-function is labeled:

```

plot.retro=function(x){
  plot(x$out[,1], x$out[, "I"], type="l", ylim=c(0,
    max(x$out[, "I"])), xlab='Day', ylab='Prevalence')
  polygon(c(x$T, x$T, x$T+x$Dt,
            x$T+x$Dt), c(-0.1,1,1,-.1), col="gray")
  lines(x$out[,1], x$out[, "I"])
  lines(x$orv[,1], x$orv[, "I"], col="red")
  title(paste("Final size: ", round(100*(x$redn),1),
              "% (R=",x$R,", target=", 100*x$P, "%)", sep=""))
  legend(x="topleft", legend=c("Natural epidemic",
    "With ORV"), col=c("black", "red"), lty=c(1,1))
  text(x=x$T+x$Dt, y=0, pos=4,
    labels=paste(x$intervention_length,
    "ORV from ", x$T))
}

```

If we assume our model is correct and that the vaccine either elicits instantaneous protection or after 2 or 4 weeks (for the antibody response to mature), the ORV is predicted to have reduce the epidemic by 25%, 15%, or 8% respectively:

```

red1=retrospec(R=1.8, 161, vaccine_efficiency=0.85,
               target_vaccination=0.5, intervention_length=10,
               mtime=250, LP=8, IP=5, N=16000)
red2=retrospec(R=1.8, 161+14, vaccine_efficiency=0.85,
               target_vaccination=0.5, intervention_length=10,
               mtime=250, LP=8, IP=5, N=16000)
red3=retrospec(R=1.8, 161+28, vaccine_efficiency=0.85,
               target_vaccination=0.5, intervention_length=10,
               mtime=250, LP=8, IP=5, N=16000)
1-red1$redn
## [1] 0.2612989
1-red2$redn
## [1] 0.1509867
1-red3$redn
## [1] 0.07827277

```

We can plot the `red1`-object to inspect the predicted epidemic curve with and without outbreak-response vaccination (Fig. 8.7). The key insight is that for ORVs to work it needs to be implemented early (Grais et al. 2008).

```
plot(red1)
```

8.9 ShinyApp

The `epimdr`-package contains the `orv.app` with a more detailed sensitivity analyses of outbreak response vaccine scenarios. The app can be launched from R through:

```

require(shiny)
orv.app

```

Final size: 73.9% ($R=1.8$, target=50%)

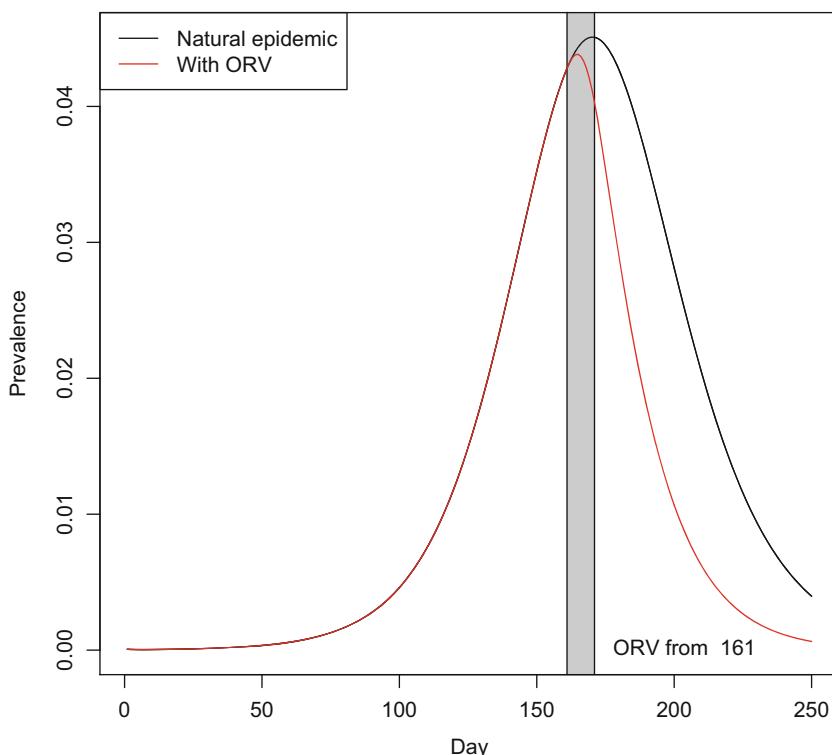


Fig. 8.7 Epidemic curve with and without outbreak-response vaccination starting on day 161 with a target of 50%, vaccine efficacy of 85%, campaign duration of 10 days, and an effective reproductive ratio of 1.8

Chapter 9

Stability and Resonant Periodicity



9.1 Preamble: Rabies

Rabies usually invades a naive host range in spatial waves. This has been documented in great detail for fox rabies in continental Europe and raccoon rabies in the Eastern USA. The rabies data set collected by CDC is the monthly number of rabid raccoons by state. The time column starts from first month of invasion for each state (Childs et al. 2000). The incidence patterns follow the characteristic pattern of major virgin epidemics followed by a fuzzy but distinct periodic recurrence intervals between 3 and 4 years (Fig. 9.1). Theory should allow us to predict such recurrence intervals.

```
data(rabies)
matplotlib(rabies[, 2:7], ylab="Cases", xlab="Month")
legend("topright", c("CT", "DE", "MD", "MA",
"NJ", "NY"), pch=as.character(1:6), col=1:6)
```

9.2 Linear Stability Analysis

Linear stability analysis is very useful for two reasons: classification of types of equilibria and the calculation of resonant periodicities (i.e., recurrence intervals) in the case of stable or unstable *foci*.

This chapter uses the following R-packages: `nleqslv`, `rootSolve`, and `polyspline`.

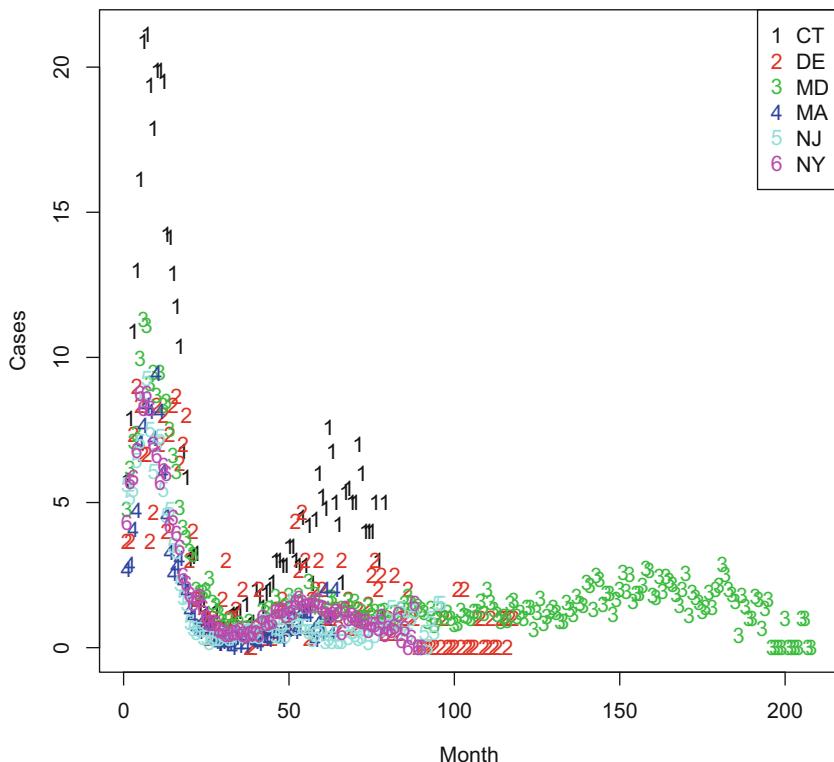


Fig. 9.1 Incidence of raccoon rabies by state since first appearance in Virginia/West Virginia in 1977 (Childs et al. 2000)

If we work with continuous-time models (ODEs like the SEIR), equilibria are stable if all the eigenvalues of the [Jacobian matrix](#)—when evaluated at the equilibrium—are smaller than 0. An equilibrium is (1) a node (i.e., all trajectories moves monotonically towards/away from the equilibrium) if the largest eigenvalue has only real parts,¹ and (2) a focus (trajectories spiral towards or away from the equilibrium) if the largest eigenvalues are a conjugate pair of complex numbers ($a \pm bi$). The resonant period of a focus is $2\pi/b$.

If we work with discrete-time models like the TSIR discussed in Chap. 7 or the Nicholson-Bailey parasitoid-host model (see Chap. 14), equilibria are stable if the absolute value of all the eigenvalues of the Jacobian —when evaluated at the equilibrium—are smaller than 1. Conditions for nodes versus foci are as for continuous-time models, but the resonant period for difference equations is $2\pi/\arctan(b/a)$.

¹ And a “center” which produces amplitude-neutral oscillations like that seen in the [Lotka-Volterra predator-prey model](#) if it has only imaginary parts.

9.3 Finding Equilibria

To carry out such calculations we need to (A) calculate the Jacobian and (B) gather the values corresponding to the equilibrium of interest. An equilibrium is where the state variables do not change. So in the case of ODEs we may consider three strategies of decreasing desirability: (i) solve analytically for when all gradient-functions are zero, (ii) solve numerically for when all gradient-functions are zero, or (iii) simulate the ODEs a long time and record the state of the system at the end. The latter is the worst because it will not find any unstable solutions, and (ii) is less good than (i) because it is less exact, but (i) may be difficult if you are a biologist working with complex models.

Let's consider the SIR model (Eq. (2.1)–(2.3)). The equilibria occurs when dS/dt , dI/dt , and dR/dt all equal zero.

Strategy 1: The disease-free equilibrium is $S^* = 1, I^* = 0, R^* = 0$. If we set $N=1$, it is easy to solve for the endemic equilibrium; The I -equation of the SIR implies that $S^* = (\gamma + \mu)/\beta = 1/R_0$ is the S steady-state, which when substituted into the S -equation gives $I^* = \mu(R_0 - 1)/\beta$, and finally, $R^* = N - I^* - S^*$. Thus for a given set of parameters we get:

```
parms   = c(mu = 1/(50*52), N = 1, beta = 2.5,
           gamma = 1/2)
N=parms ["N"]
gamma=parms ["gamma"]
beta=parms ["beta"]
mu=parms ["mu"]
Istar=as.numeric(mu*(beta/(gamma+mu)-1)/beta)
Sstar=as.numeric((gamma+mu)/beta)
Sstar
## [1] 0.2001538
Istar
## [1] 0.0006147934
```

Strategy 2: If we cannot do the math we can try to solve numerically. The nleqslv-package solves coupled equations numerically. Note, now, that the state variables, x are the unknown quantities we want to solve for so we define a function with a set of equations that we want to equal zero:

```
require(nleqslv)
rootfn=function(x, params){
  r=with(as.list(params),
         c(mu * (N - x[1]) - beta * x[1]*x[2] / N,
            beta * x[1] * x[2] / N - (mu + gamma) * x[2],
            gamma *x[2] - mu*x[3]))}
```

```

    r
}
parms   = c(mu = 1/(50*52), N = 1, beta = 2.5,
      gamma = 1/2)
ans = nleqslv(c(0.1, 0.5, 0.4), fn = rootfn,
      params=parms)
ans$x

## [1] 0.2001523463 0.0006147945 0.7992328592

```

The numerical solution is accurate to the fifth decimal place for the endemic equilibria. Let us see if we can find the disease-free equilibrium $\{S^* = 1, I^* = 0, R^* = 0\}$ numerically if we explore across a range of initial guesses.

```

ans=grid=expand.grid(seq(0,1, by=.25), seq(0,1, by=.25),
      seq(0,1, by=.25))
ans[,]=NA
for(i in 1:nrow(ans)){
  ans[i,]=nleqslv(as.numeric(grid[i,]), fn = rootfn,
    params=parms)$x
}
ans2=round(ans, 4)
ans2[!duplicated(ans2),]

##      Var1  Var2  Var3
## 1 1.0000 0e+00 0.0000
## 6 0.2002 6e-04 0.7992

```

The two equilibria show up.

Strategy 3: This should not be used for final analyses because numerically integrating differential equations are much more fraught than numerically solving non-linear equations. However as a shortcut if we already have the gradient-function defined we can use the `rootSolve`-package.

```

sirmod=function(t, y, parameters){
  S=y[1]
  I=y[2]
  R=y[3]
  beta=parameters["beta"]
  mu=parameters["mu"]
  gamma=parameters["gamma"]
  N=parameters["N"]
  dS = mu * (N - S) - beta * S * I / N
  dI = beta * S * I / N - (mu + gamma) * I
  dR = gamma*I - mu*R

```

```

res=c(dS, dI, dR)
list(res)
}

require(rootSolve)
parms  = c(mu = 1/(50*52), N = 1,
          beta = 2.5, gamma = 1/2)
equil = runsteady(y = c(S = 1 - 1E-4, I = 1E-4, R = 0),
                    times = c(0, 1E05), func = sirmod, parms = parms)
round(equil$y, 4)

##      S      I      R
## 0.2002 0.0006 0.7992

```

9.4 Evaluating the Jacobian

The R-compartment of the SIR model does not affect dynamics. So for analysis we only need to consider the coupled S-I system.

```

dS = expression(mu * (1 - S) - beta * S * I/1)
dI = expression(beta * S * I/1 - (mu + gamma) * I)
j11 = D(dS, "S")
j12 = D(dS, "I")
j21 = D(dI, "S")
j22 = D(dI, "I")

```

We define a list of parameters and steady-states for the endemic equilibrium, and then piece together the matrix by evaluating the expressions in the list.²

```

vals = list(mu = 1/(50*52), N = 1, beta = 2.5,
            gamma = 1/2, S=Sstar, I=Istar)
J=with(vals, matrix(c(eval(j11), eval(j12), eval(j21),
                      eval(j22)), ncol=2, byrow=T))
eigen(J, only.values=TRUE)$values

## [1] -0.0009608+0.02771569i -0.0009608-0.02771569i

```

The leading eigenvalues are a pair of complex conjugates with negative real parts; The equilibrium is a stable focus. The resonant frequency (in weeks) is:

² It is almost imperative to use the `with`-function here, because the working directory will otherwise get littered with defined objects that mask names corresponding to parameters and state-variables.

```
2 * pi/Im(eigen(J)$values[1])
## [1] 226.7014
```

which is just over 4 years.

Next we look at the disease-free equilibrium:

```
vals = list(mu = 1/(50*52), N = 1, beta = 2.5,
            gamma = 1/2, S=1, I=0)
J=with(vals,
       matrix(c(eval(j11), eval(j12), eval(j21),
                 eval(j22)), ncol=2, byrow=T))
eigen(J, only.values=TRUE)$values
## [1] 1.9996153846 -0.0003846154
```

The leading EV is real-only and > 0 : The disease-free equilibrium is an unstable node (because $R_0 > 1$). What if we decrease the transmission rate to 0.3?

```
vals = list(mu = 1/(50*52), N = 1, beta = 0.3,
            gamma = 1/2, S=1, I=0)
J=with(vals,
       matrix(c(eval(j11), eval(j12),
                 eval(j21), eval(j22)), ncol=2, byrow=T))
eigen(J, only.values=TRUE)$values
## [1] -0.2003846154 -0.0003846154
```

The leading EV is real-only and < 0 : The equilibrium is a stable node (because $R_0 < 1$).

9.5 Raccoon Rabies

Coyne et al. (1989) developed a compartmental model for rabies in raccoons. Using their notation the flow is from susceptible (X), infected but not-yet infectious hosts that eventually becomes rabid (H_1), infected hosts that recover with immunity (H_2), rabid raccoons (Y), immune raccoons (I), and vaccinated raccoons (V). The total number of raccoons (N) are the sum of these. The model is:

$$\frac{dX}{dt} = a(X + I + V) - \beta XY - \gamma NX - (b + c + v)X \quad (9.1)$$

$$\frac{dH_1}{dt} = \rho\beta XY - \gamma NH_1 - (b + \sigma + c)H_1 \quad (9.2)$$

$$\frac{dH_2}{dt} = (1 - \rho)\beta XY - \gamma NH_2 - (b + \sigma + c)H_2 \quad (9.3)$$

$$\frac{dY}{dt} = \sigma H_1 - \gamma NY - (b + \alpha + c)Y \quad (9.4)$$

$$\frac{dI}{dt} = \sigma H_2 - \gamma NI - (b + c)I \quad (9.5)$$

$$\frac{dV}{dt} = vX - \gamma NV - (b + c)V \quad (9.6)$$

$$N = X + H_1 + H_2 + Y + I + V \quad (9.7)$$

The parameters are defined in Table 9.1.

Table 9.1 Parameters and values for Coyne et al. (1989) rabies model

a	Birth rate	1.34/year
b	Death rate	0.836/year
r	Intrinsic rate of increase (=a-b)	0.504
K	Carrying capacity	12.69/km ²
γ	Index of density dependence (=r/K)	0.0397 km ² /year
(1- ρ)	Probability of recovery	0.20
σ	Rate of transition from latents	7.5/year
α	Disease induced mortality	66.36/year
β	Transmission rate	33.25/year
v	Vaccination rate	Variable
c	Culling rate	Variable

We will look at the slightly simplified system without vaccination (so without the V-class and v-parameter). We use `dSolve` to integrate the model using the “log-trick” introduced in Sect. 2.7—we solve the system in log-coordinates (Ellner and Guckenheimer 2011); Note, again how initial values are log-transformed in `start`, the first line in the function is `x = exp(logx)` and the last line returns `dX/X` etc. in place of `dX` which comes from the [chain-rule](#) of differentiation and the fact that $D(\log x) = 1/x$.

```
coyne=function(t, logx, parms) {
  x=exp(logx)
  X=x[1]
  H1=x[2]
  H2=x[3]
  Y=x[4]
  I=x[5]
  N = sum(x)
  with(as.list(parms), {
    dX = a * (X + I) - beta * X * Y -
      gamma * N * X - (b + c) * X
    dH1 = rho * beta * X * Y - gamma * N * H1 -
      (b + sigma + c) * H1
  })
}
```

```

dH2= (1-rho) * beta * X * Y - gamma * N * H2 -
      (b + sigma + c) * H2
dY = sigma * H1 - gamma * N * Y -
      (b + alpha + c) * Y
dI = sigma * H2 - gamma * N * I - (b + c) * I
res=c(dX/X, dH1/H1, dH2/H2, dY/Y, dI/I)
list(res)
})
}
}

```

We integrate the system:

```

times = seq(0, 50, by=1/520)
paras = c(gamma = 0.0397, b = 0.836,
          a = 1.34, sigma = 7.5,
          alpha = 66.36, beta = 33.25,
          c = 0, rho = 0.8)
start = log(c(X=12.69/2, H1=0.1, H2=0.1,
              Y = 0.1, I = 0.1))
out = as.data.frame(ode(start, times, coyne, paras))

```

and plot as time series and in the Susceptible-Infectious phase-plane anti-log'ing the state-variables to convert them to abundances from log-abundances (Fig. 9.2). The model predicts transient cycles towards the endemic equilibrium.

```

par(mfrow=c(1,2))
plot(times, exp(out$Y), ylab="Infected", xlab="Time",
      type="l")
plot(exp(out$X), exp(out$Y), ylab="Infected",
      xlab="Susceptible", type="l")

```

Childs et al. (2000) used this model to study how the predicted inter-epidemic period compare to the data (Fig. 9.1). We can repeat this analysis by calculating the resonant frequency of the system. To do this we need the 5×5 Jacobian matrix and values for the endemic equilibrium. We first differentiate the equations:

```

dX = expression(a * (X + I) - beta * X * Y -
                gamma * (X+H1+H2+Y+I) * X - (b + c) * X)
dH1= expression(rho * beta * X * Y -
                gamma * (X+H1+H2+Y+I) * H1 - (b + sigma + c) * H1)
dH2= expression((1-rho) * beta * X * Y -
                gamma * (X+H1+H2+Y+I) * H2 - (b + sigma + c) * H2)

```

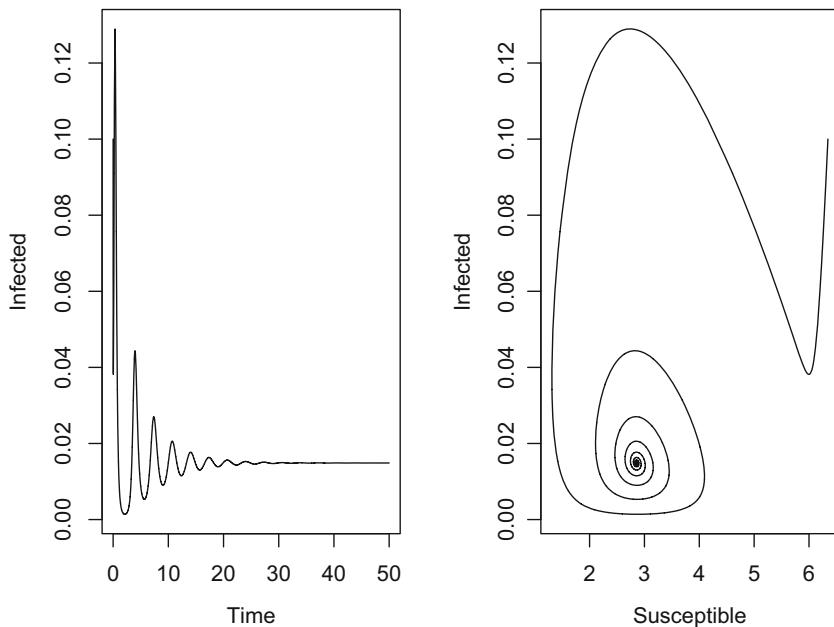


Fig. 9.2 Rabies dynamics predicted by the model of Coyne et al.'s (1989) using the parameters defined in Table 9.1

```

dY = expression(sigma * H1 - gamma * (X+H1+
H2+Y+I) * Y - (b + alpha + c) * Y)
dI = expression(sigma * H2 - gamma * (X+H1+
H2+Y+I) * I - (b + c) * I)

j11 = D(dx, "X"); j12 = D(dx, "H1"); j13 = D(dx,
"H2"); j14 = D(dx, "Y"); j15 = D(dx, "I")
j21 = D(dH1, "X"); j22 = D(dH1, "H1"); j23 = D(dH1,
"H2"); j24 = D(dH1, "Y"); j25 = D(dH1, "I")
j31 = D(dH2, "X"); j32 = D(dH2, "H1"); j33 = D(dH2,
"H2"); j34 = D(dH2, "Y"); j35 = D(dH2, "I")
j41 = D(dy, "X"); j42 = D(dy, "H1"); j43 = D(dy,
"H2"); j44 = D(dy, "Y"); j45 = D(dy, "I")
j51 = D(dI, "X"); j52 = D(dI, "H1"); j53 = D(dI,
"H2"); j54 = D(dI, "Y"); j55 = D(dI, "I")

```

Because we have already defined the `coyne-gradient` function, we use the `lazy` option (strategy 3) to find the endemic equilibrium numerically with the `runsteady`-function of the `rootSolve`-package:

```

require(rootSolve)
paras = c(gamma = 0.0397, b = 0.836, a = 1.34,
    sigma = 7.5, alpha = 66.36, beta = 33.25,
    c = 0, rho = 1)
equil=runsteady(y=log(c(X=12.69/2, H1=0.1, H2=0.1, Y =
    0.1, I = 0.1)), times=c(0,1E5),
    func=coyne, parms=paras)

```

We evaluate the Jacobian at the endemic equilibrium and calculate the dominant eigenvalue:

```

#Evaluate Jacobian elements
JJ = with(as.list(c(exp(equil$y), paras)),
c(eval(j11), eval(j12), eval(j13), eval(j14), eval(j15),
    eval(j21), eval(j22), eval(j23), eval(j24), eval(j25),
    eval(j31), eval(j32), eval(j33), eval(j34), eval(j35),
    eval(j41), eval(j42), eval(j43), eval(j44), eval(j45),
    eval(j51), eval(j52), eval(j53), eval(j54), eval(j55)))
#Populate the Jacobian matrix
J= matrix(JJ, nrow=5, byrow=T)
#Eigen decomposition
which.max(Re(eigen(J, only.values=TRUE) $values))

## [1] 3

eigen(J, only.values=TRUE) $values[3]

## [1] -0.644556+1.743531i

```

The dominant eigenvalues are the conjugate pair with a real part of -0.64 . The endemic equilibrium is a stable focus with a resonant period of

```

2 * pi/Im(eigen(J) $values[3])
## [1] 3.603714

```

Thus the model predicts recurrent outbreaks with a mean period of 3.6 years during the rabies invasion. This is 6 months shorter than observed in the data. Childs et al. (2000) varied the ρ parameter (the fraction of exposed raccoons escaping infection with immunity) to find a model that more closely matches the data (around 48 months).

9.6 Influenza

The mystery of the annual epidemics of influenza that peak in the Northern hemisphere in the Northern winter and in the Southern hemisphere in the Southern winter

(Hope-Simpson 1981) has been discussed as an exemplar par excellence in resonant periodicities in epidemiology (Dushoff et al. 2004; Bjørnstad and Viboud 2016). Seasonal influenza epidemics are caused by subtype B, A/H3N2, and A/H1N1 in various mixtures in any given year. At the aggregate level, the “flu” can be modeled as a susceptible-infected-recovered-(re)susceptible (SIRS) system (Axelsen et al. 2014) with transient immune protection upon recovery lasting around 4 years due to “epochal” evolution (Koelle et al. 2006). The SIRS model is:

$$\frac{dS}{dt} = \mu(N - S) - \frac{\beta IS}{N} + \omega R \quad (9.8)$$

$$\frac{dI}{dt} = \frac{\beta IS}{N} - (\mu + \gamma)I \quad (9.9)$$

$$\frac{dR}{dt} = \gamma I - (\mu + \omega)R, \quad (9.10)$$

where ω is the rate of loss of immunity ($\sim 0.25 \text{ year}^{-1}$). For the infectious period ($1/\gamma$) we use 3.8 days (Carrat et al. 2008) and assume an R_0 of 2.9 (Axelsen et al. 2014). We model the population fractions ($N = 1$), thus SIRS flu-appropriate parameters (per week) are:

```
N = 1
gamma = 7/3.8
omega = 1/(52 * 4)
mu = 1/(52 * 70)
R0 = 2.9
```

We back-calculate β to get the right R_0 and gather parameters:

```
#R0=beta/(gamma+mu)
beta=R0*(gamma+mu)
paras=c(beta=beta, gamma=gamma, mu=mu, omega=omega)
```

For the SIRS model there is an approximate equation for the dampening period (Keeling and Rohani 2008): $T = 4\pi/\sqrt{(4(R_0 - 1)/(G_I G_R) - ((1/G_R) - (1/A))^2)}$, where A is the mean age of infection ($= \frac{\rho + \mu + \gamma}{(\rho + \mu)(\beta - \gamma - \mu)}$), G_I is the infectious period ($= 1/(\gamma + \mu)$), and G_R is the average duration of immunity ($= 1/(\rho + \mu)$). The approximate equation predicts the dampening period for influenza to be just under a year:

```
A=(omega+mu+gamma)/((omega+mu)*(beta-gamma-mu))
GI=1/(gamma+mu)
GR = 1/(omega+mu)
T=4*pi/sqrt(4*(R0-1)/(GI*GR)-((1/GR)-(1/A))^2)
T
## [1] 47.11307
```

We can check the accuracy of the approximation against the resonant frequency of the linearized system. The endemic equilibrium of the SIRS model is $S^* = 1/R_0$, $I^* = \frac{\mu(1-1/R_0)}{\gamma+\mu-\frac{\omega\gamma}{\omega+\mu}}$, and $R^* = \gamma I^*/(\omega + \mu)$:

```
Sstar=1/R0
Istar=mu*(1-1/R0)/(gamma+mu-(omega*gamma)/(omega+mu))
Rstar=gamma*Istar/(omega+mu)
eq=list(S=Sstar, I=Istar, R=Rstar)
eq

## $S
## [1] 0.3448276
##
## $I
## [1] 0.001802665
##
## $R
## [1] 0.6533697
```

The Jacobian matrix evaluated at the endemic equilibrium is:

```
F = expression(mu * (1-S) - beta * S * I / N +
               omega * R)
G = expression(beta * S * I / N - (mu + gamma) * I)
H = expression(gamma*I - (mu + omega)*R)
j11 = D(F, "S"); j12 = D(F, "I"); j13 = D(F, "R")
j21 = D(G, "S"); j22 = D(G, "I"); j23 = D(G, "R")
j31 = D(H, "S"); j32 = D(H, "I"); j33 = D(H, "R")

J=with(eq, matrix(c(eval(j11),eval(j12),eval(j13),
                   eval(j21), eval(j22), eval(j23), eval(j31),
                   eval(j32), eval(j33)), nrow=3, byrow=T))
```

Finally, the eigenvalues and resonant frequency from the imaginary part of the dominant conjugate-pair are:

```
round(eigen(J)$values, 4)

## [1] -0.0074+0.1332i -0.0074-0.1332i -0.0003+0.0000i
2 * pi/Im(eigen(J)$values)[1]

## [1] 47.17804
```

which is in close agreement with the approximate equation.

9.7 Advanced: Transfer Functions

9.7.1 SIR

We can predict the *entire* power spectrum of a linearized stochastic system using transfer functions (Priestley 1981; Nisbet and Gurney 1982; Bjørnstad et al. 2004). In matrix form, the transfer function for a coupled continuous-time system is

$$\mathbf{T}(\omega) = (\mathbf{I}\omega - \mathbf{J})^{-1}\mathbf{A} \quad (9.11)$$

where \mathbf{I} is the identity matrix, ω is the angular frequency (between 0 and π), \mathbf{J} is the Jacobian (evaluated at the equilibrium), \mathbf{A} is the matrix of gradients differentiated with respect to the stochastic term(s), and $^{-1}$ denotes the matrix inverse.³

If the stochasticity is uncorrelated “white” noise, the power spectrum is predicted by the modulus of the transfer function. Let us consider the SIR model from Sect. 2.6 and assume that variability enters through stochasticity in β . We first need the equilibrium values for the linearization:

```
parms  = c(mu = 1/(50*52), N = 1,
      beta = 2.5, gamma = 1/2)
Istar=parms["mu"]*(parms["beta"]/(parms["gamma"] +
      parms["mu"])-1)/parms["beta"]
Sstar=(parms["gamma"]+parms["mu"])/parms["beta"]
```

Next we need to calculate the derivatives required for the Jacobian matrix:

```
dS = expression(mu * (N - S) - beta * S * I/N)
dI = expression(beta * S * I/N - (mu + gamma) * I)
j11 = D(dS, "S")
j12 = D(dS, "I")
j21 = D(dI, "S")
j22 = D(dI, "I")
```

The linearized system is governed by the Jacobian evaluated at the equilibrium:

```
vals = list(mu = parms["mu"], N = parms["N"], beta =
      parms["beta"], gamma = parms["gamma"],
      S=Sstar, I=Istar)
```

³ This equation stems from the result that if $\tilde{x}(\omega)$ denotes the Fourier transform of the vector of state variables, the transform of $\frac{dx}{dt}$ will be $\tilde{x}(\omega)\omega$; Rearranging in matrix form yields $\tilde{x}(\omega) = \mathbf{T}(\omega)\mathbf{d}\tilde{\beta}(\omega)$. For discrete time systems the transfer function is $\mathbf{T}(\omega) = (\mathbf{I} - e^{-i\omega}\mathbf{J})^{-1}\mathbf{A}$ because the Fourier transform of X_{t-1} is $\tilde{X}(\omega)e^{-i\omega}$ (see Sect. 9.7.2).

```
J=with(vals,
  matrix(c(eval(j11), eval(j12),
  eval(j21), eval(j22)), ncol=2, byrow=T))
```

The A-matrix is the linearization with respect to the stochastic term. Assuming stochasticity in β we calculate the last pieces needed for the transfer function according to:

```
a1 = D(dS, "beta")
a2 = D(dI, "beta")
A = with(vals, matrix(c(eval(a1), eval(a2)), ncol = 1))
Id = matrix(c(1, 0, 0, 1), ncol = 2)
```

Finally, we can evaluate the transfer function (across 500 frequencies between 0 and π):

```
wseq = seq(0, pi, length=500)
Fr = vector("list", 500) #set up empty list of matrices
#Loop to fill matrices for each frequency
for(i in 1:500){
  #Solve gives matrix inverse
  Fr[[i]] = matrix(solve(Id*1i*wseq[i]-J) %*% A, ncol=1)
}
```

and calculate the theoretical power spectrum from the modulus of the transfer function:

```
PS = matrix(NA, ncol = 2, nrow = 500,
  dimnames=list(1:500, c("S", "I")))
#Power spectra from real and imaginary
# parts of the Fourier transform
for(i in 1:500){
  PS[i,] = sqrt(Re(Fr[[i]])^2+Im(Fr[[i]])^2)
}
plot(wseq, PS[,2], type="l", log="x",
  xlab="Frequency (in radians)", ylab="Amplitude")
#Calculate the dominant period in weeks
2*pi/wseq[which.max(PS[,2])]

## [1] 249.5
```

So the stochastic system with variability in β is predicted to oscillate with a period of 248 weeks (just shy of 5 years) (Fig. 9.3). Thus the stochastically excited cycles have a period that is comparable but slightly longer than the resonant frequency of the deterministic system (Sect. 2.6).

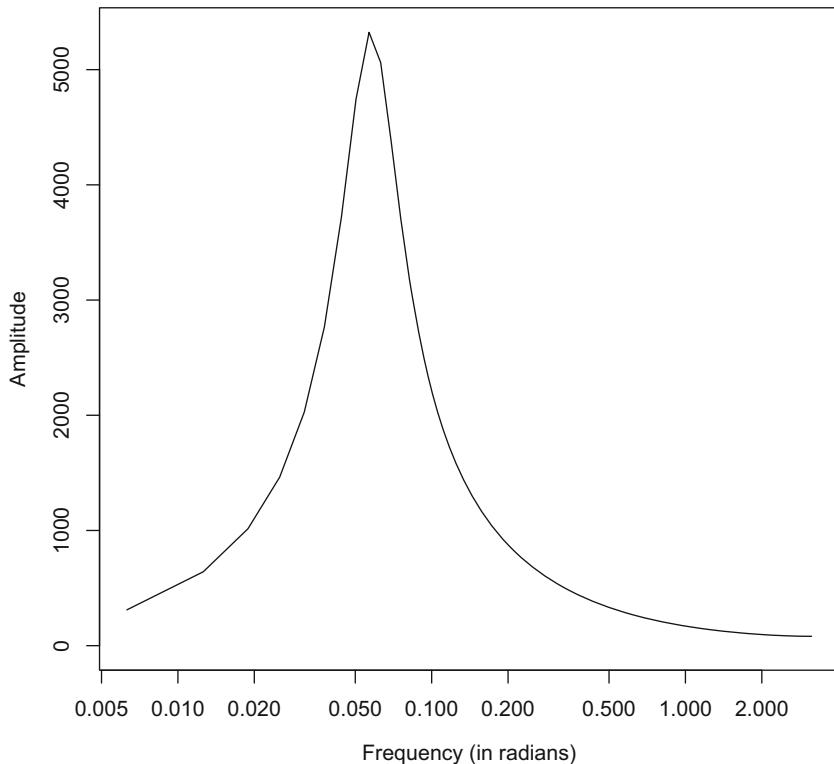


Fig. 9.3 The full power spectrum of the SEIR model with stochasticity in β as predicted from the transfer functions of the linearized system

9.7.2 The TSIR

Section 7.1 discussed how to do stochastic simulation using the TSIR model with a stochastic transmission term. We can apply the transfer function theory to this model. The TSIR model is a discrete time model, so the transfer function is $T(\omega) = (I - e^{-i\omega}J)^{-1}\mathbf{A}$. We first calculate the relevant derivatives (assuming randomness in β):

```

Seq = expression(S - beta * S * I^alpha/N + B)
Ieq = expression(beta * S * I^alpha/N)
j11 = D(Seq, "S")
j12 = D(Seq, "I")
j21 = D(Ieq, "S")
j22 = D(Ieq, "I")
jj = c(j11, j12, j21, j22)
a1 = D(Seq, "beta")
a2 = D(Ieq, "beta")

```

```
aa = c(a1, a2)
```

The equilibrium for the TSIR model is $S^* = \frac{B^{1-\alpha}N}{\beta}$ and $I^* = B$. Let us consider a city of a million people and a pathogen with an R_0 ($= \beta$) of 5 and use the compact-form equation evaluation introduced in Sect. 8.2:

```
paras = c(B = 800, beta = 5, alpha = 0.97, N=1E6)
eqs=sapply(c(quote(B^(1-alpha)*N/beta), quote(B)),
            eval, as.list(paras))
```

The Jacobian is:

```
J=matrix(sapply(jj, eval, as.list(c(paras, c(S=eqs[1],
                                         I=eqs[2])))), 2, byrow=TRUE)
evs=eigen(J)$values
2*pi/atan2(Im(evs[1]), Re(evs[1]))

## [1] 112.9908
```

Predicting a resonant period of 113 disease generations. We next piece together the transfer function:

```
A=matrix(sapply(aa, eval, as.list(c(paras,
                                     c(S=eqs[1], I=eqs[2])))), 2, byrow=TRUE)
Id=matrix(c(1,0,0,1),ncol=2)
wseq=seq(0,pi,length=500)
Fr=vector("list",500) #Set up empty list of matrices
#Loop to fill those matrices with fourier transforms
for(i in 1:500){
  #Solve gives matrix inverse
  Fr[[i]]=matrix(solve(Id-exp(1i*wseq[i])*J) %*%
                 A, ncol=1)
}
PS=matrix(NA, ncol=2, nrow=500, dimnames=list(1:500,
                                              c("S", "I")))
#Power spectra from real and imaginary parts
for(i in 1:500){
  PS[i,]=sqrt(Re(Fr[[i]])^2+Im(Fr[[i]])^2)
}
```

To compare the predicted power spectrum from the transfer functions against the simulation, we assume a standard deviation in β of one and simulate for 100 years (assuming the generation time is a week). The first 20 years of the simulation is shown in Fig. 9.4. The figure suggests a transient period of a couple of years before the dynamics settles down to stochastically excited recurrent epidemics.

```
out = SimTsir(B=800, beta=5, sdbeta=1, N=1E6,
               IT=100*52, I0=10, S0=0.3)
plot(out$I[1:1040], xlab="Biweek", ylab="Incidence",
      type="l")
```

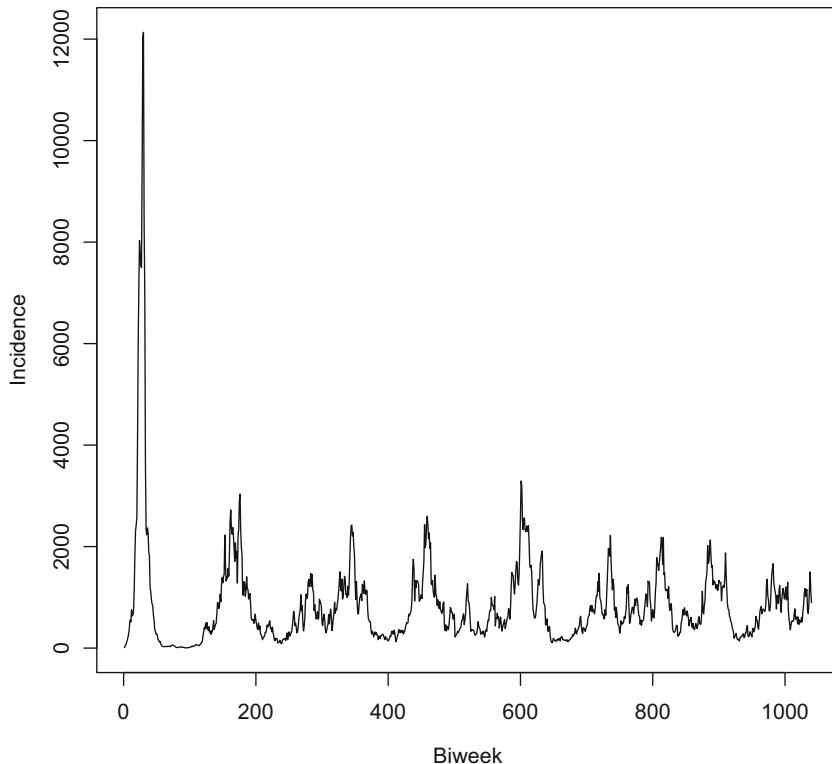


Fig. 9.4 Twenty years of simulated incidence from the TSIR model with stochasticity in transmission

We estimate the spectrum from the simulation using the periodogram (discarding the first 2 years of data).

```
sfit = spectrum(out$I[-c(1:104)])
```

As discussed in Sect. 6.3, the Schuster periodogram is the classic method to estimate the spectral density of a time series but with the drawback that it is not a “consistent method.” In statistics, a consistent method is one where the estimate converges on the truth as the sample size increases. For the Schuster periodogram the spectral density is estimated at $T/2$ frequencies, thus doubling the length of the time series, doubles the number of parameters, which defies consistency. Various

window-smoothing approaches have been proposed to ameliorate this; (see for example Priestley 1981). An alternative to smoothing is to estimate the spectral density using Kooperberg et al.'s (1995) log-spline method using the `lspec`-function of the `polyspline` package. Figure 9.5 shows the empirical estimates and the transfer-function predicted spectra for the TSIR model. Theory based on the linearized system provides an excellent approximation to the empirical estimates (Fig. 9.5).

```
require(polyspline)
sfit2=lspec(out$I[-c(1:104)])
plot(wseq, PS[,2], type="l", ylab="Amplitude",
      xlab="Frequency (in radians)", xlim=c(0,0.6))
lines(pi*sfit$freq/0.5,
      5000*sfit$spec/max(sfit$spec), col=2)
par(new=TRUE)
plot(sfit2, col=3, xlim=c(0,0.6), axes=FALSE)
legend("topright", c("Transfer fn", "Periodogram",
  "Log-spline"), lty=c(1,1,1), col=c(1,2,3))
```

9.8 (Even More) Advanced: Transfer Functions and ARMA Delay-Coordinates

The Hamilton-Caley theorem states that any stochastic, autonomous (i.e., unforced), linear, discrete-time multistate system can be rewritten as an ARMA (Sect. 6.2.1) delay-coordinate system. The significance of this jargon is that any linear(ized) system of the form

$$\mathbf{x}_t = \mathbf{J}\mathbf{x}_{t-1} + \mathbf{A}\boldsymbol{\varepsilon}_t \quad (9.12)$$

can be rewritten in a delay-form

$$x_t = c + \underbrace{\beta_1 x_{t-1} + \beta_2 x_{t-2} + \dots + \beta_d x_{t-d}}_{\text{autoregressivs}} + \boldsymbol{\varepsilon}_t + \underbrace{\alpha_1 \boldsymbol{\varepsilon}_{t-1} + \dots + \alpha_q \boldsymbol{\varepsilon}_{t-q}}_{\text{moving average}}, \quad (9.13)$$

This is useful for understanding how dynamics can induce delayed feedbacks and how the “echo” of stochastic perturbations propagate through time. It also provides an alternative route for analyzing systems for which only a subset of variables (e.g., one) has been measured. The theorem says it can always be done, but in practice it can be tedious because of the equivalence between finite AR-processes and infinite MA-processes and *vice versa*. It turns out that transfer functions (Sect. 9.7) are very helpful for such calculations (Priestley 1981; Bjørnstad et al. 2004). In its general analytic form, the discrete-time transfer function takes the form:

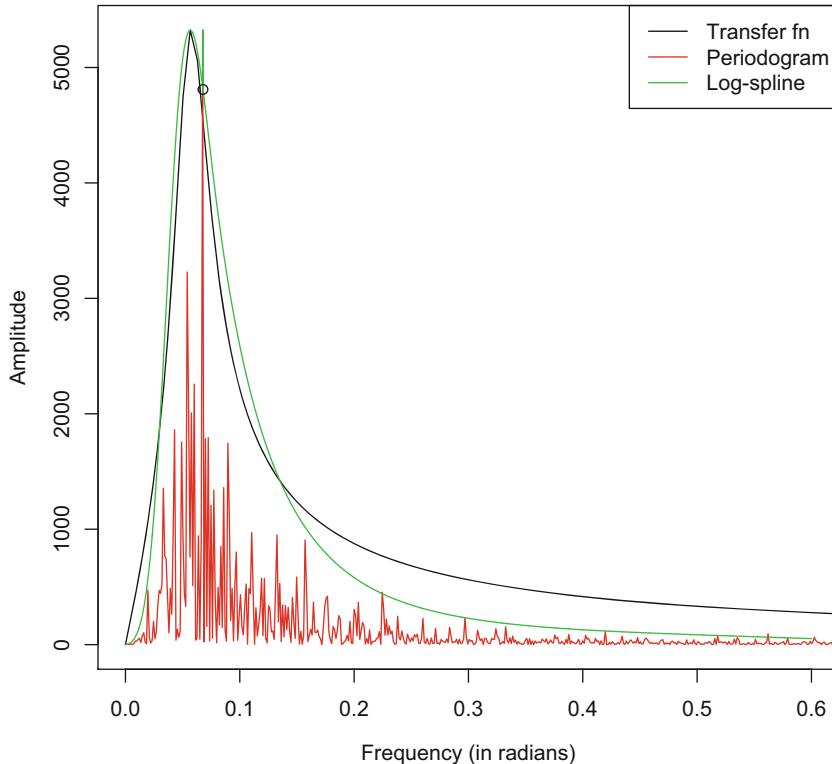


Fig. 9.5 The spectrum of the TSIR model with stochasticity in β as predicted from the transfer functions of the linearized system with the periodogram and log-spline density estimate from a simulation (Fig. 9.4) superimposed

$$T(\omega) \propto \frac{1 + \alpha_1 e^{-i\omega} + \dots + \alpha_q e^{-qi\omega}}{1 - \beta_1 e^{-i\omega} - \dots - \beta_p e^{-pi\omega}}, \quad (9.14)$$

where the α 's and β 's correspond exactly to the coefficients of the ARMA(p, q) delay-coordinate representation of the system. R does unfortunately not (yet?) have enough symbolic power to solve the transfer function analytically. However, a program like mathematica will show that the transfer functions for the I compartment of the linearized stochastic TSIR is:

$$T_I(\omega) = \frac{B}{1 + \beta} \frac{1 + (B^\alpha(\beta - 1) - 1)e^{-i\omega}}{1 - (1 + \alpha - \beta B^\alpha \beta)e^{-i\omega} + \alpha e^{-2i\omega}}. \quad (9.15)$$

Theoretically, therefore, the model predicts an ARMA(2,1) delay-coordinate structure to the time series of incidence according to:

$$I_{t+1} = \text{const} + (1 + \alpha - \beta B^\alpha \beta)I_t - \alpha I_{t-1} + \varepsilon_t + (B^\alpha(\beta - 1) - 1)\varepsilon_{t-1}. \quad (9.16)$$

Bjørnstad et al. (2001) used the state-space/delay-coordinate equivalence to study how natural enemies—viruses and parasitoids—structurally alter the pattern of delayed density-dependent feedbacks in population dynamics. Bjørnstad et al. (2004) provide some additional worked examples of using transfer functions to study population dynamics.

9.9 ShinyApp

The `TSIR.app` from Chap. 7 also calculates the transfer function-predicted power spectrum of the TSIR model and compares it with simulated data.

Chapter 10

Exotica



10.1 Introduction

Chapter 9 discussed how a linear approximation to the perennially nonlinear dynamics of infectious disease can provide important insights on invasion, stability, and resonant periodicity. As remarked by Nisbet and Gurney (1982) more generally, linear approximation can often provide remarkably useful insights for nonlinear ecological systems as long as they are not *too* nonlinear.

From a dynamical system's point of view, dynamics are (approximately) linear if the system does not "miss-behave" as it approaches (diverges) from its stable (unstable) fix-points. Thus, while the simple SIR model is mathematically speaking nonlinear (because of the $\beta SI/N$ term), its dynamics can be thought of as being "linear" because of its smooth inwards spiraling towards the endemic equilibrium (the stable focus) and logistic divergence from the disease-free equilibrium (the unstable node) when $R_0 > 1$. However, highly infectious immunizing diseases have the potential for exhibiting dynamics so nonlinear that crazy things can happen; Things that require a different set of tools. The chaotic fluctuations seen in the seasonally forced SEIR model and measles in pre-vaccination US (Sect. 5.4) are one such highly nonlinear phenomenon. There are however other dynamic "exotica" that can arise when stochasticity and nonlinearity interacts, or when there are great perturbations (such as introduction of mass vaccination) to the nonlinear epidemic clockworks. The following sections explore this and discuss some useful tools.

This chapter uses the following R-packages: `deSolve`, `pomp`, and `nlts`.

10.2 Chaos

In nonlinear systems, a perturbation will either dampen (“dissipate”) or expand as it interacts with the dynamic clockwork. The hallmark of a chaotic attractor is “sensitive dependence on initial conditions”¹: Two very nearby trajectories will diverge exponentially over time. The classic way to quantify this is through the dominant Lyapunov exponent:

$$\lambda = \lim_{T \rightarrow \infty} \frac{1}{T} \log \left(\prod_t^T J_t U_0 \right), \quad (10.1)$$

where J_t is the Jacobian evaluated on the point of the attractor at time t and for a 2D system like the TSIR U_0 is the unit vector $\{1, 0\}$. A chaotic attractor has $\lambda > 0$. The Jacobian of the TSIR model is:

$$J_t = \begin{bmatrix} 1 - \beta_s I_t^\alpha / N & -\beta_s S_t (I_t^{\alpha-1} \alpha) / N \\ \beta_s I_t^\alpha / N & \beta_s S_t (I_t^{\alpha-1} \alpha) / N \end{bmatrix}. \quad (10.2)$$

We can estimate the Lyapunov exponent numerically by simulating the TSIR a long time (Grenfell et al. 2002). We consider the measles dynamics in New York between 1920 and 1941 (Fig. 10.1) (Dalziel et al. 2016). We first fit the parameters using the protocol discussed in Chap. 7; The profile likelihood on S suggests a mean fraction of susceptibles of 0.051.

```
data(dalziel)
NY=na.omit(dalziel[dalziel$loc=="NEW YORK",])
NY=NY[NY$year %in% c(1920:1940),]
plot(NY$decimalYear, sqrt(NY$cases), type="b",
      xlab="Year", ylab="Sqrt(cases)")
#Susceptible reconstruction and
#correcting for underreporting
cum.reg = smooth.spline(cumsum(NY$rec),
                        cumsum(NY$cases), df=5)
D = - resid(cum.reg) #The residuals
rr = predict(cum.reg, deriv=1)$y
Ic = NY$cases/rr
Dc=D/rr
#Align lagged variables
seas = rep(1:26, 21)[1:545]
lInew = log(Ic[2:546])
lIold = log(Ic[1:545])
Dold = Dc[1:545]
#TSIR fit
```

¹ Interestingly Ruelle (1993) paraphrases Henri Poincaré as defining *chance* as sensitive dependence on *unknown* initial conditions as far back as 1908.

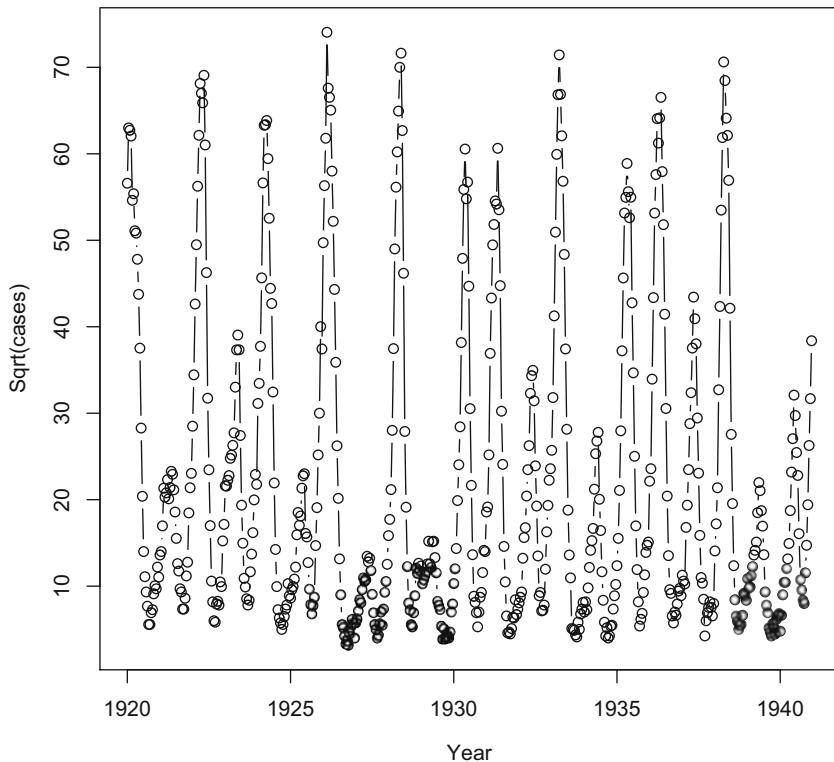


Fig. 10.1 Measles in New York city

```
N=NY$pop
offsetN=-log(N[1:545])
lSold = log(0.051*N[1:545] + Dold)
glmfit = glm(lInew ~ -1 +as.factor(seas) + lIold +
offset(lSold+offsetN))
```

We use the `SimTsir2`-function from Chap. 7 to simulate a 200-year long deterministic trajectory from the fitted model. The result is a highly erratic trajectory in the phase plane (Fig. 10.2).

```
sim2=SimTsir2(beta=exp(glmfit$coef[1:26]), alpha=0.98,
B=rep(median(NY$rec), 5200), N=median(N),
inits=list(Snull=exp(lSold[1]), Inull=Ic[1]),
type="det")
Sattr=sim2$S[2601:5200]
Iattr=sim2$I[2601:5200]
plot(Sattr, Iattr, log="y", type="l",
xlab="S", ylab="I")
```

```
points(sim2$S [seq(2601, 5200, by=26)],
      sim2$I [seq(2601, 5200, by=26)], pch=19, col="red")
legend("bottomright", c("Trajectory", "Strobe"),
       pch=c(NA, 19), lty=c(1, NA), col=c("black", "red"))
```

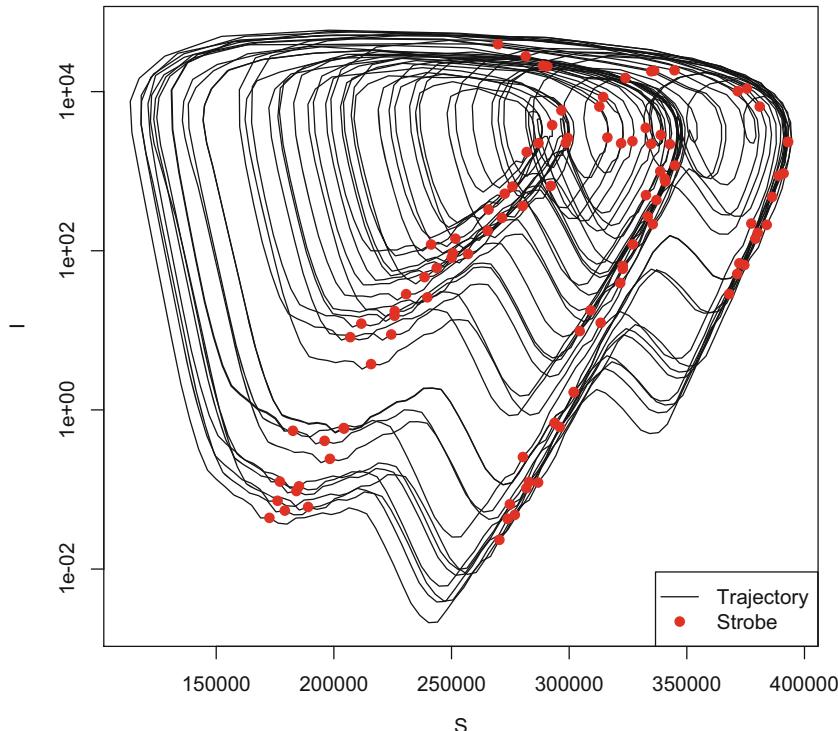


Fig. 10.2 The deterministic trajectory of the New York measles TSIR model. The black line represents the full trajectory. The red dots are the annual stroboscopic section

Calculating the Lyapunov exponent is a bit involved, so we write a function to do it. Because we will be wanting to study the attractor in greater detail, we make the function to both calculate the Lyapunov exponent and store all the Jacobian elements evaluated along the attractor.

```
TSIRlyap=function(I, S, alpha, bt, N){
  IT <- length(I)
  S <- length(bt)
  j11=rep(NA, IT); j12=rep(NA, IT)
  j21=rep(NA, IT); j22=rep(NA, IT)
  #initial unit vector
```

```

J=matrix(c(1, 0), ncol=1)
#loop over the attractor
for(i in 1:IT) {
  j11[i]=1 - bt[((i - 1) %% s) + 1] *
    I[i]^alpha/N
  j12[i]=- (bt[((i - 1) %% s) + 1] * S[i] *
    (I[i]^(alpha - 1) * alpha)/N)
  j21[i]= bt[((i - 1) %% s) + 1] * I[i]^alpha/N
  j22[i]= bt[((i - 1) %% s) + 1] * S[i] *
    (I[i]^(alpha - 1) * alpha)/N
  J<-matrix(c(j11[i],j12[i],j21[i],j22[i]),
             ncol=2, byrow=TRUE)%*%J
}
res=list(lyap=log(norm(J))/IT, j11=j11, j12=j12,
         j21=j21, j22=j22, I=I, S=S, alpha=alpha,
         bt=bt, N=N)
class(res)="lyap"
return(res)
}

```

We apply the function to the last 100 years of the simulated dynamics:

```

nylyap=TSIRlyap(I=Iattr, S=Sattr, alpha=0.98,
                 bt=exp(glmfit$coef[1:26]), N=median(N))
nylyap$lyap
## [1] 0.01223667

```

The exponent is positive indicating that the deterministic skeleton of the TSIR model for measles in New York is a chaotic attractor as concluded by Dalziel et al. (2016). This contrasts with the negative Lyapunov exponent of measles in London testifying to the stability of its biennial limit cycle (see below).

10.3 Local Lyapunov Exponents

Bailey et al. (1997) suggested that it is useful to study the *local* Lyapunov exponents to understand short-term predictability, and also how noise and nonlinearity will interact in epidemic systems. The idea is that regardless of whether dynamics is asymptotically stable, cyclic, or chaotic, there is likely to be regions in the phase plane with expansion in which stochastic divergence will be amplified and

regions of contraction were perturbations will be damped. Grenfell et al. (2002) used local Lyapunov exponents to understand the remarkable predictability of pre-vaccination measles in London. Local Lyapunov exponents are similar in nature to the global exponent, except that rather than evaluating a product of Jacobians across the attractor, the Jacobians are evaluated locally. Armed with an object produced with the `TSIR1lyap`-function it is easy to write a second function to calculate local exponents across m -steps along the attractor (or anywhere else in the phase plane, such as along a “repellor”; see Sect. 10.5). Since the TSIR is a discrete time model, contraction occurs if the largest eigenvalue of the stability matrix is inside the unit circle—thus $\log(|\lambda|) < 0$ is the cut-off between contraction and expansion. The `TSIR1lyap`-function calculates the Local Lyapunov exponents for outputs from the `TSIRlyap`-function. The parameter m controls the number of iterations along the attractor on which to calculate the product (Bailey et al. 1997).

```
TSIR1lyap=function(x, m=1){
  llyap=rep(NA, length(x$I))
  for(i in 1:(length(x$I)-m)) {
    J=matrix(c(1, 0, 0, 1), ncol=2)
    for(k in 0:(m-1)){
      J = matrix(c(x$j11[(i+k)], x$j12[(i+k)],
                  x$j21[(i+k)], x$j22[(i+k)]), ncol = 2,
                  byrow=TRUE) %*% J}
    llyap[i]=log(max(abs(eigen(J)$values)))/m
  }
  res=list(llyap=llyap, I=x$I, S=x$S)
  class(res)="llyap"
  return(res)
}
```

For ease of use we can also write a function to visualize the local exponents:

```
plot.llyap=function(x, inches=.5){
  pm=x$llyap>0
  plot(NA, xlim=range(x$S), ylim=range(x$I), xlab="S",
       ylab="I", log="y")
  symbols(x$S[pm], x$I[pm], circles=x$llyap[pm],
           inches = inches, add=TRUE)
  symbols(x$S[!pm], x$I[!pm], squares = -x$llyap[!pm],
           inches=inches, add=TRUE, bg=2)
}
```

We can study the measles NY attractor using the local Lyapunov exponents. Despite the fact that the overall attractor is chaotic, we see distinct areas of contraction associated with the collapse of epidemics and post-epidemic troughs (Fig. 10.3).

```
nylllyap = TSIRllyap(nlyap, m = 5)
plot.llyap(nylllyap, inches = 0.15)
```

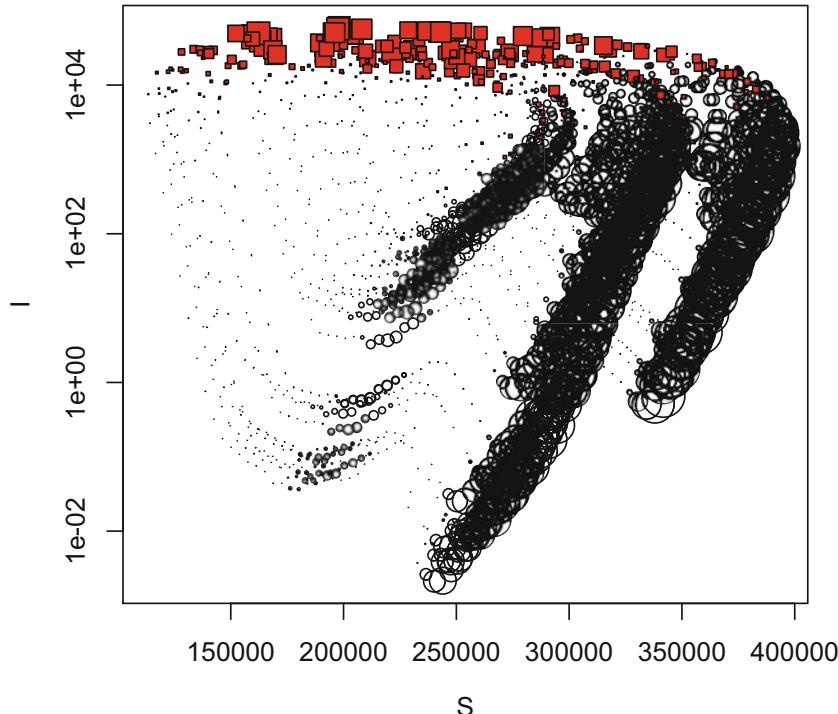


Fig. 10.3 Local Lyapunov exponents across the New York city measles attractor. Positive exponents are shown as open circles and negative exponents as red squares

Section 7.4.2 provided TSIR estimates for pre-vaccination measles in London and showed the limit-cycle nature of the dynamics. Biweekly transmission estimates were:

```
beta=c(27.71, 43.14, 37.81, 33.69, 31.64, 32.10, 30.16,
24.68, 30.19, 31.53, 30.31, 26.02, 26.57, 25.68, 23.21,
19.21, 17.50, 20.50, 29.92, 35.85, 32.65, 28.34, 31.11,
29.89, 26.89, 39.38)
```

Median biweekly birth rate for London was 2083. We can simulate the London attractor and associated global and local Lyapunov exponents.

```
sim=SimTsir2(beta=beta, alpha=0.98,
B=rep(2083, 5200), N=3.3E6, inits=list(Snull=133894,
```

```

    Inull=474))
Sattr=sim$S[5149:5200]
Iattr=sim$I[5149:5200]
lonlyap=TSIRlyap(I=Iattr, S=Sattr, alpha=0.98,
  bt=beta, N=3.3E6)
lonlyap$lyap
## [1] -0.004289374
lonllyap=TSIRllyap(lonlyap, m=1)

```

The dominant Lyapunov exponent is negative testifying to the stability of the limit cycles. We can look in greater detail across the biennial attractor (Fig. 10.4). Interestingly there is potential for significant divergence during the growth phase of the minor and major epidemics; however, the post-epidemic convergence is apparently strong enough to overcome this to result in a strongly dissipative cyclic attractor.

```

pm=lonllyap$lyap>0
plot(NA, xlim=c(1,52), ylim=range(Iattr),
  xlab="Biweek", ylab="I", log="y")
symbols((1:52)[pm], Iattr[pm], circles=
  lonllyap$lyap[pm], inches=.3, add=TRUE)
symbols((1:52)[!pm], Iattr[!pm], squares=
  -lonllyap$lyap[!pm], inches=.3, add=TRUE, bg=2)

```

To visualize this effect more clearly we can plot the long-term deterministic attractor with 20 stochastic simulation (assuming demographic stochasticity only) (Fig. 10.5). The simulations show that despite abundant variability—particularly during the minor epidemics—the trajectories exhibit long-term predictability; That is except for the rare stochastic trajectory that escapes onto the opposite-year coexisting attractor towards the end of the simulations. Grenfell et al. (2001) discuss how the area around Norwich locked on to the opposite-year coexisting attractor compared to the rest of England and Wales for about 15 years following World War II.

```

sim=SimTsir2(beta=beta, alpha=0.98, B=rep(2083, 520),
  N=3.3E6, inits=list(Snull=133894, Inull=474),
  type="det")
plot(sqrt(sim$I), ylab="Sqrt(Cases)", xlab="Biweek")
for(i in 1:20){
  sim=SimTsir2(beta=beta, alpha=0.98, B=rep(2083, 520),
    N=3.3E6, inits=list(Snull=133894, Inull=474),
    type="stoc")
  lines(sqrt(sim$I))}
sim=SimTsir2(beta=beta, alpha=0.98, B=rep(2083, 520),

```

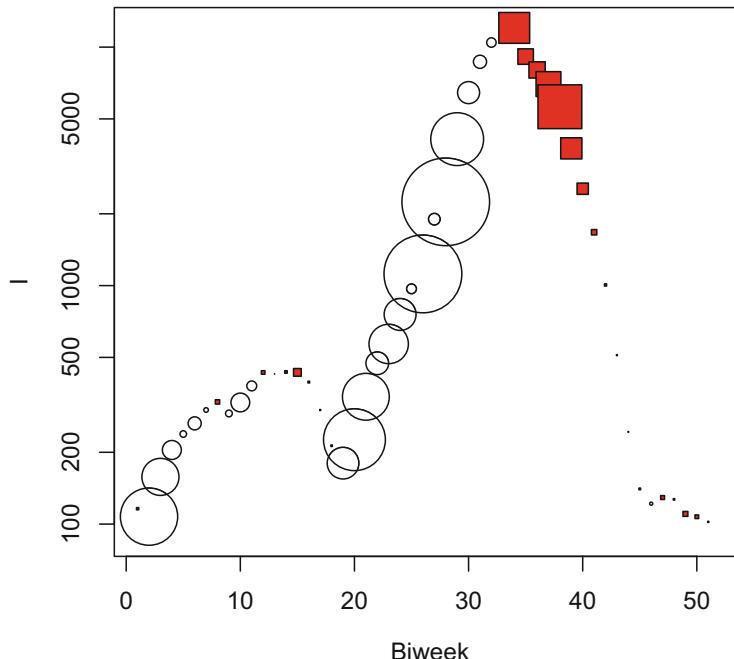


Fig. 10.4 Local Lyapunov exponents across the biennial London measles attractor. Positive exponents are shown as open circles and negative exponents as red squares

```
N=3.3E6, inits=list(Snull=133894, Inull=474),
type="det")
points(sqrt(sim$I), col=2)
```

During the first biweek of 1940, 23 cases of measles were reported in New York city. Given our estimate of the reporting rate of 22.54% in that biweek, a best guess of the incidence is 102; Correspondingly, the best guess of the number of susceptibles is 402,153. To visualize the “sensitive dependence on initial conditions” of the chaotic New York measles attractor we can forward simulate 10 years of dynamics, assuming there were either 5 more (less) infecteds (susceptibles) or 5 less (more) infecteds (susceptibles). The rapid deterministic divergence (Fig. 10.6) is a stark contrast to the predictability of the London attractor in the face of stochasticity (Fig. 10.5).

```
sim2=SimTsir2(beta=exp(glmfit$coef[1:26]), alpha=0.98,
B=rep(median(NY$rec), 260), N=median(N),
inits=list(Snull=402153, Inull=102))
sim3=SimTsir2(beta=exp(glmfit$coef[1:26]), alpha=0.98,
B=rep(median(NY$rec), 260), N=median(N),
inits=list(Snull=402153-5, Inull=102+5))
```

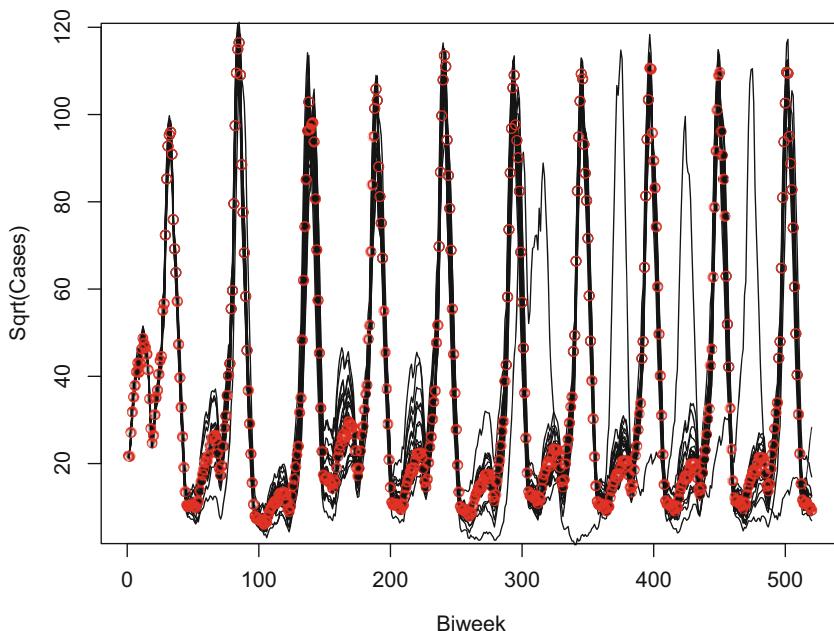


Fig. 10.5 Twenty years of deterministic dynamics (red circles) with 20 stochastic simulations of the biennial London measles attractor

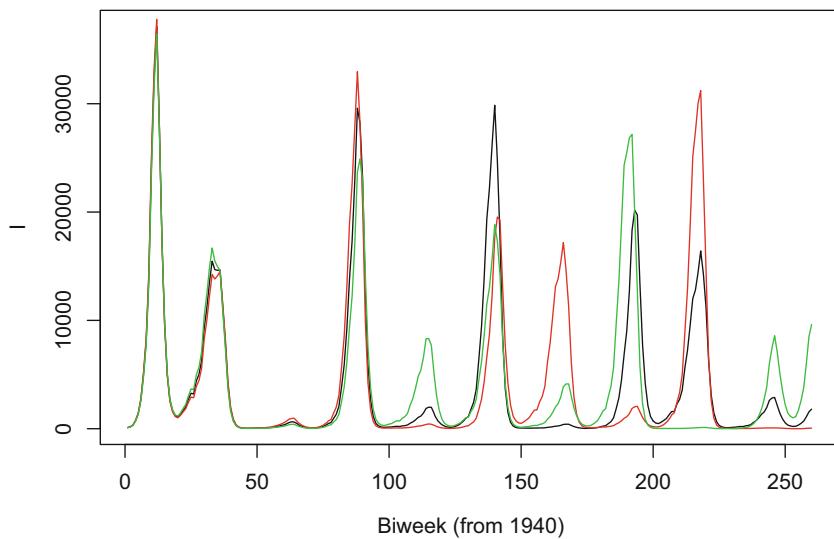


Fig. 10.6 Ten years of deterministic dynamics of the chaotic New York city measles attractor assuming three very similar initial conditions

```

sim4=SimTsir2(beta=exp(glmfit$coef[1:26]), alpha=0.98,
  B=rep(median(NY$rec), 260), N=median(N),
  inits=list(Snull=402153+5, Inull=102-5))
plot(sim2$I[1:260], type="l", ylab="I",
  xlab="Biweek (from 1940)")
lines(sim3$I[1:260], col=2)
lines(sim4$I[1:260], col=3)

```

10.4 Coexisting Attractors

Another nonlinear complication is how seasonally forced epidemic systems can exhibit coexisting attractors as, for example, seen in the seasonally forced SEIR model (Sect. 5.6). Stochastic perturbation can push dynamics between different basins of attraction leading to erratic dynamics not predicted by basic theory.

One of the many puzzles about whooping cough dynamics is the apparent contradiction between historical herd immunity and historical multi-annual epidemics *versus* current circulation in adults.² To reconcile these seemingly mutually exclusive facets of whooping cough epidemiology, Lavine et al. (2011) proposed an hypothesis that immunity to whooping cough wanes over time but re-exposure can boost immunity. This leads to the following SIRWS compartmental model where W is a waning class that is under the influence of two competing processes: return to the S class at a rate of 2ω or boost back to the R class at a rate proportional to the force of infection:

$$\frac{dS}{dt} = \mu(1-p)N - \mu S - \beta SI/N + 2\omega W \quad (10.3)$$

$$\frac{dI}{dt} = \beta SI/N - (\mu + \gamma)I \quad (10.4)$$

$$\frac{dR}{dt} = \gamma I - (\mu - 2\omega)R + \kappa\beta SW/N + \mu p R \quad (10.5)$$

$$\frac{dW}{dt} = 2\omega R - \kappa\beta WI/N - (2\omega + \mu)W \quad (10.6)$$

In the absence of boosting, immunity is expected to last for a mean of $1/\omega$ year (and distributed according to a Gamma-distribution with a shape-parameter of 2; see Sect. 2.7). The parameter κ scales how sensitive boosting is relative to infection, and p is the fraction of children vaccinated at birth. A surprising finding is that in parts of the parameter space, a limit cycle coexists with a fix point attractor. We can

² Modeling chicken pox, a herpes virus that can reactivate in older individuals in the form of zoster, Ferguson et al. (1996) showed that the SEIR model cannot sustain multiannual (or chaotic) childhood dynamics in the presence of “immigration” of the virus from an adult carrier group.

use the forwards and backwards bifurcation algorithm of Sect. 5.6 to look at this. We first define the gradients (using the log-trick):

```
sirwmod=function(t, logy, parms){
  y=exp(logy)
  S=y[1]
  I=y[2]
  R=y[3]
  W=y[4]
  with(as.list(parms),{
    dS = mu * (1-p) * N - mu * S - beta * S * I / N +
      2*omega * W
    dI = beta * S * I / N - (mu + gamma) * I
    dR = gamma * I - mu * R - 2*omega * R +
      kappa * beta * W * I / N + mu*p*N
    dW = 2*omega * R - kappa * beta * W * I / N -
      (2*omega +mu)* W
    res=c(dS/S, dI/I, dR/R, dW/W)
    list(res)
  })
}
```

We assume susceptible recruitment is reduced by vaccination and bifurcate on this parameter (Fig. 10.7). The bifurcation analysis reveals the coexistence of a fix-point and a cyclic attractor when vaccination is in the 20–40% range.

```
require(deSolve)
start = log(c(S=0.06, I=0.01, R=0.92, W = 0.01))
res=matrix(NA, ncol=100, nrow=5000)
p=seq(0.01, 1, length=100)
#Forwards
for(i in 1:100){
  times = seq(0, 200, by=0.01)
  paras = c(mu = 1/70, p=p[i], N = 1, beta = 200,
            omega = 1/10, gamma = 17, kappa=30)
  out = as.data.frame(ode(start, times, sirwmod, paras))
  start=c(S=out[20001,2], I=out[20001,3],
          R=out[20001,4], W=out[20001,5])
  res[,i]=out$I[15002:20001]
  cat(i, "\r")
}
#Backwards
```

```

res2=matrix(NA, ncol=100, nrow=5000)
start = c(S=-1.8, I=-5.8, R=1.9, W=-1.9)
for(i in 100:1){
  paras = c(mu = 1/70, p=p[i], N = 1, beta = 200,
            omega = 1/10, gamma = 17, kappa=30)
  out = as.data.frame(ode(start, times, sirwmod, paras))
  start=c(S=out[20001,2], I=out[20001,3],
           R=out[20001,4], W=out[20001,5])
  res2[,i]=out$I[15002:20001]
  cat(i, "\r")
}
plot(NA, xlim=range(p), ylim=range(res),
      ylab="Log(I)", xlab="p")
for(i in 1:100) points(rep(p[i], 2), range(res[,i]))
for(i in 1:100) points(rep(p[i], 2),
      range(res2[,i])), col=2)

```

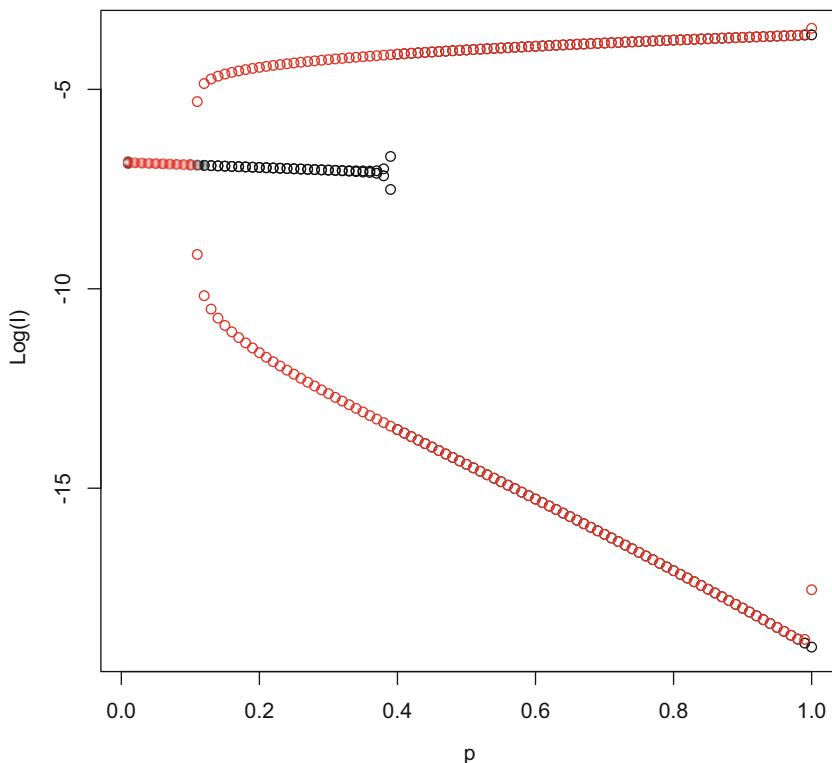


Fig. 10.7 Bifurcation diagram of the SIRWS model across the range of vaccination rates

Figure 10.8 shows trajectories towards the two attractors assuming 20% vaccination but with different initial conditions. For the given parameters the limit cycle is stable and has period of 1.8 years and the fix-point attractor is a stable focus with a damping period of 1.2 years. Using a seasonally forced version of the SIRWS model, Lavine et al. (2013) explored the hypothesis that the regime-shifts in pre-vaccination whooping cough dynamics in Copenhagen (Fig. 10.12) was due to stochastic switching between a low-amplitude noisy annual attractor and a high amplitude cyclic attractor. In the end, the best evidence suggests that the major recurrent outbreaks between 1915 and 1925 was instead a “fly-by” of an unstable multiannual “almost attractor” (Lavine et al. 2013; see Sects. 10.5 and 10.7).

```
paras = c(mu = 1/70, p=0.2, N = 1, beta = 200,
          omega = 1/10, gamma = 17, kappa=30)
start=c(S=-1, I=-5, R=3.3, W=0)
times = seq(0, 30, by=1/52)
out = as.data.frame(ode(start, times, sirwmod, paras))
plot(out$time, exp(out$I), xlab="Year",
      ylab="I", type="l", ylim=c(0,0.05))
start = c(S=-1.8, I=-5.8, R=1.9, W=-1.9)
times = seq(0, 30, by=1/52)
out = as.data.frame(ode(start, times, sirwmod, paras))
lines(out$time, exp(out$I), col=2)
```

10.5 Repellors/Unstable Manifolds

Rand and Wilson (1991) studied a seasonally forced SEIR model (Sect. 5.3) of chickenpox (assuming that shedding from zoster can be ignored); They assumed a latent period and infectious period of around 10 days and sinusoidally forced transmission with a β_0 of 537/year, $\beta_1 = 0.3$, and a birth rate of 0.02.

The integrated ODEs predict robust annual epidemics in the presence of seasonal forcing (Fig. 10.9).

```
times = seq(0, 100, by = 1/120)
start = c(S = 0.06, E = 0, I = 0.001, R = 0.939)
cparas = c(mu = 0.02, N = 1, beta0 = 537, beta1 = 0.3,
           sigma = 36, gamma = 34.3)
out = as.data.frame(ode(start, times, seirmod2, cparas))
plot(out$time, out$I, type="l", xlab="Year", ylab=
  "Prevalence", xlim=c(91,100), ylim=c(0, 0.0015))
```

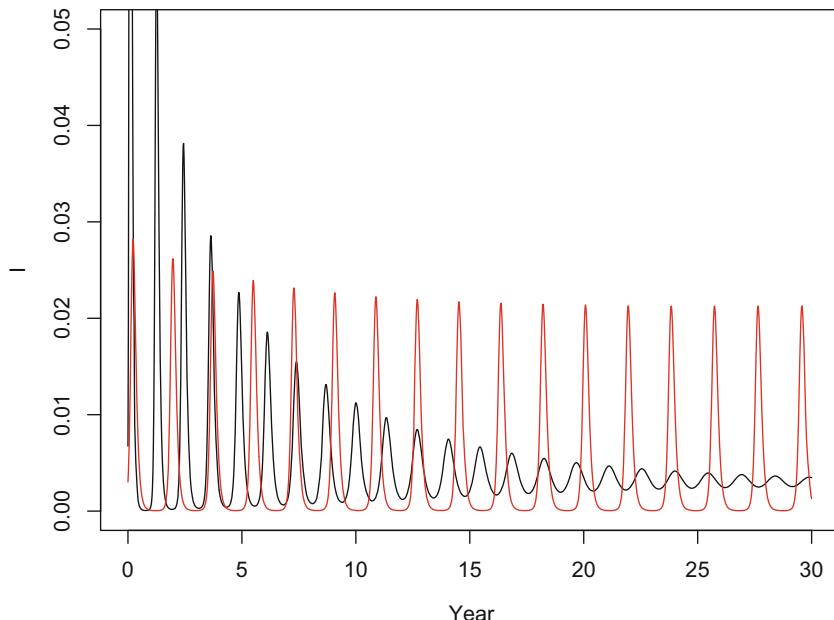


Fig. 10.8 The two coexisting attractors of the SIRWS-model with 20% vaccination at birth

In contrast, stochastic simulations of the model (see chapter Appendix for detail)—assuming stochasticity in the seasonal transmission rate—exhibit dynamics with conspicuous “regime-shifts”; Periods with the expected somewhat variable annual outbreaks are interspersed with periods of violent multiannual cycles with a period of around 4 years (Fig. 10.10) that is completely unrelated to the damping period of the unforced SEIR model (~ 2.8 years for these parameters). This appears to be a dynamical phenomenon different to what we have studied previously.

For the stochastic simulations we build a pomp object using the “C-snippets” detailed in the Appendix. The dat-data object defines the times for the stochastic simulation. We are not working with data, so the reports column is just a dummy.

```
dat=data.frame(time=seq(0, 500, by=1/52), reports=NA)
seirp=pomp(dat, times="time", t0=0,
            rprocess=euler.sim(Csnippet(rproc), delta.t=1/52/20),
            skeleton=vectorfield(Csnippet(skel)),
            rmeasure=Csnippet(robs),
            dmeasure=Csnippet(dobs),
            initializer=Csnippet(rinit),
            params=c(iota=0, beta0=537, beta1=0.3, sigma=36,
```

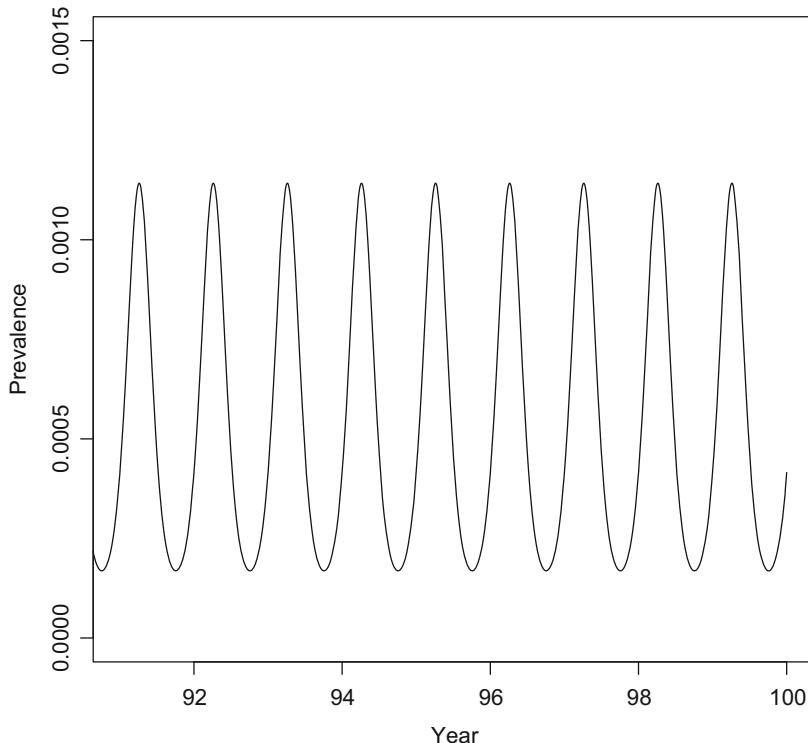


Fig. 10.9 Ten years of simulation of the forced SEIR model parameterized according to Rand and Wilson's (1991) chickenpox parameters predicts robust annual epidemics

```
gamma=34.3, alpha=0.015, rho=0.6, theta=1,
b=0.02, mu=0.02, pop0=5e8,
S0=0.06, E0=0, I0=0.001, R0=0.939),
paramnames=c("iota", "beta0", "beta1", "gamma",
"sigma", "alpha", "rho", "theta", "b", "mu", "pop0",
"S0", "E0", "I0", "R0"),
statenames=c("S", "E", "I", "R", "inc", "pop"),
zeronames="inc")
```

We simulate deterministic and stochastic trajectories to produce Figs. 10.9 and 10.10.

```
detsim<-trajectory(seirp, as.data.frame=TRUE)
plot(detsim$time, detsim$I/5E8, type="l",
xlim=c(101, 110), xlab="year", ylab="prevalence")
```

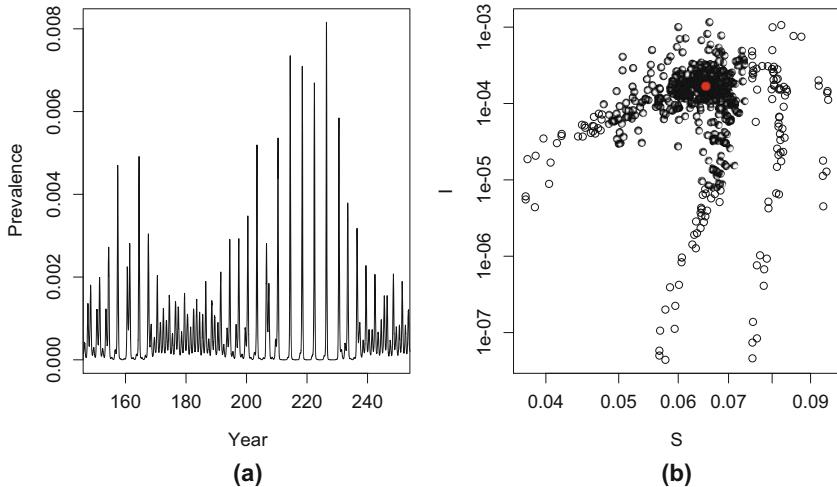


Fig. 10.10 (a) Hundred years of stochastic simulation of the forced SEIR model parameterized according to Rand and Wilson’s (1991) chickenpox parameters assuming stochasticity in transmission. (b) Annual stroboscopic section of the stochastic simulation is the S-I phase plane

The annual stroboscopic section of the deterministic and stochastic simulation is shown in Fig. 10.10b:

```
par(mfrow=c(1, 2))
stocsim<-simulate(seirp, seed=3495135,
  as.data.frame=TRUE, nsim=1)
plot(stocsim$time, stocsim$I/5E8, type="l", xlim=c(150,
  250), xlab="Year", ylab="Prevalence")
sel=seq(105, length(stocsim$I), by=52)
plot(stocsim$S[sel]/5E8, stocsim$I[sel]/5E8,
  log="xy", xlab="S", ylab="I")
sel2=sel[401:500]
points(detsim$S[sel2]/5E8, detsim$I[sel2]/5E8, col=2,
  pch=21, bg=2)
```

Rand and Wilson (1991) studied the apparent similarity of the stochastic trajectory in the S-I phase plane (Fig. 10.10) to the quasiperiodic chaotic attractor of the parametrically nearby model of Sect. 5.4 (Fig. 5.5). They stipulated that the stochastic dynamics of the deterministically annual system is intermittently governed by the weakly unstable “ghost” of the nearby 4-year quasi-periodic chaotic attractor. To study this further we turn to the notion of “invasion orbits.”

10.6 Invasion Orbits

Studying highly nonlinear, stochastic dynamical systems is complicated by the intermingling of two different sources of dynamic variability: the variability due to deterministic instability and the variability due to stochastic forcing (Bjørnstad and Grenfell 2001). In order to elucidate this complexity, we may think of the stochastic forcing as a perturbation to the nonlinear system which laws subsequently will attempt to return the system to the deterministic attractor. In Sect. 5.3, we discussed how to study the long-term asymptotic behavior of the seasonally forced SEIR system through numeric integration from arbitrary initial conditions and discarding the initial transient dynamics. “Invasion orbits” is the flip-side of this; Systematically distribute initial conditions across the phase plane and numerically integrate the transients to describe the trajectories toward the deterministic attractor. For “linear systems” or approximately linear systems—in the dynamical systems sense—the invasion orbits will be monotonic trajectories towards a stable node and smooth spirals towards a stable focus (see e.g., Fig. 2.5); The period of the inward spiral will be determined by the dampening period of the focus as discussed in Chap. 9. For highly nonlinear systems, in contrast, the approach towards the attractor may not be smooth. We can illustrate this using Rand and Wilson’s (1991) chickenpox SEIR model. We first have to set up a systematic grid of initial conditions:

```
starts=expand.grid(S=seq(0.02, 0.1, length=10),
                    E=seq(1E-8, 0.0125, length=10),
                    I=seq(1E-8, 0.005, length=10))
starts$R=1-apply(starts, 1, sum)
```

For each of these 1000 initial conditions we will simulate the system for 100 years and then store the annual pointcare section in the S-I plane. Rand and Wilson (1991) suggested this be done after discarding a short burn-in period (we use 5 years):

```
#times for integration
itimes = seq(0, 100, by=1/52)
#points for stroboscopic section
isel=seq(1, length(itimes), by=52)
#list to fill with results
cporbs=list(S=matrix(NA, ncol=dim(starts)[1],
                     nrow=length(isel)), I=matrix(NA,
                     ncol=dim(starts)[1], nrow=length(isel)))
```

We now integrate to obtain the 1000 invasion orbits:

```
for (i in 1:dim(starts)[1]){
  out2b = as.data.frame(ode(as.numeric(starts[i,]),
    itimes, seirmod2, cparas))
  cporbs$S[,i]=out2b[,2][isel]
  cporbs$I[,i]=out2b[,4][isel]
}
```

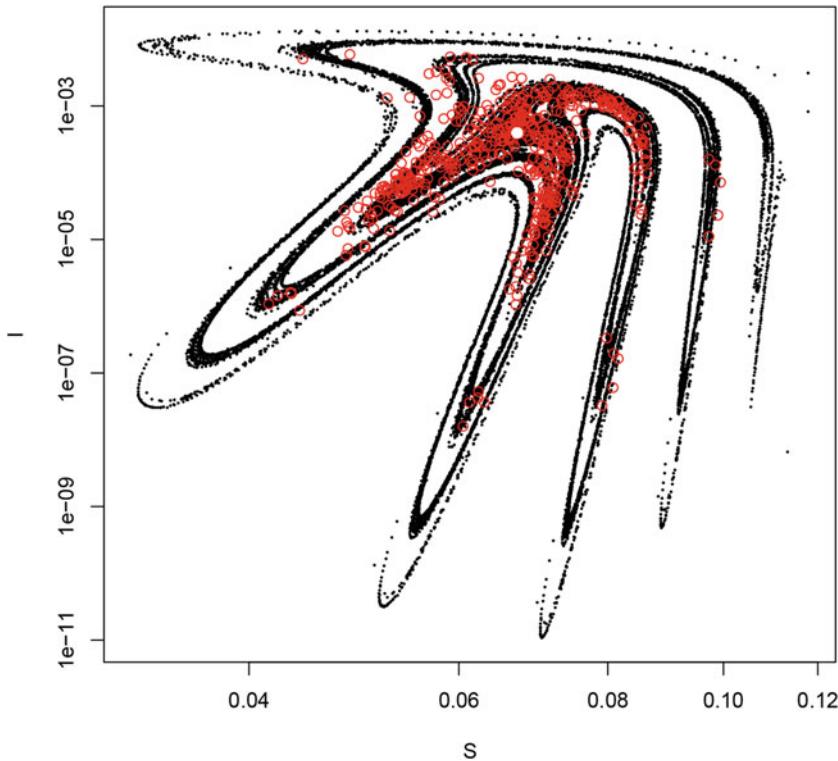


Fig. 10.11 The stroboscopic section of the invasion orbits of the forced SEIR model trace out the ghost of a chaotic attractor that has lost stability in the region of parameter space that Rand and Wilson (1991) used for their chickenpox model. The white central circle is the annual deterministic attractor, and the red open circles are annual stroboses from a stochastic simulation

We can finally plot the stroboscopic section of the invasion orbits in the S-I phase plane with the deterministic attractor superimposed (Fig. 10.11) to reveal that the stochastic simulation is largely governed by the unstable highly nonlinear structure in the phase plane dubbed variously a “repellor,” a “chaotic saddle,” or an “unstable manifold.” Eckmann and Ruelle (1985) also referred to it as an “almost attractor.”

```
#Invasion orbits
plot(as.vector(cporbs$S[-c(1:5),]),
     as.vector(cporbs$I[-c(1:5),]), pch=20, cex=0.25,
     log="xy", ylab="I", xlab="S")
#Stochastic simulation
sel=seq(105, length(stocsim$I), by=52)
points(stocsim$S[sel]/5E8, stocsim$I[sel]/5E8, col=2)
#Deterministic attractor
times = seq(0, 1000, by=1/120)
start = c(S=0.06, E=0, I=0.001, R = 0.939)
out = as.data.frame(ode(start, times, seirmod2, cparas))
sel=seq(120*100, length(times), by=120)
points(out$S[sel], out$I[sel], pch=21, col="white",
       bg="white")
```

Rohani et al. (2002) discussed how the multiannual cycles in whooping cough following vaccination may be explained as the dynamics chasing a periodic saddle (i.e., a periodic “almost attractor”).

10.7 Stochastic Resonance

Whooping cough in pre-vaccination Copenhagen generally exhibited low amplitude “fuzzy” epidemics, with the exception of a 10-year period of violent epidemics starting around 1915 (Fig. 10.12)(Lavine et al. 2013). We use wavelet analysis with the “crazy climber” ridge finding algorithm to characterize the transitions in dynamics.

```
data(pertcop)
require(Rwave)
#Wavelet decompostion
no=8
nv=16
a=2^seq(1, no+1-1/nv, by=1/nv)
wfit=cwt(sqrt(pertcop$cases), no, nv, plot=FALSE)
wspec = Mod(wfit)
#Crazy climber
crcinc<-crc(wspec, nbclimb=10, bstep=100)
fcrcinc<-cfamily(crcinc, ptile=0.5, nbchain=1000,
                  bstep=10)
## There are 1 chains.
```

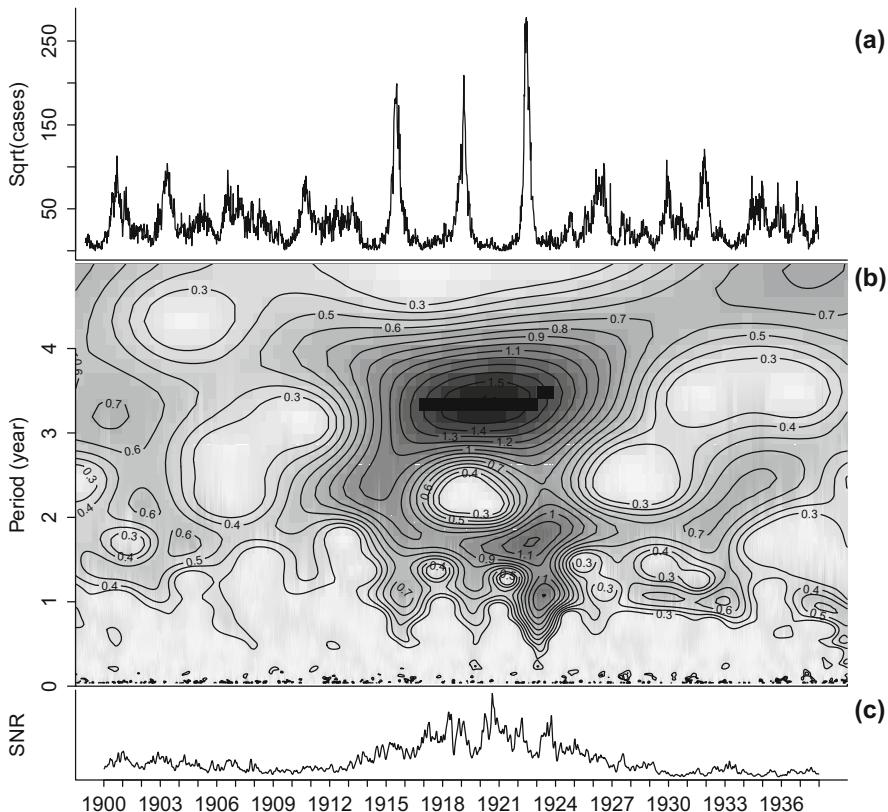


Fig. 10.12 (a) Incidence of whooping cough in Copenhagen. (b) The wavelet spectrum reveals a 10-year run of significant 3-year cycles starting around 1915. (c) The 3-year cycles are associated with increased “signal-to-noise” (SNR) ratio in the wavelet spectrum

Lavine et al. (2013) used the ratio of variation in the multiannual *versus* high-frequency part of the wavelet spectrum as a simple measure of the time varying “signal-to-noise” (SNR) ratio in the whooping cough dynamics:

```
sigind=which((a/52)>3 & (a/52)<4)
noiseind=which((a/52)<0.5)
snr=apply(wspec[, sigind], 1,
          sum)/apply(wspec[, noiseind], 1, sum)
```

We can finally make a composite plot of incidence, wavelet spectrum, and signal-to-noise ratio (Fig. 10.12) to highlight how the major epidemics are much less “noisy” than the low amplitude cycles. Lavine et al. (2013) fit a seasonally forced SIRWS (Eq. (10.4)) model to the data and concluded that the curious run of violent epidemics of whooping cough appeared to trace out an unstable multiannual “almost attractor.”

```

par(mfrow = c(3,1), mar = c(0,4,2,1))
layout(matrix(c(1,1,2,2,2,3), ncol = 1))
#Top panel
plot(as.Date(pertcop$date), pertcop$cases, xlab = "",
      ylab = "Sqrt(cases)", type = "l", bty = 'l',
      xlim = c(as.Date("1901-01-01"),
                as.Date("1938-01-01")), xaxt='n', yaxt='n')
axis(2, at=seq(0,200,by=100), labels=FALSE)
axis(2, at=seq(50,250,by=100), labels=TRUE)
#Mid panel
par(mar=c(0,4,0.25,1))
image(x=as.Date(pertcop$date, origin="1900-01-07"),
      wspec, col=gray((30:10)/32), y=a/52, ylim=c(0,5),
      ylab="Period (year)", main="", xaxt="n", yaxt="n")
contour(x=as.Date(pertcop$date, origin="1900-01-07"),
         wspec, y=a/52, ylim=c(0,5),
         zlim=c(quantile(wspec) [4], max(wspec)), add=T)
axis(2, at=0:4)
ridges<-fcrcinc[[1]]
ridges[which(ridges<1.5*10^-5)]<-NA
image(x=as.Date(pertcop$date, origin="1900-01-07"),
      y=a/52, z=ridges, add=T, col="black")
#Bottom panel
par(mar=c(3,4,0.25,1), tcl=-0.4)
plot(x=as.Date(pertcop$date, origin="1900-01-07"), snr,
      type="l", bty="l", xaxt="n", yaxt="n", ylab="SNR")
axis.Date(1, at=seq(as.Date("1900-01-01"),
                  as.Date("1938-01-01"), "years"))

```

In addition to highlighting the potential influence of unstable manifolds in disease dynamics, pre-vaccination Copenhagen whooping cough hints at another *exotic* feature of certain nonlinear dynamical systems: increased stochasticity can sometimes *increase* predictability through a process of “stochastic resonance” (Wiesenfeld and Moss 1995; Gammaiton et al. 1998). We can illustrate this phenomenon with the seasonally forced stochastic SEIR model introduced in Sect. 10.5. The pomp C-snippets are detailed in the Appendix. We simulate 500 years of weekly data across a range of 126 transmission variances between 0 and 0.025 (given by the alpha-vector). The stochastic dynamics is prone to extinction so, for each parameter set, we change the pseudorandom seed until a 500-year persistent time series is produced (using the while-loop):

```

dat=data.frame(time=seq(0, 500, by=1/52), reports=NA)
sds=rep(NA, 126)
alpha=seq(0, 0.025, by=0.0002)
Smat=Imat=matrix(NA, nrow=26001, ncol=126)

```

```

for(i in 1:126){
  seirp=pomp(dat, times="time",t0=0,
  rprocess=euler.sim(Csnippet(rproc),delta.t=1/52/20),
  skeleton=vectorfield(Csnippet(skel)),
  rmeasure=Csnippet(robs),
  dmeasure=Csnippet(dobs),
  initializer=Csnippet(rinit),
  params=c(iota=0,beta0=537,beta1=0.3,sigma=36,
    gamma=34.3,alpha=alpha[i],rho=0.6,theta=1,
    b=0.02,mu=0.02,pop0=5e8,
    S0=0.06,E0=0,I0=0.001,R0=0.939),
  paramnames=c("iota","beta0","beta1","gamma",
    "sigma","alpha","rho","theta","b","mu","pop0",
    "S0","E0","I0","R0"),
  statenames=c("S","E","I","R","inc","pop"),
  zeronames="inc"
  )
  stocsim$I[26001]=0
  j=-1
  while(stocsim$I[26001]==0){
    j=j+1
    stocsim<-simulate(seirp, seed=3495131+j,
      as.data.frame=TRUE,nsim=1)
    sds[i]=3495131+j
  }
  Imat[,i]=stocsim$I
  Smat[,i]=stocsim$S
}

```

To study *stochastic resonance* we simulate the model across a range of stochastic variability in the transmission rate. We use wavelet analysis to quantify “predictability” as a function of stochasticity (Fig. 10.13).

```

predn=rep(NA, 126)
#Set the number of "octaves" and "voices"
no = 8; nv = 10
#then calculate the corresponding periods
a = 2^seq(1,no+1-1/nv, by = 1/nv)
sel2= a>=39 & a <260 #Multiannual signal
sel=a<39 #High frequency noise
for(i in 1:126){

```

```
wfit = cwt(sqrt(Imat[,i]), no, nv, plot=FALSE)
wspec = Mod(wfit)
predn[i]=sum(wspec[,sel2])/sum(wspec[sel])
}
plot(alpha, predn, xlab="Noise variance",
      ylab='Predictability')
```

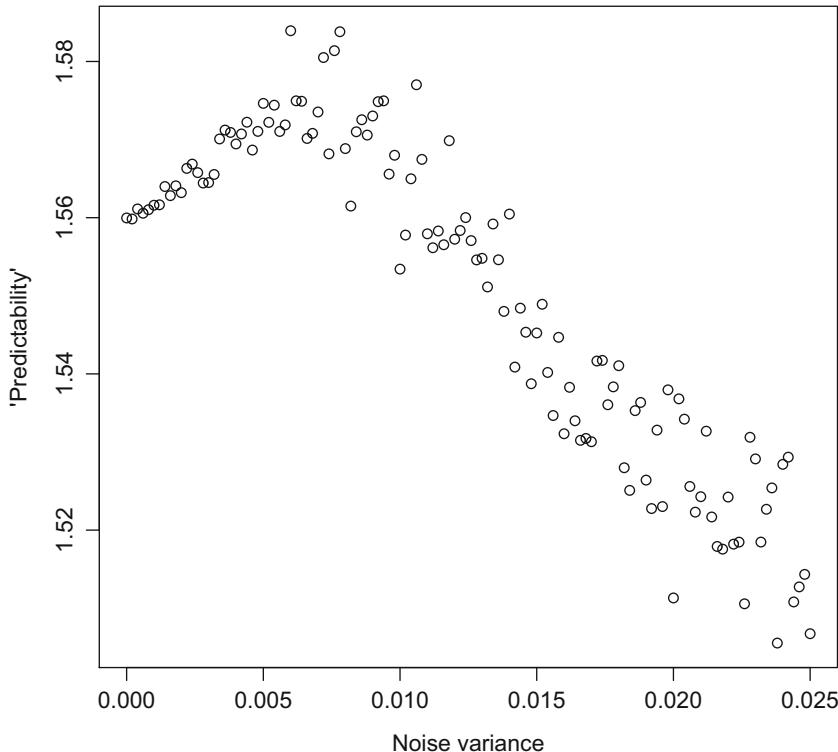


Fig. 10.13 “Predictability” measured as the ratio of the wavelet spectrum that falls in the multiannual region versus the high-frequency region as a function of the stochastic variance in the transmission rate (β_0) for the seasonally forced SEIR model

The wavelet “signal-to-noise” ratio indicates the curious phenomenon that “predictability” increases with noise up to a point and then decays (Fig. 10.13). The effect comes about because with low noise variance, the system rarely interacts with the high-amplitude quasi-periodic “almost attractor” and at high noise variance the stochasticity breaks the epidemiological clockwork. Stochasticity can push a dynamical system towards an “almost stable” multiannual cycle, as Lavine et al. (2013) argued was the case of pre-vaccination whooping cough in Copenhagen. Stochasticity may also push dynamics towards an “almost stable” fix-point, for which the

perhaps clearest ecological illustration is provided by laboratory colonies of cyclic populations of the flour beetle *Tribolium castaneum* (Cushing et al. 1998); or an “almost stable” chaotic manifold, as is the case of the seasonally forced chickenpox model of Rand and Wilson (1991). The latter phenomenon led to an interesting discussion of the meaning of “noise-induced chaos” (Yao and Tong 1994; Dennis et al. 2003; Ellner and Turchin 2005).

10.8 Predictability: Empirical Dynamic Modeling

The “predictability” measure used in the previous section is not truly a measure of the level of determinism of the dynamics. Various researchers have proposed to use some form of nonparametric autoregression—sometimes called “nonlinear forecasting” and recently “empirical dynamic modeling” (Ye et al. 2015)—in combination with leave-one-out cross-validation to quantify predictability in empirical time series. These approaches have used nearest-neighbor methods (Sugihara et al. 1990), kernel regression (Yao and Tong 1998), and local polynomials (Fan et al. 1996). The `nlts`-package has implementations of the local polynomial approaches proposed by Tong and coworkers (Cheng and Tong 1992; Fan et al. 1996; Yao and Tong 1998) building on the `locfit`-package (Loader 2006). The function `ll.order` calculates the cross-validation error across a range of kernel bandwidths and autoregressive lags.³

We consider the chickenpox SEIR model across a range of stochasticities in transmission and use `ll.order` to calculate the cross-validation predictability of annually strobbed versions of the weekly simulations (discarding the first 10 years).

```
require(nlts)
llcv=rep(NA,126)
for(i in 1:126){
  llfit=ll.order(sqrt(Iimat[seq(521,26001, by=52),i]),
    step=1, order=1:5, bandwidt = seq(0.5, 1.5,
    by = 0.5), cv=FALSE)
  llcv[i]=min(llfit$grid$GCV)
}
plot(llcv~alpha, ylab="GCV", xlab="Noise variance")
```

The action of stochastic resonance due to the “almost attractor” is readily visible, as the (generalized) cross-validation error is lowest for intermediate noise variances (Fig. 10.14).

³ The method was originally proposed as a nonparametric method to estimate the “order” of a time series (Cheng and Tong 1992).

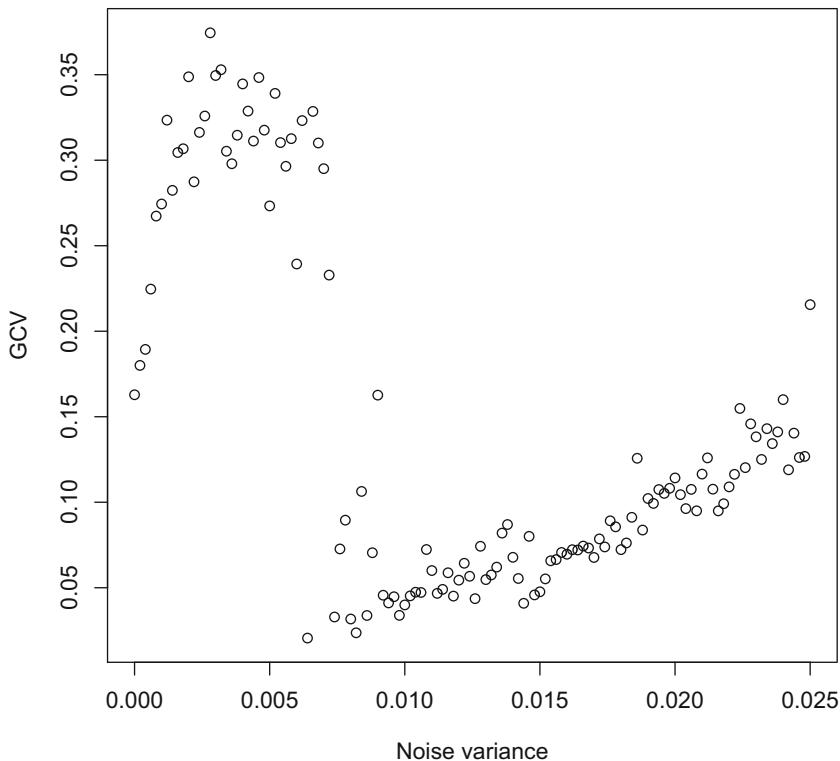


Fig. 10.14 Predictability measured as the cross-validation error of the optimized nonparametric autoregression as a function of the stochastic variance in the transmission rate (β_0) for the seasonally forced SEIR model

In an early application to epidemiology, Sugihara et al. (1990) proposed to use nonparametric prediction-error as a function of prediction lag to distinguish deterministic chaos from noisy limit cycles in measles epidemics. Nonparametric autoregression is a completely “mechanism-free” model for the disease dynamics. We can use the `11.edm`-function to check that the method produces dynamics that are in rough correspondence to the empirical patterns. Let’s consider a 10-year segment of the 10th weekly simulation. We use order (embedding dimension) 3, because this is indicated as best fit based on the order-consistent estimator (`11.order`). The resultant empirical dynamic model has roughly appropriate dynamics, though the period is slightly too short (Fig. 10.15)

```
x=sqrt(Imat[seq(521,1040, by=1),10])
sim=11.edm(x=x, order=3, bandwidth=0.8)
plot(x, type="b", ylab="Prevalence", xlab="Week")
lines(sim, col=2)
```

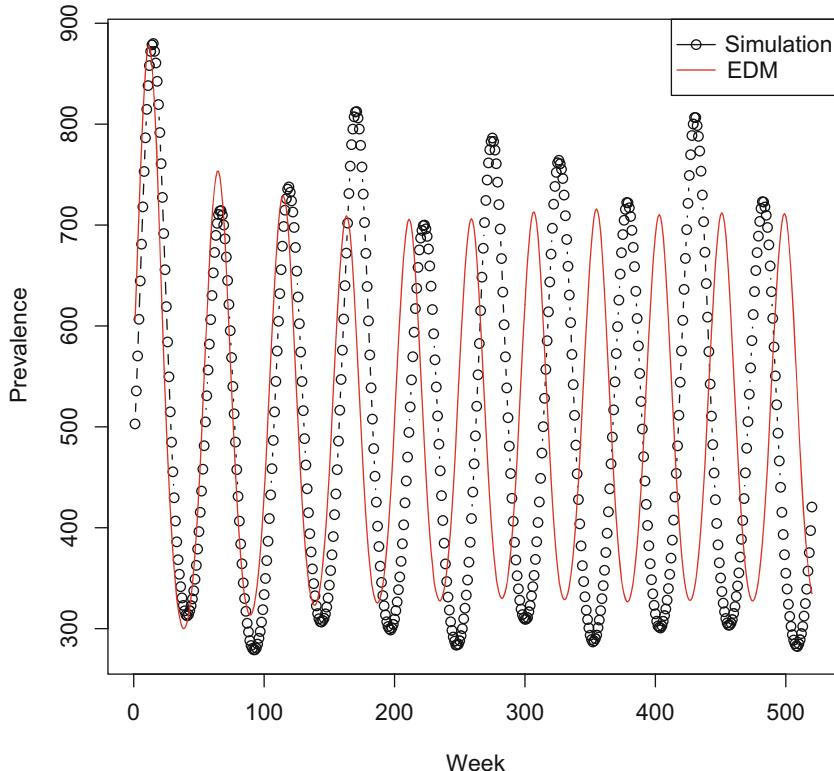


Fig. 10.15 Model simulation and dynamics predicted by the nonparametric autoregressive (“empirical dynamic”) model

```
legend("topright", legend=c("Simulation", "EDM"),
      lty=c(1,1), pch=c(1, NA), col=c("black", "red"))
```

Appendix: Making a Pomp-Simulator

Doing the computations involved in Sects. 10.5 and 10.7 are computationally expensive. The pomp-package includes a Csnippet-function that will compile C code on the fly to speed up calculations. The following provides the C code used in the simulations of the stochastic SEIR model.

```
require(pomp)
```

We first define the Csnippet for the deterministic skeleton of the unobserved process:

```

"
double rate[8]; // transition rates
double trans[8]; // transition numbers
double beta=beta0*(1+beta1*cos(2*M_PI*t));//transmission
double lambda = (iota+I*beta)/pop; // FoI

// transition rates
rate[0] = b*pop; // birth of S
rate[1] = lambda; // infection of S
rate[2] = mu; // death of S
rate[3] = sigma; // latent period of E
rate[4] = mu; // death of E
rate[5] = gamma; // recovery of I
rate[6] = mu; // death of I
rate[7] = mu; // death of R

// compute the transition numbers
trans[0] = rate[0];
trans[1] = rate[1]*S;
trans[2] = rate[2]*S;
trans[3] = rate[3]*E;
trans[4] = rate[4]*E;
trans[5] = rate[5]*I;
trans[6] = rate[6]*I;
trans[7] = rate[7]*R;

// balance the equations
DS = trans[0] - trans[1] - trans[2];
DE = trans[1] - trans[3] - trans[4];
DI = trans[3] - trans[5] - trans[6];
DR = trans[5] - trans[7];
Dinc = trans[5]; // cumulative recovery
Dpop = trans[0]-trans[2]-trans[4]-trans[6]-trans[7];
" -> skel

```

Then the Csnippet for the stochastic simulator

```

"
double rate[8]; // transition rates
double trans[8]; // transition numbers

double beta=beta0*(1+beta1*cos(2*M_PI*t));//transmission
double dW = rgammawn(alpha,dt); // white noise
double lambda = (iota+I*beta*dW/dt)/pop;

```

```

// transition rates
rate[0] = b*pop; // birth of S
rate[1] = lambda; // infection of S
rate[2] = mu; // death of S
rate[3] = sigma; // latent period of E
rate[4] = mu; // death of E
rate[5] = gamma; // recovery of I
rate[6] = mu; // death of I
rate[7] = mu; // death of R

// compute the transition numbers
trans[0] = rpois(rate[0]*dt); // births are Poisson
reulermultinom(2, S, &rate[1], dt, &trans[1]);
reulermultinom(2, E, &rate[3], dt, &trans[3]);
reulermultinom(2, I, &rate[5], dt, &trans[5]);
reulermultinom(1, R, &rate[7], dt, &trans[7]);

// balance the equations
S += trans[0] - trans[1] - trans[2];
E += trans[1] - trans[3] - trans[4];
I += trans[3] - trans[5] - trans[6];
R += trans[5] - trans[7];
inc += trans[5]; // cumulative recovery
pop = S + E + I + R;
" -> rproc

```

pomp wants Csnippets for the observational process also (even if we only use the object for simulation).

```

## Observation model simulator (negative binomial)
"
double mean = rho*inc;
double size = 1/theta;
reports = rnbinom_mu(size,mean);
" -> robs

## Observation model likelihood (negative binomial)
"
double mean = rho*inc;
double size = 1/theta;
lik = dnbinom_mu(reports,size,mean,give_log);
" -> dobs

```

We need initial conditions

```
""
S = nearbyint(pop0*S0);
E = nearbyint(pop0*E0);
I = nearbyint(pop0*I0);
R = nearbyint(pop0*R0);
pop = S+E+I+R;
inc = 0;
" -> rinit
```

Finally we can build the `pomp` object. The `dat`-data object defines the times for the stochastic simulation. We are not working with data, so the `reports` column is just a dummy.

```
dat=data.frame(time=seq(0, 500, by=1/52), reports=NA)
seirp=pomp(dat, times="time", t0=0,
rprocess=euler.sim(Csnippet(rproc), delta.t=1/52/20),
skeleton=vectorfield(Csnippet(skel)),
rmeasure=Csnippet(robs),
dmeasure=Csnippet(dobs),
initializer=Csnippet(rinit),
params=c(iota=0,beta0=537,beta1=0.3,sigma=36,
gamma=34.3,alpha=0.015,rho=0.6,theta=1,
b=0.02,mu=0.02,pop0=5e8,
S0=0.06,E0=0,I0=0.001,R0=0.939),
paramnames=c("iota","beta0","beta1","gamma","sigma",
"alpha","rho","theta","b","mu","pop0",
"S0","E0","I0","R0"),
statenames=c("S","E","I","R","inc","pop"),
zeronames="inc")
```

The `pomp`-package has numerous functions to simulate deterministic and stochastic trajectories from `pomp`-objects.

Chapter 11

Spatial Dynamics



11.1 Dispersal Kernels

Pathogens move in space because of movement of transmission stages and infected/susceptible hosts. Spatial pattern arises from landscape heterogeneities, dispersal and “reaction-diffusion” dynamics among spatially dispersed susceptible, and infected individuals. The probability distribution that governs dispersal distances is often referred to as the dispersal kernel. A variety of functional forms have been proposed in the ecological and epidemiological literature (e.g., Mollison 1991; Clark 1998; Bjørnstad and Bolker 2000; Smith et al. 2002). From the point of view of basic theory, it is often assumed that dispersal takes an exponential (the probability of dispersing a distance $d \propto \exp(-d/a)$, where a is the range) or Gaussian ($\propto \exp(-(d/a)^2)$) shape. The exponential model arises, for example, if we assume dispersal happens in a constant direction with a constant stopping rate. The Gaussian model arises if the stopping rate is constant but movement direction changes randomly like a Brownian motion. However, other kernels are relevant; Broadbent and Kendall (1953) calculated the movement probabilities of infectious larvae of a gut nematode of sheep, *Trichostrongylus retortaeformis*, that performs a random walk until it encounters a leaf of grass. Assuming the location of the leaves are according to a spatially random point process, they showed that the random walk leads to a dispersal distance distributions that follows a Bessel K_0 -function. Ferrari et al. (2006b)

This chapter uses the following R-packages: ncf and animate.

A conceptual understanding of *spatial spread* is useful prior to this discussion. A 5-min epidemics-MOOC can be seen on YouTube: <https://www.youtube.com/watch?v=WPjsAdyD1Gg>.

used this kernel in a model of pollinator-vectored plant pathogens. Empirical dispersal distribution of free-living organisms typically has an over-representation of rare long-range jumps that are improbable according to these kernels; They are the so-called “fat-tailed” kernels (Clark 1998), which have important consequences for the speed of spatial spread (Kot et al. 1996).

For human infections spatially contiguous, diffusive kernels are often a poor fit to empirical patterns because spread often follows a characteristic “hierarchical” fashion (Grenfell et al. 2001); Infections usually appear in big cities early, thereafter the timing of epidemics on average happens in an order of descending size and increasing isolation. This chapter is focused on inferring the shape of the spread-kernel from spatial patterns over time, and then investigates the dynamical consequences of such spread. We start with considering the simpler diffusive kernels and then consider the more complicated patterns arising from human mobility.

11.2 *Filipendula* Rust

Jeremy Burdon and Lars Ericson surveyed presence/absence of a fungal pathogen on a wild plant, *Filipendula ulmaria*, across islands in a Swedish archipelago (Smith et al. 2003). The *filipendula* data contains observations for 1994 (\$y94) and 1995 (\$y95), with spatial coordinates \$X and \$Y. There are additionally a large number of descriptive covariates for each site. Smith et al. (2003) used the data to estimate the most likely dispersal kernel of the rust. The host plant is an herbaceous perennial with pathogen spores overwintering on dead tissue. The infections in 1995 thus arose from the spores produced in 1994.

If spores disperse according to, say, an exponential function with range, a , then the spatial force of infection on any location, i , will be $\propto \sum_j z_j \exp(-d_{ij}/a)$, where z_j is the disease status (0/1) in the previous year and d_{ij} are the distances to other locations. The idea is that in each spring, every local group of hosts will be in the accumulated spore shadow of last year’s infected individuals. This leads to a metapopulation “incidence-function” model (Hanski 1994) for the presence/absence of rust among all locations from year to year. Figure 11.1 shows the spatial data.

```
data(filipendula)
symbols(filipendula$x, filipendula$y, circles=
  rep(1,162), inches=.1, bg=filipendula$y95+1,
  xlab="X", ylab="Y")
symbols(filipendula$x, filipendula$y, circles=
  rep(1,162), inches=.05, bg=filipendula$y94+1,
  add=TRUE)
legend("topright", c("infected 94", "infected 95"),
  pch=c(21,21), pt.cex=c(1,2), pt.bg=c(2,2))
```

As for the basic catalytic (Chap. 4) and TSIR (Chap. 7) models, we can use the glm-framework to estimate the parameters. Since the response variable is binary,

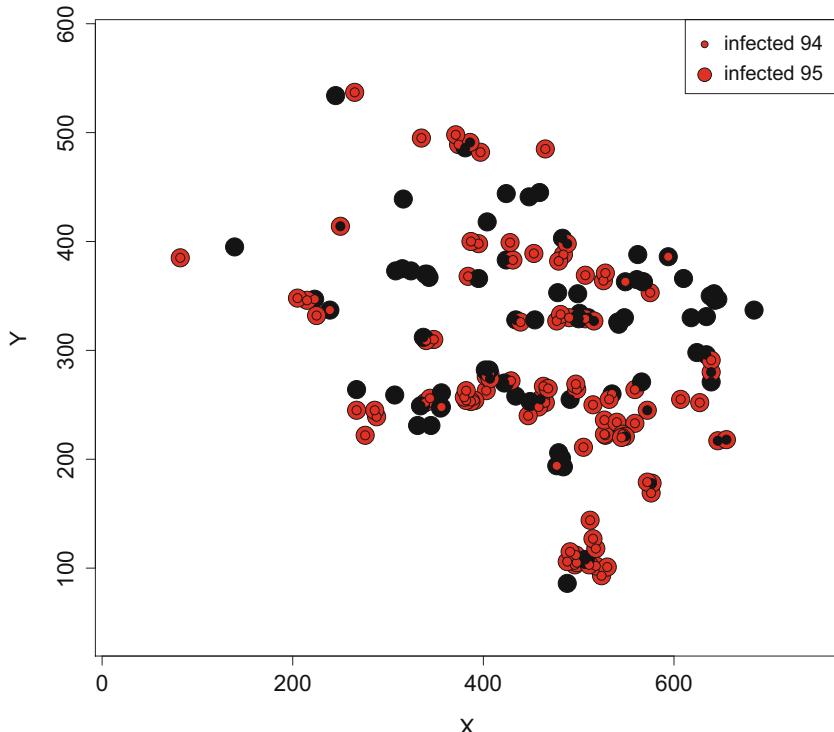


Fig. 11.1 Presence/absence of the rust on its *Filipendula ulmaria* host plant in 1994 and 1995. Red is infected. Black is uninfected

we use logistic regression to calculate a profile likelihood for a . We first calculate the distance matrix among the 162 locations:

```
dst = as.matrix(dist(filipendula[, c("X", "Y")]))
```

Arbitrarily assuming a value of a of 10 m, the 1995 FoI on each location will be proportional to:

```
a = 10
foi = apply(exp(-dst/a) * filipendula$y94, 2, sum)
```

We use `glm` to evaluate the likelihood. The deviance of the `glm` object is 2 times the negative log-likelihood.

```
lfit=glm(y95~foi, family=binomial(), data=filipendula)
lfit$deviance/2
## [1] 69.8527
```

Figure 11.2 shows the likelihood profile across candidate values for a .

```
a=seq(1, 20, length=1001)
llik=rep(NA, length(a))
for(i in 1:length(a)){
  foi=apply(exp(-dst/a[i])*filipendula$y94, 2, sum)
  lfit=glm(y95~foi, family=binomial(),
            data=filipendula)
  llik[i]=lfit$deviance/2
}
plot(a, llik, type="l", ylab="Neg. log-like")
abline(h=min(llik)+qchisq(0.95, 1)/2)
```

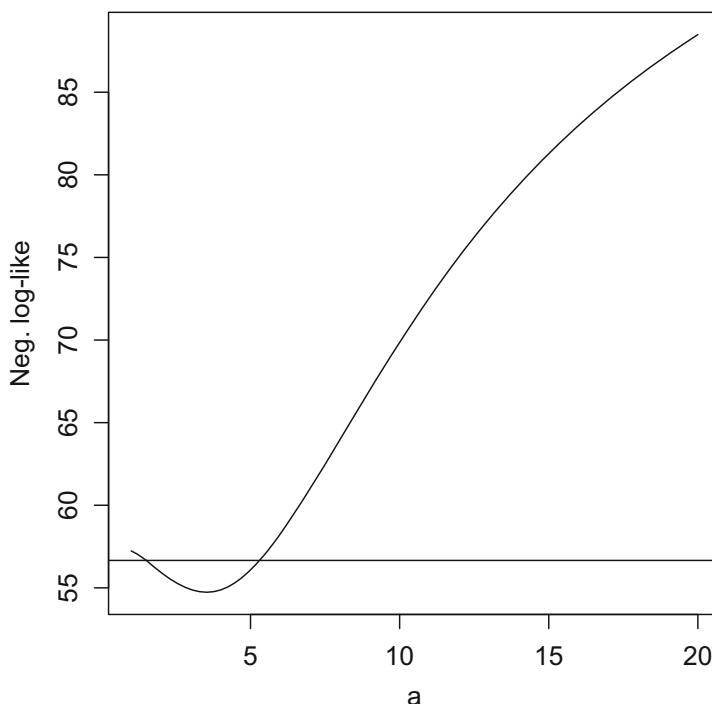


Fig. 11.2 Likelihood profile for a the parameter in the exponential dispersal kernel. The horizontal line represents the 95% cut off for the $\chi^2(1)/2$ deviation from the minimum

We can compare our best kernel model with a nonspatial model assuming a homogenous risk among hosts using likelihood-ratio tests (Sect. 8.4). Recall that for nested `glm`'s (i.e., where the simpler model is nested within the more complicated model), the difference in deviances ($= 2 \times \text{log-likelihood}$) is $\chi^2(df = \Delta p)$ -distributed,

where Δp is the number of extra parameters in the complex model. The `anova`-function provides this calculation in R. Since we first profiled on a , and then use the value \hat{a} that minimizes the negative log-likelihood, we have to correct the residual degrees of freedom of the spatial model to get the correct likelihood-ratio test.

```

ahat=a[which.min(llik)]
foi=apply(exp(-dst/ahat)*filipendula$y94,2,sum)
spmod=glm(y95~foi, family=binomial(), data=filipendula)
nullmod=glm(y95~1, family=binomial(), data=filipendula)
#correct the df of the spmod
spmod$df.residual=spmod$df.residual-1
anova(nullmod, spmod, test="Chisq")

## Analysis of Deviance Table
##
## Model 1: y95 ~ 1
## Model 2: y95 ~ foi
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       161     222.10
## 2       159     109.48  2    112.63 < 2.2e-16
##
## 1
## 2 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The spatial model gives a highly significantly better fit than the null model.

The Gaussian dispersal kernel takes the form $\propto \exp(-(d_{ij}/a)^2)$. We can estimate the parameters assuming this alternative kernel:

```

a2=seq(1,20, length=1001)
llik2=rep(NA, length(a2))
for(i in 1:length(a2)){
  foi2=apply(exp(-(dst/a2[i])^2)*filipendula$y94,2,sum)
  lfit2=glm(y95~foi2, family=binomial(),
             data=filipendula)
  llik2[i]=lfit2$deviance/2
}
ahat2=a2[which.min(llik2)]
foi2=apply(exp(-(dst/ahat2)^2)*filipendula$y94,2,sum)
spmod2=glm(y95~foi2, family=binomial(),
            data=filipendula)
spmod2$df.residual=spmod2$df.residual-1

```

Finally, we can visualize the shape of the competing probability kernels (using appropriate scaling for power exponential functions) (Fig. 11.3):

```
curve((2/(ahat2 * gamma(1/2))) * exp(-((x/ahat2)^2)),  
    0, 10, col = 2, lty = 2, ylab = "Probability",  
    xlab = "Meters")  
curve((1/(ahat) * gamma(1)) * exp(-x/ahat), 0, 10,  
    add = TRUE)  
legend("topright", c("Exponential", "Gaussian"),  
    lty = c(1, 2), col = c(1, 2))
```

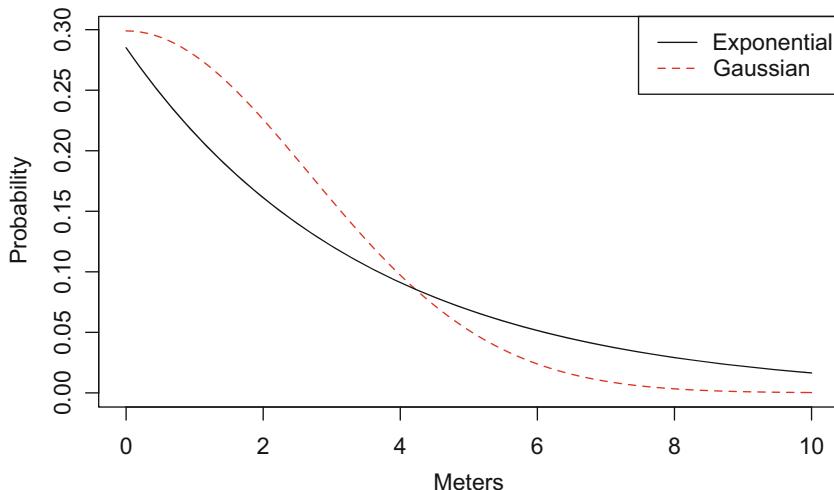


Fig. 11.3 The estimated exponential and Gaussian dispersal distance distributions for the *Filipendula* rust data

The two spatial models are not nested, but we can get model rankings using their AICs:

```
spmod$aic  
## [1] 113.4775  
spmod2$aic  
## [1] 116.6538
```

The exponential model is favored over the Gaussian.

11.3 Simulation

In addition to being a statistical method, our binomial spatial model also represents a fully specified metapopulation model for presence/absence of the rust.¹ Since we used logistic regression (the default for the binomial-family), our regression provides estimates for $\text{logit}(p) = \beta_0 + \beta_1 * \text{foi}$. The inverse-link is $p = \exp(\beta_0 + \beta_1 * \text{foi}) / (1 + \exp(\beta_0 + \beta_1 * \text{foi}))$.

We can write a simulator that stochastically projects the epidemic metapopulation forwards in time (assuming a fixed host plant distribution). We will initiate the simulation with the state of the system in 1995.

```
zprev = filipendula$y95
x = filipendula$x
y = filipendula$y
beta0 = spmod$coef[1]
beta1 = spmod$coef[2]
```

Infection probabilities for next year are:

```
foi = apply(exp(-dst/ahat) * zprev, 2, sum)
logitp = beta0 + beta1 * foi
p = exp(logitp) / (1 + exp(logitp))
```

A stochastic realization is:

```
znew = rbinom(162, 1, p)
symbols(x, y, circles = rep(1, 162), bg = znew +
  1, inches = 0.1, xlab = "X", ylab = "Y")
```

We can animate the next 100 years (if uncommented, the `Sys.sleep` argument makes the computer go to sleep for 0.1 s to help visualization):

```
simdat=matrix(NA, ncol=100, nrow=162)
for(i in 1:100){
  zprev=znew
  foi=apply(exp(-dst/ahat)*zprev,2,sum)
  logitp=beta0+beta1*foi
  p=exp(logitp)/(1+exp(logitp))
  znew=rbinom(162, 1, p)
  simdat[,i]=znew
  #symbols(x, y, circles=rep(1,162), bg=znew+1,
```

¹ Just like the chain-binomial model in Sects. 3.4 and 3.5 the spatial logistic model can be used both as a statistical method and a stochastic simulator.

```

# inches=.1, xlab="X", ylab="Y")
#Sys.sleep(0.1)
}

```

Figure 11.4 shows the predicted relative spatial risk from the stochastic simulation. The `spatial.plot`-function in the `ncf`-library is a wrapper for symbols that plots values larger (smaller) than the mean as red circles (black squares). In this case we see that spatial configuration alone can result in heterogeneous infection risk across the metapopulation. A corollary of this is that specialist plant pathogens may regulate the spatial distribution of host plant recruitment through locally density-dependent mortality and thus promote species diversity according to the Janzen-Connell hypothesis (e.g., Clark and Clark 1984; Petermann et al. 2008).

```

require(ncf)
mprev = apply(simdat, 1, mean)
spatial.plot(x, y, mprev, ctr = TRUE)

```

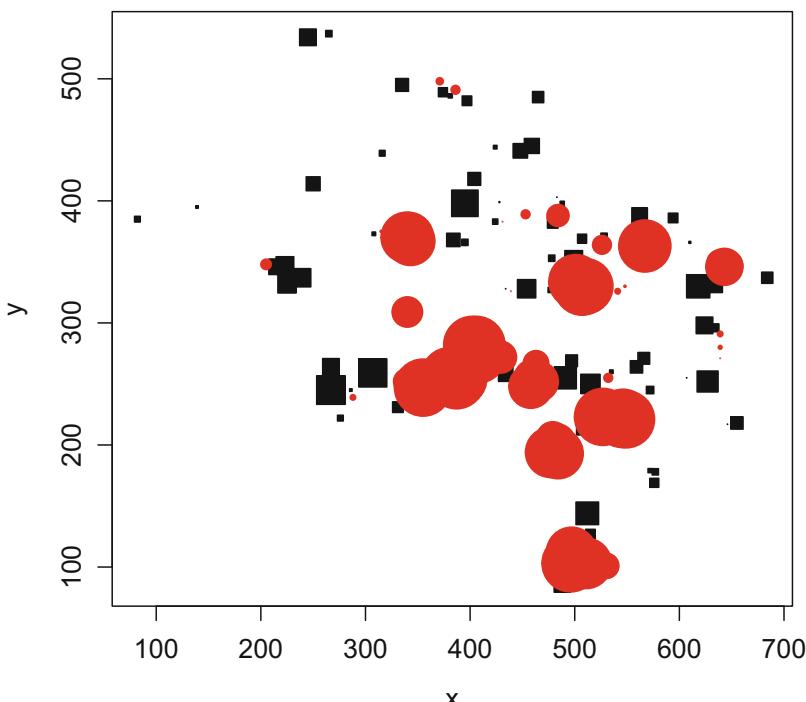


Fig. 11.4 Plot of predicted relative risk of rust infection from the metapopulation model. Risks larger (smaller) than the mean are shown as red circles (black squares). The size of the symbols reflects the deviation from the mean

11.4 Gypsy Moth

Various viruses and parasitoids of insects cause population instabilities and cycles in their hosts. The 5–10-year cycles in the gypsy moth (*Lymantria dispar*) are caused by the ldNPV-virus. Larvae get infected when ingesting viral occlusion bodies. The virus subsequently kills the larvae to release more of these infectious particles. USDA forest service conducts surveys each year of defoliation by the gypsy moth across the Northeastern USA to reveal complex spatiotemporal patterns. A web-optimized animated gif of the annual defoliation across the Northeastern USA between 1975 and 2002 can be viewed from <https://github.com/objornstad/epimdr/blob/master/mov/gm.gif>.

Spatiotemporal models can help to better understand such dynamics. There are specialized models for both the local and spatiotemporal dynamics of the gypsy moth (Dwyer et al. 2004; Abbott and Dwyer 2008; Bjørnstad et al. 2010). Here we will consider a simpler spatially extended SIR model.

11.5 Coupled Map Lattice SI Models

Coupled map lattice models² are constructed by assuming that spatiotemporal dynamics happens in two steps (Kaneko 1993; Bascompte and Solé 1995). First, local growth according to some model, for example, the seasonally forced (discrete time) SI model. Followed, second, by spatial redistribution of a fraction, m , of all individuals to other neighboring patches.

Because R is a vectorized language we can simulate CMLs using very compact code. We first write the function for the local SI dynamics according to the expectation from the chain-binomial formulation (Sect. 3.4). We assume a birth/death rate of μ and sinusoidal forcing on the transmission rate according to $\beta_0 + \beta_1 \cos(2 * \pi * t / 26)$ (so there are 26 time-steps in a year). We assume infected individual stays infected and infectious for one time step.

```
local.dyn = function(t, S, I, b0, b1, mu, N) {
  beta = b0 * (1 + b1 * cos(2 * pi * t/26))
  I = S * (1 - exp(-beta * I))
  S = (1 - mu) * S + mu * N - I
  list(S = S, I = I)
}
```

Next we construct the redistribution matrix among the $nx-by-ny$ locations (we consider a 30×30 lattice). Nearest-neighbors will be <1.5 spatial units apart (to

² The name refers to how the most stylized of these models assumes a lattice (checker board) of locations at which local numbers change from one generation to the next according to some “mapping”-rule such as the discrete logistic, the Nicholson-Baily model (see Chap. 14) or, in this case, a discrete-time seasonally forced SI model.

be exact $<\sqrt{3}$). Assume that the fraction that disperses to neighboring patches is $m = 0.25$ and that movement is random and independent of disease status.

```
m = 0.25
ny = nx = 30
# generate coordinates
xy = expand.grid(x = 1:nx, y = 1:ny)
# make distance matrix
dst = as.matrix(dist(xy))
# make redistribution matrix with zeros
redist = matrix(0, nrow = ny * nx, ncol = ny * nx)
# populate the matrix so each of the 8 neighbors
# gets their share
redist[dst < 1.5] = m/8
# the remaining fraction stays put
diag(redist) = 1 - m
```

The S and I matrices will hold the results from the simulation. We will run the model for $IT=520$ iterations ($= 20$ years). Assume that all patches have $S0 = 100$ susceptibles and that 1 infected is introduced in location {400, 1}:

```
IT = 520
S = I = matrix(NA, nrow = ny * nx, ncol = IT)
S[, 1] = 100
I[, 1] = 0
I[400, 1] = 1
```

We define the remaining parameters necessary for the local dynamics:

```
b0 = 0.04
b1 = 0.8
mu = 0.02/26
N = 1000
```

We are now ready to simulate the model. The `%*%`-operator represents matrix-multiplication and the matrix-multiplication of a vector of abundances with the redistribution-matrix moves all individuals appropriately.

```
for (t in 2:IT) {
  # local growth:
  tmp = local.dyn(t, S = S[, t - 1], I = I[, t -
  1], b0 = b0, b1 = b1, mu = mu, N = N)
  # spatial movement
```

```

S[, t] = redist %*% tmp$S
I[, t] = redist %*% tmp$I
# progress monitor
cat(t, " of ", IT, "\r")
}

```

The simulation can be visualized as an inline animation. The predicted incidence from the spatial SI-model varies so widely it is useful to transform incidence (using a fourth-root) so that low values show up better.

```

x = xy[, 1]
y = xy[, 2]
scIcubed = I^(1/4) / (max(I[, 10:IT])^(1/4))

```

```

for (k in 1:IT) {
  symbols(x, y, fg = 2, circles = scIcubed[, k],
          inches = FALSE, bg = 2, xlab = "", ylab = "")
  Sys.sleep(0.05)
}

```

Analyses of a variety of host-parasit(oid) CML models (Hassell et al. 1991; Bjørnstad et al. 1999b; Earn et al. 2000a) have revealed a variety of emergent spatiotemporal patterns including complete synchrony, waves, spatial chaos, and frozen patterns. The pattern in any given system depends on the local dynamics and mobility. We will visit on these CML models further in Chap. 14.

11.6 Making Movies

We can make permanent movies by writing the plots to a sequence of jpeg's and then use an open-source utility like [ImageMagick](#) to convert the sequence to a movie.³

```

for(k in 100:IT){
  png(filename=paste("m",1000+k,".jpg", sep=""))
  symbols(x, y, fg=2, circles=scIcubed[,k],
          inches=FALSE, bg=2,xlab="",ylab="")
  dev.off()
}

```

³ The `system()`-function in R passes the `convert` and `xm` calls to the command-line. A web-optimized version of the animated gif can be viewed on <https://github.com/objornstad/epimdr/blob/master/mov/simovie.gif>.

```
system("convert m*.jpg simovie.gif")
system("rm m*.png")
#For mp4-animation:
#system("convert -delay 5 m*.jpg simovie.mp4")
```

Alternatively we can incorporate the animation directly into a pdf—though for this to work we need to work with [LaTeX](#) and use the LaTeX animate-package.

```
require("animation")
oopt = ani.options(interval = 0.02, nmax = 100)
test.function = function (xy, I, nmax) {
  x = xy[,1]
  y = xy[,2]
  scIcubed = I^(1/4)/(max(I[,10:IT]^(1/4)))
  for (i in seq_len(ani.options("nmax"))) {
    dev.hold()
    symbols(x,y,fg=2,circles=I[,i],inches=0.1,bg=2,
            xlab="",ylab="")
    ani.pause()
  }
}

saveLatex({
  test.function(xy=xy, I=I, nmax=50)
},
  ani.basename = "BM", ani.opts = "controls,loop",
  width=0.8\textwidth", ani.first =
  par(mar = c(3, 3, 1, 0.5), mgp = c(2, 0.5, 0),
  tcl = -0.3, cex.axis = 0.8, cex.lab = 0.8,
  cex.main = 1), latex.filename = "test.tex",
  pdflatex = "/usr/texbin/pdflatex",
  img.name = "Xplot")
ani.options(oopt)
```

11.7 Nonparametric Covariance Functions for Spatiotemporal Data

Keeling et al. (2002) discuss how we may understand the emergent complicated spatiotemporal dynamics of models of natural enemies in terms of the spatial variance (or associated autocorrelation) and covariance of the interacting species.⁴ Bjørnstad and Bascompte (2001) proposed to calculate auto- and cross-correlation functions

⁴ Seabloom et al. (2005) provide similar calculations for spatial competition models.

from simulated or real data. We can use the `Sncf`-function in the `ncf`-package to calculate the “multivariate” spatial correlation function (Bjørnstad et al. 1999b) among the simulated time series (see Chap. 13 for further details on this and other geostatistical methods). We can further look at the spatial *cross-correlation* function between susceptibles and infected (Fig. 11.5). The background synchrony for both compartments (of around 0.3) is due to the common seasonal forcing. The locally higher autocorrelation at shorter distances is due to emergence of dispersal-induced aggregations of infected individuals. The negative local cross-correlation is due to the local S-I cycles.

```
fitI = Sncf(x = xy[, 1], y = xy[, 2], z = sqrt(I[, 261:520]), resamp = 500)
fitsS = Sncf(x = xy[, 1], y = xy[, 2], z = sqrt(S[, 261:520]), resamp = 500)
fitSI = Sncf(x = xy[, 1], y = xy[, 2], z = sqrt(S[, 261:520]), w = sqrt(I[, 261:520])), resamp = 500
par(mfrow = c(1, 3))
plot(fitI, ylim = c(-0.1, 1))
plot(fitsS, ylim = c(-0.1, 1))
plot(fitSI, ylim = c(-0.2, 0.2))
```

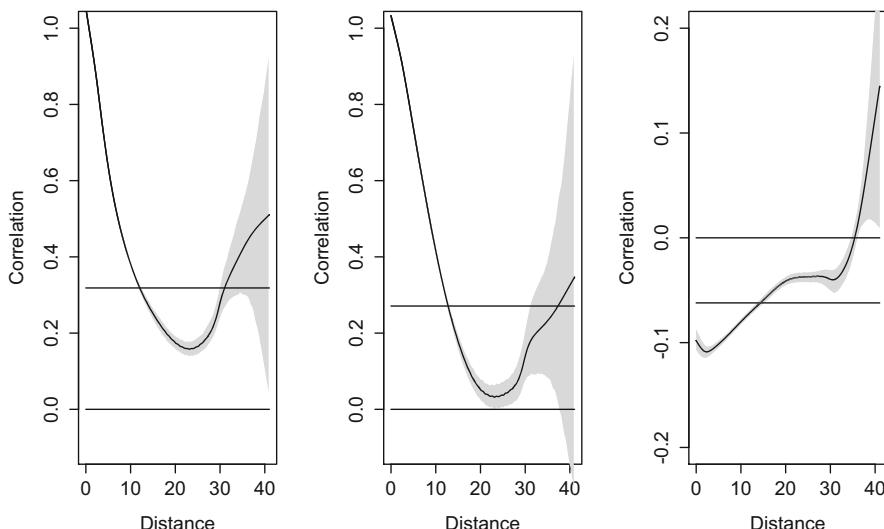


Fig. 11.5 Spatial correlation (1) infecteds, (2) susceptibles, and (3) S-I cross-correlation as a function of distance

One interesting additional application is the so-called time-lagged spatial correlation function (Bjørnstad et al. 2002a). This analysis may help quantify wave-like spread. For example we can look at the spatiotemporal relationship between the infecteds and themselves 5 time-steps later (Fig. 11.6). The peak in correlation is offset from the origin by somewhere between 5 and 10 units. This makes sense, since we assume nearest neighbor dispersal, so the leading edge should move 5 units vertically/horizontally and $5 * \sqrt{2} = 7.1$ units diagonally during 5 time steps.

```
fitIlag = Sncf(x = xy[, 1], y = xy[, 2], z = I[, 261:515], w = I[, 266:520], resamp = 100)
plot(fitIlag, ylim = c(-0.2, 0.2))
```

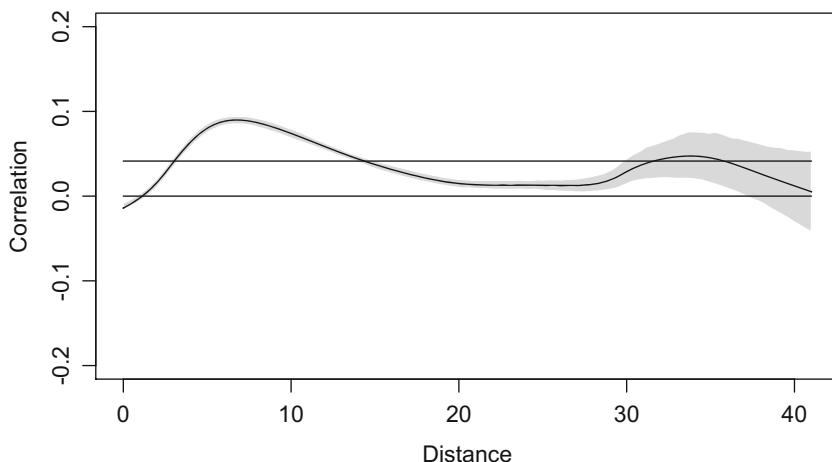


Fig. 11.6 The time-lagged spatial cross-correlation function of predicted prevalence of the SI cml model (with a 5-year time lag)

Bjørnstad et al. (2002b) used time-lagged spatial correlation functions to show that parasitoid-host interactions (see Chap. 14) lead to waves of larch tree defoliation that travels at 210 km per year in a north-easterly direction across the European Alps. Traveling waves have also been documented in the dynamics of dengue (Cummings et al. 2004) and influenza A (Gog et al. 2014).

11.8 Gravity Models

Regional spread of human pathogens rarely forms a simple diffusive pattern because human mobility patterns are more complex—movement may be distant dependent, but overall flow between any two communities also typically depend on the size

(and desirability) of both “donor” and “recipient” location (Erlander and Stewart 1990; Fotheringham 1984). Grenfell et al. (2001), for example, showed that the spatiotemporal dynamics of measles across all cities and villages in pre-vaccination England and Wales exhibited “hierarchical waves,” in which the timing of epidemics relative to the big urban conurbations (the donors) depended negatively on distance but positively on the size of the recipient. Viboud et al. (2006) demonstrated similar hierarchical spread of seasonal influenza across the states of continental USA.

Xia et al. (2004) and Viboud et al. (2006) subsequently showed that a metapopulation model where movement among communities followed a “generalized gravity model” approximates the dynamic patterns; The “gravity model” is a model of mobility/transportation from transportation science that posits that transportation volume between two communities depends inversely on distance, d , but bilinearly on the size, N , of the communities (Erlander and Stewart 1990; Fotheringham 1984). Gravity-like models have since been applied to study the spatial dynamics of a variety of human infection settings (e.g., Mari et al. 2012; Truscott and Ferguson 2012; Gog et al. 2014).

The generalized gravity model quantifying the spatial interaction between locations i and j (commonly) take the form $\theta N_i^{\tau_1} N_j^{\tau_2} d_{ij}^{-\rho}$, where θ , τ_1 , τ_2 , and ρ are nonnegative parameters shaping the topology of the spatial interaction network. The gravity model has at least two important special cases: $\rho = 0$, $\tau_1 = \tau_2 = 1$ representing a mean field model and $\tau_1 = \tau_2 = 0$ representing simple spatial diffusion.

Viboud et al. (2006) proposed a stochastic multipatch SIR model for the spread of seasonal influenza among the states of the continental USA. We will consider a simpler SIR version of the model (ignoring susceptible recruitment)⁵:

$$\frac{dS_i}{dt} = -(\beta I_i + \sum_{j \neq i} \iota_{j,i} I_j) S_i \quad (11.1)$$

$$\frac{dI_i}{dt} = (\beta I_i + \sum_{j \neq i} \iota_{j,i} I_j) S_i - \gamma I_i \quad (11.2)$$

$$\frac{dR_i}{dt} = \gamma I_i, \quad (11.3)$$

where $\iota_{j,i} I_j$ is the gravity-weighted force of infection exerted by state j on state i . The corresponding R-code is:

```
require(deSolve)
SIR.space = function(t, y, pars) {
  i = c(1:L)
```

⁵ Note that we assume that spatial transmission does not dilute local transmission. Keeling and Rohani (2002) provide a discussion of this issue.

```

S = y[i]
I = y[L + i]
R = y[2 * L + i]
with(pars, {
  beta = beta[i]
  dS = -(beta * I + m * G %% I) * S
  dI = (beta * I + m * G %% I) * S - gamma *
    I
  dR = gamma * I
  list(c(dS, dI, dR))
})
}

```

G is the spatial interaction matrix and m is a scaling factor. Combining state-level ILI-data with county-level commuter census data, Viboud et al. (2006) estimated the gravity parameters to be $\tau_1 = 0.3$, $\tau_2 = 0.6$, and $\rho = 3$.⁶ The `usflu` data contains coordinates and populations for each of the contiguous lower 48 states plus the District of Columbia. The `gdist`-function of the `ncf`-package generates spatial distance matrices from latitude/longitude data:

```

require(ncf)
data(usflu)
usdist = gdist(usflu$Longitude, usflu$Latitude)

```

We define a function to generate the spatial interaction matrix given parameters and distances:

```

gravity = function(tau1, tau2, phi, pop, distance) {
  gravity = outer(pop^tau1, pop^tau2)/distance^phi
  diag(gravity) = 0
  gravity
}
G = gravity(0.3, 0.6, 3, usflu$Pop, usdist)

```

We finally define initial conditions and parameters (scaling β such that R_0 will be the same in all states). Viboud et al. (2006) were interested in exploring spread in a pandemic setting. We therefore assume that everybody is susceptible, with 1 initial index case arriving in New York:

```

gamma = 1/3.5
R0 = 1.8
beta = R0 * gamma/usflu$Pop

```

⁶ Viboud et al. (2006) showed that the commuter flows has a heavier tail than predicted by this gravity model which we, for expedience, ignore.

```
m = 1/1000/sum(usflu$Pop)
parms = list(beta = beta, m = m, gamma = gamma, G = G)
L = length(usflu$Pop)

S = usflu$Pop
R = I = rep(0, length(usflu$Pop))
I[31] = 1
inits = c(S = S, I = I, R = R)
```

We are now set to simulate a spatial SIR pandemic across the USA (Fig. 11.7):

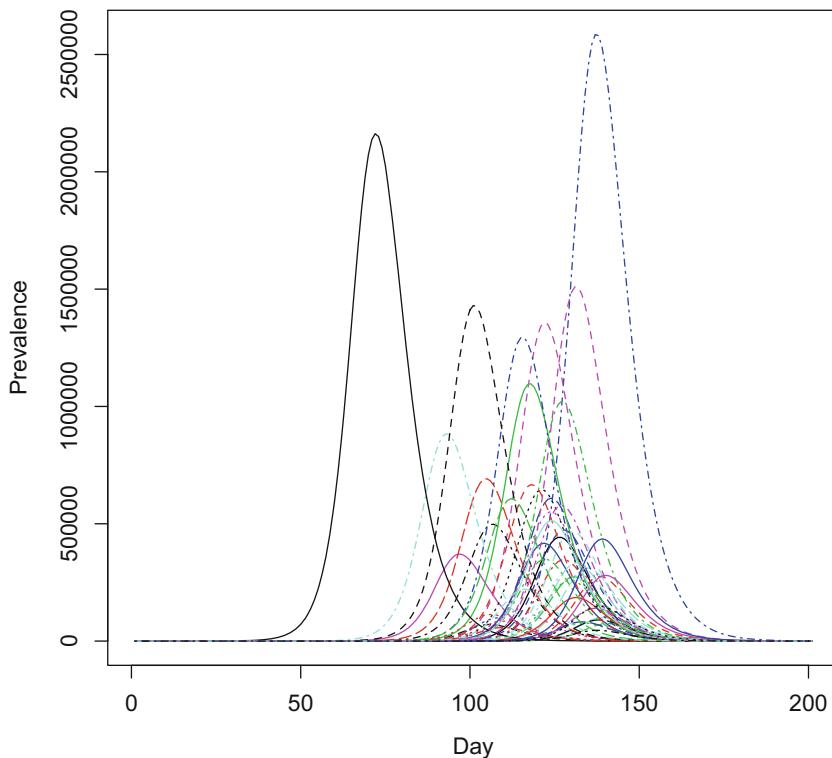


Fig. 11.7 Simulated influenza dynamics across the continental USA using a multipatch SIR model with gravity coupling parameterized according to Viboud et al. (2006)

```
require(deSolve)
times = 0:200
out = ode(inits, times, SIR.space, parms)
matplot(out[,50+(1:L)], type="l", ylab="Prevalence",
        xlab="Day")
```

The outbreak peaks are predicted to be staggered because of the spatial diffusion of the infection across the continent.

Chapter 12

Transmission on Networks



12.1 S Preamble: Objects, Classes, and Functions

The S-language which is the foundation of R was constructed using an “object”-based logic where each object is assigned a “class.” The class, in turn, controls printing, plotting, and summarizing each object. There are many excellent introductions to S programming (e.g., Venables and Ripley 2013), in this chapter we will use S3-class programming to streamline our analysis of epidemics on networks. The basic idea is this: if we label the result of some calculation as class `foo`, then R will look for functions `print.foo()`, `summary.foo()`, and `plot.foo()` in the search-path when further interacting with the result of the calculation. Let’s illustrate with a silly example:

```
foo = function(x) {  
  res = x  
  class(res) = "foo"  
  return(res)  
}  
  
print.foo = function(obj) {  
  cat("foo is:\n", obj)  
}
```

This chapter uses the following R-package: `statnet`.

A conceptual understanding of *social networks* is useful prior to this discussion. Five-minute epidemics-MOOC intros can be watched from YouTube:

Structure of networks: <https://www.youtube.com/watch?v=hLwasjKxFoc>

Networks and control <https://www.youtube.com/watch?v=GBQqhtGAzGc>.

```
summary.foo = function(obj) {
  cat("In summary, foo is:\n", obj)
}

plot.foo = function(obj) {
  plot(NA, type = "n", ylim = c(0, 1), xlim = c(0,
  1), ylab = "")
  text(x = seq(0.1, 0.9, by = 0.1), y = seq(0.1,
  0.9, by = 0.1), as.character(obj))
}
```

The result is a fully functional S3-class of R objects:

```
zz = foo("pibble")
```

which we can print,

```
zz
## foo is:
## pibble
```

summarize,

```
summary(zz)
## In summary, foo is:
## pibble
```

and plot (Fig. 12.1):

```
plot(zz)
```

And that is the basics of S3-class programming ...

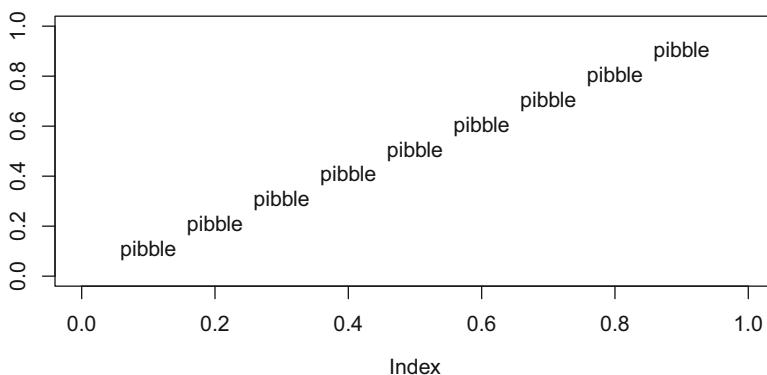


Fig. 12.1 A plot of objects of class `foo`

12.2 Networks

Transmission on social networks bears conceptual similarities to spatial transmission. The only difference being that in spatial models transmission occurs among neighbors in space, and transmission on networks occurs among neighbors in social space. We can thus use the type of compact code we used for CML models (Sect. 11.5) to simulate epidemics on networks. A key determinant of invasibility and speed of spread is the average and variance in the number of contacts on the networks (Newman 2002; Keeling and Eames 2005; Bansal et al. 2007). As we saw in the network of spread of gonorrhea (Sect. 3.8.3), there is often substantial variation in the number of sexual partners. Section 4.7 further highlighted age-specific variation in contact rates. It is easiest to consider static networks (networks for which contact patterns do not change over time) for which the contact distribution is characterized by the “degree distribution,” contacts are mapped onto “edges,” and individuals are “nodes.”

12.3 Models of Networks

In the previous spatial coupled map lattice models, transmission was restricted to the eight nearest neighbors on the lattice, so we can think of this as an example of a network with fixed degree of 8. In network theory, an analogous fixed-degree network is constructed as a ring lattice. The associated matrix that flags neighbors is a particular type of [Toeplitz matrix](#). We can define a `ringlattice`-function to generate such networks with N nodes and $2 \times K$ degrees. We label the result to be of class `cm` (short for contact matrix):

```
ringlattice = function(N, K) {
  # N is the number of nodes K is the number of
  # neighbors on each side to which each node is
  # connected so degree = 2xK
  CM = toeplitz(c(0, rep(1, K), rep(0, N - 2 *
    K - 1), rep(1, K)))
  class(CM) = "cm"
  return(CM)
}
```

Trigonometry provides a basic way to visualize a ring network...or any other object that is defined as class `cm`.

```
plot.cm=function(CM) {
  N=dim(CM)[1]
  theta=seq(0,2*pi,length=N+1)
  x=cos(theta[1:N])
  y=sin(theta[1:N])
  symbols(x,y, fg=0, circles=rep(1, N),
           inches=0.1, bg=1, xlab="", ylab="")
  segx1=as.vector(matrix(x, ncol=length(x),
                         nrow=length(x), byrow=TRUE))
  segx2=as.vector(matrix(x, ncol=length(x),
                         nrow=length(x), byrow=FALSE))
  segy1=as.vector(matrix(y, ncol=length(x),
                         nrow=length(x), byrow=TRUE))
  segy2=as.vector(matrix(y, ncol=length(x),
                         nrow=length(x), byrow=FALSE))
  segments(segx1,segy1, segx2,
           segy2, lty=as.vector(CM))
}
```

Figure 12.2 depicts a ring-lattice with 20 individuals and a fixed-degree of eight.

```
cm = ringlattice(N = 20, K = 4)
plot(cm)
```

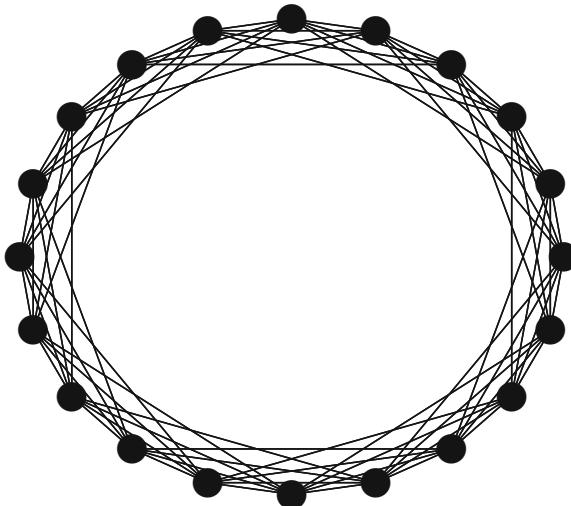


Fig. 12.2 A 20-node ring-lattice of degree 8

12.3.1 Watts-Strogatz Networks

Real social networks have heterogeneities in contact rates and usually exhibit much lower social separation than predicted by the ring lattice. In the study of small-world networks, [Watts and Strogatz](#) (Watts and Strogatz 1998) proposed an algorithm for generating more realistic networks by randomly rewiring a fraction Prw of the edges of a ring lattice.

```
WattsStrogatz=function(N, K, Prw) {
  # Build a Watts-Strogatz contact matrix from
  # a ring lattice, Prw is the rewiring probability
  CM=ringlattice(N=N, K=K)
  CMWS=CM
  tri=CM[upper.tri(CM)]
  Br=rbinom(length(tri), 1, Prw) # Break edges
  a=0
  for(i in 1:(N-1)){
    for(j in (i+1):N){
      a=a+1
      if(Br[a]==1 & CMWS[i,j]==1){ #If "Br == 1"
        CMWS[i,j]=CMWS[j,i]=0 # break edge
        tmp=i
        tmp2=c(i, which(CMWS[i,]==1))
        #new edge, if already present try again
        while(any(tmp2==tmp)){
          tmp=ceiling(N*runif(1))
          CMWS[i,tmp]=CMWS[tmp,i]=1 # make new edge
        }
      }
    }
  }
  class(CMWS)="cm"
  return(CMWS)
}
```

Figure 12.3 depicts a Watts-Strogatz network with 20 individuals, a mean degree of 8, and a rewiring probability of 0.3.

```
cm2 = WattsStrogatz(N = 20, K = 4, Prw = 0.3)
plot(cm2)
```

We can extend the notion of writing generic functions for class `cm`-objects, to define a `summary`-function that calculates and optionally plots (Fig. 12.4) the degree distribution.

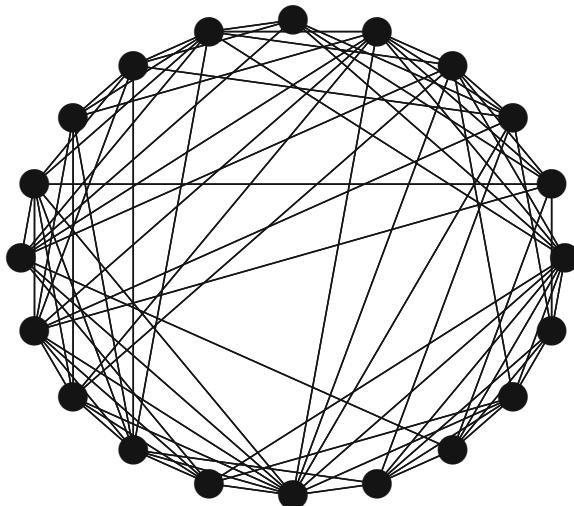


Fig. 12.3 A Watts-Strogatz network with 20 individuals, mean degree of 8, and a rewiring probability of 0.3

```

summary.cm = function(x, plot = FALSE) {
  x = table(apply(x, 2, sum))
  res = data.frame(n = x)
  names(res) = c("degree", "freq")
  if (plot)
    barplot(x, xlab = "degree")
  return(res)
}
summary(cm2, plot = TRUE)

##   degree freq
## 1      5    1
## 2      6    2
## 3      7    3
## 4      8    6
## 5      9    6
## 6     10    2

```

The Watts-Strogatz model scales the degree distribution from fixed to the “random graph”—when the rewiring probability is set to 1—which has a Poisson-distributed degree distribution. The random graph corresponds to the [Erdos-Renyi model](#) (Erdös and Rényi 1959).

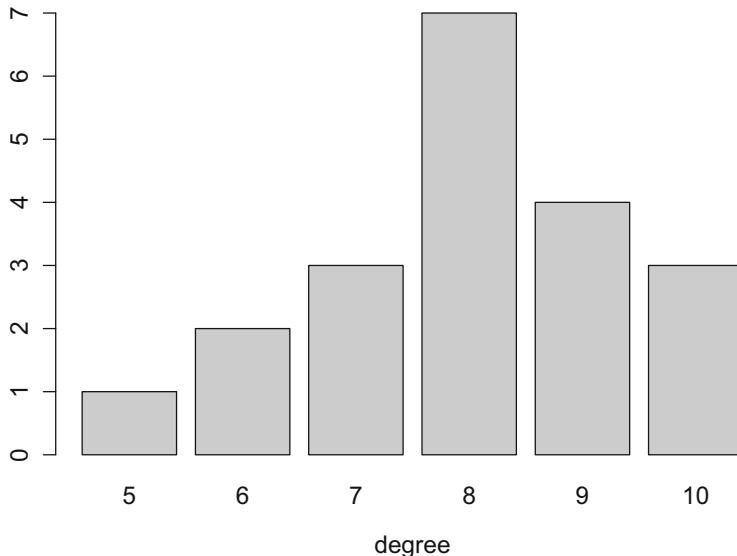


Fig. 12.4 The degree distribution of the Watts-Strogatz network with 20 individuals, a mean degree of 8, and a rewiring probability of 0.3 as generated by `summary.cm(..., plot=TRUE)`

12.3.2 Barabasi-Albert Networks

The Watts-Strogatz model can at most have Poisson-like variance in degree distribution, so it cannot mimic heavy-tailed distributions seen in “scale-free” networks. [Barabasi and Albert \(1999\)](#) proposed that such behavior arises from preferential attachment (“rich-get-richer”) dynamics. We can write a function that generates a network with N -nodes and mean degree $2 \times K$. The log-log plot (Fig. 12.5) shows the power-law heterogeneity in contacts predicted by the Barabasi-Albert algorithm.

```
BarabasiAlbert=function(N, K){
  CM=matrix(0, ncol=N, nrow=N)
  CM[1,2]=1
  CM[2,1]=1
  for(i in 3:N) {
    probs=apply(CM, 1, sum)
    link=unique(sample(c(1:N)[-i],
                      size=min(c(K, i-1)), prob=probs[-i]))
    CM[i, link]=CM[link, i]=1
  }
  class(CM)="cm"
  return(CM)
}
```

```
cm3 = BarabasiAlbert(200, 4)
ed = summary(cm3)
plot(as.numeric(ed$degree), ed$freq, log = "xy",
     xlab = "Degree", ylab = "Frequency")
```

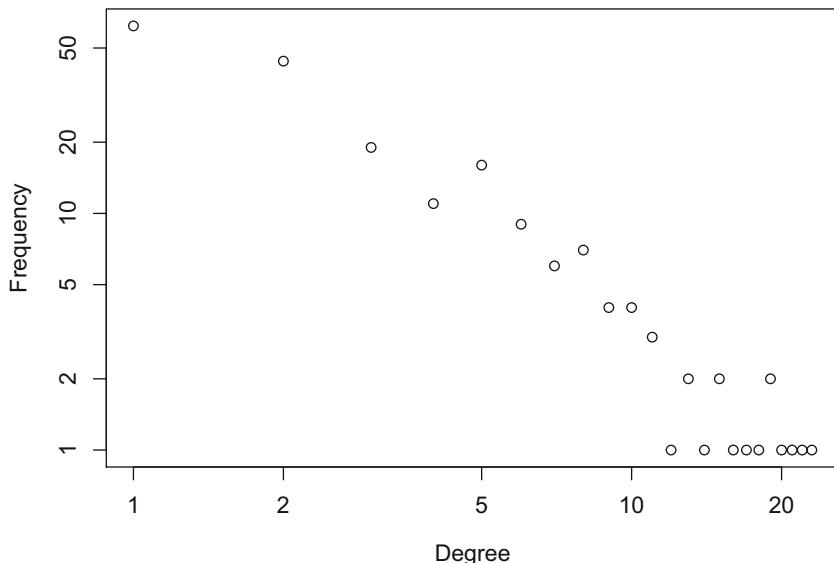


Fig. 12.5 A log-log plot of degree-distribution from a Barabasi-Albert network

For fancier visualization of networks we can use the plotting functions in the `statnet`-package (Fig. 12.6). The `network`-function in the `statnet`-package converts the contact matrix (of class CM) to a network-class object.

```
require(statnet)
plot(network(cm3, directed = FALSE))
```

12.4 Epidemics on Networks

We can run SIR-like epidemics across networks by assuming that an infection is transmitted across an S-I edge with a probability, τ , per time step (e.g., Barbour and Mollison 1990; Ferrari et al. 2006a). Following, the Reed-Frost version of the chain-binomial model (Abbey 1952), the probability of any given susceptible becoming infected is $p = 1 - (1 - \tau)^y$ where y is the number of infected neighbors. We may further assume infecteds are removed with a constant probability, γ , leading to a geometrically distributed infectious period. Spread of infections on networks depends

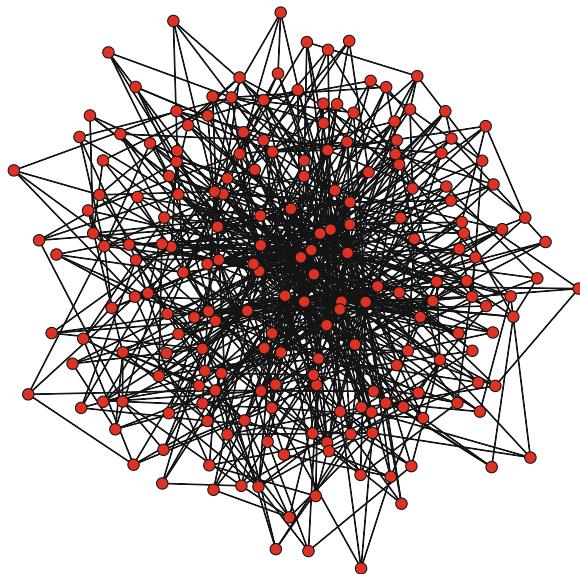


Fig. 12.6 Visualizing the Barabasi-Albert network using the `statnet`-package

on both mean number of contacts \bar{k} and heterogeneity in that number (quantified by $\overline{k^2}$) according to $R_0 = (\tau/(\tau + \gamma))(\bar{k}^2 - \bar{k})/\bar{k}$ (e.g., Bansal et al. 2007).

The `NetworkSIR` function will simulate a closed SIR epidemic on arbitrary contact matrices and return an object of class `netsIR`.

```
NetworkSIR=function(CM,tau,gamma){
  #generate SIR  epidemic on a CM-network
  #CM = contact matrix
  #tau = probability of infection across an edge
  #gamma = probability of removal per time step
  N=dim(CM)[1]
  I=matrix(rep(0,N),nrow=N,ncol=1) #First infecteds
  S=matrix(rep(1,N),nrow=N,ncol=1) #First susceptibles
  R=matrix(rep(0,N),nrow=N,ncol=1) #First removed
  I1=sample(1:N, size=1) #Pick first random infected
  I[I1,1]=1
  S[I1,1]=0
  t=1
  while(sum(I[,t-1])>0 | t==1) {
    t=t+1
    infneigh=CM%*%I[,t-1]
```

```

pinf=1-(1-tau)^infneigh
newI=rbinom(N, S[,t-1], pinf)
newR=rbinom(N, I[,t-1], gamma)
nextS=S[,t-1]-newI
nextI=I[,t-1]+newI-newR
nextR=R[,t-1]+newR
I=cbind(I, nextI)
S=cbind(S, nextS)
R=cbind(R, nextR)
}
res=list(I=I, S=S, R=R)
class(res)="netSIR"
return(res)
}

```

We can define `summary`- and `plot`-functions for the `netSIR` class.

```

summary.netSIR=function(x) {
  t=dim(x$S) [2]
  S=apply(x$S, 2, sum)
  I=apply(x$I, 2, sum)
  R=apply(x$R, 2, sum)
  res=data.frame(S=S, I=I, R=R)
  return(res)
}

plot.netSIR=function(x) {
  y=summary(x)
  plot(y$S, type="b", xlab="time", ylab="")
  lines(y$I, type="b", col="red")
  lines(y$R, type="b", col="blue")
  legend("right", legend=c("S", "I", "R"),
    lty=c(1,1,1), pch=c(1,1,1),
    col=c("black", "red", "blue"))
}

```

Figure 12.7 shows epidemic spread on (1) scale-free, (2) Watts-Strogatz, (3) Poisson, and (4) ring lattice networks.

```

cm1=BarabasiAlbert(N=200, K=2)  # (i)
cm2=WattsStrogatz(N=200, K=2, Prw=.1) # (ii)
cm3=WattsStrogatz(N=200, K=2, Prw=1) # (iii)
cm4=ringlattice(N=200, K=2)      # (iv)
sim1=NetworkSIR(cm1, .3, 0.1)
sim2=NetworkSIR(cm2, .3, 0.1)

```

```

sim3=NetworkSIR(cm3,.3,0.1)
sim4=NetworkSIR(cm4,.3,0.1)
plot(apply(sim1$I,2,sum), type="l", xlab="Time",
      ylab="Infected")
lines(apply(sim2$I,2,sum), type="l", col="red")
lines(apply(sim3$I,2,sum), type="l", col="red", lty=2)
lines(apply(sim4$I,2,sum), type="l", col="blue")
legend("topright", legend=c("Scale-free", "WS(0.1)",
    "Poisson", "Lattice"), lty=c(1,1, 2, 1),
    col=c("black", "red", "red", "blue"))

```

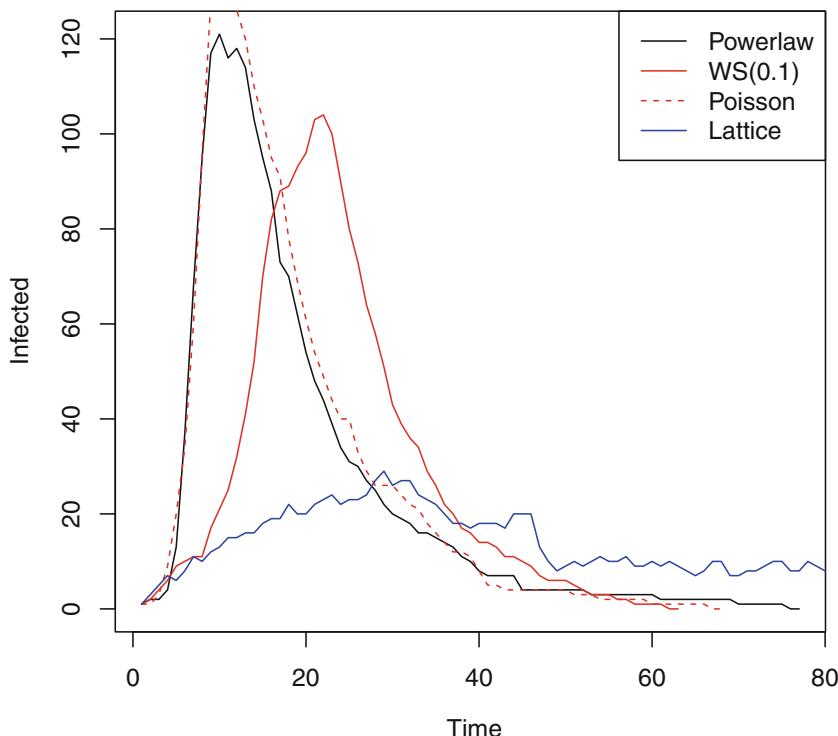


Fig. 12.7 Simulated closed epidemics on (1) scale-free, (2) Watts-Strogatz, (3) Poisson, and (4) lattice models. All with a mean degree of 4

The difference in spread can be understood in terms of how network geometry molds R_0 even when all else (including the mean number of contacts) is constant (Bansal et al. 2007). The `r0fun`-function calculates R_0 for any given network and apply it to each simulated networks. The greater the heterogeneity, the greater the R_0 :

```
r0fun=function(CM, tau, gamma){
x=apply(CM, 2, sum)
(tau/(tau+gamma) * (mean(x^2) - (mean(x))) / mean(x)
}
r0fun(cm1, 0.3, 0.1)

## [1] 4.79471

r0fun(cm2, 0.3, 0.1)

## [1] 2.30625

r0fun(cm3, 0.3, 0.1)

## [1] 2.77125

r0fun(cm4, 0.3, 0.1)

## [1] 2.25
```

We can combine the functionality of the `statnet`-package with the above results on network heterogeneity to revisit on the gonorrhea contact tracing study of De et al. (2004) from Sect. 3.8.3.

```
data(gonnet)
nwt = network(gonnet, directed = TRUE)
x = degree(nwt)[2:89]
mean(x)

## [1] 1.920455
```

The mean degree is 1.92, but the inflation factor due to the network heterogeneity is predicted to almost double the R_0 of a STD spreading across this network:

```
(mean(x^2) - (mean(x))) / mean(x)

## [1] 1.940828
```

To simulate an epidemic on the empirical contact-tracing study from Sect. 3.8.3, we first have to construct an undirected contact network among the 89 members and next apply the `NetworkSIR` model to plot the time trajectory and final infection status of the network:

```
#Undirected network
cmg=gonnet+t(gonnet)
#Simulate epidemic
cep=NetworkSIR(cmg, 0.3, 0.1)
sm=summary(cep)
par(mfrow=c(1,2))
```

```

inf=ifelse(apply(cep$I,1,sum)>0,2,1)
nwt=network(cmg, directed=FALSE)
plot(nwt, vertex.col=inf)
matplot(sm, ylab="Numbers")
legend("right", c("S", "I", "R"),
      pch=c("1", "2", "3"), col=c(1,2,3))

```

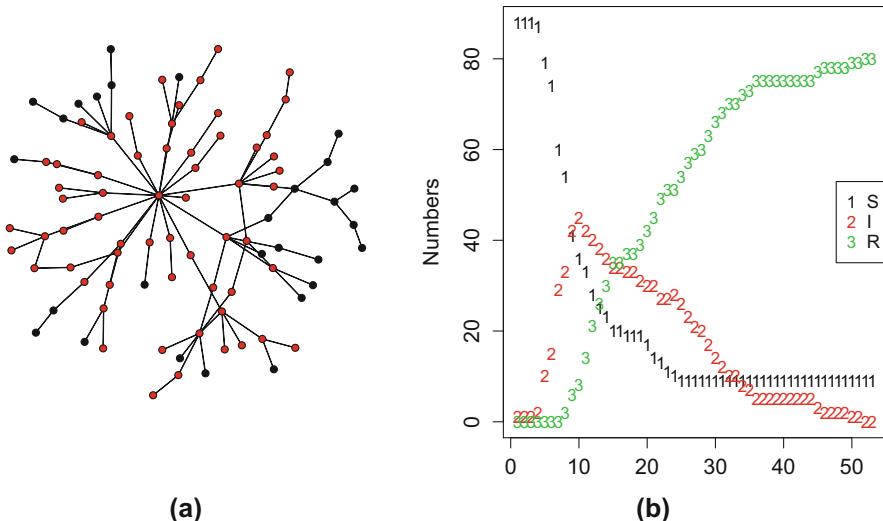


Fig. 12.8 A simulated closed epidemics on the gonorrhea contact-tracing network. (a) Network infection history (red nodes = infected, black nodes = escapees). (b) Outbreak trajectory

The infection history of the network (Fig. 12.8) reveals the feature that the core-group of the network is likely to always be infected, but peripheral individuals may escape infection by getting surrounded by immune individuals before getting infected. Ferrari et al. (2006a) discuss how the geometry of a network shapes the likelihood of a given individual escaping infection.

Models of networks and epidemics on networks is a vast literature, so the above should at best be considered a teaser. The statnet project and associated statnet-package have a rich set of resources for network analysis and modeling epidemics on networks including how to generate dynamic networks.

Chapter 13

Spatial and Spatiotemporal Patterns



13.1 Introduction

Spatial and spatiotemporal data analysis is of great importance in disease dynamics for a number of reasons such as looking for space-time clustering, hot-spot detection, characterizing invasion waves, and quantifying spatial synchrony. Spatial synchrony—the level of correlation in outbreak dynamics at different locations—is of particular significance to acute immunizing infections, because asynchrony may permit regional persistence of infections despite local chains-of-transmission breaking during post-epidemic troughs (Keeling et al. 2004). Conversely, spatial synchrony can exacerbate the economic and public health burden because the resulting regionalized outbreaks can overwhelm logistical capabilities as was evident in the early part of the 2013–2014 West African ebola outbreak.

Spatial statistics is also important in order to correct for the problem of spurious associations between incidence and environmental data because spatial autocorrelation violates the assumption of independence. We will discuss this in Sect. 15.2.

13.2 A Plant-Pathogen Case Study

Jennifer Koslow carried out an experiment with a foliar, nonsystemic rust (*Coleosporium asterum*) infecting the flat-top goldenrod (*Euthamia graminifolia*). The `gra` data present the severity of disease expression (`$score`, from 0 to 10) on host-plants planted within mesocosms (`$plot`) in an old field near Ithaca, NY, USA. The mesocosms were in a checkerboard grid with locations specified by

This chapter uses the following R-package: `ncf`.

coordinates `$xloc` and `$yloc`. Each mesocosm contained three focal *E. graminifolia* plants. The field also contained naturally occurring *E. graminifolia*, as well as several other hosts of the rust notably the Canada goldenrod (*Solidago canadensis*). Two different treatments, species composition (`$comp`, with three levels) and watering treatment (`$water`, with two levels), were applied to the mesocosms in a fully factorial design. Finally, to account for spatial variation across the field, there were four blocks with treatment combinations randomly assigned within each block.

We have to jitter the coordinates for some of the analyses because the three plants within each plot were not given separate coordinates. Figure 13.1 the spatial layout of the study. The vertical lines mark the blocks.

```
data(gra)
gra$jx=jitter(gra$xloc)
gra$jy=jitter(gra$yloc)
symbols(y=gra$xloc, x=gra$yloc, circles=gra$score,
        inches=0.1, xlab="y", ylab="x")
abline(v=47.5,col=2)
abline(v=97.5,col=2)
abline(v=147.5,col=2)
```

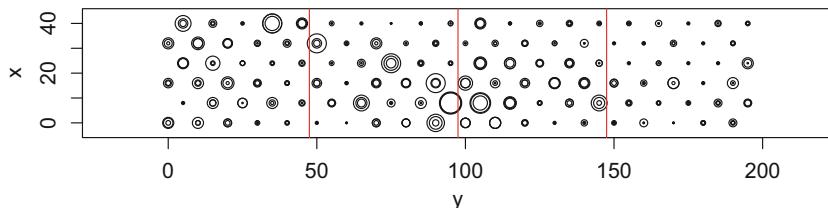


Fig. 13.1 Rust scores from Keslow's experiment

13.3 Spatial Autocorrelation

Spatial statistics is a very rich field. We will focus on a subset of methods that are more (or less) commonly used in disease ecology. Many of these involve the notion of spatial autocorrelation in one form or another. Legendre (1993) is a great introduction to the use of spatial autocorrelation in ecological studies in general. While all the methods we will be discussing—such as Mantel tests, parametric and nonparametric correlation functions, local indicators of spatial association, etc.—come in canned packages (this chapter uses the `ncf`-package), it is useful to spend a bit of time on the underlying ideas.

Many geostatistical methods to describe spatial pattern are focused on either spatial *variance* (Gary's C) or spatial *correlation* (Moran's I). We will discuss the family of “correlational” methods. We start off with considering the regular (Pearson’s) product-moment correlation between two random variables, Z_1 and Z_2 , which we denote by ρ_{12} and defined as:

$$\rho_{12} = \frac{(Z_1 - \mu_1)}{\sigma_1} \frac{(Z_2 - \mu_2)}{\sigma_2}$$

where μ ’s are expectations and σ ’s are standard deviations. *Autocorrelation* has exactly the same definition and is used when the Z ’s are measurements of the same quantity (e.g., prevalence, incidence, presence/absence, etc.) at different spatial locations (or different times).

To calculate the autocorrelation we need to know (or have an estimate of) the values of the μ ’s and σ ’s. In the case of single snapshot spatial data we use the *marginal* mean and *marginal* standard deviation.¹ Let’s explore using the graminifolia rust data (Fig. 13.1).

```
n = length(gra$score)
# marginal mean:
mu = mean(gra$score)
# marginal MLE sd:
sig = sd(gra$score) * (n - 1)/n
```

The estimated “autocorrelation matrix” (`rho`) among all 360 plants is then²:

```
# rescale Zs
zscale = (gra$score - mu)/sig
# autocorrelation matrix
rho = outer(zscale, zscale)
```

Note that these individual values are not constrained to be between -1 and 1 . This is not a worry, though, because the various geostatistical methods we will be discussing involve relatively simple manipulations of this matrix. For several of the methods we also need some sort of spatial distance matrix. Most commonly used is the Euclidian distance for UTM coordinates and [greater-circle distance](#) for latitude/longitude coordinates. The Euclidean distance matrix among all 360 plants is:

```
dst = as.matrix(dist(gra[, c("xloc", "yloc")]))
```

¹ Note that the geostatistical methods usually use the Maximum Likelihood Estimator (MLE) of the `sd` rather than the Best Linear Unbiased estimator (BLUE): i.e., the denominator is n rather than $n - 1$.

² The `outer`-function provides all pairwise products of two vectors.

To understand the different geostatistical methods we will consider the plot of the first 1000 pairs as a function of their spatial distance (Fig. 13.2). Plotting all the 64,620 pairs would clutter up the screen.

```
plot(dst[1:1000], rho[1:1000], ylab="Pairwise rho",
     xlab="Pairwise distance (m)")
```

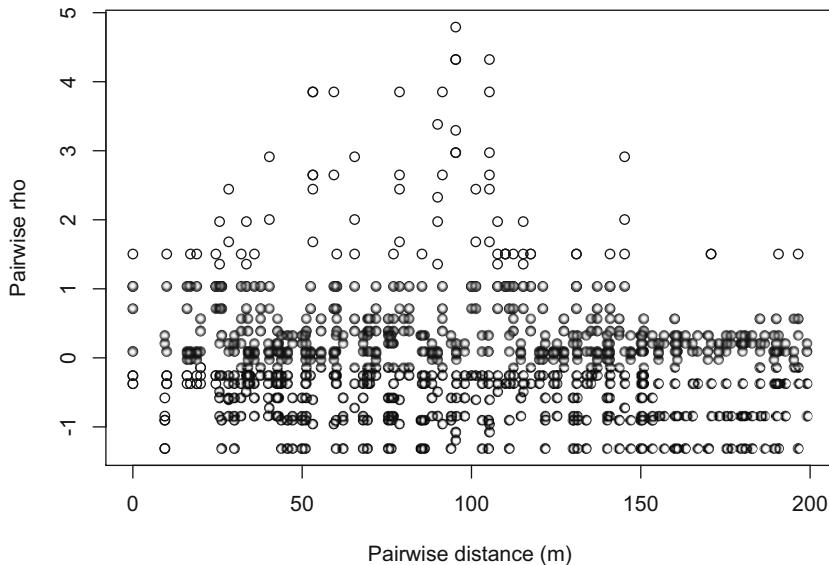


Fig. 13.2 Scatterplot of pairwise- ρ versus pairwise-distance

With this we are ready to conceptually understand many different geostatistical methods:

- **Mantel test:** An overall test for whether there is any significant relationship between the elements in the two matrices. This is essentially a test for significant correlation between ρ and distance.
- **Correlogram:** The most classic tool of testing how autocorrelation depends on distance without assuming any particular function—break the x-axis into segments (given by specifying some distance increment) and calculate the average within each distance class.³
- **Parametric correlation functions:** Assume the relationship follows some parametric relationship—such as Exponential, Gaussian, or Spherical functions—and do the appropriate nonlinear regression of ρ on distance; Sect. 15.2 provides an example of such fitting via the `nlme`-function of the `nlme`-library.

³ The semivariogram is similar to the correlogram but instead of using the “autocorrelation similarity” measure it uses the “semivariance dissimilarity” measure: $(Z_i - Z_j)^2 / 2$.

- **Nonparametric correlation function:** Fit a “nonparametric regression” (usually a smoothing spline or a kernel smoother) to the relationship (Hall and Patil 1994). This also goes by the name of the “spline.correlogram” (Bjørnstad and Falck 2001).
- **LISA:** Local indicators of spatial association (Anselin 1995): A test for “hotspots.” Specify a neighborhood size, and for each location calculate the average ρ with all the other locations that belongs to its neighborhood to find areas of significant above-average values.

There are a bunch of other named methods that are variations of these. Several of which are extensions to when there is multiple observations at each location (such as a time series), in which case it is natural to estimate the “autocorrelation matrix” using the regular correlation matrix. The “modified correlogram” of Koenig (1999) is the multivariate extension of the correlogram (e.g., Bjørnstad et al. 1999b). The “time-lagged spatial cross-correlation function” has been used to study waves of spread (see below and Sect. 11.7). Various directional versions allow the spatial correlation function to vary by cardinal direction (so-called anisotropic correlograms) to investigate directional patterns (e.g., Bjørnstad et al. 2002b).

13.4 Testing and Confidence Intervals

An important reason why specialized methods are needed for these analyses—despite most being conceptually simple—is because while the n original data-points may (or may not) be statistically independent, the n^2 numbers in the autocorrelation matrix is obviously very statistically not-independent and the interdependence is very intricate. None of the usual ways of testing for significance or generating confidence intervals are therefore applicable. Testing is usually done using permutation tests under the null-hypothesis of no spatial patterns. The correlogram (or Mantel test, or ...) of the real data should look no different than that of a random reallocation of observations to the spatial coordinates if the null hypothesis is true. Statistical significance is calculated by comparing the observed estimate to the distribution of estimates for, say, 999 different randomized data sets.⁴ If the observed is more extreme than 950 (990) of the randomized we conclude that there is significant deviation from spatial randomness at a nominal 5%-level (1%-level). For some of the methods it is possible to generate *confidence intervals* using bootstrapping (resampling with replacement) (e.g., Bjørnstad and Falck 2001).

All the above methods are available in the ncf-package.

```
require(ncf)
```

⁴ This produces a total of 1000 known possible outcomes; The 999 we randomly generated + the one nature provided.

13.5 Mantel Test

We continue using Keslow's data as a case study.

```
test1 = mantel.test(M1 = rho, M2 = dst)
```

```
test1
## $correlation
## [1] -0.04603662
##
## $p
## [1] 0.000999001
##
## $call
## [1] "mantel.test(M1 = rho, M2 = dst)"
##
## attr(),"class")
## [1] "Mantel"
```

We see that there is a significant negative association between similarity and distance. This is a crude tool but it does reveal that locations near each other tend to be more similar in disease status than those separated by a greater distance.

If we, instead of having two matrixes, have spatial coordinates and observations, the syntax is:

```
test = mantel.test(x = . . . , y = . . . , z = . . . )
```

13.6 Correlograms

The correlogram shows how the autocorrelation is a function of distance (Fig. 13.3). The shape of the correlogram can indicate random *versus* diffusive *versus* clinal patterns. Legendre and Fortin (1989) provide probes for patterns using various visual characteristics of the correlogram.

```
test2=correlog(x=gra$xloc, y=gra$yloc, z=gra$score,
               increment=10)
plot(test2)
```

The first distance class is significantly positive, and the estimated distance to which the local positive distance decays to zero (the *x-intercept*) is 44 m, indicative of significant local similarity. There is further evidence of significantly *negative* autocorrelation at long distances suggestive of a gradient (Legendre and Fortin 1989) across the field (Fig. 13.3).

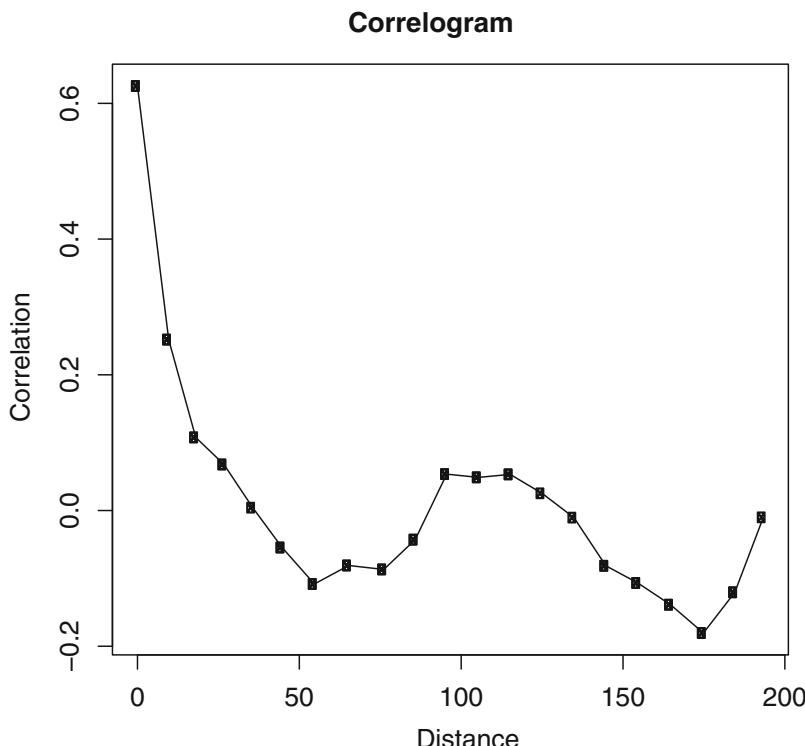


Fig. 13.3 The spatial correlogram of Keslow's rust data. Values that significantly deviate from that expected under the null hypothesis of complete spatial randomness are represented by filled black circles

13.7 Nonparametric Spatial Correlation Functions

We can get a bit finer resolution and confidence intervals for the underlying spatial correlation function using a nonparametric spatial covariance function (Hall and Patil 1994) as implemented in the spline correlogram (Bjørnstad and Falck 2001).

```
test3=spline.correlog(x=gra$xloc, y=gra$yloc,
                      z=gra$score)

summary(test3)

## $call
## [1] "spline.correlog(x = gra$xloc, y = gra$yloc,
## z = gra$score)"
##
## $estimate
##           x          e          Y
## estimate 36.53433 5.981471 0.5824953
##
## $quantiles
##           x          e          Y
## 0      -1.805418 0.000000 -0.02511758
## 0.025 23.252053 0.000000 0.14654935
## 0.25   33.067324 0.000000 0.28755750
## 0.5    36.555305 1.316775 0.38985499
## 0.75   39.924299 5.880383 0.48984864
## 0.975 44.163650 11.783945 0.75428625
## 1     49.466042 14.797740 0.98590369
```

The spline correlogram returns a bunch of stuff—in fact all the summary statistics I thought might be of relevance in some previous spatial analyses. These are:

- estimates: a vector of benchmark statistics
- x: is the lowest value at which the function is = 0.⁵
- e: is the lowest value at which the function is = 1/e (i.e., the spatial scale parameter in the presence of exponential or Gaussian spatial correlation).
- y: is the extrapolated value at x = 0.
- quantiles: A matrix summarizing the quantiles in the bootstrap distributions of the benchmark statistics. The 2.5- and 97.5-percentiles represent the 95% confidence interval.

```
plot(test3)
```

Figure 13.4 shows the estimated correlation function with its bootstrap 95% confidence intervals. The confidence intervals allow us to compare correlation functions for different data sets to test for significant differences (e.g., Bjørnstad et al. 1999a).

⁵ If correlation is initially negative, the distance calculated appears as a negative measure. This may seem a little strange, but some locally inhibitory processes predict significant negative local auto- or cross-correlation (e.g., Seabloom et al. 2005).

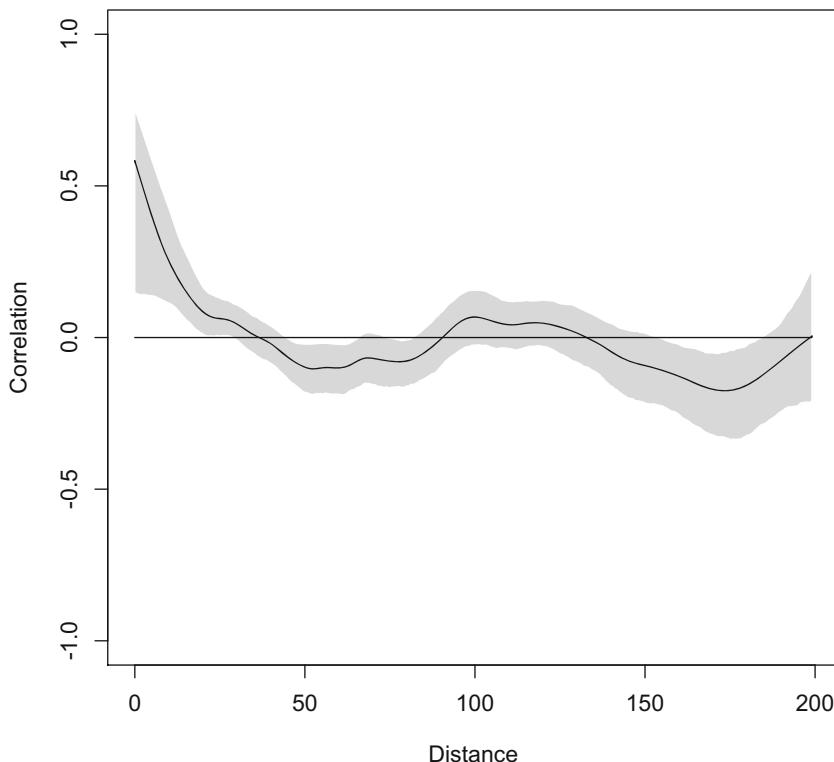


Fig. 13.4 The spline correlogram of Keslow's rust data. The outer lines represent the 95% bootstrap confidence interval

13.8 LISA

The previous methods average across all locations to study how similarity depends on distance. Local indicators of spatial association (Anselin 1995) quantify how similar observations are within neighborhoods of each observation—this can be used to test for *significant* spatial hot-/cold-spots of disease (Fig. 13.5). For this we have to define the radius of the neighborhood. Spatial dependence in the Koslow-data decay to zero at around 40 m (Fig. 13.4), so we use 20 m:

```
test4=lisa(x=gra$yloc, y=gra$xloc, z=gra$score,
neigh=20)
```

```
plot(test4)
```

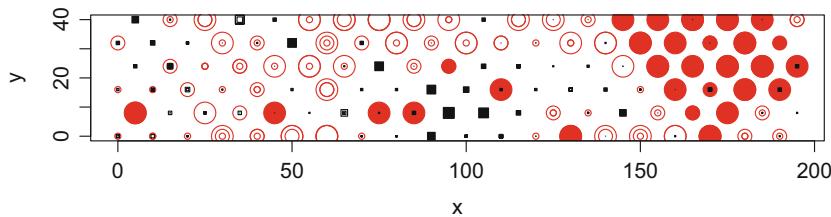


Fig. 13.5 LISA analysis of Koslow's rust data (with a 20 m neighborhood). Filled red circles are significant spatial hot-spots. Squares are cold-spots

Significant hot-spots show up as filled red circles and cold-spots as filled squares. The size of the symbols reflects how much the disease-score deviates from the mean.

13.9 Cross-Correlations

Janis Antonovics and his colleagues have done road-side surveys of antler smut disease counting number of healthy and diseased wild campions (*silene alba*) at the Mountain Lake Biological field station for more than 20 years (Antonovics 2004). The *silene2*-data contains the mean number of healthy (`$hmean`) and diseased (`$dmean`) individuals for each road segment, as well as latitude (`$lat`) and longitude (`$lon`) (Fig. 13.6).

```
data(silene2)
symbols(silene2$lon, silene2$lat, circles =
  sqrt(silene2$dmean), inches=.2, xlab="Longitude",
  ylab="Latitude")
```

Most geostatistical methods can be extended to consider spatial *cross-correlation* between different variables. We can use the *silene* data set to investigate if prevalence is spatially cross-correlated with abundance using the spline cross-correlogram (Fig. 13.7).

```
silene2$ab=silene2$dmean+silene2$hmean
silene2$prev=silene2$dmean/(silene2$dmean+silene2$hmean)
```

We square-root transform the abundance measure before analyses. There is significant positive cross-correlation within a 1 km range (95% CI: {0.6, 2.9} km).

```
testcc=spline.correlog(x=silene2$lon, y=silene2$lat,
  z=silene2$prev, w=sqrt(silene2$ab),
  latlon=TRUE, na.rm=TRUE)
plot(testcc)
```

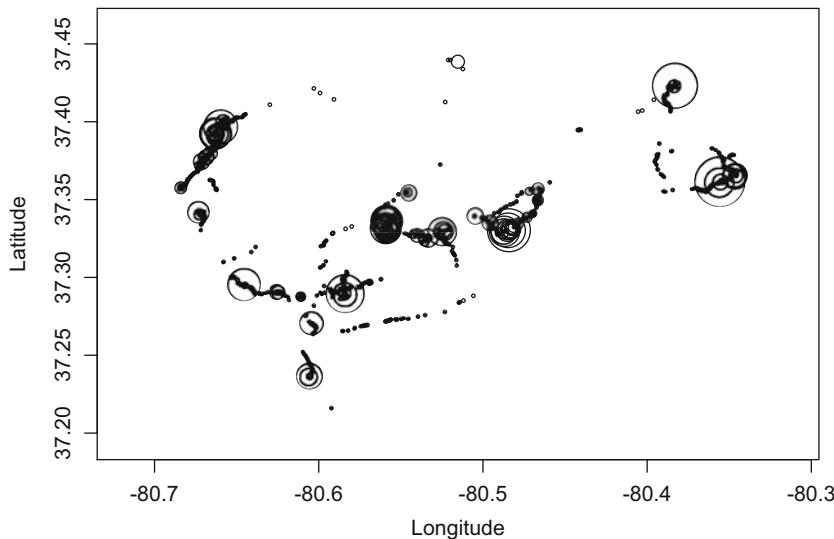


Fig. 13.6 Burden of antler smut on wild campion at Mt. Lake field station (Antonovics 2004)

We can use a spatial cross-correlogram (using 25 m distance increments) to study if presence/absence of rust is spatiotemporally cross-correlated between 1994 and 1995 in the filipendula data set we discussed in Sect. 11.2.

```
data(filip)
testcc2=correlog(x=filip$X, y=filip$Y, z=filip$Y94,
                  w=filip$Y95, increment=25)
```

The local inter-year correlation (`corr0`) is 0.75 and the first cross-correlation is significantly positive with a cross-correlogram x-intercept of 148 m⁶:

```
testcc2$corr0
## [1] 0.7651124
testcc2$x.intercept
## (Intercept)
##      148.939
```

Locations heavily affected in 1994 were thus also heavily affected in 1995 (testifying to the importance of local contagion and/or habitat heterogeneity in infection risk). This is an example of a “time-lagged cross-correlogram” (e.g., Bjørnstad et al. 2002b).

⁶ The spline cross-correlogram would give bootstrap confidence intervals on these quantities.

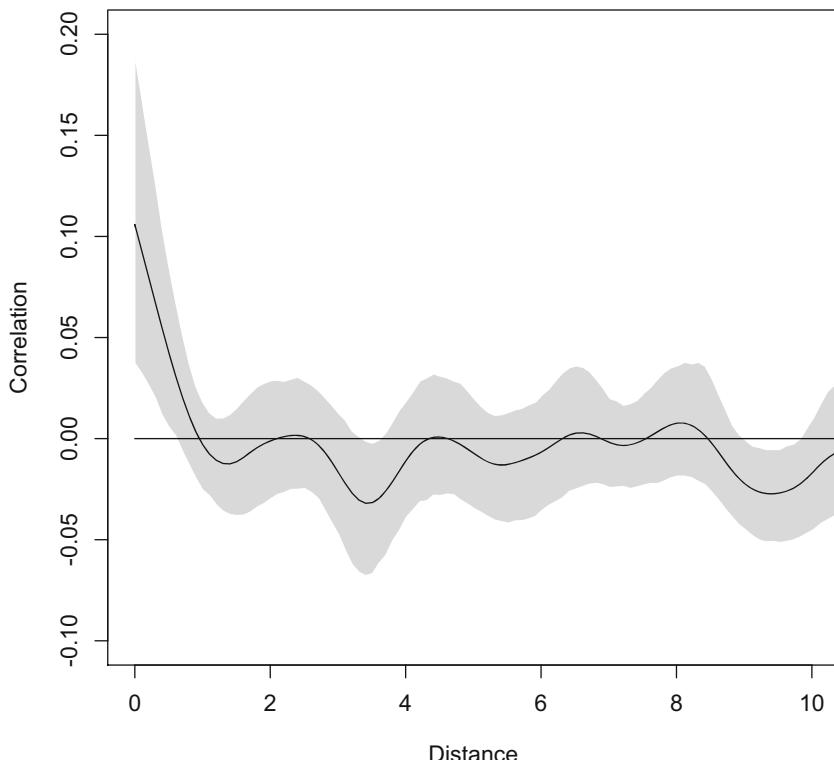


Fig. 13.7 Spatial cross-correlation of prevalence and abundance in the silene data

13.10 Gypsy Moth

The gypsy moth was introduced to the northeastern USA in the late 1860s and has spread at a rate of 10–20 km/year since. The larvae eats leaves of a wide range of trees and shrubs and reach outbreak (defoliating) densities usually around every 10 years. The outbreaks end through epizootics of the *Lymantria dispar* nuclear polyhedrosis virus and more recently the entomopathogenic fungus *Entomophaga maimaiaga* that together kills virtually all larvae following outbreaks. Bjørnstad et al. (2010) used the nonparametric spatial covariance function to study the spatiotemporal patterns in these outbreaks. The gm-data set contains UTM coordinates and fraction of forests defoliated each year between 1975 and 2002 in 20×20 km grid cells across the northeastern USA. We characterize the patterns of synchrony and time-lagged cross correlation in the outbreak time series.

```
data(gm)
sel=apply(gm[3:30], 1, sum) != 0
#Synchrony:
```

```

fit1=Sncf(gm[sel,1]/1000, gm[sel,2]/1000,
           gm[sel,3:30], resamp=500)
#Lag 1 cross-correlation
fit2=Sncf(gm[sel,1]/1000, gm[sel,2]/1000,
           z=gm[sel,3:29], w=gm[sel,4:30], resamp=500)
#Lag 2 cross-correlation
fit3=Sncf(gm[sel,1]/1000, gm[sel,2]/1000,
           z=gm[sel,3:28], w=gm[sel,5:30], resamp=500)

```

The outbreaks are highly synchronized out to 200 km, with a regional average outbreak correlation of around 0.2. The time lagged cross-correlation function shows significant local cross-correlation at the 1-year lag but not 2-year lag, indicating that outbreaks tend to persist spatially for 2 years before collapsing (Fig. 13.8):

```

par(mfrow = c(1, 3))
plot(fit1, ylim = c(-0.1, 1))
plot(fit2, ylim = c(-0.1, 1))
title("Lag 1")
plot(fit3, ylim = c(-0.1, 1))
title("Lag 2")

```

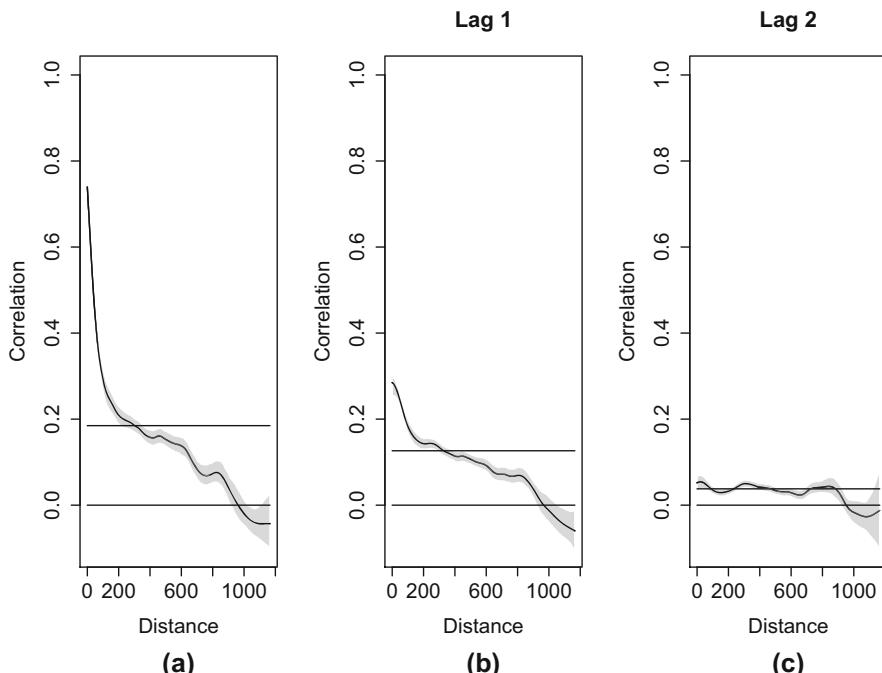


Fig. 13.8 The (a) nonparametric spatial covariance function, (b) lag-1, and (c) lag-2 cross-correlation function of gypsy moth outbreak data from the northeastern USA between 1975 and 2002

Chapter 14

Parasitoids



14.1 Parasitoid-Host Dynamics

Many of the classic studies of the spatiotemporal dynamics of natural enemies and their hosts consider parasitoid-host interactions. Parasitoids represent a fascinating group of insect “infections.” Adults are free-living and lay their eggs in larvae (or eggs) of host insects. Hosts die when the parasitoid(s) complete their development and adults emerge from the infected hosts. From a dynamical systems point of view parasitoid-host interactions share many features of infectious disease dynamics. It is therefore instructive to cap our discussion of spatiotemporal dynamics with a discussion of this ecological interaction.

Burnett (1958) conducted a cage experiment involving greenhouse white flies (*Trialeurodes vaporariorum*) and its parasitoid *Encarsia formosa*. The population was followed for 21 generations (Fig. 14.1). The two populations oscillated in increasingly violent cycles until the parasitoid went extinct.

```
data(burnett)
plot(burnett$Generation,
      burnett$NumberofHostsParasitized, type="b",
      ylab="Numbers", xlab="Generation")
lines(burnett$Generation,
      burnett$NumberofHostsUnparasitized, type="b",
      col=2, pch=2)
legend("topleft", legend=c("Parasitoid", "Host"),
      lty=c(1,1), pch=c(1,2), col=c(1,2))
```

Nicholson and Bailey (1935) developed the first mathematical model for this interaction. Assuming random search (with a searching efficiency a) by the parasitoids, the probability of escaping parasitization is $\exp(-aP_t)$ and the number of host, H , and parasitoids, P , in the next generation is:

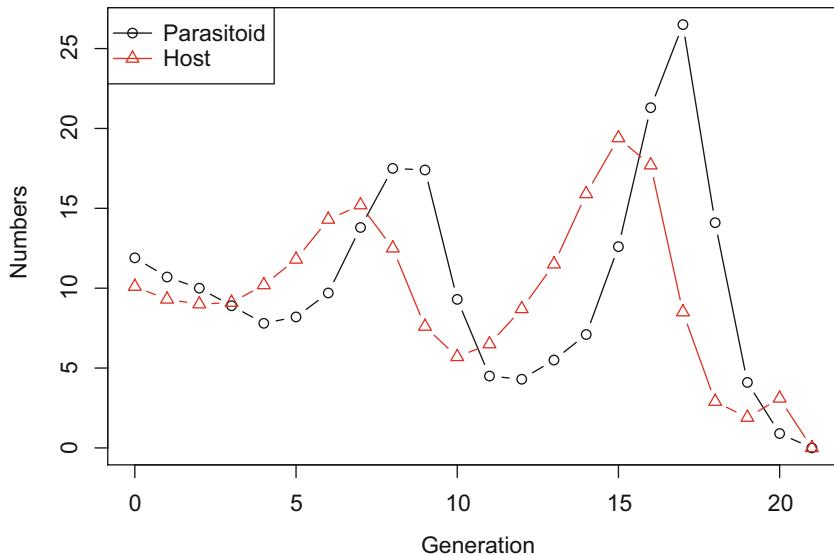


Fig. 14.1 Parasitoid-host dynamics of *T. vaporariorum* parasitized by *E. formosa*

$$H_{t+1} = RH_t \exp(-aP_t) \quad (14.1)$$

$$P_{t+1} = RH_t(1 - \exp(-aP_t)), \quad (14.2)$$

where R is the average number of offspring per hosts. A function for the host-parasitoid is:

```
NB = function(R, a, T = 100, H0 = 10, P0 = 1){
  #T is length of simulation (number of time-steps)
  #H0 and P0 are initial numbers
  H=rep(NA, T) #Host series
  P=rep(NA, T) #Parasitoid series
  H[1] = H0 #Initiating the host series
  P[1] = P0 #Initiating the parasitoid series

  for(t in 2:T) {
    H[t] = R * H[t-1] * exp(- a * P[t-1])
    P[t] = R * H[t-1] * (1-exp(- a * P[t-1]))
  }

  res= list(H = H, P = P)
  return(res)
}
```

The Nicholson-Bailey models show that density-independent growth of the host (in the absence of parasitism) and random search by parasitoids predicts cycles with

ever-increasing amplitude until the host and/or parasitoid goes extinct as seen in Burnett's (1958) experiment. Let's assume a host growth rate R of 1.1 and a parasitoid searching efficiency a of 0.1. We use the `NB()`-function to simulate the Nicholson-Bailey model and plot host/parasitoid abundance against time, and host-parasitoids in the phase plane (Fig. 14.2).

```
sim = NB(R=1.1,a=0.1)
time = 1:100
par(mfrow=c(1,2))
plot(time, sim$H, type= "l", xlab = "Generations",
      ylab = "Host abundance", ylim = c(0,14))
points(time, sim$P, type = "l", col = "red")
plot(sim$H,sim$P, type = "l", xlab = "Host abundance",
      ylab = "Parasitoid abundance")
```

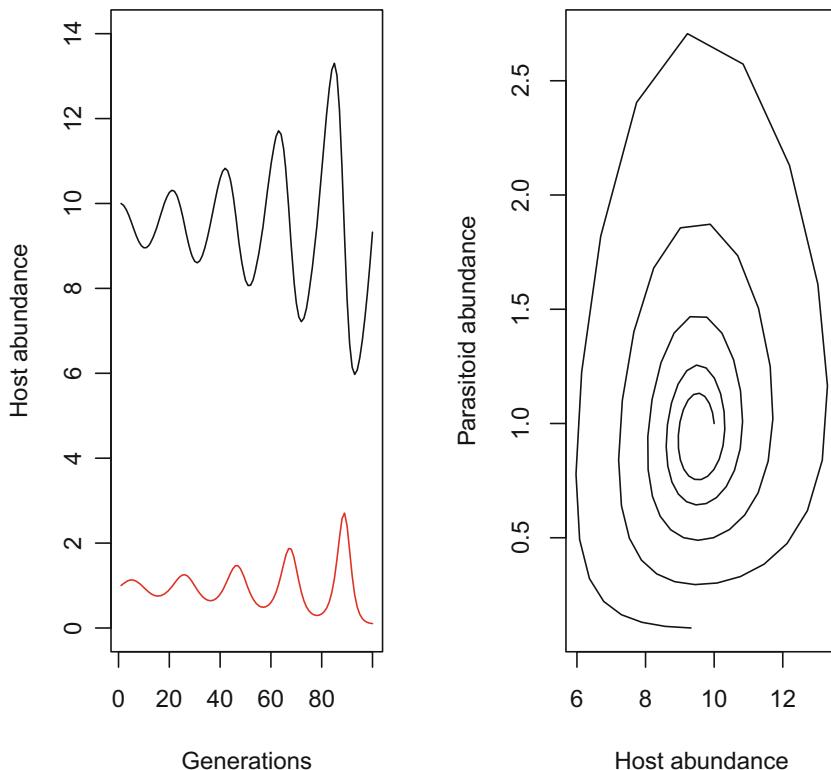


Fig. 14.2 Simulation from the Nicholson-Bailey model with $R = 1.1$ and $a = 0.1$

We can assume a sequence of searching efficiencies between 0 and 1 to explore how the time to extinction of the parasitoid depends on the searching efficiency. We

make use of the functions which and min to store the time to extinction. Persistence time is greatest at intermediate search efficiency (Fig. 14.3).

```
aVals = seq(0,1,by=0.01)
tte = rep(NA,length(aVals))
for (i in c(1:length(aVals))){
  sim = NB(R=1.1,a=aVals[i],T=500)
  tte[i] = min(which(sim$P==0))
}
plot(aVals,tte,type = "b", ylab="TTE",
  xlab="Search efficiency")
```

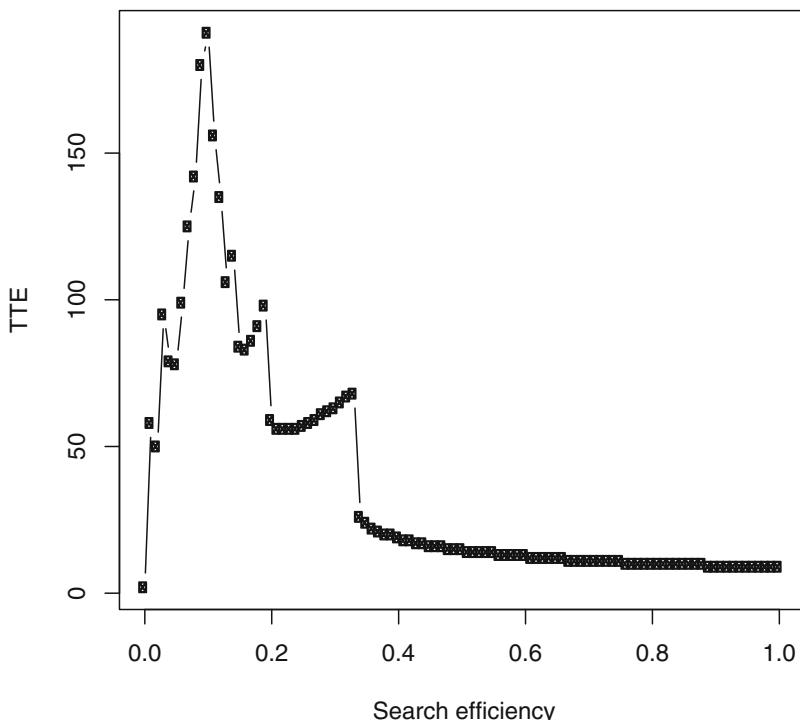


Fig. 14.3 Time to extinction of the parasitoid as a function of search efficiency in the Nicholson-Bailey model

Burnett (1958) suggested that $R = 2$ and $a = 0.067$ were appropriate values for this system. Let's check if we agree by minimizing sum-of-square-errors between observed and predicted abundances. We estimate R and a on a log-scale to make sure they are strictly positive quantities.

```

ssfn=function(par) {
  R=exp(par[1])
  a=exp(par[2])
  sim=NB(R, a, T=22, H0=10.1, P0=11.9)
  ss=sum((burnett$NumberofHostsUnparasitized-sim$H)^2+
         (burnett$NumberofHostsParasitized-sim$P)^2)
  return(ss)
}

par=log(c(2, 0.05))
fit=optim(par, ssfn)
exp(fit$par)

## [1] 2.16767130 0.06812596

```

Our fit is close to Burnett's. Figure 14.4 shows the model prediction using our best-guess parameters.

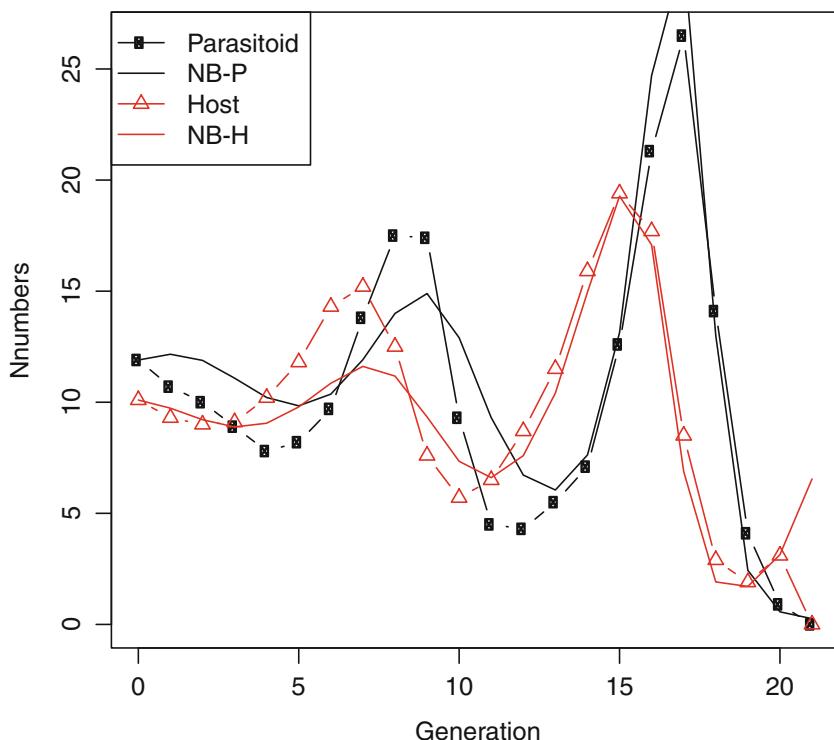


Fig. 14.4 Burnett's data and predictions by the Nicholson-Bailey model with $R = 2.17$ and $a = 0.07$

```

sim= NB(R=2.16767, a=0.06812, T=22, H0=10.1, P0=11.9)
plot(burnett$Generation,
      burnett$NumberofHostsParasitized, type="b",
      ylab="Numbers", xlab="Generation")
lines(burnett$Generation, sim$P)
lines(burnett$Generation,
      burnett$NumberofHostsUnparasitized, type="b",
      col=2, pch=2)
lines(burnett$Generation, sim$H, col=2)
legend("topleft", legend=c("Parasitoid", "NB-P",
  "Host", "NB-H"), lty=c(1,1,1,1), pch=c(1,NA,2,NA),
  col=c(1,1,2,2))

```

14.2 Stability and Resonant Periodicity

Nicholson and Bailey (1935) did a detailed mathematical analysis of the model and showed that the equilibrium is an *unstable focus* regardless of parameter values. We can revisit on the concepts from Chap. 9 for this model. The equilibrium of the Nicholson-Bailey model is $P^* = \log(R)/a$, $H^* = \log(R)/(a(R-1))$. The eigenvalues of the Jacobian evaluated at the equilibrium are:

```

F=expression(R*H*exp(-a * P))
G=expression(R*H*(1-exp(-a * P)))
j11=D(F, "H"); j12=D(F, "P")
j21=D(G, "H"); j22=D(G, "P")
R=2.17; a=0.068
params=c(R=R, a=a, P=log(R)/a, H= log(R)/(a*(R-1)))
J=with(as.list(params),
       matrix(c(eval(j11), eval(j12), eval(j21),
       eval(j22)), ncol=2, byrow=T))

eigen(J, only.values =TRUE)$values
## [1] 0.83108+0.8638247i 0.83108-0.8638247i
max(abs(eigen(J)$values))
## [1] 1.198702

```

It is an unstable focus since the eigenvalues are a pair of complex conjugates whose absolute value is greater than one.¹ Since this is a difference model, the predicted period of the outwards spiral is $2\pi/\tan^{-1}(\frac{\text{Im}(\lambda)}{\text{Re}(\lambda)})$:

```
2*pi/atan2(Im(eigen(J)$values[1]),
  Re(eigen(J)$values[1]))  
## [1] 7.807961
```

Let us explore how the cycle period depends on host growth rate (Fig. 14.5):

```
RVals = seq(1.1, 3, by=0.1)
per = rep(NA, length(RVals))
```

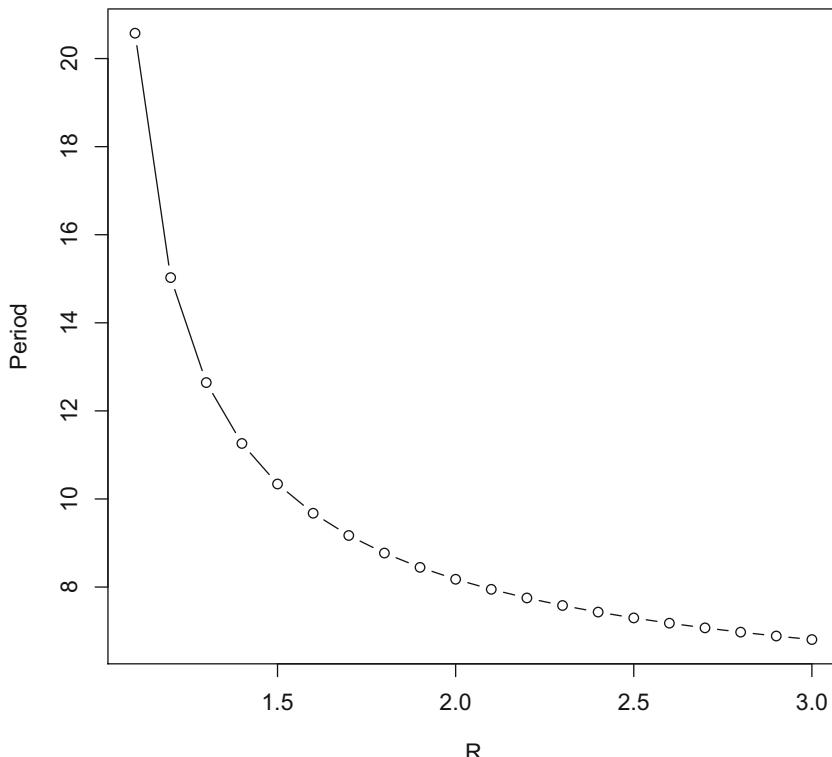


Fig. 14.5 Resonant period of the unstable Nicholson-Bailey model as a function of host growth rate

¹ Recall that according to [local stability theory](#), stability of discrete time models requires the absolute value of the largest eigenvalue of the Jacobian evaluated at the equilibrium to be smaller than 1—as opposed to continuous time models for which the requirement is that the real part must be smaller than 0.

```

for(i in 1:length(RVals)){
R=RVals[i]
a=0.068
params=c(R=R, a=a, P=log(R)/a,
H= log(R)/(a*(R-1)))
J=with(as.list(params),
matrix(c(eval(j11), eval(j12), eval(j21),
eval(j22)), ncol=2, byrow=T))
per[i]=2*pi/atan2(Im(eigen(J)$values[1]),
Re(eigen(J)$values[1]))
}
plot(RVals, per, type="b", xlab="R", ylab="Period")

```

The higher the host growth rate, the faster the outwards spiral.

14.3 Biological Control

Parasitoids have been used for biocontrol of agricultural pest through the ages (Murdoch et al. 1985). Successful biocontrol requires that the natural enemy—in this case parasitoid—keeps the pest consistently below an economic threshold. The inherent instability predicted by the Nicholson-Bailey model is at odds with successful bio-control by parasitoids. Many different model modifications have been analyzed to see when stable regulation can happen. These include: (1) long-lived adult hosts, (2) density-dependent host growth, (3) heterogeneity in risk such as aggregated attack rates, spatial heterogeneity, host refugia, and (4) interference among parasitoids (Murdoch et al. 2003). May (1978) showed—by replacing the Poisson-attack assumption with a negative binomial distribution—how heterogeneity in risk stabilizes dynamics. He coined the CV²-rule which says that if the coefficient-of-variation in attack rate is greater than 1, the parasitoid-host dynamics stabilizes. May’s (1978) model is:

$$H_{t+1} = RH_t \left(1 + \frac{aP_t}{k}\right)^{-k} \quad (14.3)$$

$$P_{t+1} = RH_t \left(1 - \left(1 + \frac{aP_t}{k}\right)^{-k}\right). \quad (14.4)$$

A Shiny-app of the negative binomial model, `May.app`, can be found in the `epimdr`-package.

As with infectious diseases, parasitoids-host interactions may persist even with local non-persistence through regional persistence and regulation in a consumer-resource metapopulation...

14.4 Larch Bud Moth

Parasitoids cause violent fluctuations in the dynamics of the larch bud moth across the European Alps. Historical records show recurrent traveling waves of defoliation every 9 years for centuries (Bjørnstad et al. 2002b; Johnson et al. 2010).² Turchin (2003) developed a model of the interactions among the larch bud worm, its parasitoids, and the host plant. Johnson et al. (2004) showed that a spatial extension of this model predicts the observed waves. However, for the more general purpose of considering spatiotemporal host-parasitoid dynamics we will use the simpler spatially extended Nicholson-Bailey model.

14.5 Host-Parasitoid Metapopulation Dynamics

Coupled map lattice models have a long history in the study of parasitoid-host dynamics (e.g., Hassell et al. 1991; Bjørnstad and Bascompte 2001; Johnson et al. 2004). Hassell et al. (1991) highlighted the importance of allowing for different mobility of the host and parasitoid. We define D_h as the proportion of host that disperses to neighboring patches, and D_p the proportion of parasitoid that disperses. Hassell et al. (1991) showed that changing these can shift the spatial dynamics between spatial chaos, spiral waves, or stable spatial heterogeneity. We construct a Nicholson-Bailey CML along the lines introduced in Chap. 11:

```
#Dh is proportion of hosts that disperses
#Dp is proportion of parasitoids that disperses
Dh = 0.5
Dp = 0.7
#xlen is width of the lattice (E-W)
#ylen is height of the lattice (N-S)
xlen = 30
ylen = 30
```

The `hp.dyn`-function defines the function to update the local abundances of hosts and parasitoids according to the Nicholson-Bailey model. Previous densities of host, `h`, and parasitoids, `p`, need to be supplied as arguments to the function, in addition to the host growth rate (`R`) and parasitoid search efficiency `a`.

```
hp.dyn = function(h, p, R, a) {
  # hnew is the post-interaction host density
  hnew = R * h * exp(-a * p)
```

² <https://github.com/objornstad/epimdr/blob/master/mov/lbm.gif> shows an animated gif of Larch bud moth defoliation between 1960 and 2000.

```

# pnew is the post-interaction parasitoid density
pnew = R * h * (1 - exp(-a * p))
# the two vectors of results are stored in a 'list'
res = list(h = hnew, p = pnew)
return(res)
}

```

We generate spatial coordinates and calculate the distance between all populations:

```

xy = expand.grid(1:xlen, 1:ylen)
dmat = as.matrix(dist(xy))

```

The redistribution matrix is calculated by checking if the distance in dmat is smaller than 1.5, thus flagging all populations that are first neighbors. If the distance is < 1.5 we assign a value of $D_h/8$ and $D_p/8$. The fractions that do not disperse ($1-D_h$ and $1-D_p$) are along the diagonal of the redistribution matrices.

```

kh = ifelse(dmat < 1.5, Dh/8, 0)
kp = ifelse(dmat < 1.5, Dp/8, 0)
diag(kh) = 1 - Dh
diag(kp) = 1 - Dp

```

We finally construct matrices to store results and set starting conditions for the simulation. IT is number of iterations. The initial conditions are zeros everywhere (i.e., the first column in each matrix gets zeros), except for an arbitrary population (in this case 23) which starts with 4 hosts and 1 parasitoid.

```

IT = 600
hmat = matrix(NA, nrow = xlen * ylen, ncol = IT)
pmat = matrix(NA, nrow = xlen * ylen, ncol = IT)
hmat[, 1] = 0
pmat[, 1] = 0
hmat[23, 1] = 4
pmat[23, 1] = 1

```

We simulate from generation 2 to IT, storing the results on the way. We first simulate growth, using the function hp.dyn. Next, we redistribute the host parasitoids according to their dispersal matrices. The vector of pre-dispersal hosts, tmp\$h, is redistributed through matrix multiplying the vector by the redistribution matrix, kh. The same is done for the parasitoid. The function cat() provides a progress monitor.

```

for(i in 2:IT) {
  #growth
  tmp = hp.dyn(h = hmat[, (i-1)], p = pmat[, (i-1)],
    R = 2, a = 1)
  #redistribution
  hmat[, i] = tmp$h%*%kh;
  pmat[, i] = tmp$p%*%kp;
  cat(i, " of ", IT, "\r")
}

```

Finally, we are ready to plot our results. Here is the code to make an animation of the last 100 generations for the parasitoid:

```

#plot the last 100 generations for the parasitoid
for(i in 1:100){
  x=xy[,1]
  y=xy[,2]
  z=pmat[,i+500]
  symbols(x,y, fg=2, circles=z, inches=0.1,
    bg=2, xlab="", ylab="")
  Sys.sleep(.1) #this is to slow down the plotting
}

```

Low mobility of both host and parasitoid (e.g., $D_p=D_h=0.1$) leads to spatially chaotic dynamics and high mobility (e.g., $D_h=0.5$, $D_p=0.7$) leads to spiral waves (Hassell et al. 1991). Animated gifs of the two dynamic regimes are on:
<https://github.com/objornstad/epimdr/blob/master/mov/cml1.gif> and
<https://github.com/objornstad/epimdr/blob/master/mov/cml2.gif>.

14.6 ShinyApp

The `epimdr`-package contains a Shiny-app to study the negative-binomial parasitoid-host model. The app can be launched through:

```

require(shiny)
May.app

```

Chapter 15

Non-independent Data



15.1 Introduction

Many infectious disease experiments result in non-independent data because of spatial autocorrelation across fields (such as discussed in Chap. 13), repeated measures on experimental animals (such as the in-host *Plasmodium* data discussed in Sect. 7.7), or other sources of correlated experimental responses among experimental units (such as the possibility of correlated infection fates among the rabbit littermates discussed in Sect. 4.3). Statistical methods that assume independence of observations are not strictly valid and/or fully effective on such data (e.g., Legendre 1993; Keitt et al. 2002). “Mixed-effects models” and “Generalized linear mixed-effects models” (GLMMs) have been/are being developed to optimize the analysis of such data (Pinheiro and Bates 2006).

While this full topic is outside the main scope of this text, it is very pertinent to analyses of disease data, so we will consider the three case studies.

```
require(nlme)
require(ncf)
require(lme4)
require(splines)
```

15.2 Spatial Dependence

We use the rust example introduced in Sect. 13.2 (Fig. 13.1) to illustrate two approaches to accounting for spatial dependence in disease data: (1) random blocks vs (2) spatial regression. This experiment looked at severity of a foliar rust infection

This chapter uses the following R-packages: `nlme`, `ncf`, `lme4`, and `splines`.

on three focal individuals of flat-top goldenrods in each of 120 plots across a field divided into four blocks. The experimental treatments were (1) watering or not and (2) whether surrounding non-focal host plants were conspecifics only, a mixture of conspecifics and an alternative host (the Canadian goldenrod) or the alternative host only.

15.2.1 Random Blocks

As in our spatial pattern analysis, we jitter the coordinates because some methods require unique coordinates for each data point.

```
data(gra)
gra$jx = jitter(gra$xloc)
gra$jy = jitter(gra$yloc)
```

We first use lme to fit two random effect models. The first considers individuals in blocks. The second considers plots nested in blocks.

```
fit=lme(score~comp+water, random = ~1 | block,
         data= gra, na.action=na.omit)
fit2=lme(score~comp+water, random = ~1 | block / plot,
          data= gra, na.action=na.omit)
```

We next do a likelihood ratio-test to check for the better fit. The likelihood ratio test (provided by anova) shows that the nested model provides the best fit.

```
anova(fit, fit2)

##      Model df      AIC      BIC    logLik   Test
## fit     1 6 1186.175 1209.424 -587.0874
## fit2    2 7 1077.579 1104.704 -531.7895 1 vs 2
##            L.Ratio p-value
## fit
## fit2 110.5959 <.0001
```

The intervals-call shows that the between-plot variance is about twice as large as the between-block variance, and watered plots have a significantly higher rust burden.

```
intervals(fit2)

## Approximate 95% confidence intervals
## Fixed effects:
##           lower       est.       upper
## (Intercept) 0.8678624 1.4180556 1.9682487
```

```

##  compSOL      -0.2517755  0.2083333  0.6684422
##  compSYM     -0.1726089  0.2875000  0.7476089
##  watermesic   0.2548782  0.6305556  1.0062329
##  attr(,"label")
##  [1] "Fixed effects:"
##
##  Random Effects:
##    Level: block
##                      lower       est.       upper
##  sd((Intercept)) 0.154977  0.4101308  1.08537
##    Level: plot
##                      lower       est.       upper
##  sd((Intercept)) 0.7901556  0.9302044  1.095076
## 
##  Within-group standard error:
##        lower       est.       upper
##  0.7317349  0.8001735  0.8750132

```

15.2.2 Spatial Regression

The above randomized block mixed-effects models are the classic solution to analyzing experiments with spatial structure. An alternative is to formulate a regression model that considers the spatial dependence among observations as a function of separating distance. To investigate how proximate observations on different experimental treatments may be spatially autocorrelated, we can explore the spatial dependence among the *residuals* from a simple linear analysis of the data. We use the nonparametric spatial covariance function (as implemented in the `spline.correlogram()`-function in the ncft-package) discussed in Chap. 13. We first fit the simple regression model that ignores space altogether.

```
fitlm = lm(score ~ comp + water, data = gra)
```

Next we calculate the spatial correlation function among the residuals of the fit (Fig. 15.1).

```
fitc = spline.correlog(gra$x, gra$y, resid(fitlm))
```

The nonparametric spatial correlation function reveals strong spatial autocorrelation that decays to zero around 38 m (with a CI of 31–43 m).

```
plot(fitc, ylim = c(-0.5, 1))
```

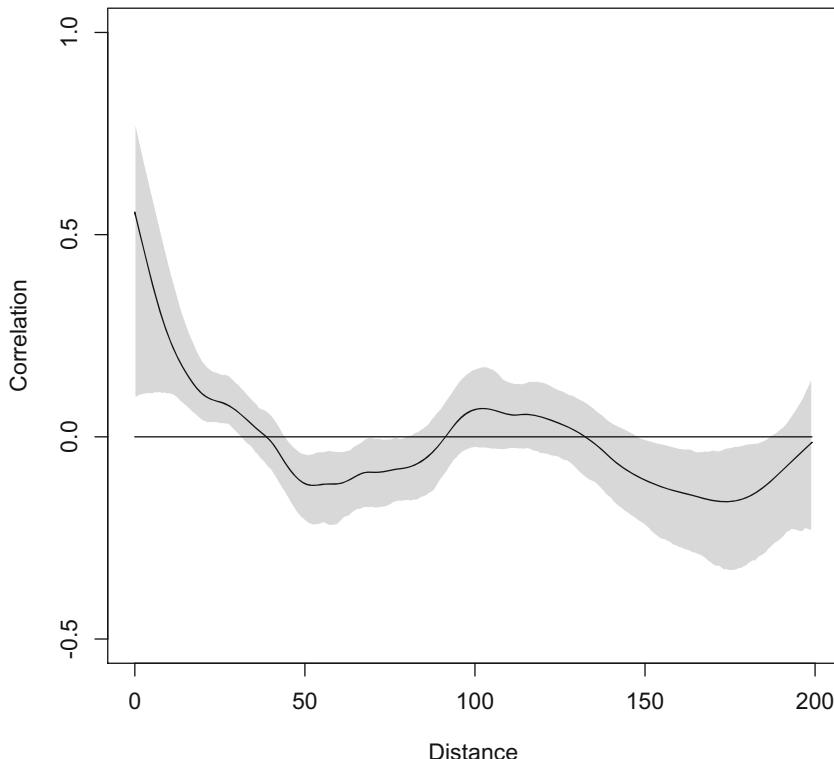


Fig. 15.1 The spline correlogram of the residuals from the regression model of Keslow's rust data

To fit the spatial regression model we use the `gls`-function from the `nlme`-package (Pinheiro and Bates 2006). This function fits mixed models from data that have a single dependence group, i.e., one spatial map, one time series, etc.; With multiple groups we use the `lme`-function discussed (see Sect. 15.3). There are many possible models for spatial dependence. We compare the exponential model (which assumes the correlation to decay with distance according to $\exp(-d/a)$ where d is distance and a is the scale) and the Gaussian model ($\exp(-(d/a)^2)$). [The nugget-flag means that the function is not anchored at one at distance zero]. We compare these to the nonspatial model (`fitn`) and the best random block model (`fit2`) using AIC.

```

fite=gls(score~comp+water, corr = corSpatial(form =
    ~jx + jy, type="exponential", nugget=TRUE),
    data=gra, na.action=na.omit)
fitg=gls(score~comp+water, corr = corSpatial(form =
    ~jx + jy, type="gaussian", nugget=TRUE), data=gra,
    na.action=na.omit)
fitn=gls(score~comp+water, data=gra, na.action=na.omit)
AIC(fite, fitg, fitn, fit2)

##      df      AIC
## fite   7 1061.725
## fitg   7 1064.522
## fitn   5 1209.500
## fit2   7 1077.579

```

The AICs show that the exponential model provides the best fit. Moreover, the spatial regression model provides a better fit than the nested random effect model. This is presumably because of the gradual decay in correlation with distance (Fig. 15.1).

```

summary(fite, corr = FALSE)

## Generalized least squares fit by REML
## Model: score ~ comp + water
## Data: gra
##          AIC      BIC      logLik
## 1061.725 1088.849 -523.8623
##
## Correlation Structure: Exponential spatial
## correlation
## Formula: ~jx + jy
## Parameter estimate(s):
##        range     nugget
## 9.9222621 0.3210873
##
## Coefficients:
##                  Value Std.Error t-value p-value
## (Intercept) 1.4914991 0.2595408 5.746685 0.0000
## compSOL      0.1776521 0.2030045 0.875114 0.3821
## compSYM      0.2068005 0.2015687 1.025955 0.3056
## watermesic   0.4998769 0.1589941 3.143996 0.0018
##
## Correlation:
##             (Intr) cmpSOL cmpSYM
## cmpSOL      -0.393

```

```

##  compSYM      -0.397   0.547
##  watermesic  -0.291   0.041   0.022
##
##  Standardized residuals:
##          Min           Q1           Med           Q3
##  -1.3253792 -0.7737412 -0.1546712  0.6258009
##          Max
##  3.9090911
##
##  Residual standard error: 1.281276
##  Degrees of freedom: 360 total; 356 residual

```

The parametrically estimated range of 9.8 m is a bit longer (but within the confidence interval) of the e-folding scale (5.5 m) estimated by the spline correlogram; $1-\text{nugget} = 0.64$ is comparable (but a little greater) than the 0.55 y-intercept. We can use the `Variogram`-function from the `nlme`-package to see if the spatial model adequately captures reflects the spatial dependence (Fig. 15.2). It looks like a plausible fit.

```
plot(Variogram(fite))
```

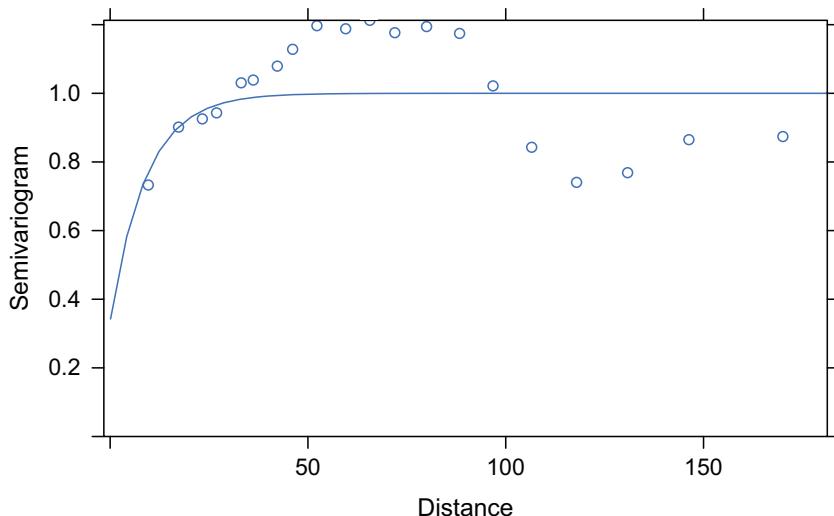


Fig. 15.2 A variogram plot of the fitted and observed spatial dependence for the spatial regression model

15.3 Repeated Measures of In-Host Mouse Malaria

Repeated measurements usually result in non-independent data because of the inherent serial dependence. Consider Huijben's data on anemia of mice infected by five different strains of *Plasmodium chaubodii* introduced in Sect. 7.7 with lots of measurements taken on days 3 through 21, 24, 26, 28, 31, 33, and 35. We will study the red blood cell counts (RBCs) of mice infected by one of five different clones as well as the control group. The sample sizes per treatment were 10 for AQ, BC, CB, and ER, 7 for AT and 5 for control. Eleven of the animals died. SH9 has the data (in long format).¹ For the analysis we strip some unnecessary columns 1, 3, 4, 7, 8, and 11 that are extraneous to focus on the RBC count:

```
data(SH9)
SH9RBC = SH9[, -c(1, 3, 4, 7, 8, 10, 11)]
```

For the repeated measures analyses we create a groupedData-object from the data frame using the nlme-package. The below call declares how the RBC counts represent time series for each mouse. Note that mice that died are scored by zero RBC count in the data set and that these zeros end up dominating patterns, we therefore rescore these data as missing (NA), and plot the grouped data object to visualize the anemia by treatment (Fig. 15.3).

```
RBC = groupedData(RBC ~ Day | Ind2, data = SH9RBC)
RBC$RBC[RBC$RBC == 0] = NA
plot(RBC, outer = ~Treatment, key = FALSE)
```

The main difference is between control and treatments, but the maximum anemia varies somewhat among strains. To test for significant differences we use lme to build a repeated measures model. In the simplest case we follow standard convention and model the time series using day as an ordered factor and assume the treatment effect to be additive. The random= $\sim 1 | \text{Ind2}$ -call in the formula indicates that we assume there to be individual variation in the intercept (but not the slopes) among individuals. We then use the ACF function to look for evidence of serial dependence in the residuals from the fit. As is apparent from the ACF plot there is temporal autocorrelation in the residuals out to at least 4 days (Fig. 15.4).

```
mle.rbc=lme(RBC~Treatment+ordered(Day), random =
  ~1|Ind2, data=RBC, na.action=na.omit, method="ML")
plot(ACF(mle.rbc))
```

There are many models for serial dependence. We use a first order autoregressive process (AR1). This is specified by the correlation=corAR1 (form= \sim

¹ With repeated measures data we often use both long-format with one line for each observation and wide-format with one line for each experimental unit.

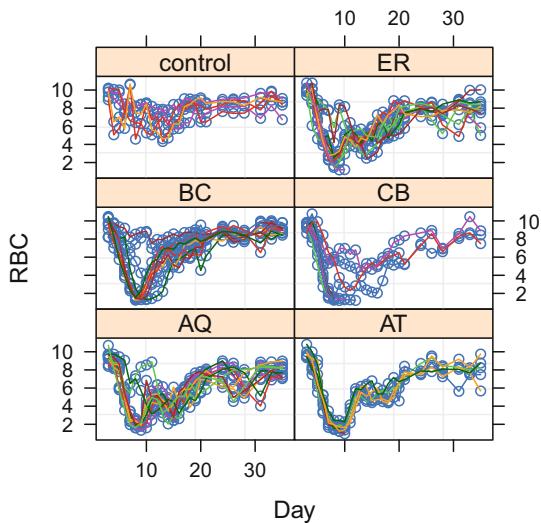


Fig. 15.3 RBC counts of control and *P. chaubodii*-infected mice. Each panel represents a different parasite strain

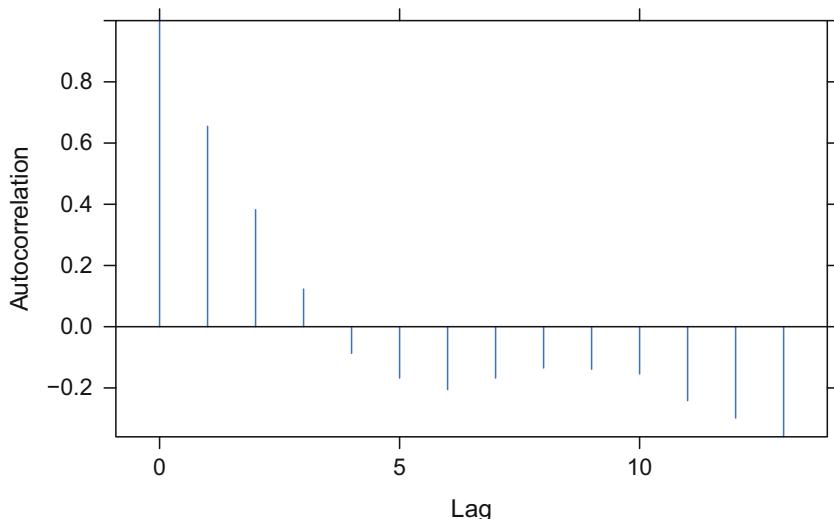


Fig. 15.4 Serial dependence as quantified using the ACF-function on the repeated measures mixed-effects model of the SH9RBC data

`Day | Ind2)` function call. Note that this is one of a variety of time-series models available in the `n1me`-package, the most general of which is the ARMA(p, q) model discussed in Sect. 6.2.1.

```
mle.rbc2=lme(RBC~Treatment+ordered(Day), random=~1|Ind2, data=RBC, correlation=corAR1(form=~Day|Ind2), na.action=na.omit, method="ML")  
mle.rbc2  
  
## Linear mixed-effects model  
## Data: RBC  
## Log-likelihood: -1568.255  
## Fixed: RBC ~ Treatment + ordered(Day)  
## (Intercept) TreatmentAT TreatmentBC  
## 5.860494309 0.024586193 0.947853117  
## TreatmentCB Treatmentcontrol TreatmentER  
## -0.022048465 1.560872851 0.325308683  
## ordered(Day).L ordered(Day).Q ordered(Day).C  
## 3.339300000 6.015597509 -5.057192257  
## ordered(Day)^4 ordered(Day)^5 ordered(Day)^6  
## 1.498354649 0.067695099 -0.600409959  
## ordered(Day)^7 ordered(Day)^8 ordered(Day)^9  
## 1.352000127 -1.122142721 -0.394162545  
## ordered(Day)^10 ordered(Day)^11 ordered(Day)^12  
## 0.312998475 -0.673514349 -0.122937927  
## ordered(Day)^13 ordered(Day)^14 ordered(Day)^15  
## 0.219014886 0.378460147 0.191963472  
## ordered(Day)^16 ordered(Day)^17 ordered(Day)^18  
## 0.180627944 -0.024392052 0.032617128  
## ordered(Day)^19 ordered(Day)^20 ordered(Day)^21  
## -0.142080994 -0.046539002 -0.054854991  
## ordered(Day)^22 ordered(Day)^23 ordered(Day)^24  
## -0.039333282 -0.210031799 0.006591632  
##  
## Random effects:  
## Formula: ~1 | Ind2  
## (Intercept) Residual  
## StdDev: 0.0002332905 1.327223  
##  
## Correlation Structure: ARMA(1,0)  
## Formula: ~Day | Ind2  
## Parameter estimate(s):  
## Phil  
## 0.7088701  
## Number of Observations: 1104  
## Number of Groups: 52
```

The Phi1 parameter of 0.7088 represents the estimated day to day correlation, which is substantial. We can plot the predicted and observed correlation. The AR1-model seems to be a nice fit (Fig. 15.5).

```
tmp = ACF(mle.rbc2)
plot(ACF ~ lag, data = tmp)
lines(0:15, 0.7088^(0:15))
```

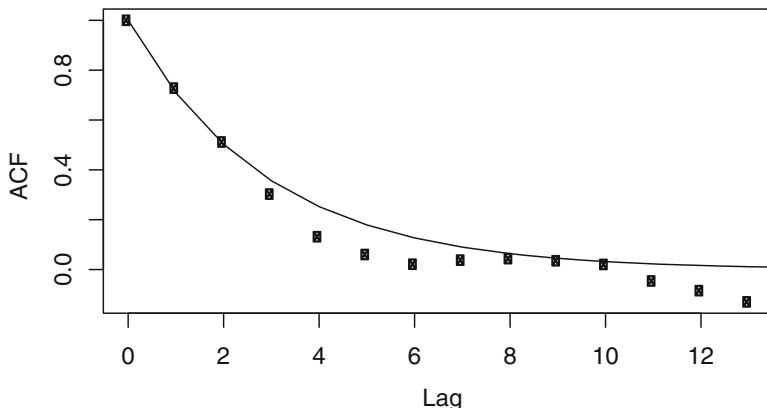


Fig. 15.5 An ACF plot of the fitted and observed serial dependence for the repeated measures regression model

Moreover, a formal likelihood-ratio test provided by the `anova` function reveals that the correlated error model provides a significantly better fit to the data:

```
anova(mle.rbc, mle.rbc2)

##               Model df      AIC      BIC    logLik
## mle.rbc       1 32  3834.369 3994.583 -1885.184
## mle.rbc2     2 33  3202.510 3367.731 -1568.255
##                   Test  L.Ratio p-value
## mle.rbc
## mle.rbc2 1 vs 2 633.8586 <.0001
```

Statistically, the time-by-treatment interaction model, rather than the additive model, is better still:

```
options(width=50)
mle.rbc3=lme(RBC~Treatment*ordered(Day), random=
~1|Ind2, data=RBC, correlation=corAR1(form=
~Day|Ind2), na.action=na.omit, method="ML")
anova(mle.rbc2, mle.rbc3)
```

```

##               Model   df      AIC      BIC    logLik
## mle.rbc2      1  33 3202.510 3367.731 -1568.255
## mle.rbc3      2 153 3163.654 3929.679 -1428.827
##               Test  L.Ratio p-value
## mle.rbc2
## mle.rbc3 1 vs 2 278.8557 <.0001

```

Finally we can plot the predicted values against time (filtering out predictions for the missing values in the original data) (Fig. 15.6). There is a distinct ordering in the virulence of the strains:

```

pr=predict(mle.rbc3)
RBC$pr=NA
RBC$pr[!is.na(RBC$RBC)]=pr
plot(RBC$pr~RBC$Day, col=as.numeric(RBC$Treatment),
     pch=as.numeric(RBC$Treatment), xlab="Day",
     ylab="RBC count")
legend("bottomright",
       legend=c("AQ", "AT", "BC", "CB", "Control", "ER"),
       pch=unique(as.numeric(RBC$Treatment)), col=1:6)

```

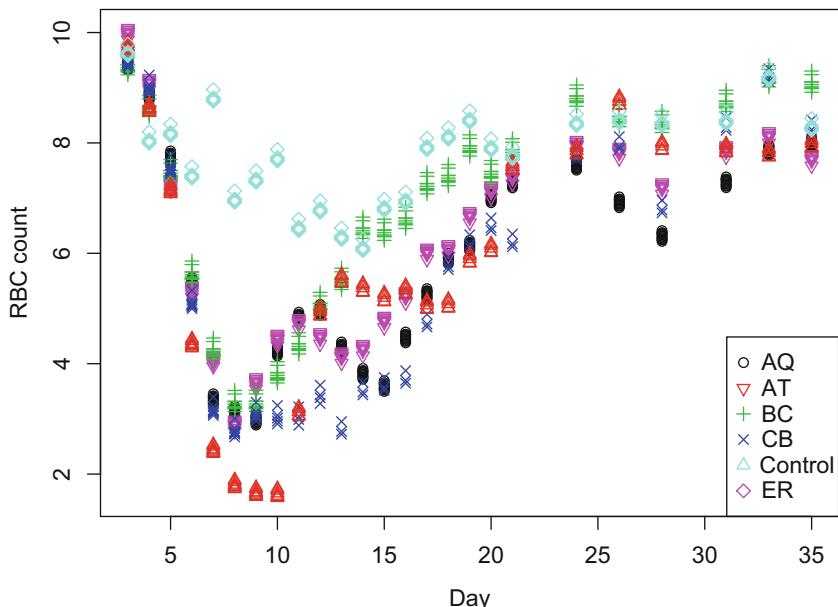


Fig. 15.6 Predicted and observed for the repeated measures RBC data

Modeling time as an ordered factor is quite parameter wasteful (the full interaction model has 153 parameters). A flexible yet more economic approach may be to model time using smoothing splines. The following example uses a B-spline with 5 degrees-of-freedom (Fig. 15.7). The qualitative features are similar to the more parameter rich model (Fig. 15.6)

```
require(splines)
mle.rbc4=lme(RBC~Treatment*bs(Day, df=5), random=
  ~1|Ind2, data=RBC, correlation=corAR1(form=
  ~Day|Ind2), na.action=na.omit, method="ML")
pr=predict(mle.rbc4)
RBC$pr=NA
RBC$pr[!is.na(RBC$RBC)]=pr
plot(RBC$pr~RBC$Day, col=as.numeric(RBC$Treatment),
  pch=as.numeric(RBC$Treatment), xlab="Day",
  ylab="RBC count")
legend("bottomright",
  legend=c("AQ", "AT", "BC", "CB", "Control", "ER"),
  pch=unique(as.numeric(RBC$Treatment))), col=1:6)
```

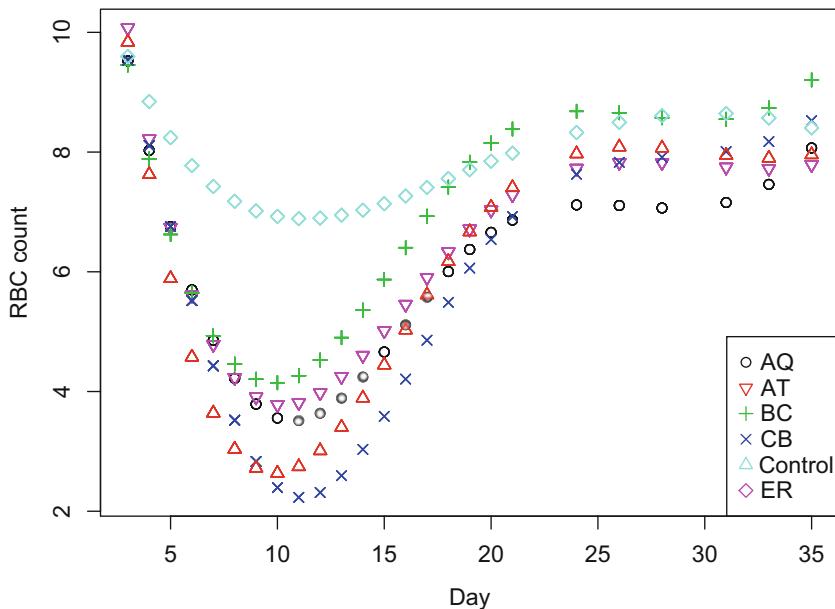


Fig. 15.7 Predicted and observed for the repeated measures RBC data using a spline model in time

15.4 *B. bronchiseptica* in Rabbits

Bordetella bronchiseptica is a respiratory infection of a range of mammals (e.g., Bjørnstad and Harvill 2005). Its congeners, *B. pertussis* and *B. parapertussis*, cause whooping cough in humans, but *B. bronchiseptica* is usually relatively asymptomatic (though it can cause snuffles in rabbits and kennel cough in dogs). The data comes from a commercial rabbitry which breeds NZW rabbits to study transmission paths in the colony. The data is from the same study as we used to study the age-specific force of infection in Sect. 4.3. Nasal swabs of female rabbits and their young were taken at weaning (~ 4 weeks old). A total of 86 does and 408 kits were included in the study (Long et al. 2010).

```
data(litter)
```

To investigate if (a) offspring of the infected mothers have an increased instantaneous risk of becoming infected and (b) if offspring of the same litter tended to have the same infection fate because of within-litter transmission, we use a random effect (generalized linear mixed model, GLMM) logistic regression, with litter as a random effect. We first do some data formatting.

```
tdat=data.frame(lsize=as.vector(table(litter$Litter)),
  Litter=names(table(litter$Litter)),
  anysick=sapply(split(litter$sick,litter$Litter),sum))
ldat=merge(litter, tdat, by="Litter")
ldat$othersick=ldat$any sick-ldat$sick
ldat$anyothersick=ldat$othersick>0
ldat$X=1:408
```

Here, the concern is with whether littermates share correlated fates. Unlike for spatial or temporal autocorrelation, there are no canned functions to quantify this correlation. However, following our discussion of autocorrelation in Sect. 13.3, it is easy to customize our own calculations. In the below, the first double-loop makes a sibling-sibling “contact-matrix,” `tmp`, that flags kits according to litter membership. After, `tmp2` rescales the binary `sick` vector that flags whether or not an animal was infected, and `tmp3` generates the correlation matrix. Finally `mean(tmp3 * tmp)` provides the within-litter autocorrelation in infection status averaged across all litters.

```
tmp=matrix(NA, ncol=length(ldat$Litter),
  nrow=length(ldat$Litter))
for(i in 1:length(ldat$Litter)){
  for(j in 1:length(ldat$Litter)){
    if(ldat$Litter[i]==ldat$Litter[j]){
      tmp[i,j]=1
```

```

        }
    }
}

diag(tmp) =NA
tmp2=scale(ldat$sick) [,1]
tmp3=outer(tmp2, tmp2, "*")
mean(tmp3*tmp, na.rm=TRUE)

## [1] 0.5302508

```

The within-litter correlation of 0.53 represents a substantial interdependence among littermates. Since the response variable is binary (infected vs noninfected) we cannot use `lme`. Instead we use the `lmer`-function from the `lme4`-package and specify using the “family” argument that the response is binomial. Using AICs we contrast the fit with within-litter correlation (`fitL`) with the fit that assumes independence (`fit0`); The appropriate independence fit is generated by declaring that each of the 408 individuals are in their own group (variable X in the data set).

```

require(lme4)
fitL=glmer(sick~msick+lsize+Facility+anyothersick+
  (1|Litter), family=binomial(), data=ldat)
fit0=glmer(sick~msick+lsize+Facility+anyothersick+
  (1|X), family=binomial(), data=ldat)
AIC(fitL, fit0)

##      df      AIC
## fitL  7 291.0263
## fit0  7 316.5853

```

The litter-dependent model is clearly best (no surprise given the strong empirical intra-litter correlation). The summary of the best model reveals that the key predictor of infection fate is whether or not a sibling was infected (`anyothersick`TRUE). The infection status of the mother was insignificant. The mixed-effect logistic regression thus reveals that the most important route of infection is likely to be sib-to-sib transmission (Long et al. 2010).

```

summary(fitL, corr = FALSE)

## Generalized linear mixed model fit by maximum
##   likelihood (Laplace Approximation) [glmerMod]
##   Family: binomial ( logit )
##   Formula:
##   sick ~ msick + lsize + Facility + anyothersick +
##     (1 | Litter)  Data: ldat

##

```

```
##      AIC      BIC  logLik deviance df.resid
##    291.0    319.1   -138.5     277.0      400
##
## Scaled residuals:
##      Min      1Q  Median      3Q     Max
## -1.7277 -0.3199 -0.1333 -0.0386 13.2186
##
## Random effects:
##   Groups Name      Variance Std.Dev.
##   Litter (Intercept) 2.077    1.441
## Number of obs: 407, groups: Litter, 52
##
## Fixed effects:
##                   Estimate Std. Error z value
## (Intercept)      -3.43236  2.32298 -1.478
## msick            2.74171  1.65447  1.657
## lsize           -0.37908  0.19153 -1.979
## FacilityT3       1.15833  0.80626  1.437
## FacilityT9      -0.01773  0.68553 -0.026
## anyothersickTRUE  2.88387  0.71564  4.030
##                  Pr(>|z|)
## (Intercept)      0.1395
## msick            0.0975 .
## lsize            0.0478 *
## FacilityT3       0.1508
## FacilityT9       0.9794
## anyothersickTRUE 5.58e-05 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Chapter 16

Quantifying In-Host Patterns



16.1 The Experiments

In addition to the mouse malaria data discussed in Sects. 7.7 and 15.3, we consider a coinfection study of FIV in cats (Roy et al. 2009). The experiment showed that disease in cats caused by infection with a virulent feline immunodeficiency viruses (FIV_f) can be attenuated by prior infection with strains of lower pathogenicity from cougars. The data are from twenty cats that were experimentally infected with two strains of FIV, the virulent house cat (*Felix*) strain (FIV_f) and a mild wild cougar (*Puma*) strain (FIV_p). On day 0, 10 cats were infected with FIV_p and 10 were sham inoculated. On day 28 five cats from each group were inoculated with FIV_f and the other ten cats were again sham inoculated. This resulted in four treatment groups: C (control, only sham inoculation), P (FIV_p on day 0 and sham on day 28), F (sham on day 0, FIV_f on day 28), and D (dual infection, FIV_p on day 0 and FIV_f on day 28). A variety of cytokines and cell counts that were thought to relate to protective immunity were measures approximately every 7 days. Details of the experiment can be found in Roy et al. (2009).

16.2 Data

For our FIV analysis we focus on the multivariate measures on days 31 and 59 to create two data sets `Day31` and `Day59`, 3 and 30 days respectively, after the second treatment (FIV_f infection). We strip some unnecessary columns 1, 14, 15, and 16 that are extraneous or were not measured on these days, remove lines with missing

This chapter uses the following R-packages: `ade4` and `MASS`.

values (using `na.omit`), and make sure that each row is labeled with the correct animal Id (using `dimnames`).

```
data(fiv)
Day31 = fiv[fiv$Day == 31, ]
dimnames(Day31)[[1]] = Day31$Id
Day31 = na.omit(Day31[, -c(1, 14, 15, 16)])
Day59 = fiv[fiv$Day == 59, ]
dimnames(Day59)[[1]] = Day59$Id
Day59 = na.omit(Day59[, -c(1, 14, 15, 16)])
```

For our malaria analysis we also strip some unnecessary columns 1, 3, 4, 7, 8, and 11 that are extraneous and focus on the red blood cell count (RBC).

```
data(SH9)
SH9RBC = SH9[, -c(1, 3, 4, 7, 8, 10, 11)]
```

In addition to the long-format used for the repeated measures analysis (Sect. 15.3), we need “wide” formatted data (denoted `...w`) for both the principal component analysis (PCA) and linear discriminant analysis (LDA) we will use to study the dynamics. We make the wide-formatted data using `reshape`. The `-seq(4,50,by=2)` strips extraneous columns generated by the `reshape`. The `names(...)[2] = "Treatment"` renames column 2.

```
SH9RBCw = reshape(SH9RBC, idvar = "Ind2",
                   direction = "wide", timevar="Day")
SH9RBCw=SH9RBCw[, -seq(4,50,by=2)]
names(SH9RBCw)[2] = "Treatment"
```

16.3 PCA of FIV Day 31 and 59 Data

The FIV data has counts of various effector cells (lymphocytes, neutrophils, CD4, CD8B, and CD25), virus load (provirus and overall viremia), and measurements on a number of cytokines (IFN γ , IL-4, IL-10, IL-12, TNF- α). The goal of the experiment was to elucidate what immunological conditions best distinguished sever from attenuated infections. The `ade4`-package has refined statistical and graphical methods to explore multivariate patterns. According to the “French protocol” (Dray and Dufour 2007), as implemented in the `ade4`-package, biplot-like decompositions are referred to as “duality-diagrams” (because of the arrows and points); Thus the naming of `dudi.pca` for principal component analysis. We use the `dudi.pca` function to elaborate on the biplot. By providing an explicit “group” annotation we can add group means as well as group ellipses (which reflect within-group variability) to the biplot using the `s.class` function. The `add.scatter.eig` function adds the eigenvalue histogram to the bottom right corner that shows the relative importance of each PCA axis (Fig. 16.1).

```

require(ade4)
pca31=dudi.pca(Day31[,1:11], scannf = FALSE, nf = 5)
#select 5 axes
groups=Day31$Treatment
s.arrow(dfxy=pca31$co[,1:2]*8, ylim=c(-7,9),
        sub="Day 31", possub="topleft", csub=2)
s.class(dfxy=pca31$li[,1:2], fac=groups, cellipse=2,
        axesell=FALSE, cstar=0 , col=c(2:5), add.plot=TRUE)
add.scatter.eig(pca31$eig, xax=1, yax=2,
                posi="bottomright")

```

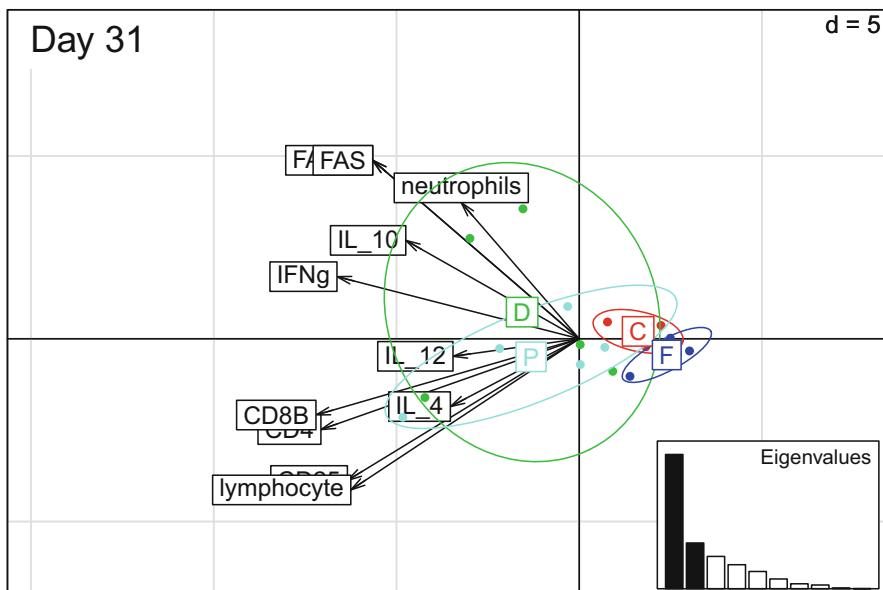


Fig. 16.1 A biplot of in-host measurement in the FIV experiment on day 31 with group ellipses and eigenvalues superimposed

On Day 59 patterns are starting to resolve and treatment units are starting to separate with FIV_f infected cats having the lowest white blood cell counts (Fig. 16.2).

```

pca59=dudi.pca(Day59[,1:11], scannf = FALSE, nf = 5)
groups=Day59$Treatment
s.arrow(dfxy=pca59$co[,1:2]*8, ylim=c(-7,9),
        sub="Day 59", possub="topleft", csub=2)

```

```
s.class(dfxy=pca59$li[,1:2], fac=groups, cellipse=2,
        axesell=FALSE, cstar=0 , col=c(2:5), add.plot=TRUE)
add.scatter.eig(pca59$eig, xax=1, yax=2,
                posi="bottomright")
```

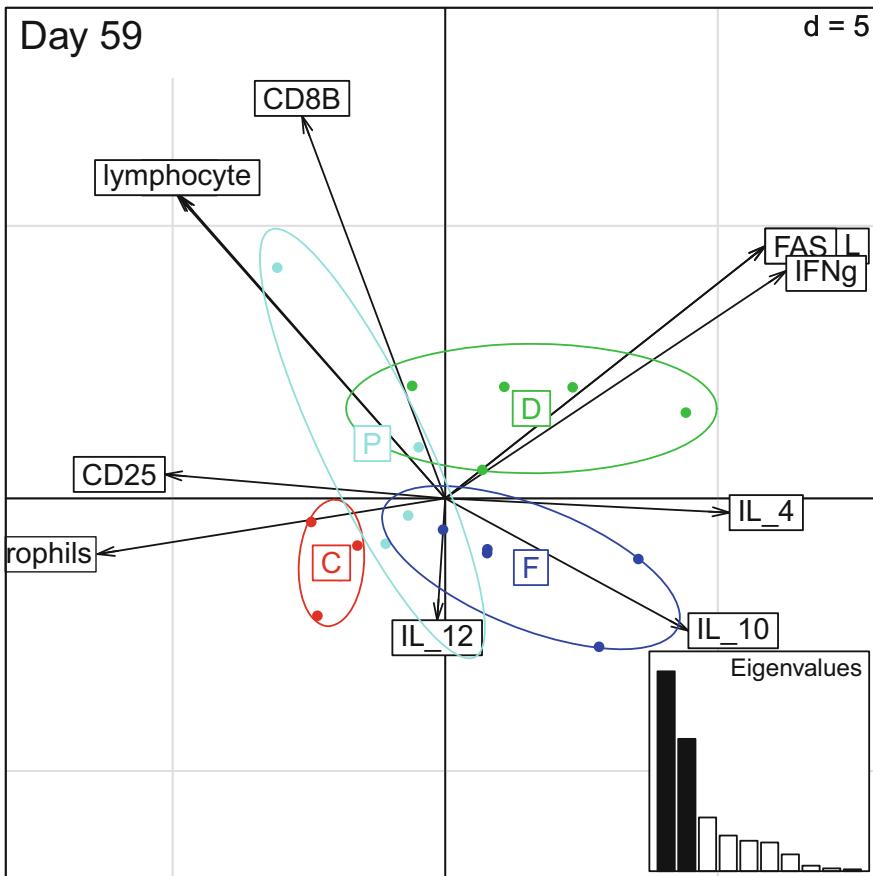


Fig. 16.2 A biplot of in-host measurement in the FIV experiment on day 59 with group ellipses and eigenvalues superimposed

16.4 LDA of FIV Day 31 and 59 Data

As opposed to PCA which broadly explores the overall variability in the multivariate data, discriminant analysis explicitly considers “group membership” (such as experimental treatment or other types of grouping) and asks what linear combination of response variables (a kin to the components in PCA) allows for the best discrimina-

tion among groups. The MASS-package has the `lda`-function to do such analysis. Since the variables are heterogeneous we normalize each prior to the analysis by applying the `scale` function to each of the first eleven columns of the data set.

```
require(MASS)
Day31sc = Day31
Day31sc[, 1:11] = apply(Day31[, 1:11], 2, scale)
```

The `lda`-function uses the group response formulation as its argument. The LDA plot depicts the discrimination among the groups along the discriminant axes (Fig. 16.3).

```
lda31 = lda(Treatment ~ CD4 + CD8B + CD25 + FAS +
             IFNg + IL_10 + IL_12 + lymphocyte + neutrophils +
             TNF_a, data = Day31sc)
plot(lda31)
```

Figure 16.3 shows how discriminant axis 1 clearly discriminates between the Dual (D)/Cougar (P) and the Control (C)/Feline (F) groups. Axis 2 separates the Dual (D) group from the Cougar (P) group. Axis 3 provides imperfect separation between the Control (C) group and the Feline (F) group. We can further check how the predicted LDA group assignments compare to the true treatment groupings:

```
pr = predict(lda31, method = "plug-in")$class
table(pr, Day31sc$Treatment)

##
## pr C D F P
##   C 2 0 1 0
##   D 0 5 0 0
##   F 1 0 3 0
##   P 0 0 0 5
```

For the most part the discrimination is good, but as Fig. 16.3 suggests there is some difficulty in discriminating between the C and F group on day 31; There is one misclassification among the groups.

To see how the group-informed LDA ordination differs from the PCA we can represent the LDA analysis as a biplot (Fig. 16.4). (The first two lines in the below code calculates the coordinates of each cat along the first two LDA axes to be compatible with the ADE4-package.) The discrimination is largely along LDA axis one.

```
ld1=as.matrix(Day31sc[, attr(lda31$terms,
    "term.labels")])%*%matrix(lda31$scaling[,1], ncol=1)
ld2=as.matrix(Day31sc[, attr(lda31$terms,
    "term.labels")])%*%matrix(lda31$scaling[,2], ncol=1)
groups=Day31$Treatment
```

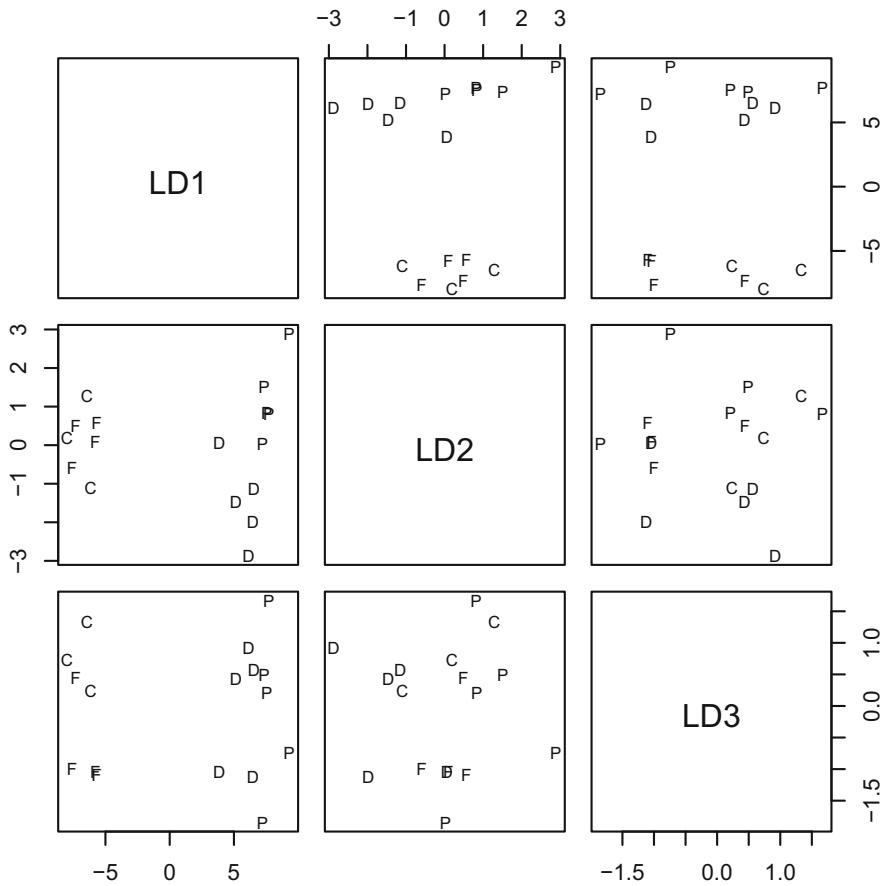


Fig. 16.3 The LDA of in-host measurement in the FIV experiment on day 31

```

contribs = lda31$svd/sum(lda31$svd)
s.arrow(dfxy=lda31$scaling[,1:2], sub="Day 31",
        possub="topleft", csub=2)
s.class(dfxy=cbind(ld1, ld2)*2.5, fac=groups,
        cellipse=2, axesell=FALSE, cstar=0,
        col=c(2:5), add.plot=TRUE)
add.scatter.eig(contribs, xax=1, yax=2,
               posi="bottomright")

```

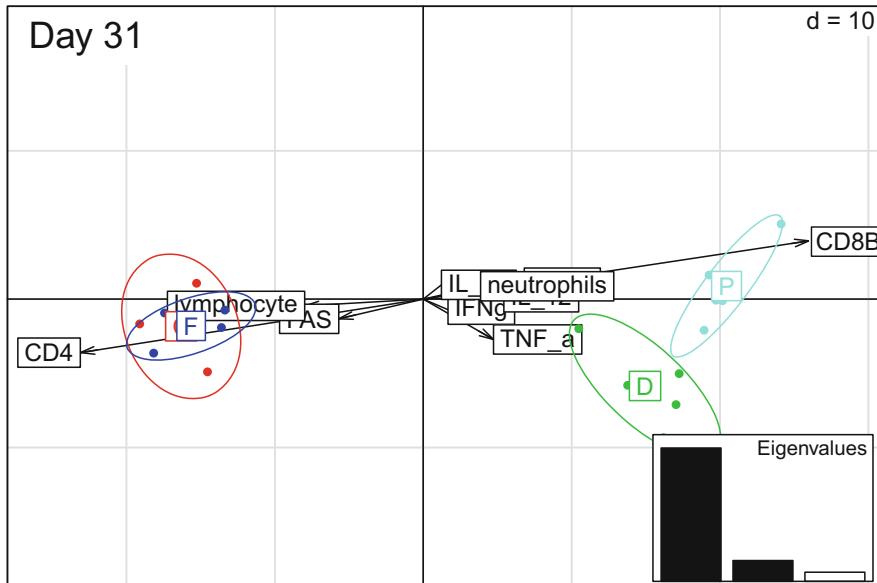


Fig. 16.4 The LDA of day 31 as a biplot

We repeat the analysis for the data from day 59 to see that discrimination among all four groups are very good by this time (Fig. 16.5). The linear discriminant (LD) 1 separates treatments C from D/F and P and LD2 separates F from the other treatments.

```
Day59sc=Day59
Day59sc[,1:11]=apply(Day59[,1:11],2,scale)
lda59 = lda(Treatment ~ CD4 + CD8B + CD25 + FAS +
    IFNg + IL_10 + IL_12 + lymphocyte + neutrophils +
    TNF_a, data = Day59sc)
pr=predict(lda59, method="plug-in")$class
table(pr, Day59sc$Treatment)

##
## pr  C D F P
##   C 5 0 0 0
##   D 0 5 0 0
##   F 0 0 5 0
##   P 0 0 0 4

ld1=as.matrix(Day59sc[,attr(lda59$terms,
    "term.labels" )])%*%matrix(lda59$scaling[,1], ncol=1)
ld2=as.matrix(Day59sc[,attr(lda59$terms,
    "term.labels" )])%*%matrix(lda59$scaling[,2], ncol=1)
```

```
groups=Day59$Treatment
```

```
contribs = lda59$svd/sum(lda59$svd)
s.arrow(dfxy=lda59$scaling[,1:2], sub="Day 59",
    possub="topleft", csub=2)
s.class(dfxy=cbind(l1d1, l2d2), fac=groups, cellipse=2,
    axesell=FALSE, cstar=0, col=c(2:5), add.plot=TRUE)
add.scatter.eig(contribs, xax=1, yax=2,
    posi="bottomright")
```

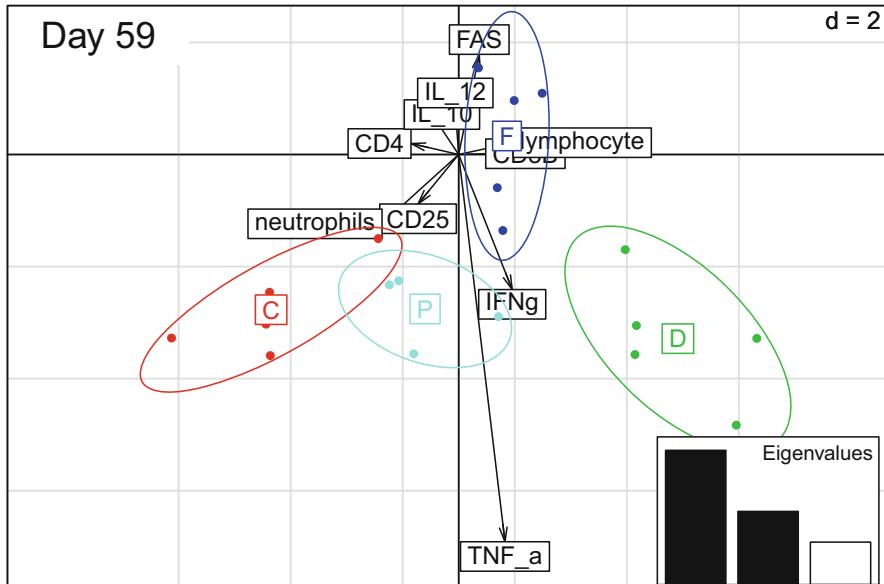


Fig. 16.5 The LDA of day 59 as a biplot

The severe disease (treatment F) is associated with reduction in counts of several cell types and modulation of the expression of various cytokines.

16.5 MANOVA of FIV Day 59 Data

In addition to the exploratory analysis provided by PCA and LDA we may also want to do a formal multivariate test between our treatment groups. The most traditional approach is through the use of multivariate analysis of variance (manova). The `manova`-function has many test options—The Hotelling T^2 is the multivariate version of the t-test. According to the R help pages, the Pillai-Bartlett statistic

is recommended by Hand and Taylor (1987) and is the default. There are many assumptions involved (including multivariate normality).

```

Y=cbind(Day59sc$CD4, Day59sc$CD8B, Day59sc$CD25,
         Day59sc$FAS, Day59sc$IFNg, Day59sc$IL_10,
         Day59sc$IL_12, Day59sc$lymphocyte,
         Day59sc$neutrophils, Day59sc$TNF_a)
X=Day59$Treatment
mova59=manova(Y~X)
summary(mova59, test="Pillai")

##                 Df Pillai approx F num Df den Df
## X              3 2.1078   1.8901     30    24
## Residuals    15
##                  Pr(>F)
## X            0.05676 .
## Residuals
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

16.6 PCA of Mouse Malaria

A preliminary PCA of the RBC time series reveals that the fate of the animals completely dominates the patterns since RBCs were scored as 0 after death (Fig. 16.6).

```

require(ade4)
dead=ifelse(SH9RBCw[,27]==0, "dead", "alive")
pcaRBC=dudi.pca(SH9RBCw[,3:27], scale=FALSE,
      scannf = FALSE, nf = 5)
s.arrow(dfxy=pcaRBC$co[,1:2]*3, xlim=c(-10, 10),
        ylim=c(-5,5), sub="RBC", possub="topleft", csub=2)
s.class(dfxy=pcaRBC$li[,1:2]*.3, fac=as.factor(dead),
        cellipse=2, axesell=FALSE, cstar=0 ,
        col=c(2:7), add.plot=TRUE)
add.scatter.eig(pcaRBC$eig, xax=1, yax=2,
               posi="bottomright")

```

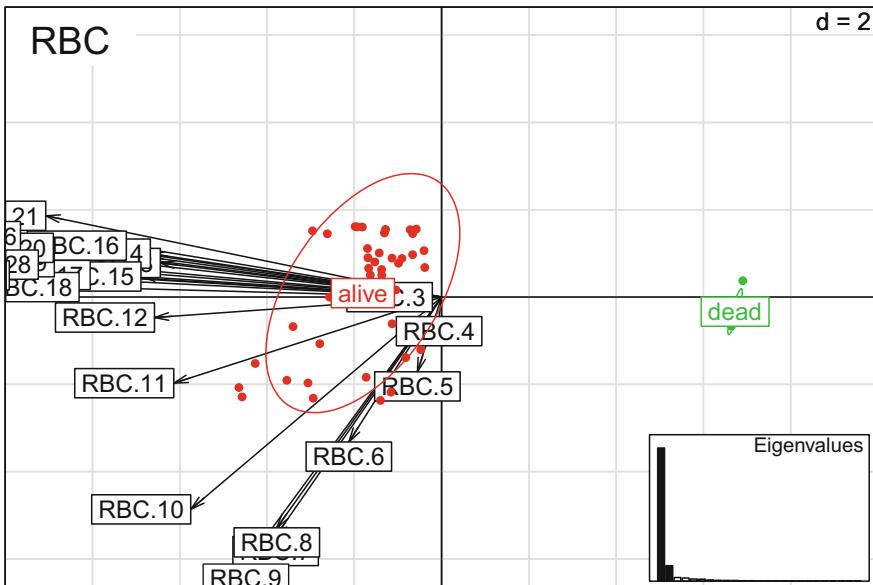


Fig. 16.6 The biplot of the RBC time series of the mouse malaria experiment

We therefore omit the 11 animals that died and redo the analysis (but note that these were nonrandom with respect to treatment; the dead were 7 CB, 2 AT, 1 BC, 0 AQ, and 0 control).¹

```
SH9RBCw2=SH9RBCw [dead=="alive",]
groups=SH9RBCw2$Treatment
pcaRBC=dudi.pca(SH9RBCw2[,3:27], scale=FALSE, scannf =
  FALSE, nf = 5)
s.arrow(dfxy=pcaRBC$co[,1:2]*3, xlim=c(-4,9),
  ylim=c(-5,5), sub="RBC", possub="topleft", csub=2)
s.class(dfxy=pcaRBC$li[,1:2]*.3, fac=groups, cellipse=2,
  axesell=FALSE, cstarc=0, col=c(2:7), add.plot=TRUE)
add.scatter.eig(pcaRBC$eig, xax=1, yax=2,
  posi="bottomright")
```

All the arrows are pointing in the same direction for axis one (Fig. 16.7). This axis is therefore broadly a “means” effect, meaning that individuals with more positive axis one scores overall tend to have more RBCs. Clearly the main driver of this variation is control versus treatment animals. There is further some level of separation among the treatment animals along axis two, with BC generally having negative values.

¹ An approach that uses all available data would be to code dead RBCs as NAs and do a PCA with missing data using nonlinear iterative partial least-squares (`nipals`) as done by Roy et al. (2009).

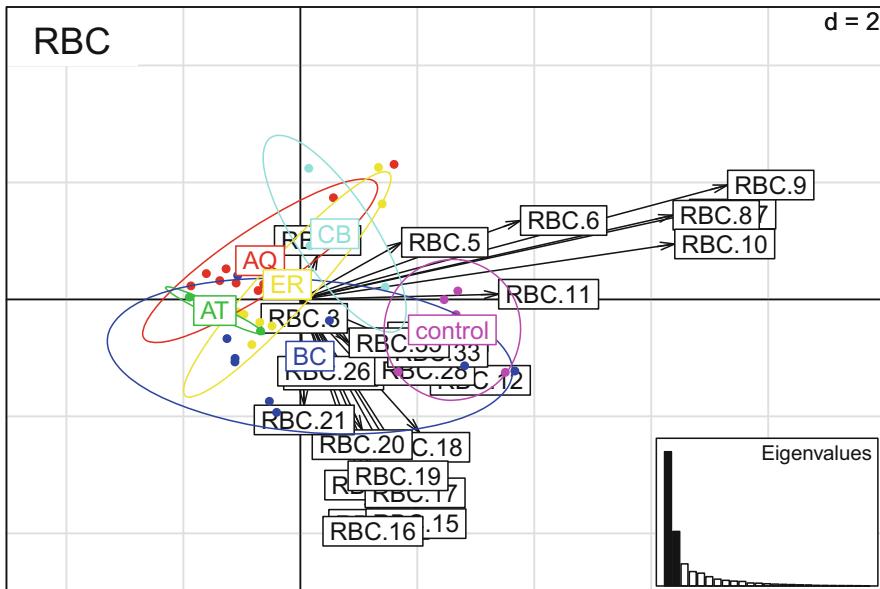


Fig. 16.7 The biplot of the RBC time series of the mouse malaria experiment excluding animals that died

16.7 FDA of Mouse Malaria

We can get some deeper insights into the differences revealed by the PCA by considering how the mouse data is of a “functional” nature. That is, we can consider each of the time series of RBC counts as sampled along a curve through time. We can ask how each curve can be thought of as being generated by adding or subtracting underlying component curves. Generally speaking, this multivariate approach is referred to as functional data analysis (FDA; Ramsay and Silverman 1997).

While specialized packages exist, we can treat our PCA as a simple FDA by considering the loadings along each axis to comprise a component time series (a so-called “empirical orthogonal function,” EOF), and the score for each individual as a weight of how much of that EOF to add or subtract to reconstitute the data. Figure 16.8 depicts the loadings of axis one and two as EOFs along the top row. The bottom row shows how adding or subtracting—corresponding to having positive or negative scores—these EOFs modulates the shape of the overall average curve among all experimental animals.

```
par(mfrow=c(1, 2))
#Gets the experimental days
day=unique(SH9$Day)
#Calculate the average time series
avg=apply(SH9RBCw2[,3:27], 2, mean)
```

```

plot(day, avg, type="b", ylim=range(SH9RBCw[, 3:27]),
     ylab = "RBC", xlab="Day")
title("Mean +/- 1 SD eof 1")
lines(day, avg+1*pcaRBC$co[,1], col=2,
      type="b", pch="+")
lines(day, avg-1*pcaRBC$co[,1], col=2,
      type="b", pch="-")
plot(day, avg, type="b", ylim=range(SH9RBCw[, 3:27]),
     ylab = "RBC", xlab="Day")
title("Mean +/- 1 SD eof 2")
lines(day, avg+1*pcaRBC$co[,2], col=2,
      type="b", pch="+")
lines(day, avg-1*pcaRBC$co[,2], col=2,
      type="b", pch="-")

```

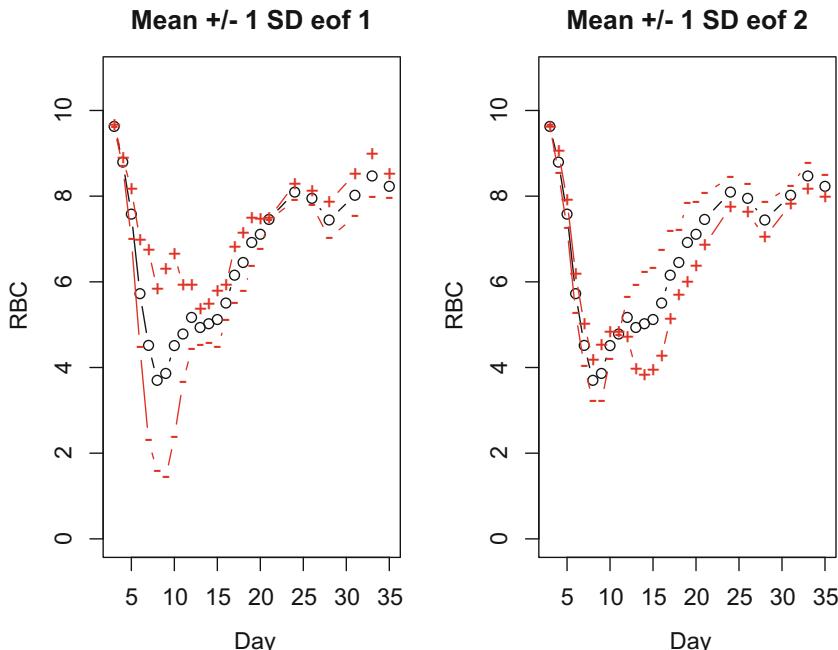


Fig. 16.8 The PCA of the RBC time series represented as a functional data analysis

The analysis offers some interesting insights. As previously suggested, axis one measures the overall anemia. Animals with positive scores experience less anemia. Axis two, in contrast, is more interesting as it reveals that the second most important pattern broadly distinguishes between animals that have peak anemia before day 10 (negative scores; broadly comprised of individuals infected with the BC clone) *versus* the other more slowly progressing infections (positive scores) that have peak

anemia around day 15. To confirm our interpretation we plot the actual time series for the 10 most extreme mice along EOF1 and EOF2 axes (Fig. 16.9).

```
par(mfrow=c(1, 2))
so=order(pcarBC$li[, 1])
plot(day, t(SH9RBCw2[so[1], 3:27]), type="l", ylab="RBC",
     xlab="Day")
for (i in 1:5) lines(day, t(SH9RBCw2[so[i], 3:27]))
for (i in 36:41) lines(day, t(SH9RBCw2[so[i], 3:27]),
                        col=2, lty=2)
so=order(pcarBC$li[, 2])
plot(day, t(SH9RBCw2[so[1], 3:27]), type="l", ylab="RBC",
     xlab="Day")
for (i in 1:5) lines(day, t(SH9RBCw2[so[i], 3:27]))
for (i in 36:41) lines(day, t(SH9RBCw2[so[i], 3:27]),
                        col=2, lty=2)
```

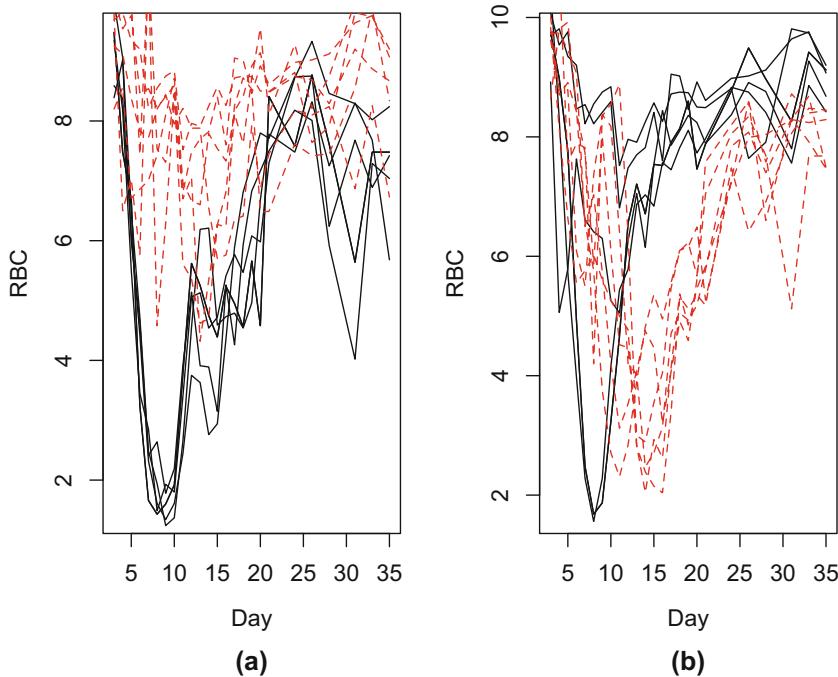


Fig. 16.9 The RBC time series for the five animals with highest (lowest) scores along the (a) first and (b) second axes. Black lines represent the five mice with the most negative scores and red dashed lines represent the mice with the most positive scores

References

- Abbey, H. (1952). An examination of the reed-frost theory of epidemics. *Human biology*, 24(3), 201.
- Abbott, K. C., & Dwyer, G. (2008). Using mechanistic models to understand synchrony in forest insect populations: The North American gypsy moth as a case study. *The American Naturalist*, 172(5), 613–624.
- Aitkin, M. A., Francis, B., & Hinde, J. (2005). *Statistical modelling in GLIM 4* (Vol. 32). New York, NY: Oxford University Press.
- Althaus, C. (2014). Estimating the reproduction number of ebola virus (EBOV) during the 2014 outbreak in West Africa. *PLoS Currents*, 6.
- Althouse, B. M., & Scarpino, S. V. (2015). Asymptomatic transmission and the resurgence of bordetella pertussis. *BMC Medicine*, 13(1), 146.
- Altizer, S., Dobson, A., Hosseini, P., Hudson, P., Pascual, M., & Rohani, P. (2006). Seasonality and the dynamics of infectious diseases. *Ecology Letters*, 9(4), 467–484.
- Anderson, R., Jackson, H., May, R., & Smith, A. (1981). Population dynamics of fox rabies in Europe. *Nature*, 289(5800), 765–771.
- Anderson, R. M., & May, R. M. (1982). Directly transmitted infectious diseases: Control by vaccination. *Science*, 215, 1053–1060.
- Anderson, R. M., & May, R. M. (1991). *Infectious diseases of humans: Dynamics and control*. Oxford: Oxford University Press.
- Anselin, L. (1995). Local indicators of spatial association - LISA. *Geographical Analysis*, 27(2), 93–115.
- Antonovics, J. (2004). Long-term study of a plant-pathogen metapopulation. In I. Hanski & O. Gaggiotti (Eds.), *Ecology, genetics, and evolution of metapopulations* (pp. 471–488). Amsterdam: Elsevier.
- Aron, J. L., & Schwartz, I. B. (1984). Seasonality and period-doubling bifurcations in an epidemic model. *Journal of Theoretical Biology*, 110(4), 665–679.

- Axelsen, J. B., Yaari, R., Grenfell, B. T., & Stone, L. (2014). Multiannual forecasting of seasonal influenza dynamics reveals climatic and evolutionary drivers. *Proceedings of the National Academy of Sciences*, 111(26), 9538–9542.
- Bailey, B. A., Ellner, S., & Nychka, D. W. (1997). Chaos with confidence: Asymptotics and applications of local lyapunov exponents. *Nonlinear dynamics and time series: Building a bridge between the natural and statistical sciences* (pp. 115–133). Providence, RI: American Mathematical Society.
- Bailey, N. T. J. (1956). On estimating the latent and infectious periods of measles: I. Families with two susceptibles only. *Biometrika*, 43(1/2), 15–22.
- Bailey, N. T. J. (1957). *The mathematical theory of epidemics*. London: Griffin.
- Bailey, N. T. J., & Alff-Steinberger, C. (1970). Improvements in the estimation of the latent and infectious periods of a contagious disease. *Biometrika*, 57(1), 141–153.
- Bansal, S., Grenfell, B. T., & Meyers, L. A. (2007). When individual behaviour matters: Homogeneous and network models in epidemiology. *Journal of the Royal Society Interface*, 4(16), 879–891.
- Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439), 509–512.
- Barbour, A. & Mollison, D. (1990). Epidemics and random graphs. In *Stochastic processes in epidemic theory* (pp. 86–89). Berlin: Springer.
- Bartlett, M. (1956). Deterministic and stochastic models for recurrent epidemics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 4, p. 109).
- Bartlett, M. S. (1960a). The critical community size for measles in the U. S. *Journal of Royal Statistical Society A*, 123, 37–44.
- Bartlett, M. S. (1960b). *Stochastic population models in ecology and epidemiology*. New York: Wiley.
- Bascompte, J., & Solé, R. V. (1995). Rethinking complexity: Modelling spatiotemporal dynamics in ecology. *Trends in Ecology & Evolution*, 10(9), 361–366.
- Bjørnstad, O. N., & Bascompte, J. (2001). Synchrony and second-order spatial correlation in host-parasitoid systems. *Journal of Animal Ecology*, 70(6), 924–933.
- Bjørnstad, O. N., & Bolker, B. (2000). Canonical functions for dispersal-induced synchrony. *Proceedings of the Royal Society of London B: Biological Sciences*, 267(1454), 1787–1794.
- Bjørnstad, O. N., & Falck, W. (2001). Nonparametric spatial covariance functions: Estimation and testing. *Environmental and Ecological Statistics*, 8(1), 53–70.
- Bjørnstad, O. N., Finkenstadt, B. F., & Grenfell, B. T. (2002a). Dynamics of measles epidemics: Estimating scaling of transmission rates using a time series sir model. *Ecological Monographs*, 72(2), 169–184.
- Bjørnstad, O. N., & Grenfell, B. T. (2001). Noisy clockwork: Time series analysis of population fluctuations in animals. *Science*, 293(5530), 638–643.
- Bjørnstad, O. N., & Harvill, E. T. (2005). Evolution and emergence of bordetella in humans. *Trends in Microbiology*, 13(8), 355–359.

- Bjørnstad, O. N., Ims, R. A., & Lambin, X. (1999b). Spatial population dynamics: Analyzing patterns and processes of population synchrony. *Trends in Ecology & Evolution*, 14(11), 427–432.
- Bjørnstad, O. N., Nelson, W. A., & Tobin, P. C. (2016). Developmental synchrony in multivoltine insects: Generation separation versus smearing. *Population Ecology*, 58(4), 479–491.
- Bjørnstad, O. N., Nisbet, R. M., & Fromentin, J.-M. (2004). Trends and cohort resonant effects in age-structured populations. *Journal of Animal Ecology*, 73(6), 1157–1167.
- Bjørnstad, O. N., Peltonen, M., Liebhold, A. M., & Baltensweiler, W. (2002b). Waves of larch budmoth outbreaks in the European Alps. *Science*, 298(5595), 1020–1023.
- Bjørnstad, O. N., Robinet, C., & Liebhold, A. M. (2010). Geographic variation in North American gypsy moth cycles: Subharmonics, generalist predators, and spatial coupling. *Ecology*, 91(1), 106–118.
- Bjørnstad, O. N., Sait, S. M., Stenseth, N. C., Thompson, D. J., & Begon, M. (2001). The impact of specialized enemies on the dimensionality of host dynamics. *Nature*, 409(6823), 1001.
- Bjørnstad, O. N., Stenseth, N. C., & Saitoh, T. (1999a). Synchrony and scaling in dynamics of voles and mice in Northern Japan. *Ecology*, 80(2), 622–637.
- Bjørnstad, O. N., & Viboud, C. (2016). Timing and periodicity of influenza epidemics. *Proceedings of the National Academy of Sciences*, 113, 201616052.
- Black, F. (1959). Measles antibodies in the population of New Haven, Connecticut. *Journal of Immunology*, 83, 74–83.
- Blumberg, S., & Lloyd-Smith, J. O. (2013). Comparing methods for estimating r_0 from the size distribution of subcritical transmission chains. *Epidemics*, 5(3), 131–145.
- Blythe, S., Nisbet, R., & Gurney, W. (1984). The dynamics of population models with distributed maturation periods. *Theoretical Population Biology*, 25(3), 289–311.
- Bobashev, G. V., Ellner, S. P., Nychka, D. W., & Grenfell, B. T. (2000). Reconstructing susceptible and recruitment dynamics from measles epidemic data. *Mathematical Population Studies*, 8(1), 1–29.
- Bolker, B., & Grenfell, B. (1993). Chaos and biological complexity in measles dynamics. *Proceedings of the Royal Society of London B*, 251(1330), 75–81.
- Bolker, B. M. (2008). *Ecological models and data in R*. Princeton: Princeton University Press.
- Breda, D., Diekmann, O., De Graaf, W., Pugliese, A., & Vermiglio, R. (2012). On the formulation of epidemic models (an appraisal of Kermack and Mckendrick). *Journal of Biological Dynamics*, 6(Suppl. 2), 103–117.
- Broadbent, S., & Kendall, D. G. (1953). The random walk of trichostrongylus retortaeformis. *BiometRICS*, 9(4), 460–466.
- Bukreyev, A., Yang, L., Fricke, J., Cheng, L., Ward, J. M., Murphy, B. R., et al. (2008). The secreted form of respiratory syncytial virus g glycoprotein helps the virus evade antibody-mediated restriction of replication by acting as an antigen

- decoy and through effects on fc receptor-bearing leukocytes. *Journal of Virology*, 82(24), 12191–12204.
- Burnett, T. (1958). A model of host-parasite interaction. In *Proceedings of the 10th International Congress, Entomology* (Vol. 2, pp. 679–686).
- Carrat, F., Vergu, E., Ferguson, N. M., Lemaitre, M., Cauchemez, S., Leach, S., et al. (2008). Time lines of infection and disease in human influenza: A review of volunteer challenge studies. *American Journal of Epidemiology*, 167(7), 775–785.
- Cheng, B., & Tong, H. (1992). On consistent nonparametric order determination and chaos. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54, 427–449.
- Childs, J. E., Curns, A. T., Dey, M. E., Real, L. A., Feinstein, L., Bjørnstad, O. N., et al. (2000). Predicting the local dynamics of epizootic rabies among raccoons in the united states. *Proceedings of the National Academy of Sciences*, 97(25), 13666–13671.
- Choi, K., & Thacker, S. B. (1981). An evaluation of influenza mortality surveillance, 1962–1979: I. time series forecasts of expected pneumonia and influenza deaths. *American Journal of Epidemiology*, 113(3), 215–226.
- Clark, D. A., & Clark, D. B. (1984). Spacing dynamics of a tropical rain forest tree: Evaluation of the Janzen-Connell model. *The American Naturalist*, 124(6), 769–788.
- Clark, J. S. (1998). Why trees migrate so fast: confronting theory with dispersal biology and the paleorecord. *The American Naturalist*, 152(2), 204–224.
- Clark, J. S., & Bjørnstad, O. N. (2004). Population time series: Process variability, observation errors, missing values, lags, and hidden states. *Ecology*, 85(11), 3140–3150.
- Codeço, C. T. (2001). Endemic and epidemic dynamics of cholera: The role of the aquatic reservoir. *BMC Infectious Diseases*, 1, 1.
- Cowling, B. J., Fang, V. J., Riley, S., Peiris, J. S. M., & Leung, G. M. (2009). Estimation of the serial interval of influenza. *Epidemiology*, 20(3), 344–347.
- Coyne, M., Smith, G., & McAllister, F. (1989). Mathematic model for the population biology of rabies in raccoons in the mid-Atlantic states. *American Journal of Veterinary Research*, 50(12), 2148–2154.
- Cummings, D. A., Irizarry, R. A., Huang, N. E., Endy, T. P., Nisalak, A., Ungchusak, K., et al. (2004). Travelling waves in the occurrence of dengue haemorrhagic fever in Thailand. *Nature*, 427(6972), 344–347.
- Cushing, J., Dennis, B., Desharnais, R., & Costantino, R. (1998). Moving toward an unstable equilibrium: Saddle nodes in population systems. *Journal of Animal Ecology*, 67, 298–306.
- Dalziel, B. D., Bjørnstad, O. N., van Panhuis, W. G., Burke, D. S., Metcalf, C. J. E., & Grenfell, B. T. (2016). Persistent chaos of measles epidemics in the prevaccination United States caused by a small change in seasonal transmission patterns. *PLoS Computational Biology*, 12(2), e1004655.

- De, P., Singh, A. E., Wong, T., Yacoub, W., & Jolly, A. (2004). Sexual network analysis of a gonorrhoea outbreak. *Sexually Transmitted Infections*, 80(4), 280–285.
- De Castro, F., & Bolker, B. (2005). Mechanisms of disease-induced extinction. *Ecology Letters*, 8(1), 117–126.
- de Jong, M., Diekmann, O., & Heesterbeek, H. (1995). How does transmission of infection depend on population size? In D. Mollison (Ed.), *Epidemic models: Their structure and relation to data* (pp. 84–94). Cambridge: Cambridge University Press.
- de Valpine, P., Scranton, K., Knape, J., Ram, K., & Mills, N. J. (2014). The importance of individual developmental variation in stage-structured population models. *Ecology Letters*, 17(8), 1026–1038.
- Dennis, B., Desharnais, R. A., Cushing, J., Henson, S. M., & Costantino, R. (2003). Can noise induce chaos? *Oikos*, 102(2), 329–339.
- Diekmann, O., Heesterbeek, J. A. P., & Metz, J. A. J. (1990). On the definition and the computation of the basic reproduction ratio r_0 in models for infectious diseases in heterogeneous populations. *Journal of Mathematical Biology*, 28(4), 365–382.
- Dietz, K. (1993). The estimation of the basic reproduction number for infectious diseases. *Statistical Methods in Medical Research*, 2, 23–41.
- Dietz, K., & Heesterbeek, J. (2002). Daniel Bernoulli's epidemiological model revisited. *Mathematical Biosciences*, 180(1), 1–21.
- Dietz, K., & Schenzle, D. (1985). Proportionate mixing models for age-dependent infection transmission. *Journal of Mathematical Biology*, 22(1), 117–120.
- Dray, S., & Dufour, A.-B. (2007). The ade4 package: Implementing the duality diagram for ecologists. *Journal of Statistical Software*, 22(4), 1–20.
- Dushoff, J., Plotkin, J. B., Levin, S. A., & Earn, D. J. (2004). Dynamical resonance can account for seasonality of influenza epidemics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(48), 16915–16916.
- Dwyer, G., Dushoff, J., & Yee, S. H. (2004). The combined effects of pathogens and predators on insect outbreaks. *Nature*, 430(6997), 341–345.
- Earn, D. J., Levin, S. A., & Rohani, P. (2000a). Coherence and conservation. *Science*, 290(5495), 1360–1364.
- Earn, D. J., Rohani, P., Bolker, B. M., & Grenfell, B. T. (2000b). A simple model for complex dynamical transitions in epidemics. *Science*, 287(5453), 667–670.
- Eckmann, J.-P., & Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics*, 57(3), 617.
- Ellner, S., Bailey, B., Bobashev, G., Gallant, A., Grenfell, B., & Nychka, D. (1998). Noise and nonlinearity in measles epidemics: combining mechanistic and statistical approaches to population modeling. *The American Naturalist*, 151(5), 425–440.
- Ellner, S., & Turchin, P. (2005). When can noise induce chaos and why does it matter: A critique. *Oikos*, 111(3), 620–631.
- Ellner, S. P., & Guckenheimer, J. (2011). *Dynamic models in biology*. Princeton: Princeton University Press.

- Ellner, S. P., Seifu, Y., & Smith, R. H. (2002). Fitting population dynamic models to time-series data by gradient matching. *Ecology*, 83(8), 2256–2270.
- Erdős, P., & Rényi, A. (1959). On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6, 290–297.
- Erlander, S., & Stewart, N. F. (1990). *The gravity model in transportation analysis: Theory and extensions* (Vol. 3). Rancho Cordova, CA: VSP.
- Fan, J., Yao, Q., & Tong, H. (1996). Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1), 189–206.
- Ferguson, N. M., Anderson, R. M., & Garnett, G. P. (1996). Mass vaccination to control chickenpox: The influence of zoster. *Proceedings of the National Academy of Sciences*, 93(14), 7231–7235.
- Ferguson, N. M., Keeling, M. J., Edmunds, W. J., Gani, R., Grenfell, B. T., Anderson, R. M., & Leach, S. (2003). Planning for smallpox outbreaks. *Nature*, 425(6959), 681–685.
- Ferrari, M. J., Bansal, S., Meyers, L. A., & Bjørnstad, O. N. (2006a). Network frailty and the geometry of herd immunity. *Proceedings of the Royal Society of London B: Biological Sciences*, 273(1602), 2743–2748.
- Ferrari, M. J., Bjørnstad, O. N., & Dobson, A. P. (2005). Estimation and inference of r_0 of an infectious pathogen by a removal method. *Mathematical Biosciences*, 198(1), 14–26.
- Ferrari, M. J., Bjørnstad, O. N., Partain, J. L., & Antonovics, J. (2006b). A gravity model for the spread of a pollinator-borne plant pathogen. *The American Naturalist*, 168(3), 294–303.
- Ferrari, M. J., Djibo, A., Grais, R. F., Grenfell, B. T., & Bjørnstad, O. N. (2010). Episodic outbreaks bias estimates of age-specific force of infection: A corrected method using measles as an example. *Epidemiology and Infection*, 138(1), 108–116.
- Ferrari, M. J., Grais, R. F., Bharti, N., Conlan, A. J., Bjørnstad, O. N., Wolfson, L. J., et al. (2008). The dynamics of measles in sub-Saharan Africa. *Nature*, 451(7179), 679–684.
- Ferrari, M. J., Perkins, S. E., Pomeroy, L. W., & Bjørnstad, O. N. (2011). Pathogens, social networks, and the paradox of transmission scaling. *Interdisciplinary Perspectives on Infectious Diseases*, 2011, Article ID 267049, 10 pp.
- Fine, P. E. M., & Clarkson, J. A. (1982). Measles in england and wales. I. An analysis of factors underlying seasonal patterns. *International Journal of Epidemiology*, 11, 5–15.
- Finkenstädt, B. F., Bjørnstad, O. N., & Grenfell, B. T. (2002). A stochastic model for extinction and recurrence of epidemics: Estimation and inference for measles outbreaks. *Biostatistics*, 3(4), 493–510.
- Finkenstädt, B. F., & Grenfell, B. T. (2000). Time series modelling of childhood diseases: A dynamical systems approach. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(2), 187–205.
- Fotheringham, A. S. (1984). Spatial flows and spatial patterns. *Environment and Planning A*, 16(4), 529–543.

- Gammaitoni, L., Hänggi, P., Jung, P., & Marchesoni, F. (1998). Stochastic resonance. *Reviews of Modern Physics*, 70(1), 223.
- Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25), 2340–2361.
- Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4), 1716–1733.
- Glass, G. E., Cheek, J. E., Patz, J. A., Shields, T. M., Doyle, T. J., Thoroughman, D. A., et al. (2000). Using remotely sensed data to identify areas at risk for hantavirus pulmonary syndrome. *Emerging Infectious Diseases*, 6(3), 238.
- Glass, K., Xia, Y., & Grenfell, B. (2003). Interpreting time-series analyses for continuous-time biological models – measles as a case study. *Journal of Theoretical Biology*, 223(1), 19–25.
- Gog, J. R., Ballesteros, S., Viboud, C., Simonsen, L., Bjørnstad, O. N., Shaman, J., et al. (2014). Spatial transmission of 2009 pandemic influenza in the US. *PLoS Computational Biology*, 10(6), e1003635.
- Grais, R., Dubray, C., Gerstl, S., Guthmann, J., Djibo, A., Nargaye, K., et al. (2007). Unacceptably high mortality related to measles epidemics in niger, nigeria, and chad. *PLoS Medicine*, 4(1), e16.
- Grais, R. F., Conlan, A. J. K., Ferrari, M. J., Djibo, A., Le Menach, A., Bjørnstad, O. N., et al. (2008). Time is of the essence: Exploring a measles outbreak response vaccination in niamey, niger. *Journal of the Royal Society Interface*, 5(18), 67–74.
- Grenfell, B., & Anderson, R. (1989). Pertussis in england and wales: An investigation of transmission dynamics and control by mass vaccination. *Proceedings of the Royal Society of London B: Biological Sciences*, 236(1284), 213–252.
- Grenfell, B., Bjørnstad, O., & Kappey, J. (2001). Travelling waves and spatial hierarchies in measles epidemics. *Nature*, 414(6865), 716–723.
- Grenfell, B., & Harwood, J. (1997). (Meta)population dynamics of infectious diseases. *Trends in Ecology and Evolution*, 12(10), 395–399.
- Grenfell, B. T., & Anderson, R. M. (1985). The estimation of age-related rates of infection from case notifications and serological data. *Journal of Hygiene (Cambridge)*, 95, 419–436.
- Grenfell, B. T., Bjørnstad, O. N., & Finkenstadt, B. F. (2002). Dynamics of measles epidemics: Scaling noise, determinism, and predictability with the tsir model. *Ecological Monographs*, 72(2), 185–202.
- Gupta, S., Ferguson, N., & Anderson, R. (1998). Chaos, persistence, and evolution of strain structure in antigenically diverse infectious agents. *Science*, 280(5365), 912–915.
- Hall, P., & Patil, P. (1994). Properties of nonparametric estimators of autocovariance for stationary random fields. *Probability Theory and Related Fields*, 99(3), 399–424.
- Hand, D. J., & Taylor, C. C. (1987). *Multivariate analysis of variance and repeated measures: A practical approach for behavioural scientists* (Vol. 5). Boca Raton, FL: CRC Press.
- Hanski, I. (1994). A practical model of metapopulation dynamics. *Journal of Animal Ecology*, 63, 151–162.

- Härdle, W. (1990). *Applied nonparametric regression* (Vol. 19). Cambridge: Cambridge University Press.
- Hassell, M. P., Comins, H. N., & May, R. M. (1991). Spatial structure and chaos in insect population dynamics. *Nature*, 353(6341), 255–258.
- Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models* (Vol. 43). Boca Raton, FL: CRC Press.
- Heesterbeek, J., & Dietz, K. (1996). The concept of ro in epidemic theory. *Statistica Neerlandica*, 50(1), 89–110.
- Heisey, D. M., Joly, D. O., & Messier, F. (2006). The fitting of general force-of-infection models to wildlife disease prevalence data. *Ecology*, 87(9), 2356–2365.
- Hens, N., Aerts, M., Faes, C., Shkedy, Z., Lejeune, O., Van Damme, P., et al. (2010). Seventy-five years of estimating the force of infection from current status data. *Epidemiology and Infection*, 138(06), 802–812.
- Hope-Simpson, R. (1952). Infectiousness of communicable diseases in the household. *Lancet*, 2, 549–554.
- Hope-Simpson, R. (1981). The role of season in the epidemiology of influenza. *Journal of Hygiene*, 86(01), 35–47.
- Iacono, G. L., Cunningham, A. A., Fichet-Calvet, E., Garry, R. F., Grant, D. S., Khan, S. H., et al. (2015). Using modelling to disentangle the relative contributions of zoonotic and anthroponotic transmission: The case of lassa fever. *PLoS Neglected Tropical Diseases*, 9(1), e3398.
- Johnson, D. M., Bjørnstad, O. N., & Liebhold, A. M. (2004). Landscape geometry and travelling waves in the larch budmoth. *Ecology Letters*, 7(10), 967–974.
- Johnson, D. M., Büntgen, U., Frank, D. C., Kausrud, K., Haynes, K. J., Liebhold, A. M., et al. (2010). Climatic warming disrupts recurrent alpine insect outbreaks. *Proceedings of the National Academy of Sciences*, 107(47), 20576–20581.
- Jones, C. G., Ostfeld, R. S., Richard, M. P., Schauber, E. M., & Wolff, J. O. (1998). Chain reactions linking acorns to gypsy moth outbreaks and lyme disease risk. *Science*, 279(5353), 1023–1026.
- Kaneko, K. (1993). *Theory and applications of coupled map lattices* (Vol. 12). Chichester: Wiley.
- Keeling, M., Wilson, H., & Pacala, S. (2002). Deterministic limits to stochastic spatial models of natural enemies. *The American Naturalist*, 159(1), 57–80.
- Keeling, M. J., Bjørnstad, O. N., & Grenfell, B. T. (2004). Metapopulation dynamics of infectious diseases. In I. Hanski & O. Gaggiotti (Eds.), *Ecology, genetics, and evolution of metapopulations* (pp. 415–445). Amsterdam: Elsevier.
- Keeling, M. J., & Eames, K. T. (2005). Networks and epidemic models. *Journal of the Royal Society Interface*, 2(4), 295–307.
- Keeling, M. J., & Grenfell, B. (1997). Disease extinction and community size: Modeling the persistence of measles. *Science*, 275(5296), 65–67.
- Keeling, M. J., & Rohani, P. (2002). Estimating spatial coupling in epidemiological systems: A mechanistic approach. *Ecology Letters*, 5(1), 20–29.
- Keeling, M. J., & Rohani, P. (2008). *Modeling infectious diseases in humans and animals*. Princeton, NJ: Princeton University Press.

- Keeling, M. J., Rohani, P., & Grenfell, B. T. (2001). Seasonally forced disease dynamics explored as switching between attractors. *Physica D: Nonlinear Phenomena*, 148(3), 317–335.
- Keitt, T. H., Bjørnstad, O. N., Dixon, P. M., & Citron-Pousty, S. (2002). Accounting for spatial pattern when modeling organism-environment interactions. *Ecotherapy*, 25(5), 616–625.
- Kendall, B. E., Briggs, C. J., Murdoch, W. W., Turchin, P., Ellner, S. P., McCauley, E., et al. (1999). Why do populations cycle? A synthesis of statistical and mechanistic modeling approaches. *Ecology*, 80(6), 1789–1805.
- Kennedy, D. A., & Read, A. F. (2017). Why does drug resistance readily evolve but vaccine resistance does not? *Proceedings of the Royal Society B: Biological Sciences*, 284, 20162562.
- Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 115(772), 700–721.
- King, A. A., de Celles, M. D., Magpantay, F. M. G., & Rohani, P. (2015a). Avoidable errors in the modelling of outbreaks of emerging pathogens, with special reference to ebola. *Proceedings of the Royal Society B Biological Sciences*, 282(1806), 20150347.
- King, A. A., Ionides, E. L., Pascual, M., & Bouma, M. J. (2008). Inapparent infections and cholera dynamics. *Nature*, 454(7206), 877–880.
- King, A. A., Nguyen, D., & Ionides, E. L. (2015b). Statistical inference for partially observed markov processes via the R package pomp. ArXiv preprint, arXiv:1509.00503.
- Kirimanjeswara, G. S., Agosto, L. M., Kennett, M. J., Bjornstad, O. N., & Harvill, E. T. (2005). Pertussis toxin inhibits neutrophil recruitment to delay antibody-mediated clearance of bordetella pertussis. *The Journal of Clinical Investigation*, 115(12), 3594–3601.
- Koelle, K., Cobey, S., Grenfell, B., & Pascual, M. (2006). Epochal evolution shapes the phylodynamics of interpandemic influenza a (H3N2) in humans. *Science*, 314(5807), 1898–1903.
- Koenig, W. D. (1999). Spatial autocorrelation of ecological phenomena. *Trends in Ecology & Evolution*, 14(1), 22–26.
- Kooperberg, C., Stone, C. J., & Truong, Y. K. (1995). Logspline estimation of a possibly mixed spectral distribution. *Journal of Time Series Analysis*, 16(4), 359–388.
- Kot, M., Lewis, M. A., & van den Driessche, P. (1996). Dispersal data and the spread of invading organisms. *Ecology*, 77(7), 2027–2042.
- Kucharski, A. J., Conlan, A. J., & Eames, K. T. (2015). School's out: Seasonal variation in the movement patterns of school children. *PLoS One*, 10(6), e0128070.
- Lavine, J. S., King, A. A., Andreasen, V., & Bjørnstad, O. N. (2013). Immune boosting explains regime-shifts in prevaccine-era pertussis dynamics. *PLoS One*, 8(8), e72086.

- Lavine, J. S., King, A. A., & Bjørnstad, O. N. (2011). Natural immune boosting in pertussis dynamics and the potential for long-term vaccine failure. *Proceedings of the National Academy of Sciences*, 108(17), 7259–7264.
- Legendre, L., Frechette, M., & Legendre, P. (1981). The contingency periodogram: A method of identifying rhythms in series of nonmetric ecological data. *The Journal of Ecology*, 69, 965–979.
- Legendre, P. (1993). Spatial autocorrelation: Trouble or new paradigm? *Ecology*, 74(6), 1659–1673.
- Legendre, P., & Fortin, M. J. (1989). Spatial pattern and ecological analysis. *Vegetatio*, 80(2), 107–138.
- Legrand, J., Grais, R. F., Boelle, P. Y., Valleron, A. J., & Flahault, A. (2007). Understanding the dynamics of ebola epidemics. *Epidemiology and Infection*, 135(4), 610–621.
- Lipsitch, M., Cohen, T., Cooper, B., Robins, J. M., Ma, S., James, L., et al. (2003). Transmission dynamics and control of severe acute respiratory syndrome. *Science*, 300(5627), 1966–1970.
- Lloyd, A. L. (2001). Destabilization of epidemic models with the inclusion of realistic distributions of infectious periods. *Proceedings of the Royal Society of London B: Biological Sciences*, 268(1470), 985–993.
- Lloyd-Smith, J. O., George, D., Pepin, K. M., Pitzer, V. E., Pulliam, J. R. C., Dobson, A. P., et al. (2009). Epidemic dynamics at the human-animal interface. *Science*, 326(5958), 1362–1367.
- Loader, C. (2006). *Local regression and likelihood*. Berlin: Springer Science & Business Media.
- Lomb, N. R. (1976). Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space Science*, 39(2), 447–462.
- Long, G. H., Sinha, D., Read, A. F., Pritt, S., Kline, B., Harvill, E. T., et al. (2010). Identifying the age cohort responsible for transmission in a natural outbreak of bordetella bronchiseptica. *PLoS Pathogens*, 6(12), e1001224.
- Lowen, A. C., Mubareka, S., Steel, J., & Palese, P. (2007). Influenza virus transmission is dependent on relative humidity and temperature. *PLoS Pathogens*, 3(10), e151.
- Luis, A. D., Douglass, R. J., Mills, J. N., & Bjørnstad, O. N. (2015). Environmental fluctuations lead to predictability in sin nombre hantavirus outbreaks. *Ecology*, 96(6), 1691–1701.
- Mahmud, A., Metcalf, C., & Grenfell, B. (2017). Comparative dynamics, seasonality in transmission, and predictability of childhood infections in Mexico. *Epidemiology & Infection*, 145(3), 607–625.
- Mari, L., Bertuzzo, E., Righetto, L., Casagrandi, R., Gatto, M., Rodriguez-Iturbe, I., & Rinaldo, A. (2012). Modelling cholera epidemics: The role of waterways, human mobility and sanitation. *Journal of the Royal Society Interface*, 9(67), 376–388.
- Martinez-Bakker, M., Bakker, K. M., King, A. A., & Rohani, P. (2014). Human birth seasonality: Latitudinal gradient and interplay with childhood disease dynamics. *Proceedings of the Royal Society B*, 281, 20132438.

- May, R. M. (1978). Host-parasitoid systems in patchy environments: A phenomenological model. *The Journal of Animal Ecology*, 47, 833–844.
- McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models: Vol. 37. Monographs on statistics and applied probability* (2nd ed.). London: Chapman and Hall.
- Metcalf, C., Bjørnstad, O., Ferrari, M., Klepac, P., Bharti, N., Lopez-Gatell, H., et al. (2011a). The epidemiology of rubella in mexico: Seasonality, stochasticity and regional variation. *Epidemiology and Infection*, 139(07), 1029–1038.
- Metcalf, C., Graham, A., Huijben, S., Barclay, V., Long, G., Grenfell, B., et al. (2011b). Partitioning regulatory mechanisms of within-host malaria dynamics using the effective propagation number. *Science*, 333(6045), 984–988.
- Metcalf, C. J. E., & Barrett, A. (2016). Invasion dynamics of teratogenic infections in light of rubella control: Implications for zika virus. *PLoS Currents*, 8.
- Metcalf, C. J. E., Bjørnstad, O. N., Grenfell, B. T., & Andreasen, V. (2009). Seasonality and comparative dynamics of six childhood infections in pre-vaccination Copenhagen. *Proceedings of the Royal Society of London B: Biological Sciences*, 276(1676), 4111–4118.
- Metcalf, C. J. E., Munayco, C. V., Chowell, G., Grenfell, B. T., & Bjørnstad, O. N. (2011c). Rubella metapopulation dynamics and importance of spatial coupling to the risk of congenital rubella syndrome in Peru. *Journal of the Royal Society Interface*, 8(56), 369–376.
- Metz, J., & Diekmann, O. (1991). Exact finite dimensional representations of models for physiologically structured populations. I: The abstract foundations of linear chain trickery. *Differential equations with applications in biology, physics and engineering: Vol. 133. Lecture notes in pure and applied mathematics* (pp. 269–289). New York: Marcel Dekker.
- Mideo, N., Reece, S. E., Smith, A. L., & Metcalf, C. J. E. (2013). The cinderella syndrome: Why do malaria-infected cells burst at midnight? *Trends in Parasitology*, 29(1), 10–16.
- Miller, C. L., & Fletcher, W. (1976). Severity of notified whooping cough. *British Medical Journal*, 1(6002), 117–119.
- Miramontes, O., & Rohani, P. (1998). Intrinsically generated coloured noise in laboratory insect populations. *Proceedings of the Royal Society of London B: Biological Sciences*, 265(1398), 785–792.
- Mollison, D. (1991). Dependence of epidemic and population velocities on basic parameters. *Mathematical Biosciences*, 107(2), 255–287.
- Mossong, J., Hens, N., Jit, M., Beutels, P., Auranen, K., Mikolajczyk, R., et al. (2008). Social contacts and mixing patterns relevant to the spread of infectious diseases. *PLoS Medicine*, 5(3), e74.
- Muensch, H. (1959). *Catalytic models in epidemiology*. Cambridge, MA: Harvard University Press.
- Murdoch, W. W., Briggs, C. J., & Nisbet, R. M. (2003). *Consumer-resource dynamics* (Vol. 36). Princeton, NJ: Princeton University Press.
- Murdoch, W. W., Chesson, J., & Chesson, P. L. (1985). Biological control in theory and practice. *The American Naturalist*, 125(3), 344–366.

- Newman, M. E. (2002). Spread of epidemic disease on networks. *Physical Review E*, 66(1), 016128.
- Nicholson, A. J., & Bailey, V. A. (1935). The balance of animal populations. Part I. *Proceedings of the Zoological Society of London*, 105(3), 551–598.
- Nisbet, R. M., & Gurney, W. (1982). *Modelling fluctuating populations*. Chichester: John Wiley and Sons Limited.
- Panagiotopoulos, T., Berger, A., & Valassi-Adam, E. (1999). Increase in congenital rubella occurrence after immunisation in greece: Retrospective survey and systematic review. *British Medical Journal*, 319(7223), 1462–1467.
- Peel, A. J., Pulliam, J., Luis, A., Plowright, R., O’Shea, T., Hayman, D., et al. (2014). The effect of seasonal birth pulses on pathogen persistence in wild mammal populations. *Proceedings of the Royal Society of London B: Biological Sciences*, 281(1786), 20132962.
- Petermann, J. S., Fergus, A. J., Turnbull, L. A., & Schmid, B. (2008). Janzen-connell effects are widespread and strong enough to maintain diversity in grasslands. *Ecology*, 89(9), 2399–2406.
- Pinheiro, J., & Bates, D. (2006). *Mixed-effects models in S and S-PLUS*. Berlin: Springer Science & Business Media.
- Pitzer, V. E., Patel, M. M., Lopman, B. A., Viboud, C., Parashar, U. D., & Grenfell, B. T. (2011). Modeling rotavirus strain dynamics in developed countries to understand the potential impact of vaccination on genotype distributions. *Proceedings of the National Academy of Sciences*, 108(48), 19353–19358.
- Polacheck, T., Hilborn, R., & Punt, A. E. (1993). Fitting surplus production models: comparing methods and measuring uncertainty. *Canadian Journal of Fisheries and Aquatic Sciences*, 50(12), 2597–2607.
- Pomeroy, L. W., Bjørnstad, O. N., Kim, H., Jumbo, S. D., Abdoukkadiri, S., & Garabed, R. (2015). Serotype-specific transmission and waning immunity of endemic foot-and-mouth disease virus in Cameroon. *PLoS One*, 10(9), e0136642.
- Priestley, M. B. (1981). *Spectral analysis and time series*. Cambridge, MA: Academic.
- Ramsay, J., & Silverman, B. (1997). *Functional data analysis*. Berlin: Springer.
- Rand, D., & Wilson, H. (1991). Chaotic stochasticity: A ubiquitous source of unpredictability in epidemics. *Proceedings of the Royal Society of London B: Biological Sciences*, 246(1316), 179–184.
- Riley, S., Fraser, C., Donnelly, C. A., Ghani, A. C., Abu-Raddad, L. J., Hedley, A. J., et al. (2003). Transmission dynamics of the etiological agent of sars in Hong Kong: Impact of public health interventions. *Science*, 300(5627), 1961–1966.
- Roberts, M., & Heesterbeek, H. (1993). Bluff your way in epidemic models. *Trends in Microbiology*, 1(9), 343–348.
- Rohani, P., Keeling, M. J., & Grenfell, B. T. (2002). The interplay between determinism and stochasticity in childhood diseases. *The American Naturalist*, 159(5), 469–481.
- Rohani, P., & King, A. A. (2010). Never mind the length, feel the quality: The impact of long-term epidemiological data sets on theory, application and policy. *Trends in Ecology & Evolution*, 25(10), 611–618.

- Roy, S., Lavine, J., Chiaromonte, F., Terwee, J., VandeWoude, S., Bjornstad, O., et al. (2009). Multivariate statistical analyses demonstrate unique host immune responses to single and dual lentiviral infection. *PLoS One*, 4(10), e7359.
- Ruelle, D. (1993). *Chance and chaos*. Princeton, NJ: Princeton University Press.
- Ruiz-Moreno, D., Pascual, M., Bouma, M., Dobson, A., & Cash, B. (2007). Cholera seasonality in Madras (1901–1940): Dual role for rainfall in endemic and epidemic regions. *EcoHealth*, 4(1), 52–62.
- Schenzle, D. (1984). An age-structured model of pre-and post-vaccination measles transmission. *Mathematical Medicine and Biology: A Journal of the IMA*, 1(2), 169–191.
- Seabloom, E. W., Bjørnstad, O. N., Bolker, B. M., & Reichman, O. J. (2005). Spatial signature of environmental heterogeneity, dispersal, and competition in successional grasslands. *Ecological Monographs*, 75(2), 199–214.
- Shaman, J., & Kohn, M. (2009). Absolute humidity modulates influenza survival, transmission, and seasonality. *Proceedings of the National Academy of Sciences*, 106(9), 3243–3248.
- Shrestha, S., Bjørnstad, O. N., & King, A. A. (2014). Evolution of acuteness in pathogen metapopulations: Conflicts between “classical” and invasion-persistence trade-offs. *Theoretical Ecology*, 7(3), 299–311.
- Smith, D. L., Ericson, L., & Burdon, J. J. (2003). Epidemiological patterns at multiple spatial scales: An 11-year study of a *Triphragmium ulmariae*–*Filipendula ulmaria* metapopulation. *Journal of Ecology*, 91(5), 890–903.
- Smith, D. L., Lucey, B., Waller, L. A., Childs, J. E., & Real, L. A. (2002). Predicting the spatial dynamics of rabies epidemics on heterogeneous landscapes. *Proceedings of the National Academy of Sciences*, 99(6), 3668–3672.
- Stern, A., Nickel, P., Meyer, T. F., & So, M. (1984). Opacity determinants of neisseria gonorrhoeae: Gene expression and chromosomal linkage to the gonococcal pilus gene. *Cell*, 37(2), 447–456.
- Sugihara, G., Grenfell, B., & May, R. M. (1990). Distinguishing error from chaos in ecological time series. *Philosophical Transactions of the Royal Society of London Series B*, 330, 235–250.
- Swinton, J. (1998). Extinction times and phase transitions for spatially structured closed epidemics. *Bulletin of Mathematical Biology*, 60(2), 215–230.
- Takahashi, S., Liao, Q., Van Boeckel, T. P., Xing, W., Sun, J., Hsiao, V. Y., et al. (2016). Hand, foot, and mouth disease in China: Modeling epidemic dynamics of enterovirus serotypes and implications for vaccination. *PLoS Medicine*, 13(2), e1001958.
- Tettelin, H., Saunders, N. J., Heidelberg, J., Jeffries, A. C., Nelson, K. E., Eisen, J. A., et al. (2000). Complete genome sequence of *Neisseria meningitidis* serogroup B strain MC58. *Science*, 287(5459), 1809–1815.
- Tilman, D. (1976). Ecological competition between algae: Experimental confirmation of resource-based competition theory. *Science*, 192, 463–465.
- Torrence, C., & Compo, G. P. (1998). A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79(1), 61–78.

- Truscott, J., & Ferguson, N. M. (2012). Evaluating the adequacy of gravity models as a description of human mobility for epidemic modelling. *PLoS Computational Biology*, 8(10), e1002699.
- Turchin, P. (2003). *Complex population dynamics: A theoretical/empirical synthesis* (Vol. 35). Princeton, NJ: Princeton University Press.
- Venables, W., Ripley, B. D. (2013). *S programming*. Berlin: Springer Science & Business Media.
- Viboud, C., Bjørnstad, O. N., Smith, D. L., Simonsen, L., Miller, M. A., & Grenfell, B. T. (2006). Synchrony, waves, and spatial hierarchies in the spread of influenza. *Science*, 312(5772), 447–451.
- Vink, M. A., Bootsma, M. C. J., & Wallinga, J. (2014). Serial intervals of respiratory infectious diseases: A systematic review and analysis. *American Journal of Epidemiology*, 180(9), 865–875.
- Warfel, J. M., Zimmerman, L. I., & Merkel, T. J. (2014). Acellular pertussis vaccines protect against disease but fail to prevent infection and transmission in a nonhuman primate model. *Proceedings of the National Academy of Sciences*, 111(2), 787–792.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684), 440–442.
- White, L. F., & Pagano, M. (2008). A likelihood-based method for real-time estimation of the serial interval and reproductive number of an epidemic. *Statistics in Medicine*, 27(16), 2999–3016.
- WHO Ebola Response Team (2014). Ebola virus disease in west africa—the first 9 months of the epidemic and forward projections. *The New England Journal of Medicine*, 371(14), 1481–1495.
- Wiesenfeld, K., & Moss, F. (1995). Stochastic resonance and the benefits of noise: From ice ages to crayfish and squids. *Nature*, 373(6509), 33–36.
- Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310), 1102–1104.
- Xia, Y., Bjørnstad, O. N., & Grenfell, B. T. (2004). Measles metapopulation dynamics: A gravity model for epidemiological coupling and dynamics. *The American Naturalist*, 164(2), 267–281.
- Yao, Q., & Tong, H. (1994). On prediction and chaos in stochastic systems. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 348(1688), 357–369.
- Yao, Q., & Tong, H. (1998). A bootstrap detection for operational determinism. *Physica D: Nonlinear Phenomena*, 115(1–2), 49–55.
- Ye, H., Beamish, R. J., Glaser, S. M., Grant, S. C., Hsieh, C.-H., Richards, L. J., et al. (2015). Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling. *Proceedings of the National Academy of Sciences*, 112(13), E1569–E1576.

Index

A

- ACF, 95, 96
- Age-incidence, 1, 49, 69, 79, 80
- Animation, 220, 265
- ARMA, 97, 98, 176
- Autocorrelation, 95, 97, 221, 241–247, 267, 270, 273, 279

B

- Basic reproductive ratio, 4, 11

C

- Catalytic model, 1, 59
- Chain-binomial model, 23, 36, 41, 42, 117, 120, 215, 234
- Chaos, 204, 219
- Chickenpox, 192, 194–197, 203
- Correlogram, 244–249, 270
- Coupled-map lattice models, 219, 229, 263
- Critical community size, 4
- Cycles, 2, 11, 20, 22, 81, 89, 99, 106, 110, 111, 134, 144, 166, 172, 193, 198, 199, 204, 217, 255

D

- Dampening period, 21, 22, 93, 101, 169, 196

E

- Ebola, 32, 33, 43, 45, 49, 54, 241
- Effective reproductive ratio, 31, 33, 50, 157
- Eigenvalues, 21–23, 88, 160, 163, 168, 170, 260, 261, 285, 286

F

- FDA, 293
- FIV, 283–286, 288, 290

Force of infection, 36, 53, 58, 59, 61–74, 78, 206, 210, 211, 223, 279

Fox, 31, 94, 159

G

- Gillespie algorithm, 137–140
- Gonorrhea, 5, 6, 50, 229, 238, 239
- Gravity model, 223–225
- Gypsy moth, 217, 252, 253

H

- Hessian matrix, 146
- HIV, 5

I

- Infectious period, 11, 12, 16, 23–25, 27, 35, 42, 48, 84, 139, 142, 146, 169, 192
- Influenza, 3, 32, 41, 49, 81, 82, 97–99, 151, 168, 169, 222, 223, 225
- Isoclines, 18, 20

J

- Jacobian matrix, 21, 22, 88, 160, 161, 166, 168, 170, 171, 174, 180, 182, 260, 261

L

- Larch bud moth, 263
- Latent period, 23, 47, 57, 84, 142, 147, 192
- LDA, 287–290
- Likelihood, 243
- Lyapunov exponent, 180, 182–187

M

- Malaria, 2, 131, 283, 284, 292, 293
- Mantel test, 242, 245

M
Measles, 2, 5–7, 23, 31, 34, 35, 37, 40, 57, 61, 82–84, 89, 93, 102, 104–108, 110, 111, 114, 119, 121, 125–129, 150, 153, 179, 180, 182–185, 187, 188, 204, 223
MLE, 40, 145, 243

N
Network, 239

O
ODE, 21, 27, 88

P
Parasitoid, 255–258, 262–265
PCA, 284, 286, 287, 290–294
Phase plane, 18, 20, 85, 86, 90, 92, 119, 181, 183, 195–197, 257

POLYMOD, 74, 76
Profile likelihood, 125, 130, 180, 211
Project Tycho, 107, 129

R
Rabbit, 62, 279
Rabies, 31, 93, 159, 160, 164, 165, 168
Raccoon, 159, 164, 168
Recurrent epidemics, 6, 35, 92
Resonant frequency, 163, 166, 170, 172
Rubella, 3, 69–72, 129
Rust, 210, 211, 214, 216, 241–243, 247, 249–251, 267, 268, 270

S
Scarlet fever, 117, 129, 132
Seasonality, 12, 76, 81–86, 89, 91, 93, 97, 122
Seroprevalence, 60, 69, 124

Simple epidemic, 35
SIR model, 1, 3, 9–12, 15, 16, 18, 19, 21, 24, 26, 27, 35, 36, 47, 77, 79, 85, 95, 97, 117, 120, 121, 137, 138, 140, 149, 161, 163, 171, 179, 217, 223, 225, 235

SIRS model, 169, 170
Spatiotemporal dynamics, 217, 219, 220, 223, 241, 252, 255, 263

Spectral analysis, 95, 100, 107
Splines, 66, 67
Stability analysis, 159
Stochasticity, 81, 85, 97, 120, 121, 127, 143, 171–173, 175, 177, 179, 186, 187, 193, 195, 200–202
Susceptible reconstruction, 122
Synchrony, 131, 221, 241, 252

T
Transfer function, 171–173, 176, 177
Transmission rate, 12, 48, 85, 146, 164, 165, 193, 202, 204, 217
TSIR model, 35, 117, 119, 121, 125–129, 131, 144, 160, 173–178, 180, 182–185, 210

V
Vaccination, 13, 35, 41, 50, 69, 72, 78, 93, 153, 154, 156, 157, 165, 190–193, 198

W
WAIFW, 75, 77, 78
Wavelet analysis, 95, 100–108, 111, 113, 198, 199, 201, 202
Whooping cough, 2, 107, 109, 189, 192, 198–200, 202, 279