

Statistics for Biology and Health

Series Editors:

M. Gail

K. Krickeberg

J. M. Samet

A. Tsiatis

W. Wong

Statistics for Biology and Health

- Bacchieri/Cioppa:* Fundamentals of Clinical Research
Borchers/Buckland/Zucchini: Estimating Animal Abundance: Closed Populations
Burzykowski/Molenberghs/Buyse: The Evaluation of Surrogate Endpoints
Duchateau/Janssen: The Frailty Model
Everitt/Rabe-Hesketh: Analyzing Medical Data Using S-PLUS
Ewens/Grant: Statistical Methods in Bioinformatics: An Introduction, 2nd ed.
Gentleman/Carey/Huber/Irizarry/Dudoit: Bioinformatics and Computational Biology Solutions Using R and Bioconductor
Hougaard: Analysis of Multivariate Survival Data
Keyfitz/Caswell: Applied Mathematical Demography, 3rd ed.
Klein/Moeschberger: Survival Analysis: Techniques for Censored and Truncated Data, 2nd ed.
Kleinbaum/Klein: Survival AnalysisL A Self-Learning Text, 2nd ed.
Kleinbaum/Klein: Logistic Regression: A Self-Learning Text, 2nd ed.
Lange: Mathematical and Statistical Methods for Genetic Analysis, 2nd ed.
Lazar: The Statistical Analysis of Functional MRI Data
Manton/Singer/Suzman: Forecasting the Health of Elderly Populations
Martinussen/Scheike: Dynamic Regression Models for Survival Data
Moyé: Multiple Analyses in Clinical Trials: Fundamentals for Investigators
Nielsen: Statistical Methods in Molecular Evolution
O'Quigley: Proportional Hazards Regression
Parmigiani/Garrett/Irizarry/Zeger: The Analysis of Gene Expression Data: Methods and Software
Proschan/LanWittes: Statistical Monitoring of Clinical Trials: A Unified Approach
Siegmund/Yakir: The Statistics of Gene Mapping
Simon/Korn/McShane/Radmacher/Wright/Zhao: Design and Analysis of DNA Microarray Investigations
Sorensen/Gianola: Likelihood, Bayesian, and MCMC Methods in Quantitative Genetics
Stallard/Manton/Cohen: Forecasting Product Liability Claims: Epidemiology and Modeling in the Manville Asbestos Case
Sun: The Statistical Analysis of Interval-censored Failure Time Data
Therneau/Grambsch: Modeling Survival Data: Extending the Cox Model
Ting: Dose Finding in Drug Development
Vittinghoff/Glidden/Shiboski/McCulloch: Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models
Wu/Ma/Casella: Statistical Genetics of Quantitative Traits: Linkage, Maps, and QTL
Zhang/Singer: Recursive Partitioning in the Health Sciences
Zuur/Ieno/Smith: Analysing Ecological Data
Zuur/Ieno/Walker/Saveliev/Smith: Mixed Effects Models and Extensions in Ecology with R

Alain F. Zuur · Elena N. Ieno · Neil J. Walker ·
Anatoly A. Saveliev · Graham M. Smith

Mixed Effects Models and Extensions in Ecology with R



Alain F. Zuur
Highland Statistics Ltd.
Newburgh
United Kingdom
highstat@highstat.com

Elena N. Ieno
Highland Statistics Ltd.
Newburgh
United Kingdom
bio@highstat.com

Neil J. Walker
Central Science Laboratory
Gloucester
United Kingdom
n.walker@csl.gov.uk

Anatoly A. Saveliev
Kazan State University
Kazan
Russia
saa@ksu.ru

Graham M. Smith
Bath Spa University
Bath
United Kingdom
graham.smith@myotis.co.uk

Series Editors

M. Gail
National Cancer Institute
Rockville, MD 20892
USA

K. Krickeberg
Le Chatelet
F-63270 Manglieu
France

J. Samet
Department of Preventive
Medicine
Keck School of Medicine
University of Southern
California
1441 Eastlake Ave. Room
4436, MC 9175
Los Angeles, CA 90089

A. Tsiatis
Department of Statistics
North Carolina State University
Raleigh, NC 27695
USA

W. Wong
Department of Statistics
Stanford University
Stanford, CA 94305-4065
USA

ISSN 1431-8776
ISBN 978-0-387-87457-9
DOI 10.1007/978-0-387-87458-6

e-ISBN 978-0-387-87458-6

Library of Congress Control Number: 2008942429

© Springer Science+Business Media, LLC 2009

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden. The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

springer.com

*Thanks to my parents for sharing the burden
of my university fees – Alain F. Zuur*

*To my friends, colleagues, and former students
who are actively committed to the protection and
care of the environment – Elena N. Ieno*

*Thanks to my wife Tatiana for her patience
and moral support – Anatoly A. Saveliev*

*I would like to thank all family and friends for
help and support through times good and bad
during the writing of this book – Neil J. Walker*

*To my parents who, even now, continue to support
me in everything I do – Graham M. Smith*

Preface

No sooner, it seems, had our first book *Analysing Ecological Data* gone to print, than we embarked on the writing of the nearly 600 page text you are now holding. This proved to be a labour of love of sorts – we felt that there were certain issues sufficiently common in the analysis of ecological data that merited more detailed description and analysis. Thus the present book can be seen as a ‘sequel’ to *Analysing Ecological Data* but with much greater emphasis on these very issues so commonly encountered in the collection of, and analysis of, ecological data. In particular, we look at different ways of analysing nested data, heterogeneity of variance, spatial and temporal correlation, and zero-inflated data.

The original plan was to write a text of about 350 pages, but to do justice to the sheer range of problems and ideas we have well exceeded that original target (as you can see!). Such is the scope of applied statistics in ecology. In particular, partly on the back of reviewer’s comments, we have included a chapter on Bayesian Monte-Carlo Markov-Chain applications in generalized linear modelling. We hope this serves as an informative introduction (but no more than an introduction!) to this interesting and increasingly relevant area of statistics.

We received lots of positive feedback on the approach and style we used in *Analysing Ecological Data*, especially the combination of case studies and a theory section. We have therefore followed the same approach with this book. This time, however, we have provided the R code used for the analysis. Most of this R code is included in the text, but where the code was particularly long, it is only available from the book’s website at www.highstat.com. In the case studies, we also included advice on what to write in a paper.

Newburgh, United Kingdom
Newburgh, United Kingdom
Gloucester, United Kingdom
Kazan, Russia
Bath, United Kingdom
December 2008

Alain F. Zuur
Elena N. Ieno
Neil J. Walker
Anatoly A. Saveliev
Graham M. Smith

Acknowledgements

The material in this book has been taught in various courses in 2007 and 2008, and we are greatly in debt to all participants who helped improving the material. We would also like to thank a number of people who read and commented on parts of earlier drafts, namely Chris Elphick, Alex Douglas, and Graham Pierce. The manuscript was reviewed by Loveday Conquest (University of Washington), Sarah Goslee (USDA), Thomas Kneib (LMU Munich), Bret Larget (University of Wisconsin), Ruth Salway (University of Bath), Jing Hua Zhao (University of Cambridge), and several anonymous referees. We thank them all for their positive, encouraging, and useful reviews. Their comments and criticisms greatly improved the book.

The most difficult part of writing a book is finding public domain data which can be used in theory chapters. We are particularly thankful to the following persons for donating data sets. Sonia Mendes and Graham Pierce for the whale data, Gerard Janssen for the benthic data, Pam Sikkink for the grassland data, Graham Pierce and Jennifer Smith for the squid data, Alexandre Roulin for the barn owl data, Michael Reed and Chris Elphick for the Hawaiian bird data, Tatiana Rogova for the Volzhsko-Kamsky forestry data, Robert Cruikshanks, Mary Kelly-Quinn and John O'Halloran for the Irish (sodium dominance index) river data, Chris Elphick for the sparrow and California bird data, Michael Penston for the sea lice data, Joaquín Vicente and Christian Gortázar for the wild boar and deer data, Ken Mackenzie for the cod data, and António Mira for the snake data. The proper references are given in the text. We also would like to thank all people involved in the case study chapters; they are credited where relevant.

Michelle Cronin provided the seal photo on the back cover, Joaquin Vicente the deer photo, and Malena Sabatino gave us the bee photo. The photograph of the koalas was provided by Australian Koala Foundation (www.savethekoala.com). The photo on the front cover is from © Wayne Lynch/Arcticphoto.com.

Finally, we would like to thank John Kimmel for giving us the opportunity to write this book and for patiently accepting the 6-month delay. Up to the next book.

Contents

1	Introduction	1
1.1	What Is in the Book?	1
1.1.1	To Include or Not to Include GLM and GAM	3
1.1.2	Case Studies	4
1.1.3	Flowchart of the Content	4
1.2	Software	5
1.3	How to Use This Book If You Are an Instructor	6
1.4	What We Did Not Do and Why	6
1.5	How to Cite R and Associated Packages	7
1.6	Our R Programming Style	8
1.7	Getting Data into R	9
1.7.1	Data in a Package	10
2	Limitations of Linear Regression Applied on Ecological Data	11
2.1	Data Exploration	12
2.1.1	Cleveland Dotplots	12
2.1.2	Pairplots	14
2.1.3	Boxplots	15
2.1.4	xyplot from the Lattice Package	15
2.2	The Linear Regression Model	17
2.3	Violating the Assumptions; Exception or Rule?	19
2.3.1	Introduction	19
2.3.2	Normality	19
2.3.3	Heterogeneity	20
2.3.4	Fixed X	21
2.3.5	Independence	21
2.3.6	Example 1; Wedge Clam Data	22
2.3.7	Example 2; Moby's Teeth	26
2.3.8	Example 3; Nereis	28
2.3.9	Example 4; Pelagic Bioluminescence	30
2.4	Where to Go from Here	31

3 Things Are Not Always Linear; Additive Modelling	35
3.1 Introduction	35
3.2 Additive Modelling	36
3.2.1 GAM in gam and GAM in mgcv	37
3.2.2 GAM in gam with LOESS	38
3.2.3 GAM in mgcv with Cubic Regression Splines	42
3.3 Technical Details of GAM in mgcv	44
3.3.1 A (Little) Bit More Technical Information on Regression Splines	47
3.3.2 Smoothing Splines Alias Penalised Splines	49
3.3.3 Cross-Validation	51
3.3.4 Additive Models with Multiple Explanatory Variables	53
3.3.5 Two More Things	53
3.4 GAM Example 1; Bioluminescent Data for Two Stations	55
3.4.1 Interaction Between a Continuous and Nominal Variable	59
3.5 GAM Example 2: Dealing with Collinearity	63
3.6 Inference	66
3.7 Summary and Where to Go from Here?	67
4 Dealing with Heterogeneity	71
4.1 Dealing with Heterogeneity	72
4.1.1 Linear Regression Applied on Squid	72
4.1.2 The Fixed Variance Structure	74
4.1.3 The VarIdent Variance Structure	75
4.1.4 The varPower Variance Structure	78
4.1.5 The varExp Variance Structure	80
4.1.6 The varConstPower Variance Structure	80
4.1.7 The varComb Variance Structure	81
4.1.8 Overview of All Variance Structures	82
4.1.9 Graphical Validation of the Optimal Model	84
4.2 Benthic Biodiversity Experiment	86
4.2.1 Linear Regression Applied on the Benthic Biodiversity Data	86
4.2.2 GLS Applied on the Benthic Biodiversity Data	89
4.2.3 A Protocol	90
4.2.4 Application of the Protocol on the Benthic Biodiversity Data	92
5 Mixed Effects Modelling for Nested Data	101
5.1 Introduction	101
5.2 2-Stage Analysis Method	103
5.3 The Linear Mixed Effects Model	105
5.3.1 Introduction	105
5.3.2 The Random Intercept Model	106
5.3.3 The Random Intercept and Slope Model	109
5.3.4 Random Effects Model	111

5.4	Induced Correlations	112
5.4.1	Intraclass Correlation Coefficient	114
5.5	The Marginal Model	114
5.6	Maximum Likelihood and REML Estimation	116
5.6.1	Illustration of Difference Between ML and REML	119
5.7	Model Selection in (Additive) Mixed Effects Modelling	120
5.8	RIKZ Data: Good Versus Bad Model Selection	122
5.8.1	The Wrong Approach	122
5.8.2	The Good Approach	127
5.9	Model Validation	128
5.10	Begging Behaviour of Nestling Barn Owls	129
5.10.1	Step 1 of the Protocol: Linear Regression	130
5.10.2	Step 2 of the Protocol: Fit the Model with GLS	132
5.10.3	Step 3 of the Protocol: Choose a Variance Structure	132
5.10.4	Step 4: Fit the Model	133
5.10.5	Step 5 of the Protocol: Compare New Model with Old Model	133
5.10.6	Step 6 of the Protocol: Everything Ok?	134
5.10.7	Steps 7 and 8 of the Protocol: The Optimal Fixed Structure	135
5.10.8	Step 9 of the Protocol: Refit with REML and Validate the Model	137
5.10.9	Step 10 of the Protocol	139
5.10.10	Sorry, We are Not Done Yet	139
6	Violation of Independence – Part I	143
6.1	Temporal Correlation and Linear Regression	143
6.1.1	ARMA Error Structures	150
6.2	Linear Regression Model and Multivariate Time Series	152
6.3	Owl Sibling Negotiation Data	158
7	Violation of Independence – Part II	161
7.1	Tools to Detect Violation of Independence	161
7.2	Adding Spatial Correlation Structures to the Model	166
7.3	Revisiting the Hawaiian Birds	171
7.4	Nitrogen Isotope Ratios in Whales	172
7.4.1	Moby	172
7.4.2	All Whales	174
7.5	Spatial Correlation due to a Missing Covariate	177
7.6	Short Godwits Time Series	182
7.6.1	Description of the Data	182
7.6.2	Data Exploration	183
7.6.3	Linear Regression	184
7.6.4	Protocol Time	186
7.6.5	Why All the Fuss?	190

8 Meet the Exponential Family	193
8.1 Introduction	193
8.2 The Normal Distribution	194
8.3 The Poisson Distribution	196
8.3.1 Preparation for the Offset in GLM	198
8.4 The Negative Binomial Distribution	199
8.5 The Gamma Distribution	201
8.6 The Bernoulli and Binomial Distributions	202
8.7 The Natural Exponential Family	204
8.7.1 Which Distribution to Select?	205
8.8 Zero Truncated Distributions for Count Data	206
9 GLM and GAM for Count Data	209
9.1 Introduction	209
9.2 Gaussian Linear Regression as a GLM	210
9.3 Introducing Poisson GLM with an Artificial Example	211
9.4 Likelihood Criterion	213
9.5 Introducing the Poisson GLM with a Real Example	215
9.5.1 Introduction	215
9.5.2 R Code and Results	216
9.5.3 Deviance	217
9.5.4 Sketching the Fitted Values	218
9.6 Model Selection in a GLM	220
9.6.1 Introduction	220
9.6.2 R Code and Output	220
9.6.3 Options for Finding the Optimal Model	221
9.6.4 The Drop1 Command	222
9.6.5 Two Ways of Using the Anova Command	223
9.6.6 Results	223
9.7 Overdispersion	224
9.7.1 Introduction	224
9.7.2 Causes and Solutions for Overdispersion	224
9.7.3 Quick Fix: Dealing with Overdispersion in a Poisson GLM	225
9.7.4 R Code and Numerical Output	226
9.7.5 Model Selection in Quasi-Poisson	227
9.8 Model Validation in a Poisson GLM	228
9.8.1 Pearson Residuals	229
9.8.2 Deviance Residuals	229
9.8.3 Which One to Use?	230
9.8.4 What to Plot?	230
9.9 Illustration of Model Validation in Quasi-Poisson GLM	231
9.10 Negative Binomial GLM	233
9.10.1 Introduction	233
9.10.2 Results	236

9.11 GAM	238
9.11.1 Distribution of larval Sea Lice Around Scottish Fish Farms	239
10 GLM and GAM for Absence–Presence and Proportional Data	245
10.1 Introduction	245
10.2 GLM for Absence–Presence Data	246
10.2.1 Tuberculosis in Wild Boar	246
10.2.2 Parasites in Cod	252
10.3 GLM for Proportional Data	254
10.4 GAM for Absence–Presence Data	258
10.5 Where to Go from Here?	259
11 Zero-Truncated and Zero-Inflated Models for Count Data	261
11.1 Introduction	261
11.2 Zero-Truncated Data	263
11.2.1 The Underlying Mathematics for Truncated Models	263
11.2.2 Illustration of Poisson and NB Truncated Models	265
11.3 Too Many Zeros	269
11.3.1 Sources of Zeros	270
11.3.2 Sources of Zeros for the Cod Parasite Data	271
11.3.3 Two-Part Models Versus Mixture Models, and Hippos	271
11.4 ZIP and ZINB Models	274
11.4.1 Mathematics of the ZIP and ZINB	274
11.4.2 Example of ZIP and ZINB Models	278
11.5 ZAP and ZANB Models, Alias Hurdle Models	286
11.5.1 Mathematics of the ZAP and ZANB	287
11.5.2 Example of ZAP and ZANB	288
11.6 Comparing Poisson, Quasi-Poisson, NB, ZIP, ZINB, ZAP and ZANB GLMs	291
11.7 Flowchart and Where to Go from Here	293
12 Generalised Estimation Equations	295
12.1 GLM: Ignoring the Dependence Structure	295
12.1.1 The California Bird Data	295
12.1.2 The Owl Data	299
12.1.3 The Deer Data	300
12.2 Specifying the GEE	302
12.2.1 Introduction	302
12.2.2 Step 1 of the GEE: Systematic Component and Link Function	303
12.2.3 Step 2 of the GEE: The Variance	304
12.2.4 Step 3 of the GEE: The Association Structure	304
12.3 Why All the Fuss?	309
12.3.1 A Bit of Maths	310

12.4	Association for Binary Data	313
12.5	Examples of GEE	314
12.5.1	A GEE for the California Birds.....	314
12.5.2	A GEE for the Owls	316
12.5.3	A GEE for the Deer Data.....	319
12.6	Concluding Remarks	320
13	GLMM and GAMM	323
13.1	Setting the Scene for Binomial GLMM	324
13.2	GLMM and GAMM for Binomial and Poisson Data	327
13.2.1	Deer Data	327
13.2.2	The Owl Data Revisited.....	333
13.2.3	A Word of Warning	339
13.3	The Underlying Mathematics in GLMM	339
14	Estimating Trends for Antarctic Birds in Relation to Climate Change	343
A.F. Zuur, C. Barbraud, E.N. Ieno, H. Weimerskirch, G.M. Smith, and N.J. Walker		
14.1	Introduction	343
14.1.1	Explanatory Variables	344
14.2	Data Exploration	345
14.3	Trends and Auto-correlation.....	350
14.4	Using Ice Extent as an Explanatory Variable	352
14.5	SOI and Differences Between Arrival and Laying Dates	354
14.6	Discussion	360
14.7	What to Report in a Paper	361
15	Large-Scale Impacts of Land-Use Change in a Scottish Farming Catchment	363
A.F. Zuur, D. Raffaelli, A.A. Saveliev, N.J. Walker, E.N. Ieno, and G.M. Smith		
15.1	Introduction	363
15.2	Data Exploration	365
15.3	Estimation of Trends for the Bird Data	367
15.3.1	Model Validation	368
15.3.2	Failed Approach 1	372
15.3.3	Failed Approach 2	373
15.3.4	Assume Homogeneity?	374
15.4	Dealing with Independence	374
15.5	To Transform or Not to Transform.....	378
15.6	Birds and Explanatory Variables	378
15.7	Conclusions	380
15.8	What to Write in a Paper	381

16 Negative Binomial GAM and GAMM to Analyse Amphibian Roadkills	383
A.F. Zuur, A. Mira, F. Carvalho, E.N. Ieno, A.A. Saveliev, G.M. Smith, and N.J. Walker	
16.1 Introduction	383
16.1.1 Roadkills	383
16.2 Data Exploration	385
16.3 GAM	389
16.4 Understanding What the Negative Binomial is Doing	394
16.5 GAMM: Adding Spatial Correlation	396
16.6 Discussion	397
16.7 What to Write in a Paper	397
17 Additive Mixed Modelling Applied on Deep-Sea Pelagic Bioluminescent Organisms	399
A.F. Zuur, I.G. Priede, E.N. Ieno, G.M. Smith, A.A. Saveliev, and N.J. Walker	
17.1 Biological Introduction	399
17.2 The Data and Underlying Questions	401
17.3 Construction of Multi-panel Plots for Grouped Data	402
17.3.1 Approach 1	402
17.3.2 Approach 2	407
17.3.3 Approach 3	408
17.4 Estimating Common Patterns Using Additive Mixed Modelling	410
17.4.1 One Smoothing Curve for All Stations	410
17.4.2 Four Smoothers; One for Each Month	414
17.4.3 Smoothing Curves for Groups Based on Geographical Distances	417
17.4.4 Smoothing Curves for Groups Based on Source Correlations	418
17.5 Choosing the Best Model	419
17.6 Discussion	420
17.7 What to Write in a Paper	421
18 Additive Mixed Modelling Applied on Phytoplankton Time Series Data	423
A.F. Zuur, M.J. Latuhihin, E.N. Ieno, J.G. Baretta-Bekker, G.M. Smith, and N.J. Walker	
18.1 Introduction	423
18.1.1 Biological Background of the Project	424
18.2 Data Exploration	427
18.3 A Statistical Data Analysis Strategy for DIN	429
18.4 Results for Temperature	439
18.5 Results for DIAT1	441
18.6 Comparing Phytoplankton and Environmental Trends	443

18.7	Conclusions	445
18.8	What to Write in a Paper.....	446
19	Mixed Effects Modelling Applied on American Foulbrood Affecting Honey Bees Larvae	447
	A.F. Zuur, L.B. Gende, E.N. Ieno, N.J. Fernández, M.J. Egularas, R. Fritz, N.J. Walker, A.A. Saveliev, and G.M. Smith	
19.1	Introduction	447
19.2	Data Exploration	448
19.3	Analysis of the Data	450
19.4	Discussion	458
19.5	What to Write in a Paper.....	458
20	Three-Way Nested Data for Age Determination Techniques Applied to Cetaceans	459
	E.N. Ieno, P.L. Luque, G.J. Pierce, A.F. Zuur, M.B. Santos, N.J. Walker, A.A. Saveliev, and G.M. Smith	
20.1	Introduction	459
20.2	Data Exploration	460
20.3	Data Analysis	462
20.3.1	Intraclass Correlations	466
20.4	Discussion	467
20.5	What to Write in a Paper.....	468
21	GLMM Applied on the Spatial Distribution of Koalas in a Fragmented Landscape	469
	J.R. Rhodes, C.A. McAlpine, A.F. Zuur, G.M. Smith, and E.N. Ieno	
21.1	Introduction	469
21.2	The Data	471
21.3	Data Exploration and Preliminary Analysis	473
21.3.1	Collinearity	473
21.3.2	Spatial Auto-correlation.....	479
21.4	Generalised Linear Mixed Effects Modelling	481
21.4.1	Model Selection	483
21.4.2	Model Adequacy	487
21.5	Discussion	490
21.6	What to Write in a Paper.....	492
22	A Comparison of GLM, GEE, and GLMM Applied to Badger Activity Data	493
	N.J. Walker, A.F. Zuur, A. Ward, A.A. Saveliev, E.N. Ieno, and G.M. Smith	
22.1	Introduction	493
22.2	Data Exploration	495
22.3	GLM Results Assuming Independence	497

22.4 GEE Results	499
22.5 GLMM Results	500
22.6 Discussion	501
22.7 What to Write in a Paper	502
23 Incorporating Temporal Correlation in Seal Abundance Data with MCMC	503
A.A. Saveliev, M. Cronin, A.F. Zuur, E.N. Ieno, N.J. Walker, and G.M. Smith	
23.1 Introduction	503
23.2 Preliminary Results	504
23.3 GLM	507
23.3.1 Validation	509
23.4 What Is Bayesian Statistics?	510
23.4.1 Theory Behind Bayesian Statistics	510
23.4.2 Markov Chain Monte Carlo Techniques	511
23.5 Fitting the Poisson Model in BRugs	513
23.5.1 Code in R	513
23.5.2 Model Code	514
23.5.3 Initialising the Chains	515
23.5.4 Summarising the Posterior Distributions	517
23.5.5 Inference	518
23.6 Poisson Model with Random Effects	520
23.7 Poisson Model with Random Effects and Auto-correlation	523
23.8 Negative Binomial Distribution with Auto-correlated Random Effects	525
23.8.1 Comparison of Models	528
23.9 Conclusions	528
A Required Pre-knowledge: A Linear Regression and Additive Modelling Example	531
A.1 The Data	531
A.2 Data Exploration	532
A.2.1 Step 1: Outliers	532
A.2.2 Step 2: Collinearity	533
A.2.3 Relationships	536
A.3 Linear Regression	536
A.3.1 Model Selection	540
A.3.2 Model Validation	542
A.3.3 Model Interpretation	543
A.4 Additive Modelling	546
A.5 Further Extensions	550
A.6 Information Theory and Multi-model Inference	550
A.7 Maximum Likelihood Estimation in Linear Regression Context ..	552
References	553
Index	563

Contributors

C. Barbraud Centre d'Etudes Biologiques de Chizé, Centre National de la Recherche Scientifique, 79360 Villiers en Bois, France

J.G. Baretta-Bekker Rijkswaterstaat – Centre for Water Management, P.O. Box 17, 8200 AA Lelystad, The Netherlands

F. Carvalho Unidade de Biologia da Conservação, Departamento de Biologia, Universidade de Évora, 7002-554 – Évora, Portugal

M. Cronin Coastal & Marine Resources Centre, Naval Base, Haulbowline, Cobh, Co. Cork, Ireland

M.J. Egúaras Laboratorio de Artrópodos, Departamento de Biología, Universidad Nacional de Mar del Plata, Funes 3350, (7600) Mar del Plata, Argentina

N.J. Fernández Laboratorio de Artrópodos, Departamento de Biología, Universidad Nacional de Mar del Plata, Funes 3350, (7600) Mar del Plata, Argentina

R. Fritz Laboratorio de Bromatología, Departamento de Química, Universidad Nacional de Mar del Plata, Funes 3350, segundo piso, (7600) Mar del Plata, Argentina

L.B. Gende Laboratorio de Artrópodos, Departamento de Biología, Universidad Nacional de Mar del Plata, Funes 3350, (7600) Mar del Plata, Argentina

E.N. Ieno Highland Statistics LTD., 6 Laverock Road, Newburgh, AB41 6FN, United Kingdom

M.J. Latuhihin Rijkswaterstaat – Data-ICT-Department, P.O. Box 5023, 2600 GA Delft, The Netherlands

P.L. Luque School of Biological Sciences, University of Aberdeen, Aberdeen, AB24 2TZ, United Kingdom

C.A. McAlpine The University of Queensland, School of Geography, Planning and Architecture, Brisbane, QLD 4072, Australia

A. Mira Unidade de Biologia da Conservação, Departamento de Biologia Universidade de Évora, 7002-554 – Évora, Portugal

G.J. Pierce Instituto Español de Oceanografía, Centro Oceanográfico de Vigo, P.O. Box 1552, 36200, Vigo, España and University of Aberdeen, Oceanlab, Main Street, Newburgh, AB41 6AA, United Kingdom

I.G. Priede University of Aberdeen, Oceanlab, Main Street, Newburgh, AB41 6AA, United Kingdom

D. Raffaelli Environment, University of York, Heslington, York, YO10 5DD, United Kingdom

J.R. Rhodes The University of Queensland, School of Geography, Planning and Architecture, Brisbane, QLD 4072, Australia

M.B. Santos Vázquez Instituto Español de Oceanografía, Centro Oceanográfico de Vigo, P.O. Box 1552, 36200, Vigo, España

A.A. Saveliev Faculty of Ecology, Kazan State University, 18 Kremlevskaja Street, Kazan, 420008, Russia

G.M. Smith School of Science and Environment, Bath Spa University, Newton Park, Newton St Loe, Bath, BA2 9BN, United Kingdom

N.J. Walker Woodchester Park CSL, Tinkley Lane, Nympsfield, Gloucester GL10 3UJ, United Kingdom

A. Ward Central Science Laboratory, Sand Hutton, York, YO41 1LZ, United Kingdom

H. Weimerskirch Centre d'Etudes Biologiques de Chizé, Centre National de la Recherche Scientifique, 79360 Villiers en Bois, France

A.F. Zuur Highland Statistics LTD., 6 Laverock Road, Newburgh, AB41 6FN, United Kingdom

Chapter 1

Introduction

1.1 What Is in the Book?

Does your data have repeated measurements; is it nested (hierarchical)? Is it sampled at multiple locations or sampled repeatedly over time? Or is your response variable heterogeneous? Welcome to our world, the world of mixed effects modelling. The bad news is that it is a complicated world. Nonetheless, it is one that few ecologists can avoid, even though it is one of the most difficult fields in statistics. Many textbooks describe mixed effects modelling and extensions, but most are highly mathematical, and few focus on ecology.

We have met many scientists who have proudly showed us their copy of Pinheiro and Bates (2000) or Wood (2006), but admitted that these were really too technical for them to fully use. Of course, these two books are extremely good, but probably outside the reach of most non-mathematical readers.

The aim of this book is to provide a text on mixed effects modelling (and extensions) that can be read by anyone who needs to analyse their data without the (immediate) need to delve into the underlying mathematics. In particular, we focus on the following:

1. Generalised least squares (GLS) in Chapter 4. One of the main underlying assumptions in linear regression models (which include analysis of variance models) is homogeneity (constant variance). However, our experience has shown that most ecological data sets are heterogeneous. This is a problem that can be solved by using non-parametric tests, transformations, or analysing the raw data with GLS, which extends the linear regression by modelling the heterogeneity with covariates.
2. Mixed effects models and additive mixed effects models in Chapters 5, 6, and 7. We focus on regression and smoothing models for nested data (also called panel data or hierarchical data), repeated measurements, temporal correlated data, and spatial correlated data.
3. Generalised linear modelling (GLM) and generalised additive modelling (GAM) for count data, binary data, proportional data, and zero-inflated count data in Chapters 8–11.

4. Generalised estimation equations (GEEs) in Chapter 12. GEE can be used to analyse repeated measurements and longitudinal repeated measurements (over time) data. These can be continuous, binary, proportional, or count data.
5. Generalised linear mixed models (GLMMs) and generalised additive mixed models (GAMMs) in Chapter 13. GLMMs and GAMMs are used to model nested data and temporal and spatial correlation structures in count data or binomial data. These models combine mixed effects modelling and GLM and GAM.

When writing any technical book, a common starting point is to decide on the existing expertise of your target reader. Do we assume no existing expertise or do we assume a certain level of statistical background?

We decided that the entrance level for this text would be good knowledge of linear regression. This means we have assumed a familiarity with the underlying assumptions of linear regression, the model selection process, hypothesis testing procedures (t -test, F -test, and nested models), backward and forward selection based on the Akaike information criterion (or related information criteria), and model validation (assessing the underlying assumptions based on graphical or numerical tools using the residuals). Appendix A gives a short review of these procedures, and we recommend that you first familiarise yourself with the material in this appendix before continuing with Chapter 2. If you feel uncomfortable with the information in the appendix, then we recommend that you have a look at the regression chapters in, for example, Montgomery and Peck (1992), Fox (2002), or Quinn and Keough (2002). In fact, any book on linear regression will do. Also, our own book, Zuur et al. (2007), can be used.

The next question is then to decide who the book is to be aimed at. Since 2000, the first two authors of this book have given statistical courses for environmental scientists, biologists, ecologists, and other scientists; they have seen about 5000 participants in this time. The material covered in these courses is based on modules described in Zuur et al. (2007). For example, a popular course is the following one:

- Day 1: Data exploration.
- Day 2: Linear regression.
- Day 3: GLM.
- Day 4: GAM.
- Day 5: Catching up.

This is a 40-hour course and has been incorporated into MSc and PhD courses in several countries in Europe as well as being given as in-house and open courses at many universities and research institutes, mainly at biology departments. The problem with this course is that although you can teach people how to do linear regression, GLM, or GAM, the reality is that nearly all ecological data sets contain elements like nested data, temporal correlation, spatial correlation, data with lots of zeros, and heterogeneity. Hence, most ecologists for most of the time will need to apply techniques like mixed effects modelling, GLMM, GAMM, and models that can cope with lots of zeros (zero-inflated GLM). And it is for the user of this type of data that this book is primarily aimed at.

This book is also aimed at readers who want to gain the required knowledge by working through examples by downloading the code and data and try it for themselves before applying the same methods on their own data.

Two of the authors of this book are statisticians and speaking from their experience, having a book like this that first explains complicated statistical methods in a non-mathematical context and demonstrates them in case studies before digging into the underlying mathematics can still be extremely useful, even for the statistician!

The final question was what to write? We have already partially answered this question in the paragraphs above: statistical techniques that can cope with complicated data structures like nested data, temporal and spatial correlation, and repeated measurements for all types of data (continuous, binary, proportional, counts, and counts with lots of zeros).

1.1.1 To Include or Not to Include GLM and GAM

One of our dilemmas when writing this book was whether we should require the reader to be familiar with GLM and GAM before reading this book. We decided against this and have included GLM and GAM chapters in this book for the following reasons.

1. During the pre-publication review process, it became clear that many instructors would use this book to explain the full range of methods beyond linear regression. It, therefore, made sense to include GLM and GAM, allowing students to buy a single book containing all the methods beyond linear regression.
2. Most statistical textbooks written 5 or 10 years ago tend to discuss only logistic regression (for absence–presence and proportional data) and Poisson regression (for count data). In reality, Poisson regression hardly ever works for ecological count data due to its underlying assumption that the variance equals the mean of the data. For most ecological data sets, the variance is larger than the mean; this phenomenon is called overdispersion. Negative binomial GLMs and GAMs have become increasingly popular to deal with overdispersion. However, we still cover Poisson GLM as a pre-requisite to explain the negative binomial (NB) GLM.
3. Many ecological data sets also contain large number of zeros, and during the last 5 years, a new set of models have become popular in ecology to deal with this. These include zero-inflated Poisson GLMs and GAMs and zero-inflated negative binomial GLMs and GAMs. Zero inflated means that we have a data set with lots of zeros, more than we expect based on the Poisson or negative binomial distribution. The excessive number of zeros may (or may not!) cause overdispersion. Using these zero-inflated models means that we can often solve two problems at once: overdispersion and the excessive number of zeros. But again, before we can explain these zero-inflated models, we have to ensure that the reader is fully familiar with Poisson and logistic GLMs.

This explains why we have included text on the Poisson GLM, negative binomial GLM, and zero-inflated Poisson and the increasingly useful negative binomial GLMs and GAMs.

A few applications of zero-inflated Poisson GLMMs and zero-inflated negative binomial GLMMs/GAMMs have been published recently. However, there is hardly any fully tested software around that can be used to fit these zero-inflated GLMMs and GAMMs. So, although we decided to include the zero-inflated GLMs and GAMs in this book, we leave zero-inflated GLMMs and GAMMs for a future text.

1.1.2 Case Studies

A common criticism of statistical textbooks is that they contain examples using ‘ideal’ data. In this book, you will not find ozone data or Fisher’s iris data to illustrate how well certain statistical methods work. In contrast, we have only used data sets from consultancy projects and PhD research projects, where for many our first reaction was “How are we ever going to analyse these data?”

As well as the chapters on applied theory, this book also contains ten case study chapters with each case study showing a detailed data exploration, data analysis, discussion and a ‘what to write in a paper’ section. In the data exploration and data analysis section, we describe our thinking process, and in the ‘what to write in a paper’ section, we emphasise the key points for a paper.

It should be noted that our analysis approach for these data may not be the only one; as it is often the case, multiple statistical analyses can be applied to the same data set.

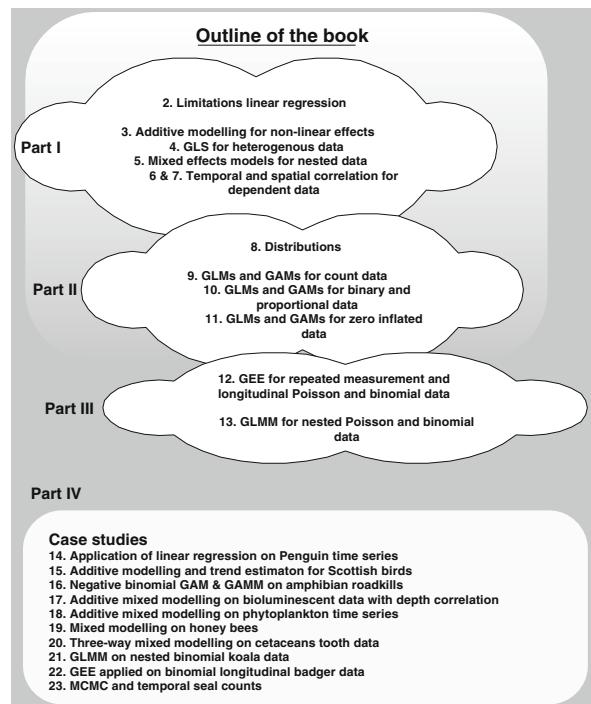
The data used in the case studies, and in the main text, are all available from the book’s website at www.highstat.com. The computer code is also available for downloading. If you want to use any of the data from this book for publications, please contact the owner of the data for permission. Contact details are given at the beginning of the book.

1.1.3 Flowchart of the Content

The flowchart in Fig. 1.1 gives a schematic overview of this book. In Part I, we start discussing the limitations of the linear regression model and show how these limitations can be solved with additive modelling, including random effects (resulting in mixed effects models), and temporal and spatial correlation. In Part II, we discuss GLM, GAM, and zero-inflated models. In Part III, we combine Parts I and II and discuss GEE, GLMM, and GAMM. Finally, in Part IV, we present ten case studies, each of them showing a detailed example using real data.

There are various ways to use this book. You can start reading the case studies, find one that matches your data, and apply the same steps on your own data. Then look up the corresponding theory. The alternative is to read the theory first, perhaps

Fig. 1.1 Outline of this book. In Part I, the limitations of linear regression are discussed, and various solutions are discussed (additive modelling for non-linear patterns, GLS for heterogeneity, mixed effects modelling for nested data, and correlation structures to deal with dependence). In the second part, GLM and GAM are introduced, and in the third part, these methods are extended towards GLMM and GAMM. In the last part, case studies are presented



concentrate on the numerous examples, and find a matching case study. Yet, a third option is to read the book from A to Z (which we obviously advise our readers).

Some sections are marked with an asterisk. These are more technical sections, or expand on ideas in the main text. They can be skipped on the first reading.

1.2 Software

There are many software packages available for mixed effects modelling, for example MLWIN, SPLUS, SAS, Stata, GENSTAT, and R. All have excellent facilities for mixed effects modelling and generalised linear mixed modelling; see West et al. (2006) for a comparison. As to GAM and GAMM, we can only recommend SPLUS or R. Stata seems to be particularly suited for negative binomial models, but has limited GAM facilities (at the time of writing).

Our choice is R (www.r-project.org), because it is good and it is free. There is no point teaching students a complicated computer language in a 2500 USD package if a future employer is unwilling to buy the same package. Because R is free, this is not an issue (unless the employer demands the use of a specific package).

If you are an instructor and use this book for teaching, we advise you start your class with an introductory course in R before starting with this book. We have tried teaching R and statistics at the same time, but have found this is rather challenging for the student.

The pre-requisite R knowledge required for this book is fairly basic and is covered in Appendix A; important commands are `boxplot`, `dotchart`, `pairs`, `lm`, `plot`, `summary`, and `anova`. Some basic R skills in data manipulating and plotting will also be useful, especially if the data contain missing values.

Instructors can contact us for an R survival guide that we wrote for our own courses. It contains all essential R code for pre-required knowledge for this book.

1.3 How to Use This Book If You Are an Instructor

We wrote this book with teaching in mind. When we teach, we tend to have groups consisting of 10–25 people (environmental scientists, biologists, etc.), mostly consisting of PhD students, post-docs, consultants, senior scientists, and the occasional brave MSc students. As people can only fully appreciate the text in this book if they have good knowledge of linear regression and basic R knowledge, our courses contain the following:

- Day 1: Revision of linear regression and R (half a day).
- Day 1 and 2: GLS.
- Day 3: Mixed effects modelling and additive mixed modelling.
- Day 4: Adding temporal and spatial correlation to linear regression, mixed effects models, and additive (mixed) models.
- Days 5 and 6: GEE, GLMM, and GAMM.

Each day is 8 hours of teaching and exercises. The case studies and detailed examples in the sections can be used as exercises. The schedule above is challenging, and depending on the pre-knowledge and number of questions, 48 hours may not be enough.

We have taught our courses in more than 20 different countries and noticed that there is a huge difference in mathematical and statistical knowledge of students. We have had groups of 60 MSc students where 20 had never seen any statistics at all, 20 were familiar with basic statistics, and 20 had done regression and GLM during their undergraduate courses and were keen to move on to GLMMs and GAMMs! This applies not only to MSc courses but also to postgraduate courses or courses at research institutes. Hence, teaching statistics is a challenge.

Before starting with the mixed effects modelling material, you need to ensure that all students are familiar with concepts like interaction, comparing full and nested models, model validation, sketching fitted values, and dealing with nominal variables.

1.4 What We Did Not Do and Why

During the writing of this book and when it was finished, we received comments from a large group of people, including the referees. This resulted in an enormous amount of ideas and suggestions on how to improve the text, and most of these

suggestions were included in the final version, but a few were not. As some of these topics are important for all readers, we decided to briefly discuss them.

Originally, our plan was to provide all the data in nicely prepared ASCII files and use the `read.table` command to import the data into R. However, data preparation is also part of the analyses, and we therefore decided to provide the data in the same format as was given to us. This means we put the reader through the same data preparation process that they would need to go through with their own data. With the `read.table` command, one has to store the data somewhere physically in a directory, e.g. on the C or D drive, and access it from there. However, not everyone may be able to store data on a C drive due to security settings or has a D drive. To avoid any confusion, we created a package (don't call it a library!) that contains all data sets used in this book. This means that any data set used in this book can be accessed with a single command (once the package has been installed). Our package is available from the book website at www.highstat.com. There, you can also find all the R code and data files in ASCII format, should you wish to use the `read.table` command.

It has also been suggested that we include appendices on matrix algebra and giving an introduction to R. We think that this would duplicate material from other books as many statistical textbooks already contain appendices on matrix algebra. As for R, we suggest you get a copy of Dalgaard (2002) and spend some time familiarising yourself with it. Appendix A shows what you need to know to get started, but R warrants spending additional time developing your expertise. We realise this means that you need to buy yet more books, but information on matrix algebra and R programming can also be obtained free from the Internet.

We have also deliberately decided not to add more mathematics into the text. If, after completing the book, you have a desire to dig further into the mathematical details, we recommend Pinheiro and Bates (2000) or Wood (2006).

1.5 How to Cite R and Associated Packages

This is an important issue. Without the effort of the people who programmed R and the packages that we have used, this book would not exist. The same holds for you; you have access to a free package that is extremely powerful. In recognition, it is appropriate therefore to cite R or any associated package that you use. Once in R, type

```
> citation()
```

and press enter. Do not type the `>` symbol. It gives the following text.

To cite R in publications use:

R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0,
URL <http://www.R-project.org>.

...

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also 'citation("pkgname")' for citing R packages.

The last lines suggest that for citing the `mgcv` or `nlme` packages (which we will use a lot), you should type

```
> citation("nlme")
> citation("mgcv")
```

It gives full details on how to cite these packages. In this book, we use a large number of packages. Citing them each time would drastically increase the number of pages; so for the sake of succinctness, we mention and cite them all below. In alphabetic order, the packages used in the book and their citations are as follows: AED (Zuur et al., 2009), BRugs (Thomas et al., 2006), coda (Plummer et al., 2007), Design (Harrell, 2007), gam (Hastie, 2006), geepack (Yan, 2002; Yan and Fine 2004), geoR (Ribeiro and Diggle, 2001), glmmML (Broström, 2008), gstat (Pebesma, 2004), lattice (Sarkar, 2008), lme4 (Bates and Sarkar, 2006), lmtest (Zeileis and Hothorn, 2002), MASS (Venables and Ripley, 2002), mgcv (Wood, 2004; 2006), ncf (Bjornstad, 2008), nlme (Pinheiro et al., 2008), pscl (Jackman, 2007), scatterplot3d (Ligges and Mächler, 2003), stats (R Development Core Team, 2008), and VGAM (Yee, 2007). The reference for R itself is R Development Core Team (2008). Note that some references may differ depending on the version of R used. While writing this book, we used versions 2.4.0–2.7.0 inclusive, and therefore, some references are to packages from 2006, while others are from 2008.

1.6 Our R Programming Style

One of the good things about R is also, perversely, a problem; everything can be done in at least five different ways. To many, of course, this is a strength of R, but for beginners it can be confusing. We have tried to adopt a style closely matching the style used by Pinheiro and Bates (2000), Venables and Ripley (2002), and Dalgaard (2002). However, sometimes these authors simplify their code to reduce its length, minimise typing, and speed up calculation. For example, Dalgaard (2002) uses the following code to print the output of a linear regression model:

```
> summary(lm(y ~ x1 + x2))
```

An experienced R user will see immediately that this combines two commands; the `lm` is used for linear regression, and its output is put directly into the `summary` command, which prints the estimated parameters, standard errors, etc. Writing optimised code, such as this, is good practice and in general something to be

encouraged. However, in our experience, while teaching statistics to R beginners, it is better to explicitly write code as easily followed steps, and we would write the above examples as

```
M1 <- lm(y ~ x1 + x2)
summary(M1)
```

We call this a – b – c programming; first a, then b, and finally c. This may not produce the most elegant or most efficient code, but its simplicity makes it easier to follow when learning R.

1.7 Getting Data into R

The most difficult thing in learning a new stats package is to import your data and start working with it. As an example of importing data in R, we use data from Cronin (2007), which is also used in Chapter 23. The following R code reads the data. We assume the data are available as a text (tab-delimited) file ‘Seals.txt’ on the C drive in the directory ‘Bookdata’. The following code reads the data into R:

```
> Seals <- read.table(file = "C:\\\\Bookdata\\\\Seals.txt",
  header = TRUE)
```

The `>` symbol is used to mimic the R commander. You should not type it into R! R commands are case sensitive; so make sure you type in commands exactly as illustrated. The `header = TRUE` option tells R that the first row contains headers (the alternative is `FALSE`). The data are stored in a data frame called `Seals`, which is a sort of data matrix. Information in a data frame can be accessed in various ways.

If you just type in `Abun` (the column with abundances), R gives an error message saying that it does not know what `Abun` is. There are various options to access the variables inside the object `Seals`. You can use commands like

```
> hist(Seals$Abun)
```

to make a histogram of the abundance. The `$` sign is used to access variables inside the object `Seals`. It is also possible to work along the lines of

```
> A <- Seals$Abund
> hist(A)
```

First, we define a new variable `A` and then work with this. The advantage is that you don’t have to use the `Seals$` all the time. Option three is to access the data via columns of the object `Seals`:

```
> A <- Seals[,1]
> hist(A)
```

A fourth option is to provide the `Seals` object as an argument to the function that you use, e.g.

```
> lm(Abun ~ factor(Site), data = Seals)
```

The `data` option specifies that R has to use the data in the object `Seals` for the linear regression. Yet, a fifth option is to use the `attach(Seals)` command. This command tells R to look also inside the object `Seals`; hence, R will have access to anything that you put in there. Its advantage is that with one command, you avoid typing in lots of data preparation commands. In writing a book, it saves space. In classroom teaching, it can be an advantage too because students don't have to type all the `$` commands.

However, at this point, the R experts tend to stand up and say that it is all wrong; they will tell you not to use the `attach` command. The reason is that you can attach multiple objects, and misery may happen if multiple objects contain the same variable names. This may cause an error message (if you are lucky). The other problem is that you may (accidentally) attach the same object twice. If you then make changes to a variable (e.g. a transformation), R may use the other (unchanged) copy during the analysis without telling you! Our advise is not to use the `attach` command, and if you decide to use it, be very careful!

1.7.1 Data in a Package

In this book, we use at least 30 different data sets. Instead of copying and pasting the `read.table` command for each example and case study, we stored all data in a package called AED (which stands for Analysing Ecological Data). It is available from the book website at www.highstat.com. As a result, all you have to do is to download it, install it (Start R, click on Packages, and select ‘Install package from local zip file’), and then type

```
> library(AED)
> data(Seals)
```

Instead of the `Seals` argument in the function `data`, you can use any of the other data sets used in this book. To save space, we tend to put both commands on one line:

```
> library(AED); data(Seals)
```

You must type the “`;`” symbol. You can even use a fancy solution, namely

```
> data(Seals, package = "AED")
```

Chapter 2

Limitations of Linear Regression Applied on Ecological Data

This chapter revises the basic concepts of linear regression, shows how to apply linear regression in R, discusses model validation, and outlines the limitations of linear regression when applied to ecological data. Later chapters present methods to overcome some of these limitations; but as always before doing any complicated statistical analyses, we begin with a detailed data exploration. The key concepts to consider at this stage are outliers, collinearity, and the type of relationships between the variables. Failure to apply this initial data exploration may result in an inappropriate analysis forcing you to reanalyse your data and rewrite your paper, thesis, or report.

We assume that the reader is ‘reasonably’ familiar with data exploration and linear regression techniques. This book is a follow-up to *Analysing Ecological Data* by Zuur et al. (2007), which discusses a wide range of exploration and analytical tools (including linear regression and its extensions), together with several related case study chapters. Other useful, non-mathematical textbooks containing regression chapters include Chambers and Hastie (1992), Fox (2002), Maindonald and Braun (2003), Venables and Ripley (2002), Dalgaard (2002), Faraway (2005), Verzani (2005) and Crawley (2002, 2005). At a considerable higher mathematical level, Ruppert et al. (2003) and Wood (2006) are excellent references for linear regression and extensions. All these books discuss linear regression and show how to apply it in R. Other good, but not based on R, textbooks include Montgomery and Peck (1992), Draper and Smith (1998) and Quinn and Keough (2002). Any of the above mentioned texts using R can be also used to learn R, but we highly recommend the book from Dalgaard (2002) or for a slightly different approach, Crawley (2005). However, even if you are completely unfamiliar with R, you should still be able to pick up the essentials from this book and ‘learn it as you go along’. It is not that difficult and, once exposed to R, you will never use anything else.

Although various linear regression examples are given in this chapter, a complete example, including all R code and aspects like interaction, model selection and model validation steps, is given in Appendix A.

2.1 Data Exploration

2.1.1 Cleveland Dotplots

The first step in any data analysis is the data exploration. An important aspect in this step is identifying outliers (we discuss these later) and useful tools for this are boxplots and/or Cleveland dotplots (Cleveland, 1993). As an example of data exploration, we start with data used in Ieno et al. (2006). To identify the effect of species density on nutrient generation in the marine benthos, they applied a two-way ANOVA with nutrient concentration as the response variable with density of the deposit-feeding polychaete *Hediste diversicolor* (*Nereis diversicolor*), and nutrient type (NH₄-N, PO₄-P, NO₃-N) as nominal explanatory variables. The data matrix consists of three columns labelled concentration, biomass, and nutrient type. The aim is to model Nereis concentration as a function of biomass and nutrient. The following R code reads the data and makes a Cleveland dotplot.

```
> library(AED); data(Nereis)
```

R commands are case sensitive; so make sure you type in commands exactly as illustrated. The data are stored in a data frame called *Nereis*, which is a sort of data matrix. Information in a data frame can be accessed in various ways. First, we need to know what is in there, and this is done by typing the following at the R prompt:

```
> names(Nereis)
```

This command gives the names of all variables in the data frame:

```
[1] "concentration" "biomass" "nutrient"
```

The following lines of code produce the Cleveland dotplot in Fig. 2.1A.

```
> dotchart(Nereis$concentration,
           ylab = "Order of observations",
           xlab = "Concentration", main = "Cleveland dotplot")
```

The *dotchart* function makes the Cleveland dotplot. Note that the arguments of the *dotchart* function are typed in over multiple rows. When the code runs over more than one line like this, you should ensure that the last symbol on such a line is a slash (\) or a comma (,). So, this works as well:

```
> dotchart(Nereis$concentration, ylab = "Order of \
           observations",
           xlab = " \
           Concentration", main = "Cleveland dotplot")
```

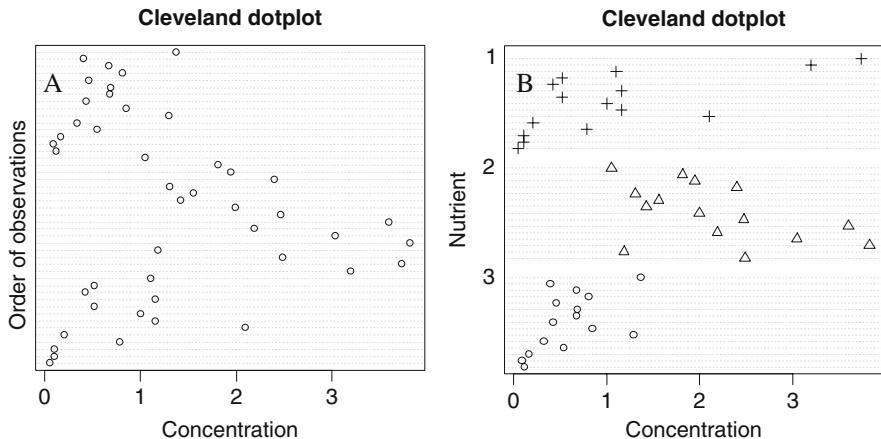


Fig. 2.1 **A:** Cleveland dotplot for Nereis concentration. **B:** Conditional Cleveland dotplot of Nereis concentration conditional on nutrient with values 1, 2 and 3. Different symbols were used, and the graph suggests violation of homogeneity. The *x*-axes show the value at a particular observation, and the *y*-axes show the observations

In a dotchart, the first row in the text file is plotted as the lowest value along the *y*-axis in Fig. 2.1A, the second observation as the second lowest, etc. The *x*-axis shows the value of the concentration for each observation. By itself, this graph is not that spectacular, but extending it by making use of the grouping option in `dotchart` (for further details type: `?dotchart` in R) makes it considerably more useful, as can be seen from Fig. 2.1B. This figure was produced using the following command:

```
> dotchart(Nereis$concentration,
           groups = factor(Nereis$nutrient),
           ylab = "Nutrient", xlab = "Concentration",
           main = "Cleveland dotplot", pch = Nereis$nutrient)
```

The `groups = factor(nutrient)` bit ensures that observations from the same nutrient are grouped together, and the `pch` command stands for point character. In this case, the nutrient levels are labelled as 1, 2 and 3. If other characters are required, or nutrient is labelled as alpha-numerical values, then you have to make a new column with the required values. To figure out which number corresponds to a particular symbol is a matter of trial and error, or looking it up in a table, see, for example, Venables and Ripley (2002).

Cleveland dotplots are useful to detect outliers and violation of homogeneity. Homogeneity means that the spread of the data values is the same for all variables, and if this assumption is violated, we call this heterogeneity. Points on the far end along the horizontal axis (extremely large or extremely small values) may be considered outliers. Whether such points are influential in the statistical analysis depends

on the technique used and the relationship between the response and explanatory variables. In this case, there are no extremely large or small values for the variable concentration values. The Cleveland dotplot in Fig. 2.1B indicates that we may expect problems with violation of homogeneity in a linear regression model applied on these data, as the spread in the third nutrient is considerable smaller than that in the other two. The mean concentration value of nutrient two seems to be larger, indicating that in a regression model, the covariate nutrient will probably play an important role.

2.1.2 Pairplots

Another essential data exploration tool is the pairplot obtained by the R command

```
> pairs(Nereis)
```

The resulting graph is presented in Fig. 2.2. Each panel is a scatterplot of two variables. The graph does not show any obvious relationships between concentration and biomass, but there seems to be a clear relationship between concentration and

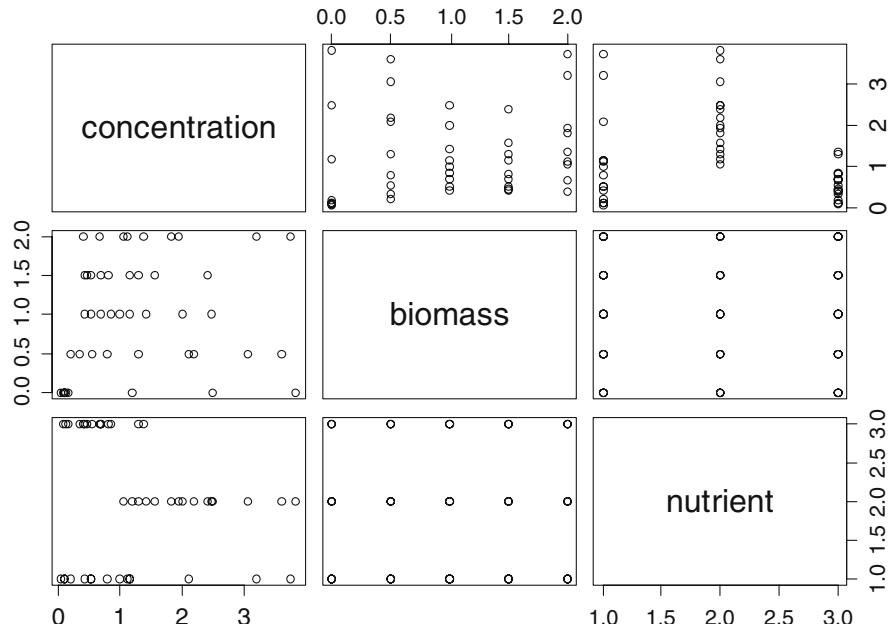


Fig. 2.2 Pairplot for concentration, biomass and nutrient. Each panel is a scatterplot between two variables. It is also possible to add regression or smoothing lines in each panel. In general, it does not make sense to add a nominal variable (nutrient) to a pairplot. In this case, there are only two explanatory variables; hence, it does not do any harm to include nutrient

nutrients, as already suggested by the Cleveland dotplot. More impressive pairplots can be made by using the `panel` option in `pairs`. The help file for `pairs` is obtained by typing: `?pairs`. It shows various examples of pairplot code that gives pairplots with histograms along the diagonal, correlations in the lower panels, and scatterplots with smoothers in the upper diagonal panels.

2.1.3 Boxplots

Another useful data exploration tool that should be routinely applied is the boxplot. Just like the Cleveland dotplot, it splits up the data into groups based on a nominal variable (for example nutrient). The boxplot of concentration conditional on nutrient is given in Fig. 2.3. The following code was used to generate the graph:

```
> boxplot(concentration ~ factor(nutrient),
  varwidth = TRUE, xlab = "nutrient",
  main = "Boxplot of concentration conditional on\\
nutrient", ylab = "concentration", data = Nereis)
```

The `varwidth = TRUE` command ensures that the width of each boxplot is proportional to the sample size per level. In this case, the sample size per nutrient (labelled 1, 2, and 3) is about the same.

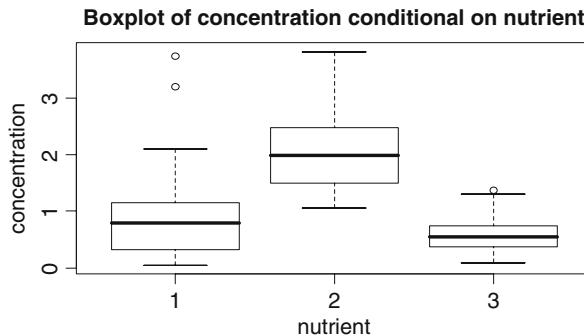


Fig. 2.3 Boxplot of concentration conditional on the nominal variable nutrient. The horizontal line in each box is the median, the boxes define the hinge (25–75% quartile, and the line is 1.5 times the hinge). Points outside this interval are represented as dots. Such points may (or may not) be outliers. One should not label them as outliers purely on the basis of a boxplot! The width of the boxes is proportional to the number of observations per class

2.1.4 xyplot from the Lattice Package

As with the Cleveland dotplot and the pairplot, the boxplot shows that there may be a nutrient effect: higher mean concentration values for nutrient level 2, but also

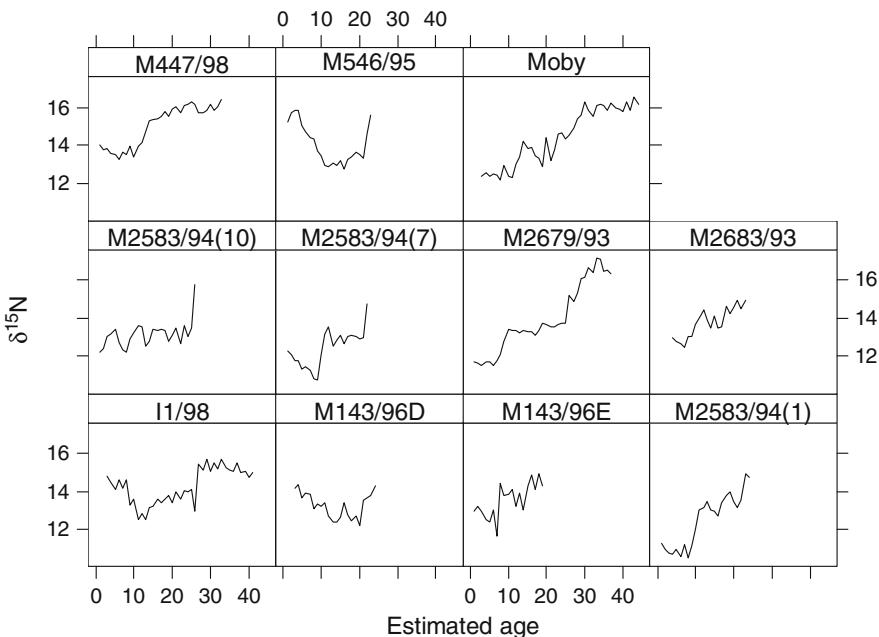


Fig. 2.4 Nitrogen concentration in teeth versus age for each of the 11 whales stranded in Scotland. The graph was made with the `xypplot` from the lattice package

less spread for nutrient level 3, indicating potential heterogeneity problems later on. We now show a more advanced data exploration method. As the Nereis data set has only two explanatory variables, this method is less appropriate for these data, and therefore we use a different data set.

Just like rings in trees, teeth of an animal have rings, and from these it is possible to extract information on how chemical variables have changed during the life of the animal. Mendes et al. (2007) measured the nitrogen isotopic composition in growth layers of teeth from 11 sperm whales stranded in Scotland. The underlying aim of the research was to ‘investigate the existence, timing, rate and prevalence of dietary and/or foraging location shifts that might be indicative of ontogenetic benchmarks related to changes in schooling behaviour, movements, environmental conditions, foraging ecology and physiology’ (Mendes et al., 2007).

Figure 2.4 shows an `xypplot` from the lattice package. The name lattice is used in R, but in SPLUS it is called a Trellis graph. It consists of a scatterplot of nitrogen isotope ratios versus age for each whale. Working with lattice graphs is difficult, and one of the few books on this topic is Sarkar (2008). One of the underlying questions is whether all whales have similar nitrogen-age relationships, and the graph suggests that some whales indeed have similar patterns. The R code to generate the graph in Fig. 2.4 is

```
> library(AED); data(TeethNitrogen)
> library(lattice)
> xyplot(X15N ~ Age | factor(Tooth), type = "l",
  xlab = "Estimated age", col = 1,
  ylab = expression(paste(delta^{15}, "N")),
  strip = function(bg = 'white', ...),
  strip.default(bg = 'white', ...),
  data = TeethNitrogen)
```

The `xyplot` makes the actual graph, and the rest of the code is merely there to extract the data. The `type = "l"` and `col = 1` means that a line in black colour is drawn. Note that the 1 in `type` stands for lines, not for the 1 from 1, 2, and 3. But the 1 for `col` is a number! The complicated bit for the y-label is needed for subscripts, and the `strip` code is used to ensure that the background colour in the strips with whale names is white. It can be difficult to figure out this type of information, but you quickly learn the coding you use regularly. To make some journal editors happy, the following code can be added before the last bracket to ensure that tick marks are pointing inwards: `scales = list(tck = c (-1, 0))`. More data exploration tools will be demonstrated later in this book.

2.2 The Linear Regression Model

In the second step of the data analysis, we have to apply some sort of model, and the ‘mother of all models’ is without doubt the linear regression model. The *bivariate* linear regression model is defined by

$$Y_i = \alpha + \beta \times X_i + \varepsilon_i \quad \text{where} \quad \varepsilon_i \sim N(0, \sigma^2)$$

The Y_i is the response (or dependent) variable, and X_i is the explanatory (or independent) variable. The unexplained information is captured by the residuals ε_i , and these are assumed to be normally distributed with expectation 0 and variance σ^2 . The parameters α and β are the population intercept and slope and are unknown. In practice, we take a sample and use this to come up with estimates a and b and confidence intervals. These confidence intervals tell us that if we repeat the experiment a large number of times, how often the real (fixed and unknown) α and β are in the interval based on the confidence bands (which will differ for each experiment!). A typical choice is the 95% confidence interval. In most cases, β (the slope) is of primary interest as it tells us whether there is a relationship between Y and X .

So, we take a sample of size N and obtain the estimators a and b plus confidence intervals. And then, we make a statement on the population parameters α and β . But this is a big thing to do! You may wonder how it is possible that we can do this. Well, the magic answer is ‘assumptions’. The fact that you take sample data and use this to make a statement on population parameters is based on a series of

assumptions, namely, normality, homogeneity, fixed X, independence, and correct model specification.

The underlying geometric principle of linear regression is shown in Fig. 2.5 (based on Figs. 5.6 and 5.7 in Zuur et al. (2007), and Fig. 14.4 in Sokal and Rohlf (1995)). The data used in this graph is from a benthic study carried out by RIKZ in The Netherlands. Samples at 45 stations along the coastline were taken and benthic species were counted. To measure diversity, the species richness (the different number of species) per site was calculated. A possible factor explaining species richness is Normal Amsterdams Peil (NAP), which measures the height of a site compared to average sea level, and represents a measure of food for birds, fish, and benthic species. A linear regression model was applied, and the fitted curve is the straight line in Fig. 2.5. The Gaussian density curves on top of the line show the probability of other realisations at the same NAP values. Another ‘realisation’ can be thought of as going back into the field, taking samples at the same environmental conditions, carry out the species identification, and again determining species richness per site. Obviously, you will not find exactly the same results. The normality assumption means that for each NAP value, we have bell-shaped curves determining the probabilities of the (species richness) values of other realisations or sub-samples. Homogeneity means that the spread of all Gaussian curves is the same at all NAP values.

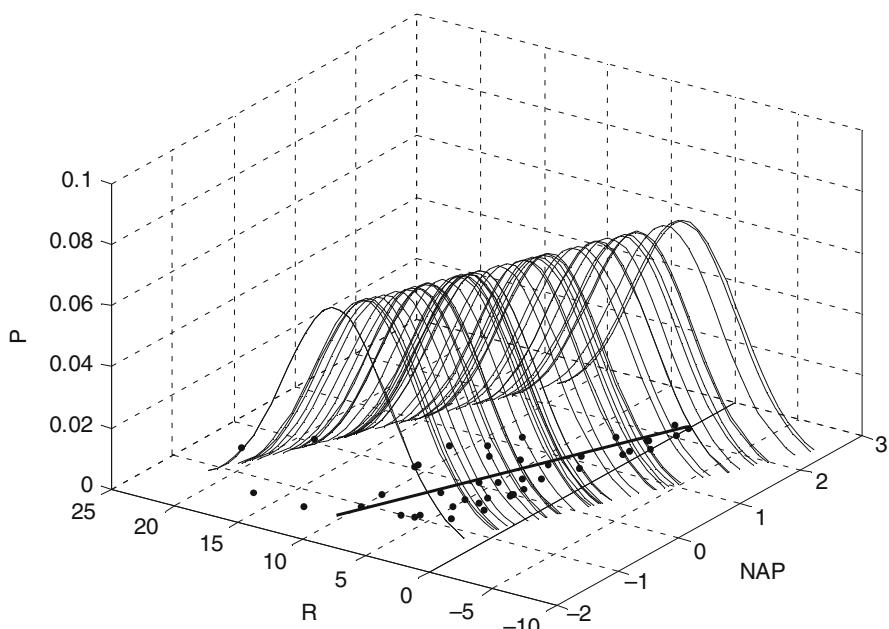


Fig. 2.5 Regression curve for all 45 observations from the RIKZ data discussed in Zuur et al. (2007) showing the underlying theory for linear regression. NAP is the explanatory variable, R (species richness) is the response variable, and the third axis labelled ‘P’ shows the probability of other realisations

Multiple linear regression is an extension of bivariate linear regression in the sense that multiple explanatory variables are used. The underlying model is given by

$$Y_i = \alpha + \beta_1 \times X_{1i} + \beta_2 \times X_{2i} + \dots + \beta_M \times X_{Mi} + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma^2)$$

There are now M explanatory variables. Visualising the underlying theory as in Fig. 2.5 is not possible, as we cannot draw a high dimensional graph on paper, but the same principle applies. Further information on bivariate and multiple linear regression are discussed in the examples below and in Appendix A.

2.3 Violating the Assumptions; Exception or Rule?

2.3.1 Introduction

One of the questions that the authors of this book are sometimes faced with is: ‘Why do we have to do all this GLM, GAM, mixed modelling, GLMM, and GAMM stuff? Can’t we just apply linear regression on our data?’ The answer is always in a ‘Yes you can, but . . .’ format. The ‘but . . .’ refers to the following. Always apply the simplest statistical technique on your data, but ensure it is applied correctly! And here is a crucial problem. In ecology, the data are seldom modelled adequately by linear regression models. If they are, you are lucky. If you apply a linear regression model on your data, then you are implicitly assuming a whole series of assumptions, and once the results are obtained, you need to verify all of them. This is called the model validation process. We already mentioned the assumptions, but will do this again; (i) normality, (ii) homogeneity, (iii) fixed X (X represents explanatory variables), (iv) independence, and (v) a correct model specification. So, how do we verify these assumptions, and what should we do, if we violate some, or all of them? We discuss how to verify these assumptions using five examples later in this section with each example violating at least one assumption. What should we do if we violate all the assumptions? The answer is simple: reject the model. But what do we do if we only violate one of the assumptions? And how much can we violate the assumptions before we are in trouble? We discuss this later.

2.3.2 Normality

Several authors argue that violation of normality is not a serious problem (Sokal and Rohlf, 1995; Zar, 1999) as a consequence of the central limit theory. Some authors even argue that the normality assumption is not needed at all provided the sample size is large enough (Fitzmaurice et al., 2004). Normality at each X value should be checked by making a histogram of all observations at *that* particular X value. Very often, we don’t have multiple observations (sub-samples) at each X value. In that case, the best we can do is to pool all residuals and make a histogram of the

pooled residuals; normality of the pooled residuals is reassuring, but it does not imply normality of the population data.

We also discuss how not to check for normality as the underlying concept of normality is grossly misunderstood by many researchers. The linear regression model requires normality of the data, and therefore of the residuals at *each X* value. The residuals represent the information that is left over after removing the effect of the explanatory variables. However, the raw data *Y* (*Y* represents the response variable) contains the effects of the explanatory variables. To assess normality of the *Y* data, it is therefore misleading to base your judgement purely on a histogram of all the *Y* data. The story is different if you have a large number of replicates at each *X* value. Summarising, unless you have replicated observations for each *X* value, you should not base your judgment of normality based on a histogram of the raw data. Instead, apply a model, and inspect the residuals.

2.3.3 Heterogeneity

Ok, apparently we can get away with a small amount of non-normality. However, heterogeneity (violation of homogeneity), also called heteroscedasticity, happens if the spread of the data is not the same at each *X* value, and this can be checked by comparing the spread of the residuals for the different *X* values. Just as in the previous subsection, we can argue that most of the time, we don't have multiple observations at each *X* value, at least not in most field studies. The only thing we can do is to pool all the residuals and plot them against fitted values. The spread should be roughly the same across the range of fitted values. Examples of such graphs are provided later. In sexual dimorphism, female species may show more variation than male species (or the other way around depending on species). In certain ecological systems, there may be more spread in the summer than in the winter, or less spread at higher toxicated sites, more spread at certain geographical locations, more variation in time due to accumulation of toxic elements, etc. In fact, we have seldom seen a data set in which there was no heterogeneity of some sort. The easiest option to deal with heterogeneity is a data transformation. And this is where the phrase 'a mean-variance stabilising' transformation comes from.

Many students have criticised us for using graphical techniques to assess homogeneity, which require some level of subjective assessment rather than using one of the many available tests. The problem with the tests reported by most statistical software packages, and we will illustrate some of them later, is that they require normality. For example, Barlett's test for homogeneity is quite sensitive to non-normality (Sokal and Rohlf, 1995). We therefore prefer to assess homogeneity purely based on a graphical inspection of the residuals.

Minor violation of homogeneity is not too serious (Sokal and Rohlf, 1995), but serious heterogeneity is a major problem. It means that the theory underlying the linear regression model is invalid, and although the software may give beautiful

p-values, *t*-values and *F*-values, you cannot trust them. In this book, we will discuss various ways to deal with heterogeneity.

2.3.4 Fixed *X*

Fixed *X* is an assumption implying that the explanatory variables are deterministic. You know the values at each sample in advance. This is the case if you a priori select sites with a preset temperature value or if you choose the amount of toxin in a basin. But if you go into the field, take at random a sample, and then measure the temperature or the toxin concentration, then it is random. Chapter 5 in Faraway (2005) gives a very nice overview how serious violation of this assumption results in biased regression parameters. The phrase ‘biased’ means that the expected value for the estimate parameter does not equal the population value. Fortunately, we can ignore the problem if the error in determining the explanatory variable is small compared to the range of the explanatory variable. So, if you have 20 samples where the temperature varies between 15 and 20 degrees Celsius, and the error of your thermometer is 0.1, then you are ok. But the age determination of the whales in Fig. 2.4 may be a different story as the range of age is from 0 to 40 years, but the error on the age reading may (or may not) be a couple of years. There are some elegant solutions for this (see the references for this in Faraway (2005)), but in Chapter 7 we (shortly) discuss the use of a brute force approach (bootstrapping).

2.3.5 Independence

Violation of independence is the most serious problem as it invalidates important tests such as the *F*-test and the *t*-test. A key question is then how do we identify a lack of independence and how do deal with it. You have violation of independence if the *Y* value at X_i is influenced by other X_i (Quinn and Keough, 2002). In fact, there are two ways that this can happen: either an improper model or dependence structure due to the nature of the data itself. Suppose you fit a straight line on a data set that shows a clear non-linear pattern between *Y* and *X* in a scatterplot. If you plot the residuals versus *X*, you will see a clear pattern in the residuals: the residuals of samples with similar *X* values are all positive or negative. So, an improper model formulation may cause violation of independence. The solution requires a model improvement, or a transformation to ‘linearise the relationship’. Other causes for violation of independence are due to the nature of the data itself. What you eat now depends on what you were eating 1 minute ago. If it rains at 100 m in the air, it will also rain at 200 m in the air. If we have large numbers of birds at time *t*, then it is likely that there were also large numbers of birds at time *t* – 1. The same holds for spatial locations close to each other and sampling pelagic bioluminescence along a depth gradient. This type of violation of independence can be taken care of by incorporating a temporal or spatial dependence structure between the observations (or residuals) in the model.

The case studies later in the book contain various examples of both scenarios, but for now we look at a series of examples where some of these important assumptions have been violated.

2.3.6 Example 1; Wedge Clam Data

Figure 2.6 shows a coplot of biomass (labelled as AFD which stands for ash free dry weight) of 398 wedge clams (*Donax hanleyanus*) plotted against length for six different months (Ieno, unpublished data). The data used in this section were measured on a beach in Argentina in 1997. An initial scatterplot of the data (not shown here) showed a clear non-linear relationship, and therefore, both AFD and length were log-transformed to linearise the relationship. Note this transformation is only necessary if we want to apply linear regression. As an alternative, the untransformed data can be analysed with additive modelling (Chapter 3). The coplot in Fig. 2.6 indicates a clear linear relationship between AFD and length in all months, and it seems sensible to apply linear regression to model this relationship. Due to different stages of the life cycle of wedge clams, the biomass-length relationship may change between months, especially before and after the spawning period in September–October and February–March. This justifies adding a length-month interaction term. This model is also known as an analysis of covariance (ANCOVA). The following R code was used for the coplot (Fig. 2.6) and the linear regression model.

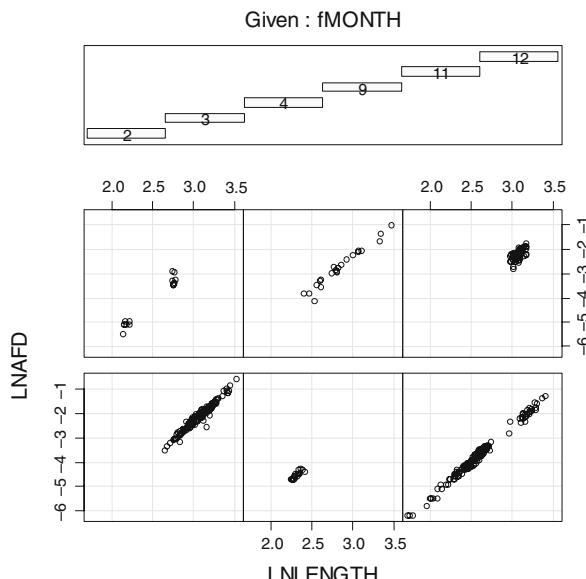


Fig. 2.6 Coplot of the wedge clam data during the spring and summer period. (The data were taken on the southern hemisphere.) The lower left panel contains the data from month 2, the lower right of month 4, the upper left from month 9, and the upper right of month 12

```
> library(AED); data(Clams)
> Clams$LNAFD <- log(Clams$AFD)
> Clams$LNLENGTH <- log(Clams$LENGTH)
> Clams$fMONTH <- factor(Clams$MONTH)
> library(lattice)
> coplot(LNAFD ~ LNLENGTH | fMONTH, data = Clams)
> M1 <- lm(LNAFD ~ LNLENGTH * fMONTH, data = Clams)
> drop1(M1, test = "F")
```

The `drop1` command compares the full model with a model in which the interaction is dropped, and an F -test is used to compare the residual sum of squares of both the models (Appendix A):

```
Single term deletions
Model: LNAFD ~ LNLENGTH * fMONTH
          Df Sum of Sq    RSS      AIC F value    Pr(F)
<none>              6.36 -1622.35
LNLENGTH:fMONTH 5     0.23   6.58 -1618.47 2.7385 0.01906
```

On the third line of this output (labelled as none), we have the output of the full model, and the last line shows the output from the model without the interaction. Note that this model is nested within the full model. The F -statistic shows that the interaction is significant at the 5% level. However, before trusting the values obtained by the F -statistic and use the ‘magic’ 5% as rejection level, we need to be confident that all model assumptions are valid. Hence, we enter the next stage of the analysis, the model validation.

2.3.6.1 Model Validation

Standard model validation graphs are (i) residuals versus fitted values to verify homogeneity, (ii) a QQ-plot or histogram of the residuals for normality, and (iii) residuals versus each explanatory variable to check independence, see Fig. 2.7. We also need to check whether there are any influential observations. The following R code was used to generate Fig. 2.7.

```
> op <- par(mfrow = c(2, 2), mar = c(5, 4, 1, 2))
> plot(M1, add.smooth = FALSE, which = 1)
> E <- resid(M1)
> hist(E, xlab = "Residuals", main = "")
> plot(Clams$LNLENGTH, E, xlab = "Log(Length)",
       ylab = "Residuals")
> plot(Clams$fMONTH, E, xlab = "Month",
       ylab = "Residuals")
> par(op)
```

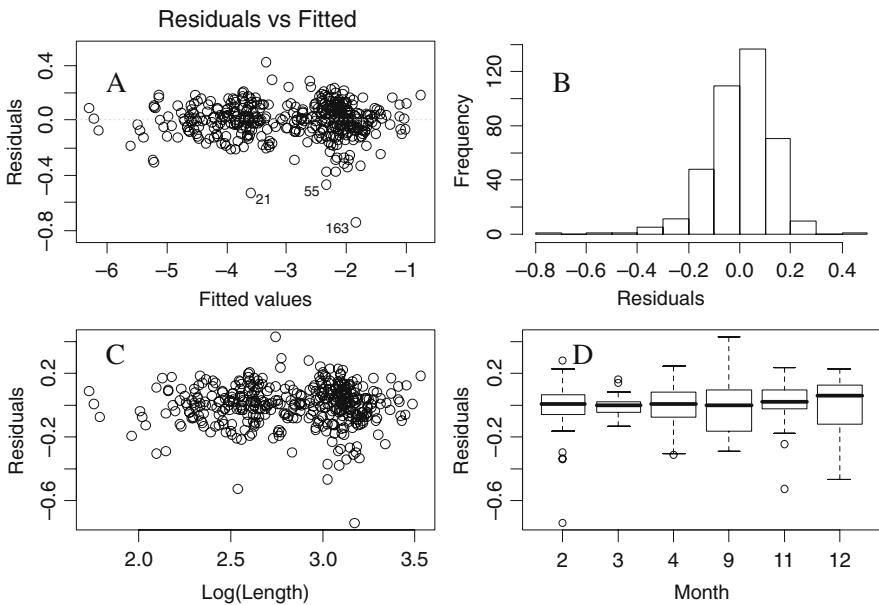


Fig. 2.7 Model validation graphs. **A:** Fitted values versus residuals (homogeneity). **B:** Histogram of the residuals (normality). **C:** Residuals versus length (independence). **D:** Residuals versus month

The first line specifies a graphical window with four panels and a certain amount of white space around each panel. The last command `par(op)` sets the graphical settings back to the default values. There seems to be minor evidence of non-normality (Fig. 2.7B), and more worrying, the spread in the residuals is not the same at all length classes and months (Fig. 2.7A, C, D). In month 3, there is less spread than in other months. A and C of Fig. 2.7 are similar in this case, but if we had a larger number of explanatory variables, these panels would no longer share this similar appearance.

The residuals play an essential part in the model validation process. Residuals are defined as observed values minus fitted values (we call these the ordinary residuals). However, it is also possible to define other types of residuals, namely standardised residuals and Studentised residuals. In Appendix A, we discuss the definition of the standardised residuals. These have certain theoretical advantages over the ordinary residuals, and it better to use these in the code above. Studentised residuals are useful for identifying influential observations. They are obtained by fitting a linear regression model using the full data set, and the same regression model on a data set in which one observation is dropped (in turn), and predicting the value of the dropped observation (Zuur et al., 2007). We do not use Studentised residuals here. However, if you do a good data exploration and deal with outliers at that stage, then ordinary, standardised, and Studentised residuals tend to be very similar (in terms of patterns).

Instead of a visual inspection, it is also possible to apply a test for homogeneity. Sokal and Rohlf (1995) describe three such tests, namely the Barlett's test for homogeneity, Hartley's F_{\max} test and the log-anova, or Scheffé-Box test. Faraway (2005) gives an example of the F -test. It uses the ratio of variances. Panel 2.7C suggests that the observations for $\log(\text{Length})$ less than 2.275 have a different spread than those larger than 2.275. The following code applies the F -ratio test, and the output is given immediately after the code.

```
> E1 <- E[Clams$LNLENGTH <= 2.75]
> E2 <- E[Clams$LNLENGTH > 2.75]
> var.test(E1, E2)

F test to compare two variances data: E1 and E2
F = 0.73, num df = 161, denom df = 235, p-value = 0.039
alternative hypothesis: true ratio of variances is not
equal to 1
95 percent confidence interval: 0.557 0.985
sample estimates: ratio of variances: 0.738
```

The null hypothesis (H_0) in this test is that the ratio of the two variances is equal to 0, and the test suggests rejecting it at the 5% level. However, $p = 0.04$ is not very convincing. On top of this, the choice for 2.275 is rather arbitrary. We can easily fiddle around with different cut-off levels and come up with a different conclusion. We could also use the F_{\max} to test whether residuals in different months have the same spread (see page 397 in Sokal and Rohlf, 1995). We will address the same question with the Bartlett test for homogeneity. The null hypothesis is that variances in all months are the same. The following code and output shows that we can reject the null hypothesis at the 5% level.

```
> bartlett.test(E, Clams$fMONTH)

Bartlett test of homogeneity of variances
data: E and MONTH
Bartlett's K-squared = 34.28, df = 5, p-value = <0.001
```

The problem with the Bartlett test is that it is rather sensitive to non-normality; hence, one should make histograms of residuals per month. Results are not presented here, but the R command `hist(E[Clams$MONTH == 12])` gives a bimodal histogram.

The conclusion of the linear regression (or ANCOVA) model is that there is a significant relationship between biomass, length, and month with a weak but significant interaction between the length and the month. However, with a p -value of 0.02 for this interaction term, we would have preferred to see no patterns at all in the residuals. Both the tests and graphical output, gave us some reasons to doubt the suitability of this model for these data. In Chapter 4, we discuss extensions of the linear regression model that can be used to test whether we need different variances per month.

2.3.7 Example 2; Moby's Teeth

Figure 2.4 showed nitrogen isotope ratios in teeth of stranded whales. One of which became famous and attracted newspaper headlines when it stranded in Edinburgh, Scotland, and was nicknamed ‘Moby the whale’. The graph in Fig. 2.4 indicates that Moby’s isotope ratios increased with age, and a linear regression was applied to model this pattern. The following code was used to access the data, rename the object with a very long name (TeethNitrogen) into something much shorter, apply linear regression on Moby’s data, and make the validation graphs in Fig. 2.8.

```
> library(AED); data(TeethNitrogen)
> TN <- TeethNitrogen
> M2 <- lm(X15N ~ Age, subset = (TN$Tooth == "Moby") ,
  data = TN)
> op <- par(mfrow = c(2, 2))
> plot(M2, add.smooth = FALSE)
> par(op)
```

Figure 2.8 is the typical graphical output produced by the `plot` command in R. Based on the QQ-plot in panel B, the residuals look normally distributed (if the points are in a line, normality can be assumed). Panel D identifies potential and

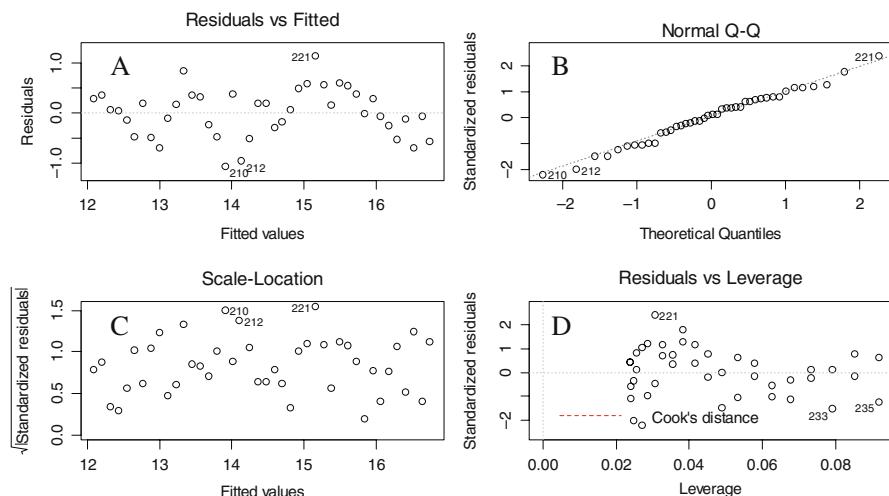


Fig. 2.8 Model validation graphs obtained by applying a linear regression model on the teeth data from Moby. Panel A and C show residuals versus fitted values; note the clear pattern! Panel B is a QQ-plot for normality, and Panel D shows the standardised residuals versus leverage and the Cook statistic is superimposed as contour plots. In this case, the Cook values are small and cannot be clearly seen

influential observations. It is a scatterplot of leverage against residuals. Leverage measures whether any observation has extreme values of the explanatory variables. If there is only one explanatory variable, then a Cleveland dotplot or boxplot will identify such points. However, an observation may have a combination of values of explanatory variables that make it unique in terms of ‘environmental’ conditions. None of the data exploration methods mentioned so far will detect this. If such a point has a ‘large’ influence on the linear regression model, we may decide to remove it. And this is measured by the Cook distance (a leave-one-out measure of influence), which is superimposed with contour lines in panel D. We will return to the Cook distance later (Appendix A) as the default output of R is not the best way to present the Cook distance. In this case, there are no observations with a Cook distance larger than 1, which is the threshold value upon one should take further action (Fox, 2002). Summarising, leverage indicates how different an individual observation is compared to the other observations in terms of the values of the explanatory variables; the Cook distance tells you how influential an observation is on the estimated parameters.

Figure 2.8A shows residuals versus fitted values. Violation of homogeneity can be detected if this panel shows any pattern in the spread of the residuals. Panel C is based on the same theme. However, in panel C, the residuals are square-root transformed (after taking the absolute values) and weighted by the leverage. Both panels A and C can be used to assess homogeneity. The spread seems to be the same everywhere; however, panel A shows a clear problem: violation of independence. There are in fact two violations to deal with here. The first one can be seen better from Fig. 2.9. It shows the observed values plotted against age with a fitted linear regression curve added. There are groups of sequential residuals that are above and below the regression line.

The graph was obtained by

```
> N.Moby <- TN$X15N[TN$Tooth == "Moby"]
> Age.Moby <- TN$Age[TN$Tooth == "Moby"]
```

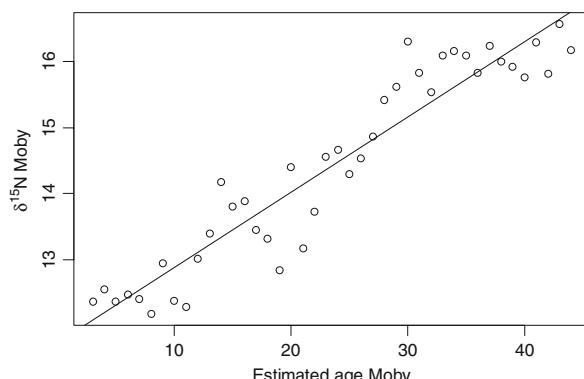


Fig. 2.9 Observed nitrogen isotope ratios plotted versus age for Moby the whale. The line is obtained by linear regression

```
> plot(y = N.Moby, x = Age.Moby,
      xlab = "Estimated age Moby",
      ylab = expression(paste(delta^{15}, "N Moby")))
> abline(M2)
```

To keep the code for the `plot` command simple, we defined the variables `N.Moby` and `Age.Moby`. The `abline` command draws the fitted regression curve. Applying an additive model (Chapter 3) or adding more covariates may solve the misfit. The other form of dependence is due to the nature of these data; high nitrogen isotope ratios at a certain age may be due to high nitrogen values at younger ages. To allow for this type of dependence, some sort of auto-correlation structure on the data is needed, and this is discussed in Chapters 5, 6, and 7.

The relevant numerical output obtained by the `summary(M2)` command is given by

	Estimate	Std. Error	t-value	p-value
(Intercept)	11.748	0.163	71.83	<0.001
Age.Moby	0.113	0.006	18.40	<0.001

Residual standard error: 0.485 on 40 degrees of freedom
 Multiple R-Squared: 0.894, Adjusted R-squared: 0.891
 F-statistic: 338.4 on 1 and 40 DF, p-value: < 0.001

The output shows the estimated intercept and slope (plus standard errors, *t*-values and *p*-values). We also get information on R^2 and the adjusted R^2 (the latter one can be used to select the best model if there are any non-significant terms in the model), the square root of the variance (residual standard error), and the *F*-statistic (which is testing the null hypothesis whether all slopes, one in this case, are equal to zero). The estimated model is given by

$$y_i = 11.748 + 0.113 \times age_i$$

The estimated slope and intercept are significantly different from 0 at the 5% level. The model explains 89% of the variation; the estimator for σ is equal to $s = 0.486$. But the problem is that we still have to reject this model because there is a clear violation of independence. Solutions will be given in Chapters 6 and 7.

2.3.8 Example 3; *Nereis*

In the third example, we present the results of a linear regression model applied on the *Nereis* data, presented earlier in this chapter. The concentration is modelled as a function of nutrient, biomass, and their interaction. This can also be called a 2-way ANOVA with interaction. The following R code accesses the data, defines

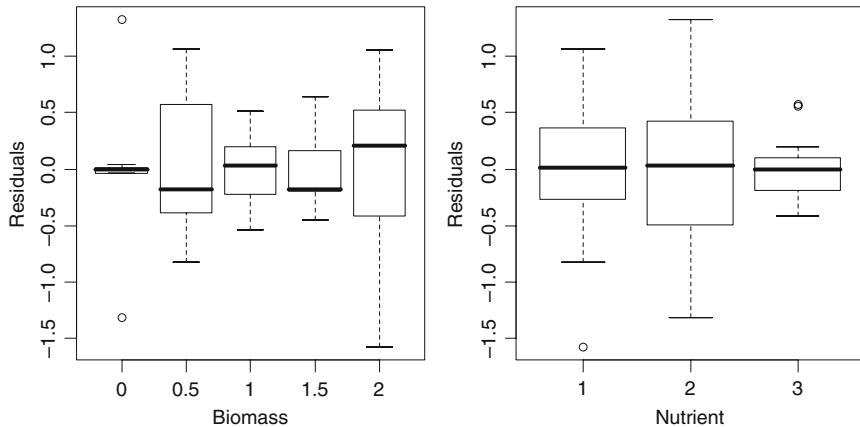


Fig. 2.10 Model validation graphs for the Nereis data showing heterogeneity. Residuals are plotted versus biomass and nutrient

the explanatory variables biomass and nutrient as factors, applies linear regression, and plots the validation graphs in Fig. 2.10. Note that homogeneity is violated!

```
> library(AED); data(Nereis)
> Nereis$fbiomass <- factor(Nereis$biomass)
> Nereis$fnutrient <- factor(Nereis$nutrient)
> M3 <- lm(concentration ~ fbiomass * fnutrient,
            data = Nereis)
> drop1(M3, test = "F")
> op <- par(mfrow = c(1, 2))
> plot(resid(M3) ~ Nereis$fbiomass, xlab = "Biomass",
       ylab = "Residuals")
> plot(resid(M3) ~ Nereis$fnutrient,
       xlab = "Nutrient", ylab = "Residuals")
> par(op)
```

The numerical output obtained by the `drop1` command is printed below and shows that the biomass-nutrient interaction term is significant at the 5% level.

```
Single term deletions
Model: concentration ~ fbiomass * fnutrient
                    Df Sum of Sq      RSS      AIC F value    Pr(F)
<none>                      13.630 -23.746
fbiomass:fnutrient     8      11.553    25.183 -12.121  3.1785 0.0099
```

However, the boxplot of (i) residuals versus nutrient and (ii) residuals versus biomass in Fig. 2.10 shows a clear violation of homogeneity. Applying a

transformation on concentration may solve this problem. The disadvantage of a transformation is that we are changing the type of relationship between response and explanatory variables. So, again we need to reject the linear regression model for these data.

2.3.9 Example 4; Pelagic Bioluminescence

In Gillibrand et al. (2007), pelagic bioluminescence along a depth gradient in the northeast Atlantic Ocean is analysed. Figure 2.11 shows an `xyplot` from the lattice package. Each panel represents a station. The underlying questions are (i) how to model the bioluminescent–depth relationship and (ii) how to deal with the data of difference stations. The following code was used to read the data and make the lattice panel.

```
> library(AED); data(ISIT)
> ISIT$fStation <- factor(ISIT$Station)
> library(lattice)
> xyplot(Sources ~ SampleDepth | fStation, data = ISIT,
  xlab = "Sample Depth", ylab = "Sources",
  strip = function(bg = 'white', ...),
  strip.default(bg = 'white', ...),
  panel = function(x, y) {
    panel.grid(h = -1, v = 2)
    I1 <- order(x)
    llines(x[I1], y[I1], col = 1)})
```

You can see this code is slightly more complicated than used for Fig. 2.4. In this code, we used a panel function that automatically splits up the data by station. When R enters this panel function, the *x* and the *y* variables are the data for one particular station. We then have a range of options in the way we can display this *x* and *y* data. First, we add a grid using the `panel.grid` command. If you don't like the grid, just remove this command. The `I1 <- order (x)` determines the order of age as we did not sort the data before importing into R. Finally, we added lines between points with sequential ages. Omitting the `order` command and removing the `[I1]` in the `llines` function produces a spaghetti plot.

There is no point in applying a linear regression model with Sources as the response variable and Depth and Station as explanatory variables (plus an interaction between them) because the relationships are not linear and the variation per station differs. Perhaps it is better to consider station as a random effect (Chapter 5). Another problem is that depth can be seen as a spatial gradient. Hence, there may be spatial correlation along the depth gradient. In Chapter 5, we discuss random effect models, and in Chapter 7 spatial correlation for smoothing models. A full analysis of this data set is presented in Chapter 17.

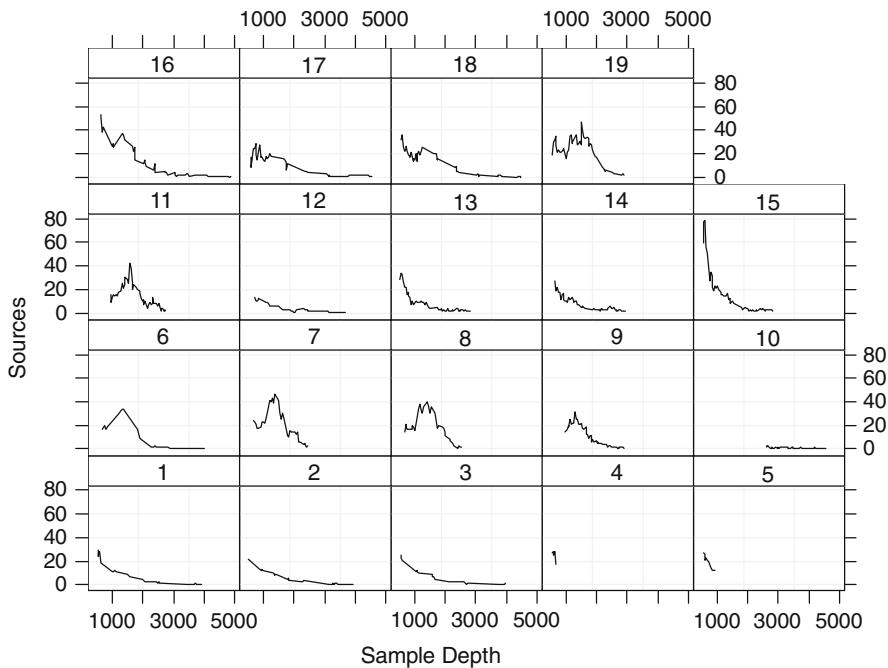


Fig. 2.11 Pelagic bioluminescence (labelled as Sources) along a depth gradient in the northeast Atlantic Ocean. Each panel represents a station

2.4 Where to Go from Here

The data exploration should filter out any typing mistakes (typos), identify possible outliers and the need for a data transformation, and provide some ideas about the follow up analyses. As for typos, these should obviously be corrected before continuing with any analysis, but do not apply a transformation on the response variable yet unless there are strong reasons to do so. Some of the methods discussed in later chapters may be able to deal with (groups) of extreme observations or heterogeneity. Many books will tell you to routinely apply a data transformation to linearise the relationship. Well, if you are particular fond on linear regression then yes, but (generalised) additive (mixed) modelling is especially designed to model non-linear relationships. Even heterogeneity, as for example encountered in Fig. 2.1B can be dealt with (as will be explained in Chapter 4); so you do not need to apply a transformation to stabilise the mean-variance relationship, provided you are willing to read the rest of this book. The only thing we cannot solve with any of the techniques discussed in later chapters is observations with extreme explanatory variables. If this happens for your data, then a transformation on the explanatory variable(s) could well be justified at this stage.

The original aim of this chapter was to simply illustrate the linear regression model for an ecological data set and discuss the numerical and graphical output. However, in preparing this book, we had access to about 15 data sets, and in Zuur et al. (2007), we had access to a further 20 data sets. In none of these real data sets could we find a non-trivial example for a linear regression model for which all assumptions held. This clearly identifies the limitation of linear regression for analysing ecological data. Hence, our choice of the title of this chapter.

So, what can we do? The problem of heterogeneity can be solved by either allowing for different variances in the linear regression model (using generalised least squares estimation) or using a different distribution and model structure (Poisson, negative binomial and Gamma distributions in GLM); the dependence problem requires the use of models that allow for more flexibility than regression (e.g. smoothing methods) and a model for the error structure (e.g. temporal, spatial correlation, or along another gradient like age or depth). We will also need to consider nested data and random effects. Taken together, all these techniques lead to mixed

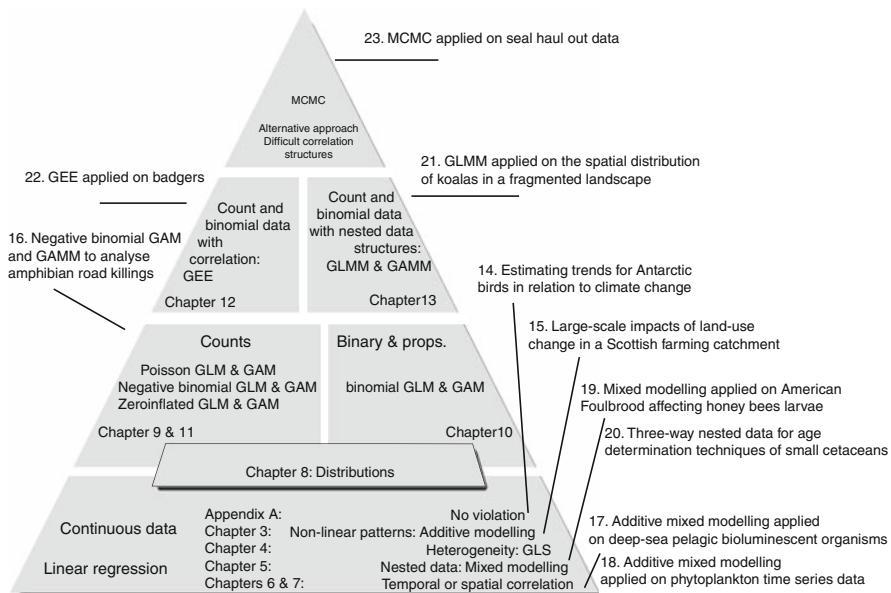


Fig. 2.12 Overview of all the chapters in this book. Linear regression is discussed in Appendix A. Additive modeling, generalised least squares (GLS), and mixed modelling techniques are presented in Chapters 4, 5, 6, and 7. Chapter 8 contains an explanation of the Poisson, negative binomial, Bernoulli, binomial, and zero-truncated distributions. GLM and GAM models are discussed in Chapters 9, 10, and 11, and finally, Chapters 12 and 13 contain GEE, GLMM, and GAMM. Associated case studies are printed outside the triangle. Chapter 23 contains an application of Markov Chain Monte Carlo (MCMC), which can be used as an alternative estimation technique or if the correlation structure is more complicated than the R functions for mixed modeling, GLMM and GAMM can cope with

modelling approach, and if combined with GLM and GAM, to generalised linear mixed modelling (GLMM) and generalised additive mixed modelling (GAMM).

Chapter 4 shows how we can deal with heterogeneity in linear regression and smoothing models, random effects for nested data are introduced in Chapter 5, and temporal and spatial correlation structures are discussed in Chapters 6 and 7. In Chapter 8, we introduce different distributions for count data, binary data, proportional data, and zero inflated count data. These are then used in Chapters 9, 10, and 11. Finally, Chapters 12 and 13 discuss how we can incorporate correlation structures and random effects in models for count data, binary data, and proportional data. See Fig. 2.12 for a schematic overview.

Before reading on, we strongly advise to read Appendix A as it provides a more detailed discussion on linear regression. It is essential that you are familiar with all steps discussed in this appendix.

Chapter 3

Things Are Not Always Linear; Additive Modelling

3.1 Introduction

In the previous chapter, we looked at linear regression, and although the word linear implies modelling only linear relationships, this is not necessarily the case. A model of the form $Y_i = \alpha + \beta_1 \times X_i + \beta_2 \times X_i^2 + \varepsilon_i$ is a linear regression model, but the relationship between Y_i and X_i is modelled using a second-order polynomial function. The same holds if an interaction term is used. For example, in Chapter 2, we modelled the biomass of wedge clams as a function of length, month and the interaction between length and month. But a scatterplot between biomass and length may not necessarily show a linear pattern.

The word ‘linear’ in linear regression basically means linear in the parameters. Hence, the following models are all linear regression models.

- $Y_i = \alpha + \beta_1 \times X_i + \beta_2 \times X_i^2 + \varepsilon_i$
- $Y_i = \alpha + \beta_1 \times \log(X_i) + \varepsilon_i$
- $Y_i = \alpha + \beta_1 \times (X_i \times W_i) + \varepsilon_i$
- $Y_i = \alpha + \beta_1 \times \exp(X_i) + \varepsilon_i$
- $Y_i = \alpha + \beta_1 \times \sin(X_i) + \varepsilon_i$

In all these models, we can define a new explanatory variable Z_i such that we have a model of the form $Y_i = \alpha + \beta_1 \times Z_i + \varepsilon_i$. However, a model of the form

$$Y_i = \alpha + \beta_1 \times X_{1i} \times e^{\beta_2 \times X_{2i} + \beta_3 \times X_{3i}} + \varepsilon_i$$

is not linear in the parameters. In Chapter 2, we also discussed assessing whether the linear regression model is suitable for your data by plotting the residuals against fitted values, and residuals against each explanatory variable. If in the biomass wedge clam example, the residuals are plotted against length, and there are clear patterns, then you have a serious problem. Options to fix this problem are as follows:

- Extend the model with interactions terms.
- Extend the model with a non-linear length effect (e.g. use length and length to the power of two as explanatory variables).

- Add more explanatory variables.
- Transform the data to linearise the relationships. You can either transform the response variables or the explanatory variables. See, for example, Chapter 4 in Zuur et al. (2007) for guidance on this. An interesting discussion with arguments against transformations can be found in Keele (pg. 6–7, 2008). One of the arguments is that a transformation affects the entire $Y - X$ relationship, whereas maybe the relationship is partly linear and also partly non-linear along the X gradient.

Now suppose you have already added all possible explanatory variables, and interactions, but you still see patterns in the graph of residuals against individual explanatory variables, and you do not want to transform the variables. Then you need to move on from the linear regression model, and one alternative is to use smoothing models, the subject of this chapter. These models allow for non-linear relationships between the response variable and multiple explanatory variables and are also called additive models. They are part of the family of generalised additive models (GAM) that we discuss in Chapters 8, 9, and 10, and an additive model can also be referred to as a GAM with a Gaussian distribution (and an identity link, but this is something we will explain in Chapter 9).

References on additive modelling, or more generally on GAM, in R, are Chambers and Hastie (1992), Bowman and Azzalini (1997), Venables and Ripley (2002), Faraway (2006), and Keele (2008). More advanced books are Hastie and Tibshirani (1990), Schimek (2000), Ruppert et al. (2003) or Wood (2006). Other books on GAM, but without R code are Fox (2000), Quinn and Keough (2002), and Zuur et al. (2007), among others. The ‘must cite’ books are Hastie and Tibshirani (1990) and Wood (2006). The Keele (2008) is an ‘easy to read’ book, though it aims at the social scientist. The Zuur et al. (2007) book has various case studies, showing applications of GAMs on various types of data, including time series. Its supporting website at www.highstat.com contains the required R code.

3.2 Additive Modelling

The linear regression model using only one explanatory variable is given by

$$Y_i = \alpha + \beta \times X_i + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma^2) \quad (3.1)$$

The relationship between Y_i and X_i is summarised by the parameter β . In additive modelling, we use a smoothing function to link Y_i and X_i . Figure 3.1A shows a scatterplot of pelagic bioluminescence along a depth gradient in the northeast Atlantic Ocean for a particular station (number 16). These data were introduced in Chapter 2 and the graph shows a non-linear pattern. One approach to model these data is to try to linearise the relationship between the two variables by applying a square root, or logarithmic, transformation, and then applying a linear regression. Although this approach may work for the data from this particular station, it will not work for data

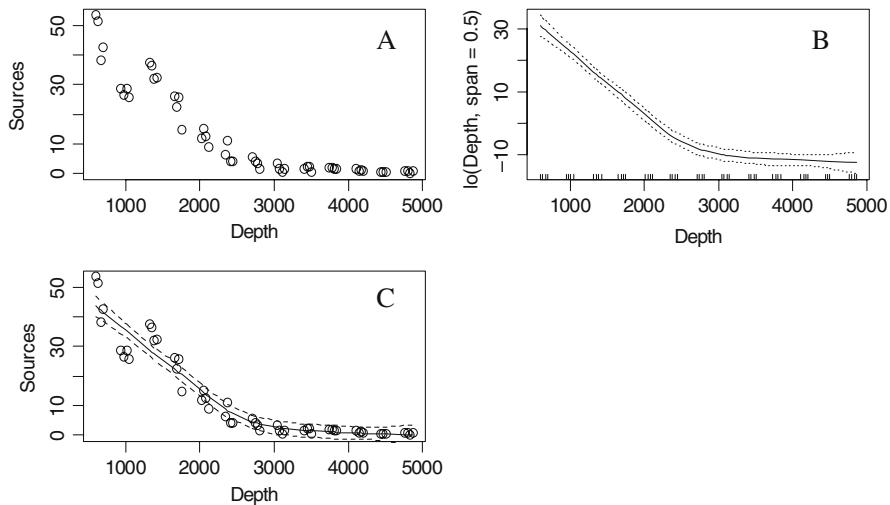


Fig. 3.1 **A:** Scatterplot of pelagic bioluminescence versus depth gradient for cruise 16. **B:** Estimated LOESS curve and pointwise 95% confidence bands obtained by the `gam` function in the `gam` package. **C:** Fitted values obtained by the LOESS smoother and observed data

of all stations in this data set, especially those showing a non-monotonic decreasing source-depth relationship (e.g. stations 6–10, see Fig. 2.11). We therefore need something that can cope with non-linear patterns, for example, the GAM.

3.2.1 GAM in `gam` and GAM in `mgcv`

The additive model fits a smoothing curve through the data. There are as many smoothing techniques as there are roads to Rome. In R, there are two main packages for GAM: The `gam` package written by Hastie and Tibshirani and the `mgcv` package produced by Wood. Each package has its own charms. Readers familiar with the classical textbook from Hastie and Tibshirani (1990) may prefer the `gam` package as it follows the theory described in the book. Estimation of smoothers is done using a method called the back-fitting algorithm (a simple but robust way to estimate one smoother at a time). The GAM in `mgcv` uses splines and these require slightly more mathematical understanding than the methods in `gam`.

The main advantage of GAM in the `gam` package is its simplicity; it is easy to understand and explain. The main advantage of GAM in `mgcv` is that it can do cross-validation and allows for generalised additive mixed modelling (GAMM) including spatial and temporal correlations as well as nested data and various heterogeneity patterns. GAMM will be discussed later in this book. Cross-validation is a process that automatically determines the optimal amount of smoothing.

3.2.2 GAM in gam with LOESS

Instead of going straight into the mathematical background of GAM, we show how to run it and explain what it does with help of an example. The following R code was used to generate Fig. 3.1A:

```
> library(AED); data(ISIT)
> op <- par(mfrow = c(2, 2), mar = c(5, 4, 1, 2))
> Sources16 <- ISIT$Sources[ISIT$Station == 16]
> Depth16 <- ISIT$SampleDepth[ISIT$Station == 16]
> plot(Depth16, Sources16, type = "p")
```

We access the variables directly from the ISIT object by using the \$ symbol. The additive model applied on the source (response variable Y_i) and depth (explanatory X_i) variable is

$$Y_i = \alpha + f(X_i) + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma^2) \quad (3.2)$$

Note that the only difference between models (3.1) and (3.2) is the replacement of $\beta \times X_i$ by the smoothing curve $f(X_i)$. However, it is fundamentally important to understand the difference. The linear regression model gives us a formula and the relationship between Y_i and X_i is quantified by an estimated regression parameter plus confidence intervals. In a GAM, we do not have such an equation. We have a smoother, and the only thing we can do with it, is plot it.¹ This does not mean that we cannot predict from this model; we can, but not with a simple equation. The smoother is the curve in Fig. 3.1B. The dotted lines are 95% point-wise confidence bands. For the moment, it is sufficient to know that the curve is a LOESS smoother obtained with the gam package. We will explain later what the abbreviation LOESS means. The R code is given below:

```
> library(gam)
> M1 <- gam(Sources16 ~ lo(Depth16, span = 0.5))
> plot(M1, se = TRUE) #Fig. 3.1B
```

The gam package does not come with the base installation of R, and you will need to download and install it. The `plot` command in the code above gives the LOESS smoother in Fig. 3.1B. To obtain a graph with fitted and observed values (Fig. 3.1C), use the following code:

¹This is not entirely true as we will see later. The smoothers used in this chapter consist of a series of local regression-type models, which do allow for prediction. We just don't get one overall equation.

```
> M2 <- predict(M2, se = TRUE)
> plot(Depth16, Sources16, type = "p")
> I1 <- order(Depth16)
> lines(Depth16[I1], M2$fit[I1], lty = 1)
> lines(Depth16[I1], M2$fit[I1] + 2 * M2$se[I1], lty = 2)
> lines(Depth16[I1], M2$fit[I1] - 2 * M2$se[I1], lty = 2)
> par(op)
```

The `predict` command creates an object containing two variables: fitted values and standard errors. These can be accessed by typing `M2$fit` and `M2$se`. The `order` command avoids a spaghetti plot as the data were not sorted by depth. The `par(op)` sets the graphical parameters back to default values.

3.2.2.1 LOESS Smoothing

LOESS smoothing is discussed in several textbooks, e.g. Cleveland (1993), Hastie and Tibshirani (1990), Fox (2000), Zuur et al. (2007), and Keele (2008) among many others. Here, we only give a short, conceptual explanation, and we strongly advise that you consult Hastie and Tibshirani (1990) if you decide to work with the GAM from the `gam` package. The principle of LOESS smoothing is illustrated in Fig. 3.2.

Suppose we have a target value of depth = 1500 m. Choose a window around this target value of a certain size, let us say 500 m on both sides of the target value so the window goes from 1000 to 2000 m. All the observations inside this window are plotted as black dots, the ones outside the window as open circles. The aim is now to obtain a value for the number of sources at the target value (denoted by the triangle). Multiple options exist; we can take the mean value of the number of sources of all

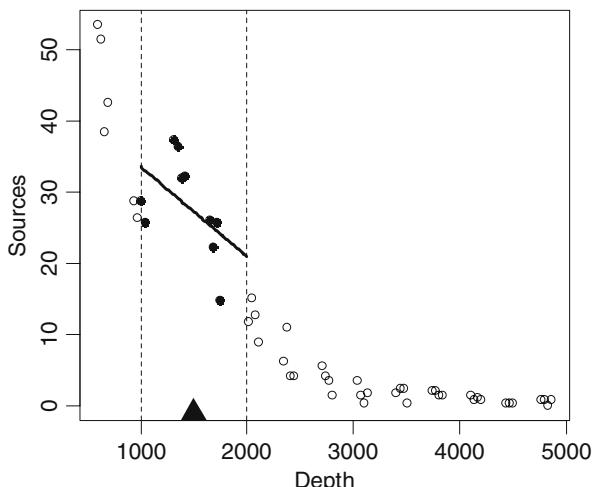


Fig. 3.2 Illustration of LOESS smoothing. We want to predict a value of sources at the target value of 1500 m depth (denoted by the *triangle*). A window around this target value is chosen (denoted by the *dotted lines*) and all points in this window (the *black dots*) are used in the local linear regression analysis

the points inside the window or the median. Alternatively, we can apply a linear regression using only the black points and predict the number of sources at depth = 1500 m from the resulting equation. Or we can apply a weighted linear regression, where the weights are determined by the distance (along the horizontal direction) of the observations from the target value. If we use the linear regression option (also called local linear regression), we get the thick linear regression line in Fig. 3.2. Using the underlying equation for this line, we can easily predict the sources at the target depth of 1500 m, which is 27.2. Repeating this whole process for a sequence of depth values (e.g. from the smallest to the largest depth value, with a total of 100 intermediate equidistant values) and each time storing the predicted value is called LOESS smoothing. In a statistical context, the abbreviation LOESS stands for local regression smoother, which now makes sense. If weights are used in the local regression, we talk about local weighted linear regression abbreviated as LOWESS. Instead of a linear regression, it is also possible to use polynomial models of order p ; a typical value for p is two. You may see the name local polynomial regression in the literature for this.

The R function to use is `loess` (or `lo`). By default, it fits a local polynomial model of order two, but you if you use as option `degree = 1`, it fits a local linear regression. Confusingly, the function `loess` is actually doing local weighted linear regression (see its help file obtained by typing `?loess`). R also has a routine that is called `lowess`, which is an older version of `loess`. The function `loess` has better support for missing values and has different default settings. Keele (2008) shows that the differences between LOESS and LOWESS are marginal.

The process outlined above produces a similar curve as the smoothing curve in Fig. 3.1. The only difference between the smoothing curve in Fig. 3.1 and the one produced by the approach above is the size of the window (resulting in a less smooth curve in Fig. 3.2).

There are two problems associated with moving a window along the depth gradient: the size of the window (also called span width) and what happens at the edges. There is not much you can do about the edges, except to be careful with the interpretation of the smoother at the edges. The size of the window is a major headache. Instead of specifying a specific size (e.g. 500 m), it is expressed as the percentage of the data inside the window. The command in R

```
> lo(Depth, span = 0.5)
```

means that the size of the window is chosen so 50% of the data is inside each window. It is the default value. If this value is chosen as 1, we obtain a nearly straight line; setting it to a very small value gives a smoother that joins every data point. So, is there any point in changing the span width in Fig. 3.1? Note that the sources at the depth range around 1000 m show groups of points being under- and over-fitted. So, perhaps we should decrease the span a little bit, and try 0.4 or 0.3. But how do you assess the optimal span width? One option is trial and error. Figure 3.3 shows the smoother for the ISIT data of station 16. It seems that the smoother obtained with `span = 0.1` over-fits the data (this is called under-smoothing). On the other extreme,

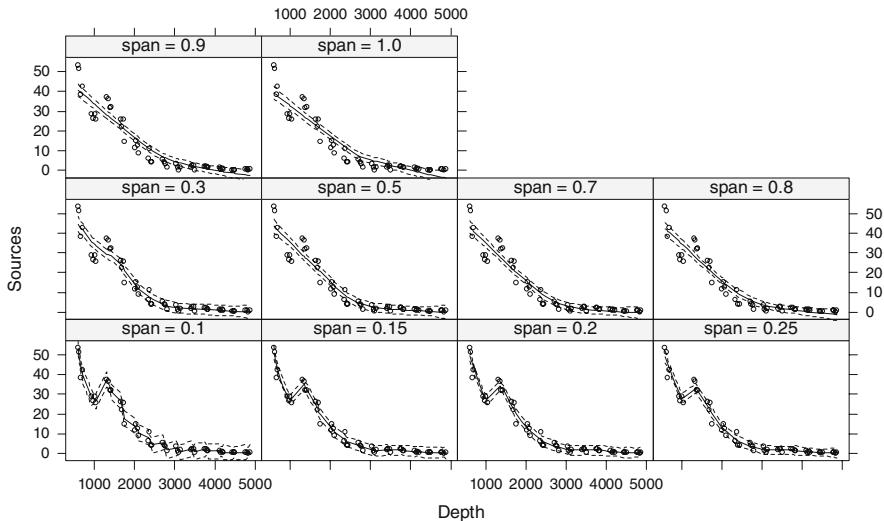


Fig. 3.3 LOESS smoothers for different span values. The solid line is the LOESS smoother and the dotted lines are 95% confidence bands. A visual inspection of the fitted lines and observed values indicates that a span of 0.2 seems to be optimal. The R code to make this graph is about two pages and is therefore presented on the book website

the model with $\text{span} = 1$ shows a clear misfit (this is called over-smoothing). It seems that the smoother obtained with $\text{span} = 0.2$ follows the pattern of the data reasonably well.

Finding the optimal span width is also a matter of bias – variance tradeoff. The smoother obtained with $\text{span} = 0.1$ has only a few data points in each window for the local regression; hence, the uncertainty around the predicted value at the target value will be large (and as a result the confidence bands around the smoothers are large). But the fit will be good! On the other hand, a smoother with a span of 0.9 or 1 will have small variances (lots of data are used to estimate the Y value at each target point), but the fit is not good.

Another way to find the optimal amount of smoothing is inspecting residual graphs. If there is still a pattern in the residuals, it may be an option to increase the span width. Yet, another option is to use the Akaike Information Criterion (AIC); see also Appendix A. In this case, the model with a span of 0.15 has the lowest AIC.

The first two options, visual inspection of smoothers and residuals, sound subjective, though with common sense they work fine in practice. It is also possible to apply automatic selection of the amount of smoothing, but instead of presenting this approach for the LOESS smoother, we will discuss this using the GAM in the mgcv package. Automatic estimation of the amount of smoothing has not been implemented (at the time of writing this book) in the GAM function in the gam package. This provides a good motivation for using the mgcv package!

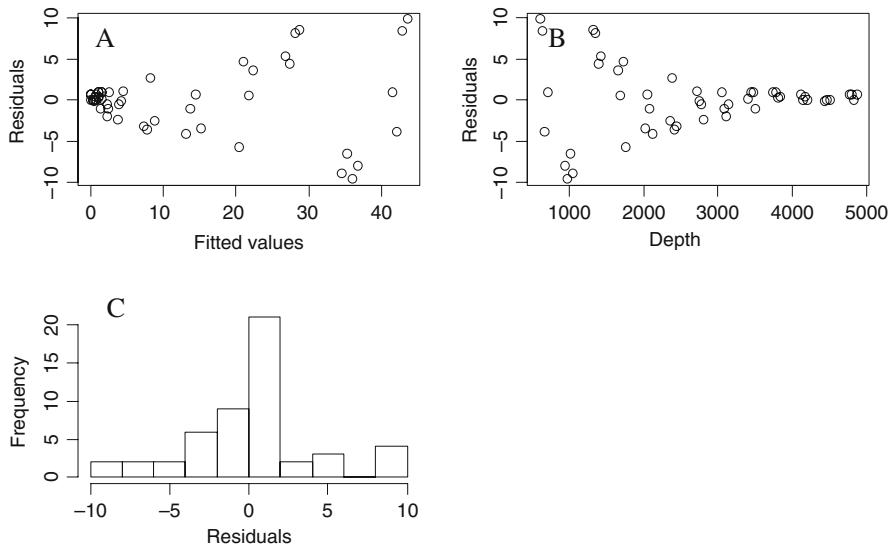


Fig. 3.4 **A:** Residuals versus fitted values. **B:** Residuals versus Depth. **C:** Histogram of residuals. The panels show that there are residual patterns, heterogeneity and non-normality

Hastie and Tibshirani (1990) show how the LOESS smoother can be written in simple matrix algebra, mimicking the matrix algebra of linear regression, and then give a justification for using hypothesis testing procedures that provide F -statistics and p -values. But just as in linear regression, we need to apply a whole series of model validation tests: homogeneity, normality, fixed X, independence, and no residual patterns. The same graphs can (and should) be used; see Fig. 3.4, where it can be seen from panels A and B that there are patterns in the residuals and the residual spread is smaller for deeper depths. The patterns may disappear if we use a span of 0.4 or 0.2. But instead of going this route, we apply GAM with the routines from the mgcv package in the next section (and remaining part of this book) as it is considerably more flexible.

3.2.3 GAM in mgcv with Cubic Regression Splines

Before explaining anything about GAM from mgcv, we show how to run it and show that it produces very similar results to results from the gam package. You can then decide whether it is worth your while digging a bit deeper into the underlying mathematics of splines, which are slightly more complicated than the LOESS smoother! Figure 3.5 contains the same graphs as in Fig. 3.1, except that we used the GAM from mgcv

The following R code was used to generate Fig. 3.5. As both packages use the same function `gam`, it may be wise to restart R; alternatively, type `detach("package:gam")`. This avoids R choosing the wrong `gam` function.

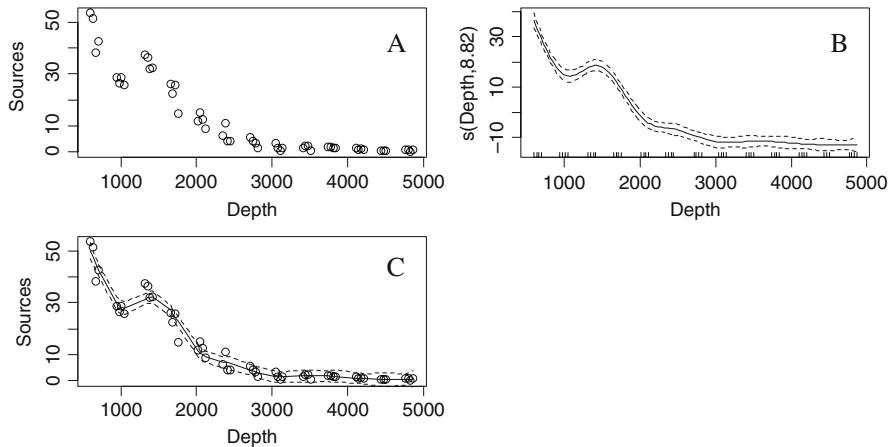


Fig. 3.5 **A:** Scatterplot of pelagic bioluminescence versus depth gradient for cruise 16. **B:** Estimated smoothing curve (cubic regression spline) and point-wise 95% confidence bands. **C:** Fitted values and observed data

```
> library(AED); data(ISIT)
> library(mgcv)
> op <- par(mfrow = c(2, 2), mar = c(5, 4, 1, 2))
> Sources16 <- ISIT$Sources[ISIT$Station == 16]
> Depth16 <- ISIT$SampleDepth[ISIT$Station == 16]
> plot(Depth16, Sources16, type = "p")
> M3 <- gam(Sources16 ~ s(Depth16, fx = FALSE, k=-1,
+                         bs = "cr"))
> plot(M3, se = TRUE)
> M3pred <- predict(M3, se = TRUE, type = "response")
> plot(Depth16, Sources16, type = "p")
> I1 <- order(Depth16)
> lines(Depth16[I1], M3pred$fit[I1], lty=1)
> lines(Depth16[I1], M3pred$fit[I1]+2*M3pred$se[I1], lty=2)
> lines(Depth16[I1], M3pred$fit[I1]-2*M3pred$se[I1], lty=2)
```

This is nearly the same code as before, except it uses the GAM from the mgcv package. This package is part of the base distribution; so you do not have to download anything, but the code to run the `gam` function is slightly different. The format is now similar to that of linear regression (Chapter 2). The expression $Y \sim s(X)$ means that a smoothing function is used for the explanatory variable X . The `fx = FALSE`, `k = -1` bit means that the amount of smoothing is not fixed to a preset value; hence, cross-validation is used to estimate the optimal amount of smoothing. We will explain later what cross-validation is. The `bs = "cr"` code tells R that a cubic regression spline should be used. A cubic regression spline is slightly more

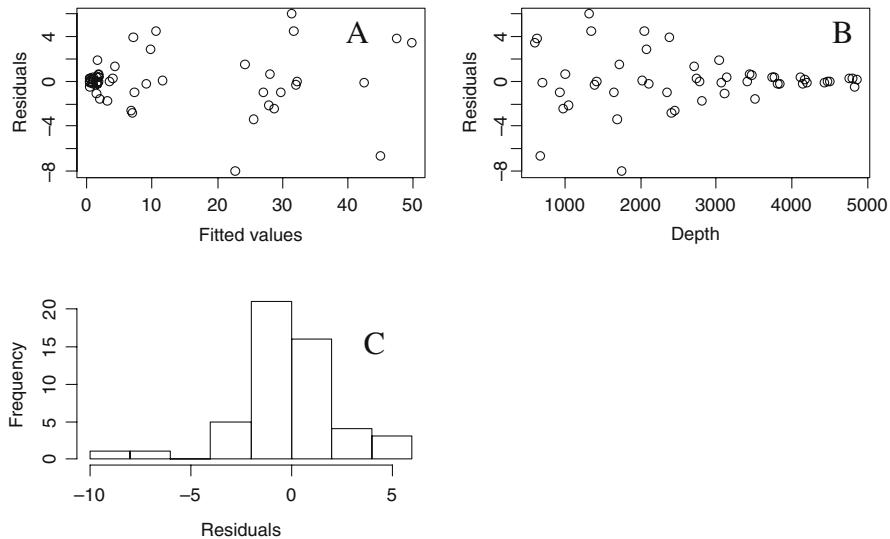


Fig. 3.6 A: Residuals versus fitted values using the `gam` function in `mgcv`. B: Residuals versus Depth. C: Histogram of residuals. The panels show that there are no residual patterns (independence), but there is heterogeneity (differences in spread)

complicated to understand than a LOESS smoother and a technical explanation is given in the next section. For now, it suffices to know that for the cubic regression spline method, the X gradient (depth) is divided into a certain number of intervals. In each interval, a cubic polynomial (this is a model of the form $Y_i = \alpha + \beta_1 \times X_i + \beta_2 \times X_i^2 + \beta_3 \times X_i^3$) is fitted, and the fitted values per segment are then glued together to form the smoothing curve. The points where the intervals connect are called knots. To obtain a smooth connection at the knots, certain conditions are imposed. These conditions involve first- and second-order derivates and require high-school mathematics to understand. The `gam` function in `mgcv` allows for other types of splines, but for most data sets, the visual differences are small.

This method, again, gives F -statistics and p -values, and as before, we need to check the assumptions on data characteristics. The validation plots for this example are presented in Fig. 3.6 and show that the residual patterns have disappeared (except for the heterogeneity). The reason for this is that `mgcv` has produced a less smooth curve due to the cross-validation. The R code that was used to create Fig. 3.6 is identical as that for Fig. 3.4 and is not shown again.

3.3 Technical Details of GAM in `mgcv`*

This section gives a more technical discussion on regression splines (hence the * in the section title) and some other aspects of GAM in the `mgcv` package. It is heavily based on Chapters 3 and 4 in Wood (2006), but at a considerably lower mathematical

level. We advise readers to have a look at his chapters after reading this section, and to avoid confusion, we have used the same mathematical notation as Wood (2006), where possible.

The family of splines is rather large, e.g. cubic splines, B-splines, natural splines, thin-splines, and smoothing splines. Here, we discuss a few of them.

We mentioned in the previous section that for a cubic polynomial, the x -axis is divided into various segments, and on each segment, a cubic regression spline is fitted. The word cubic refers to 3. In a more general context, we can consider any order for the polynomial. Recall that the smoother is given by the function $f(X_i)$ in

$$Y_i = \alpha + f(X_i) + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma^2) \quad (3.3)$$

We now give an expression for the function $f(X_i)$ so that it can be written as a linear regression model. This is done by using a ‘basis’ for $f(X_i)$. This means that $f(X_i)$ is built up in basic components, called the basis functions $b_j(X_i)$, such that:

$$f(X_i) = \sum_{j=1}^p \beta_j \times b_j(X_i) \quad (3.4)$$

This may look magic to many readers, but the principle is fairly simple. Suppose that $p = 4$. This gives

$$f(X_i) = \beta_1 \times b_1(X_i) + \beta_2 \times b_2(X_i) + \beta_3 \times b_3(X_i) + \beta_4 \times b_4(X_i) \quad (3.5)$$

Ok, this may still look magic, but let us assume for the moment that $b_1(X_i) = 1$, $b_2(X_i) = X_i$, $b_3(X_i) = X_i^2$, and $b_4(X_i) = X_i^3$. This results in

$$f(X_i) = \beta_1 \times \beta_2 \times X_i + \beta_3 \times X_i^2 + \beta_4 \times X_i^3 \quad (3.6)$$

This is a cubic polynomial, and it can produce a wide range of possible shapes, depending on the values β_1 , β_2 , β_3 , and β_4 . Six examples of such shapes are given in Fig. 3.7. We took fixed values for X_i between 0 and 1 (with equidistance values), randomly generated some values for β_1 , β_2 , β_3 , and β_4 , and each time calculated the value of the function f using the expression in Equation (3.6).

The problem is that in reality, we do not know the values β_1 , β_2 , β_3 , and β_4 , and the shapes of the function f in Fig. 3.7 are not flexible enough to model more complicated patterns. Let us solve these two problems. Note that we do know the values of X_i , as it is a measured explanatory variable.

Suppose we divide the depth gradient into four segments (identified by the dotted lines in Fig. 3.8), and on each segment, we fit the model in Equation (3.6) using ordinary least squares (OLS). The fitted curves obtained by this approach are given in Fig. 3.8. At least, we now know the betas per segment (thanks to OLS), and because we used multiple segments, more complicated patterns than in Fig. 3.7 can be fitted. So, these two problems are solved.

Fig. 3.7 Examples of the function f for six different sets of values for β_1 , β_2 , β_3 , and β_4 . The X values were chosen to have values between 0 and 1. Note the variety in possible shapes for the smoother. R code to create this graph is on the book website

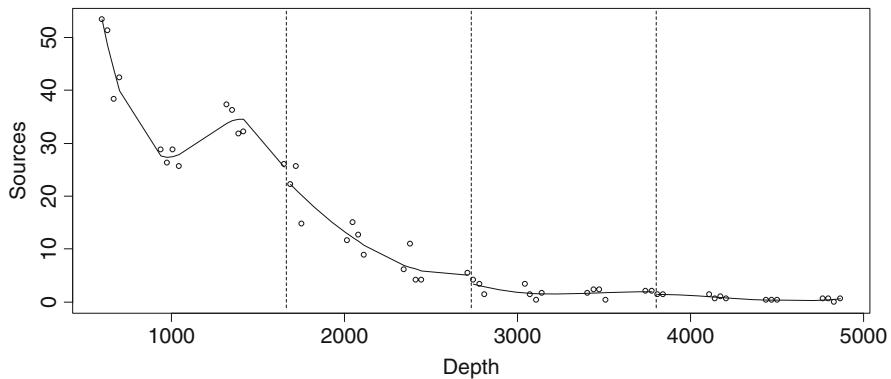
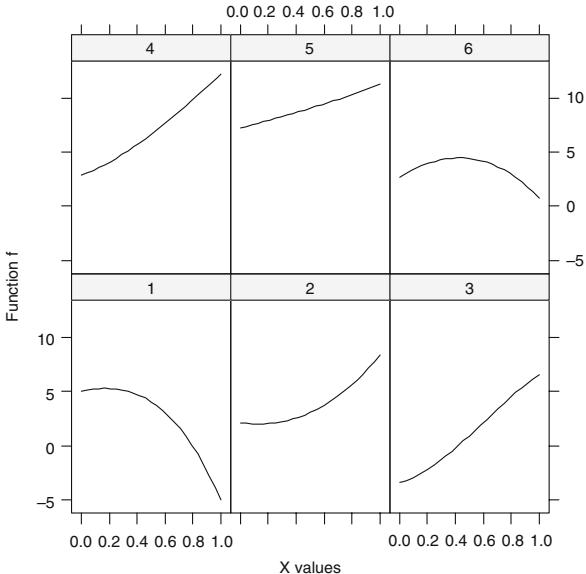


Fig. 3.8 Illustration of fitting a cubic polynomial on four segments of data using the ISIT data from station 19. We arbitrarily choose four segments along the depth gradient. The dotted lines mark these segments, and the line in each segment is the fit from the cubic polynomial model. R code to create this graph can be found on the book website

The problem with Fig. 3.8 is that, if we omit the dotted lines, any editor or referee will wonder why the line has gaps, and it seems to have discontinuities at three points (where the lines come together). A cubic regression spline ensures that the line will look smooth at the points where the individual lines (from the segments) connect. This is done with first-order and second-order derivatives. And here is where things get technical. If you are interested, you may read on or you can skip the next subsection; it is not essential for using GAMs.

3.3.1 A (Little) Bit More Technical Information on Regression Splines

In the previous paragraph, we ended with the statement that the cubic regression spline fits a third-order polynomial on segments of data, and where it connects the resulting fitted lines, it ensures that at the connection points (also called knots), the connections are smooth. This requires some high-school mathematics. It involves first-order and second-order derivatives. It can be shown that this affects the definition of the b_j s in Equation (3.4), see Wood (2006). Before showing how the b_j s look, we need to define the position of the knots (points along the x -axis where segments are defined). In Fig. 3.7, the knots are at 1665, 2732, and 3799 m. They were obtained by determining the maximum depth (4866 m) minus the minimum depth (598 m) and dividing this by 4 to obtain the segment width (= 1067 m). Once you have the segment width, you can easily calculate the position of the knots. The first knot is at 598 + 1067, the second at 598 + 2 × 1067, and the third at 598 + 3 × 1067. Let the vector \mathbf{x}^* contain these knot positions, making $\mathbf{x}^* = (1665, 2732, 3799)$. The only catch is that the formulae presented below are defined for gradients scaled between 0 and 1. Obviously, this process also influences the values of \mathbf{x}^* , which means they become $\mathbf{x}^* = (0.25, 0.5, 0.75)$; but to understand the underlying principle, it is not relevant that the gradient is scaled or not.

We are now in a position to specify how the b_j s looks for a cubic regression spline. They become:

$$\begin{aligned} b_1(X_i) &= 1 \\ b_2(X_i) &= X_i \\ b_3(X_i) &= R(X_i, x_1^*) \\ b_4(X_i) &= R(X_i, x_2^*) \end{aligned} \tag{3.7}$$

The notation x_1^* refers to the first element of \mathbf{x}^* , x_2^* to the second element, etc. The form of $R(X_i, z)$ is rather intimidating, and it is given in Wood (2006). We decided to print it here as well, but the computer calculates it for you.

$$\begin{aligned} R(X, z) &= \frac{1}{4} \times \left(\left(z - \frac{1}{2} \right)^2 - \frac{1}{12} \right) \times \left(\left(X - \frac{1}{2} \right)^2 - \frac{1}{12} \right) - \\ &\quad \frac{1}{24} \times \left(\left(|X - z| - \frac{1}{2} \right)^4 - \frac{1}{2} \left(|X - z| - \frac{1}{2} \right)^2 + \frac{7}{240} \right) \end{aligned}$$

The computer fills in the knot values of $z = 0.25$, or 0.5 , or 0.75 and calculates the values of R for given values of the covariate X . Summarising, instead of using a basis of the form $b_1(X_i) = 1$, $b_2(X_i) = X_i$, $b_3(X_i) = X_i^2$, and $b_4(X_i) = X_i^3$, we use more complicated ones that involve $R(X_i, \mathbf{x}^*)$, and all that this does is ensure that at the knots, the smoothers are neatly connected.

Substituting Equation (3.4) into (3.3) results in the following expression for the additive model.

$$Y_i = \alpha + \sum_{j=1}^p \beta_j \times b_j(X_i) + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma^2) \quad (3.8)$$

Recall that the b_j s are known values determined by the definitions in Equation (3.7) and depend on the number and the values of the knots. This means that the expression in Equation (3.8) forms a linear regression model, and the regression parameters β_j can be estimated by ordinary least squares. The number of regression parameters is equal to the number of knots plus four (for cubic bases).

In a *natural* cubic regression spline, a linear regression model is fitted in the outer two segments. This avoids spurious behaviour of the smoother at the edges.

The next question is how many knots to use. Figure 3.9 shows the effect of the number of knots on the smoothness of the smoother. We used the data from transect 19 as it shows a more non-linear pattern than the data for transect 16. The number of knots is defined as the number of splits (indicated by a vertical line) and the two endpoints. The lower left panel shows the smoother if we use two segments (three knots), the panel next to it three segments, etc. Clearly, the more knots we use, the less smooth the curve becomes.

Just as for the LOESS smoother, we can choose the ‘optimal’ amount of knots based on a visual comparison of the smoothers. It seems that the smoother with 6 knots follows the pattern of the data reasonably well. Using more knots does not seem to change much. It is also possible to use the AIC (Eilers and Marx, 1996).

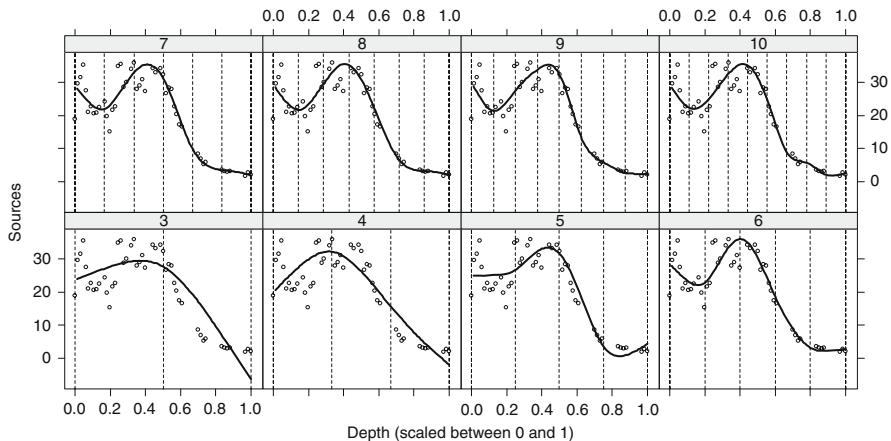


Fig. 3.9 Smoothing curves obtained by cubic regression splines using 3 knots, (lower left panel), 4 knots, 5 knots, etc., and 10 knots (upper right panel). Both end points also count as knots. The vertical lines show the position of the knots. The thick line in each panel is the smoother, and the dots the observed data for transect 19. The more knots used, the more variation in the smoother. R code to calculate the smoothing curves can be found in Wood (2006), and extended R code to present the graphs in an *xypplot* from the lattice package is on the book website

Keele (2008) gives as general recommendation to use 3 knots if there are less than 30 observations and 5 knots if there are more than 100 observations.

As to the placement of the knots, this is typically done by the software using quartiles and equidistant positions.

3.3.2 Smoothing Splines Alias Penalised Splines

Because the model in Equation (3.8) can be written as a linear regression model, it is tempting to use hypothesis testing procedures or backwards selection methods to find the optimal number of knots. After all, linear regression implies that we can find the parameters β_j by minimising the sum of squares:

$$S = \sum_{i=1}^n (Y_i - \mu_i)^2$$

where μ_i is the fitted value for observation i , and $i = 1, \dots, 789$ (there are 789 observations). A typical mathematical notation for this is

$$S = \sum_{j=1}^n (Y_j - \mu_j)^2 = \|\mathbf{Y} - \boldsymbol{\mu}\|^2 = \|\mathbf{Y} - \mathbf{X} \times \boldsymbol{\beta}\|^2$$

The notation $\| \cdot \|$ stands for the Euclidean norm, \mathbf{Y} contains all the observed data in vector format, $\boldsymbol{\beta}$ all the parameters in vector format, and \mathbf{X} all the b_j s. It looks intimidating, but it is nothing more than some matrix notation for the residual sum of squares in linear regression. This type of matrix notation for linear regression can be found in many statistical textbooks, e.g. Montgomery and Peck (1992) or Draper and Smith (1998).

However, Wood (2006) argues that it is unwise to go the route of hypothesis testing and backwards selection to obtain the optimal number of knots for a regression spline. Instead, smoothing splines (also called penalised splines) are used. These are obtained by finding the parameters $\boldsymbol{\beta}$ (and therefore the smoothers) that minimise the following criteria.

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \int f''(x)^2 dx \quad (3.9)$$

The second part of this equation is a penalty (hence the name penalised least squares and penalised smoothers) and is new. It contains λ and an integral over the second-order derivatives. Remember that the second-order derivatives of the smoothing function f , denoted by f'' , tell you how smooth a curve is. A high value of f'' means that the smoother f is highly non-linear, whereas a straight line (the perfect smooth curve) has a second-order derivative of 0. So, if λ is very large, the penalty for having a non-smooth curve is large, and the resulting smoother will be a straight line. On the other hand, if λ is small, then there is a low penalty for non-smoothness, and we are likely to end up with a considerably less smooth curve.

We started with the question of how to find the optimal amount of smoothing, and it is still not answered. Instead, we have obtained a different way of quantifying the amount of smoothing. We are now no longer looking for the optimal number of knots. Instead, our new aim is find the optimal amount of smoothing by choosing a fixed and large number of knots (and keep them fixed during the analysis) and by focusing the analysis on finding the optimal value for λ .² The word ‘optimal’ means we want to minimise the expression in Equation (3.9), where the integral measures the amount of wiggleness of the smoother, which is then controlled by the parameter λ . It is just a different approach to tackle the same problem, albeit a better one according to Wood (2006); based on mathematics, the cubic smoothing spline gives the best possible fit with the least amount of error. Minimising the expression in Equation (3.9), also called penalised least squares, is just much simpler, faster, and more robust than hypothesis testing for the number of knots.

So, how do we find the optimal value of λ ? Before addressing this question, we look at the effects of changing the value of λ on the smoother. Figure 3.10 shows the cubic regression spline for $\lambda = 10e-6$, $\lambda = 10e-5$, $\lambda = 10e-4$ up to $\lambda = 10$. As might be expected, the larger the value of λ , the more linear the smoother. Changing the number of knots has a considerably smaller effect than changing λ .

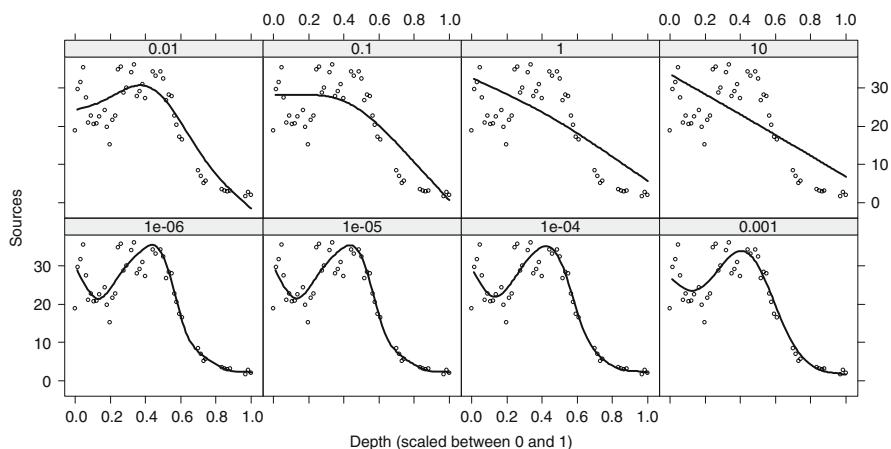


Fig. 3.10 Smoothing curve for different values of λ . The lower left panel shows the estimated smoother obtained by minimising the expression in Equation (3.9) for $\lambda = 10e-6$, the upper left panel for $\lambda = 10e-2$, and the upper right panel for $\lambda = 10$. R code to calculate the smoothing curves can be found in Wood (2006), and modified code to present the graphs in an `xyplot` from the lattice package is on the book website

²Keele (pg. 69, 2008) shows an example in which the fits of two smoothing splines with the same amount of smoothing (λ) are compared; one smoother uses four knots and the other uses sixteen knots; the difference between the curves is minimal.

Now that we have seen the effect of changing λ , we can ask the original question again: ‘What is the optimal amount of smoothing?’, or even better: ‘What is the optimal value for λ ?’ This question is answered with a process called cross-validation and is explained below.

3.3.3 Cross-Validation

To obtain the smoothers in Fig. 3.10, we *chose* values for λ and then minimised the expression in Equation (3.9). Now, we consider the minimisation of this function if *both* β and λ are unknown.

The function $f(X_i)$ is for the population, and its estimator (based on a sample) is written as:

$$\hat{f}(X_i)$$

The subtle difference is the hat $\hat{\cdot}$ on f , which denotes that it is an estimator. The estimator needs to be estimated to be as close as possible to the real value, and one criterion to judge this is (Wood, 2006)

$$M = \frac{1}{n} \sum_{i=1}^n (f(X_i) - \hat{f}(X_i))^2$$

If we would know $f(X_i)$, we could just choose a λ such that M is as small as possible. The problem is that $f(X_i)$ is unknown. Therefore, the strategy is to replace M by ‘something’ that can be calculated and minimise this ‘something’ as a function of λ . There are different options for the ‘something’ bit, and here is where things like cross-validation (CV), generalised cross-validation (GCV), unbiased risk estimator (UBRE), or Mallow’s C_p pop up. You will see these things in the numerical output of the GAM from the mgcv package if you apply automatic selection of the amount of smoothing. Let us start explaining how cross-validation, also called *ordinary* cross-validation (OCV), works. Define the OCV score as

$$V_0 = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{f}^{[-i]}(X_i))^2$$

The notation $[-i]$ means that the smoother is calculated using all observations except for observation i . Observation i is dropped, the smoother is estimated using the remaining $n - 1$ observations, the value for observation i is predicted from the estimated smoother, and the difference between the predicted and real value is calculated. This process is then repeated for each observation in turn, resulting in n prediction residuals. As always, residuals are squared and added up, which gives V_0 . Wood (2006) shows that the expected value of V_0 is approximately equal to the expected value of M plus σ^2 :

$$E[V_0] \approx E[M] + \sigma^2$$

If the aim is to find a λ that minimises M , then minimising V_0 is a reasonable approach. The process of minimising V_0 is called ordinary cross-validation. However, for a large data set, this is a time consuming process as we have to repeat the analysis n times. Luckily, a mathematical shortcut exists to calculate V_0 in one analysis! The new equation for V_0 is

$$V_0 = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{f}(X_i))^2 / (1 - A_{ii})^2$$

The A_{ii} are the diagonal elements of the so-called influence matrix. Explaining what exactly this is requires a certain degree of matrix algebra and the interested reader is referred to Section 4.5.2 of Wood (2006). However, instead of working with V_0 , we use a slightly modified version, namely, the generalised cross validation (GCV). The $1 - A_{ii}$ bit is replaced by something that involves the trace of $\mathbf{I} - \mathbf{A}$, and the justification for this requires fairly complex matrix algebra. It suffices to know that GCV is a modified version of OCV with computational advantages.

To illustrate the process, we calculated GCV for a large range of values for λ (Fig. 3.11). The lowest GCV value is obtained for $\lambda = 0.000407$ (we took this from the numerical output underlying the graph). Hence, the smoother in the third panel (from the left) at the bottom of Fig. 3.10 is close to the optimal smoother.

It is advisable not to follow blindly the results of the cross-validation. Violation of collinearity and independence and application of cross-validation on small (< 50) data sets can cause trouble (e.g. over-smoothing). It may be wise to verify the cross-validation results with smoothers in which the amount of smoothing is selected manually (or modify the results from the optimal model a little bit).

Instead of providing the output of the cross-validation in terms of λ , the `gam` function in the `mgcv` package uses a term called the *effective degrees of freedom*

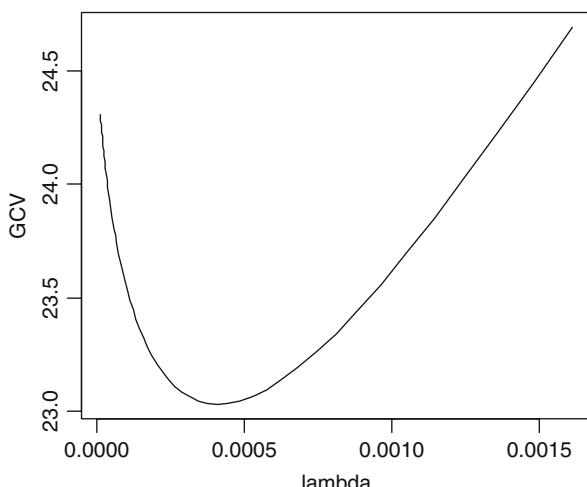


Fig. 3.11 GCV score plotted versus λ . The optimal smoother has $\lambda = 0.000407$. R code to produce this graph is given on the book website

(edf). This is a value between 0 and infinity and is a sort of mathematical transformation of λ . The higher the edf, the more non-linear is the smoothing spline.

In the cross-validation process, we assumed that the residual variance σ^2 is unknown. The UBRE approach minimizes a slightly different criterion, and it works well if σ^2 is known. Its underlying formulae are identical to Mallow's C_p .

Another issue is the actual estimation of the smoothing spline and λ . Both require iterative numerical algorithms. We can choose λ and find the smoothing spline by minimizing the expression in Equation (3.9) and put the generalised cross-validation procedure on top of this to find optimal λ . This is called outer iteration. The alternative is to do it the other way around: put the generalised cross-validation inside the iteration process of minimising the expression in Equation (3.9). This is called performance iteration. Probably you don't want to know these numerical details; they only become an issue if you get convergence problems and want to try different settings.

3.3.4 Additive Models with Multiple Explanatory Variables

Equation (3.2) showed an additive model with one explanatory variable. The same model can easily be extended to two explanatory variables:

$$Y_i = \alpha + f_1(X_i) + f_2(Z_i) + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma^2) \quad (3.10)$$

The functions $f_1(X_i)$ and $f_2(Z_i)$ are smoothing functions of the explanatory variables X_i and Z_i respectively. Both functions can be written as in Equation (3.4). Just as the model with one explanatory variable was written as a linear regression model; so the model in Equation (3.10) is written as a linear regression model. Its form is exactly the same as in (3.9), except that the matrix \mathbf{X} now also contain information on the second smoother. This also means that we can have hybrid models of the form

$$Y_i = \alpha + f_1(X_i) + \beta \times Z_i + \text{factor}(W_i) + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma^2) \quad (3.11)$$

Examples of interaction in smoothers are given in the case study chapters, and an introductory example is discussed in the next section.

3.3.5 Two More Things

Before we give an example, there are two more things we need to discuss: degrees of freedom of a smoother and other types of smoothers. As to other type of smoothers, in the previous section, we discussed the cubic regression spline. Recall that this was a smoother that fits third order polynomials on segments of data, and to ensure a nice looking curve with no sudden jumps at the knots, certain conditions were imposed.

These conditions resulted in a basis system with rather intimidating expressions of the form

$$f(X_i) = \sum_{j=1}^p \beta_j \times b_j(X_i)$$

There is actually a large collection of related smoothers, and Chapter 4 in Wood (2006) gives a detailed mathematical explanation on different types of smoothers. We do not repeat the mathematical detail here. The main difference between these smoothers is the definition of the b_j s, and also a few differ with respect to the optimisation criterion defined in Equation (3.9).

A useful smoother is the cyclic cubic regression spline. It ensures that the value of the smoother at the far left point of the gradient is the same as at the far right point of the gradient. This comes in handy if you use a smoother for month (with 12 values) or a smoother for day of the year; it wouldn't make sense to have a big jump between the January value and the December value for the month smoother. Shrinkage smoothers are also useful; they can have 0 amount of smoothing. This means that if you do a backwards selection to find the optimal model, all smoothers with 0 amount of smoothing can be dropped simultaneously from the model.

The thin plate regression spline is yet another smoother, one that apparently does quite well (except for larger data sets). The thin plate regression spline involves higher order derivates in the integral in Equation (3.9). In practise, the difference between cubic regression splines, thin plate regression splines, B-splines, P-splines, etc., is rather small.

The final point we want to discuss in this section is the amount of smoothing for smoothing splines. If a model has two smoothers, say

$$Y_i = \alpha + f_1(X_i) + f_2(Z_i) + \varepsilon_i$$

then these two smoothers have the form

$$f_1(X_i) = \sum_{j=1}^p \beta_j \times b_j(X_i) \quad \text{and} \quad f_2(Z_i) = \sum_{j=1}^p \gamma_j \times b_j(Z_i)$$

We mentioned this earlier in this section. Using two smoothers instead of one smoother has an effect on the definitions of the \mathbf{Y} , \mathbf{X} , and $\boldsymbol{\beta}$ in Equation (3.9), but in essence the form stays the same. The optimisation criterion with the penalty for the wigginess becomes

$$\|\mathbf{Y} - \mathbf{X} \times \boldsymbol{\beta}\|^2 + \lambda_1 \int f_1''(x)^2 dx + \lambda_2 \int f_2''(x)^2 dx \quad (3.12)$$

This looks rather intimidating, but all it does is allow for different amounts of wigginess per smoothing spline. As a result, some smoothers can be smooth (large λ_j), whereas others are not (small λ_j). Hence, the values of the λ_j s determine the amount of smoothing. To get these λ_j s, the optimisation criterion in Equation (3.12) can be written in matrix notation as

$$\|\mathbf{Y} - \mathbf{X} \times \boldsymbol{\beta}\|^2 + \boldsymbol{\beta}' \times \mathbf{S} \times \boldsymbol{\beta} \quad (3.13)$$

This looks a little bit friendlier, but the average reader may still be utterly confused by this stage. But you need not be too concerned as the rules for filling in the elements of \mathbf{S} are given in Wood (2006). Now we come to the reason why we explained all of this detail. The amount of smoothing of a smoother is not expressed in terms of the λ_j s, as you would expect, but as *effective degrees of freedom* for a smoother. A high value (8–10 or higher) means that the curve is highly non-linear, whereas a smoother with 1 degree of freedom is a straight line. You can think about it as a calibration parameter. Technically, the matrix \mathbf{S} , which depends on the λ s, is involved in determining the effective degrees of freedom (edf) and it mirrors the algebra underpinning linear regression.

We now give two examples, and once we are more familiar with the graphical and numerical output of GAMs, we discuss how much trust we can place in the p -values coming out of the `anova` and `summary` commands.

3.4 GAM Example 1; Bioluminescent Data for Two Stations

Earlier in this section, we used bioluminescent data, obtained from one particular station. A `xyplore` from the lattice package was given in Chapter 2, showing the data from all 19 stations along each depth gradient. In this section, we combine the data from two stations and show how GAM can be used to analyse the data. We start by presenting the data in a graph, see Fig. 3.12. The patterns in both graphs are

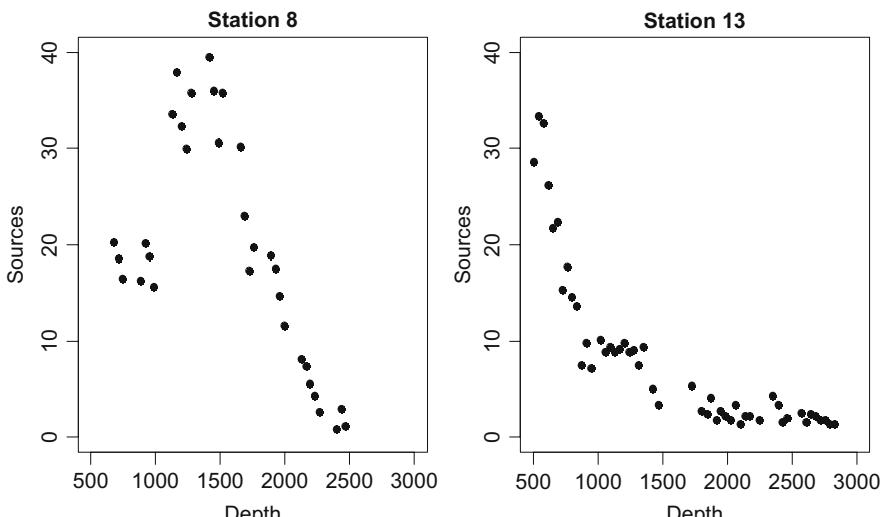


Fig. 3.12 Bioluminescent data for stations 8 and 13. Sources are plotted against depth. The axes limits were set to be equal

similar (decreasing pattern along depth), but there are also clear differences. The question that now arises is whether we can model the data with one smoother or perhaps we need two smoothers, one for each station? To answer this, we need to fit a model with one smoother and then a model with two smoothers and compare them with each other. We start with a GAM that only contains one smoother, go over the graphical and numerical output, and then return to the question of choosing the best model.

The R code used to create the graph is given below.

```
> library(AED); data(ISIT)
> S8 <- ISIT$Sources[ISIT$Station == 8]
> D8 <- ISIT$SampleDepth[ISIT$Station == 8]
> S13 <- ISIT$Sources[ISIT$Station == 13]
> D13 <- ISIT$SampleDepth[ISIT$Station == 13]
> So <- c(S8, S13); De <- c(D8, D13)
> ID <- rep(c(8, 13), c(length(S8), length(S13)))
> mi <- max(min(D8), min(D13))
> ma <- min(max(D8), max(D13))
> I1 <- De > mi & De < ma
> op <- par(mfrow = c(1, 2))
> plot(D8[I1], S8[I1], pch = 16, xlab = "Depth",
       ylab = "Sources", col = 1, main = "Station 8",
       xlim = c(500, 3000), ylim = c(0, 40))
> plot(D13[I1], S13[I1], pch = 16, xlab = "Depth",
       ylab = "Sources", col = 1, main = "Station 13",
       xlim = c(500, 3000), ylim = c(0, 40))
> par(op)
```

The first part of the code accesses the data, extracts the relevant data (`Sources` and `SampleDepth`) from stations 8 and 13, concatenates them in a long vector `So` and `De` using the concatenate command `c`, creates a vector that identifies which row corresponds to a certain station using the `rep` (repeat) command, and determines the common depth ranges for both stations (the `min` and `max` commands). These data are then plotted in two panels on the same graphical window (see the `par` command). The `[I1]` bit in the `plot` command ensures that we only use those data with the same depth ranges.

The underlying model for a GAM with one smoother is given by

$$\text{Sources}_i = \alpha + f(\text{Depth}_i) + \text{factor}(\text{Station}_i) + \varepsilon_i \quad \varepsilon_i \sim N(0, \sigma^2) \quad (3.14)$$

Station is a nominal explanatory variable with two levels, and depth is a continuous explanatory variable. The index i runs from 1 to 75. To apply this GAM model in the `mgcv` package, use the following code.

```
> library(mgcv)
> M4 <- gam(So ~ s(De) + factor(ID), subset = I1)
> summary(M4)
> anova(M4)
```

The `subset = I` part is merely used to ensure that we use observations within the same depth ranges. The `library` command loads the `mgcv` package. The argument within the `gam` function is similar to linear regression; `So` (sources of both stations) is modelled as a smoothing function (using the `s` function) of `De` (concatenated depth data of both stations) and the nominal variable `ID` (identifying the station). Both the `summary` and `anova` commands provide useful numerical output. The `summary` output is as follows.

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	19.198	1.053	18.236	< 2e-16
factor(ID)13	-12.296	1.393	-8.825	6.86e-13

Approximate significance of smooth terms:

	edf	Est.rank	F	p-value
<code>s(De)</code>	4.849	9	10.32	7.37e-10

R-sq. (adj) = 0.695 Deviance explained = 71.9%
GCV score = 38.802 Scale est. = 35.259 n = 75

The explained deviance (R^2) is 71.9%, the variance of the residuals is 35.26, the smoother for depth is significant at the 5% level, the estimated degrees for the smoother is 4.8, the intercept has a value of 19.2, and station 13 is 12.3 units lower than station 8. The output from the `anova` command is more compact and is useful if the model contains nominal variables with more than two levels as it gives one *p*-value for all the levels using an *F*-test. It is given below.

Parametric Terms:

	df	F	p-value
<code>factor(ID)</code>	1	77.88	6.86e-13

Approximate significance of smooth terms:

	edf	Est.rank	F	p-value
<code>s(De)</code>	4.849	9.000	10.32	7.37e-10

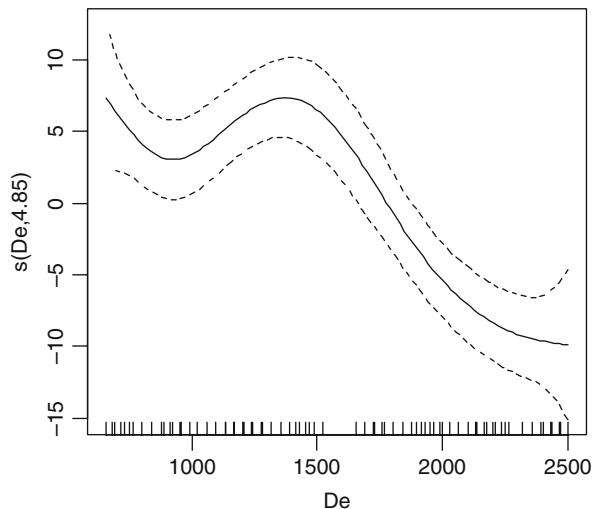
In practice, both the `summary` and `anova` commands would be used. The `plot` (`M4`) command produces the fitted smoother in Fig. 3.13. To get the fitted values for observations from station 8, we use

$$Sources_i = 19.2 + f(De_i) + \varepsilon_i \quad \varepsilon_i \sim N(0, 5.9^2) \quad (3.15A)$$

And for observations of station 13 we use

$$Sources_i = 19.2 + f(De_i) - 12.3 + \varepsilon_i \quad \varepsilon_i \sim N(0, 5.9^2) \quad (3.15B)$$

Fig. 3.13 Estimated smoothing curve. The x-axis shows the values of depth (vertical lines) and the y-axis the contribution of the smoother to the fitted values. The solid line is the smoother and the dotted lines 95% confidence bands. The little vertical lines along the x-axis indicate the depth values of the observations



The values 19.2 and 12.3 can be subtracted to give a new intercept of 6.9. In both cases, the smoothing function $f(\text{Depth}_i)$ is the solid curve in Fig. 3.13. The mgcv package has several useful tools to visualise the results. The following two commands produce the 3-dimensional plot in Fig. 3.14.

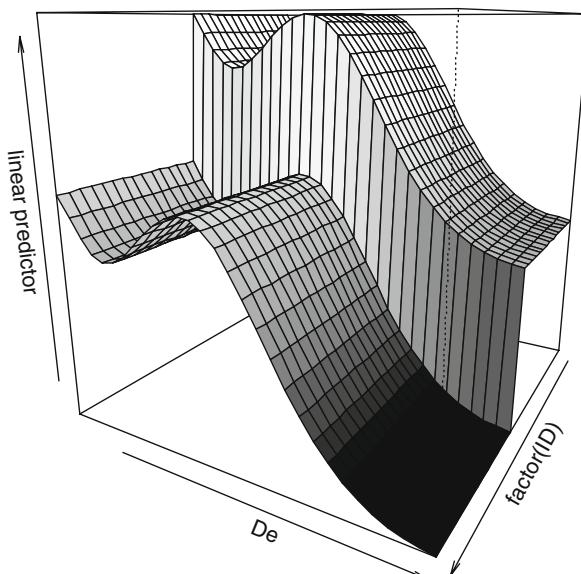


Fig. 3.14 Three-dimensional graph showing the fitted values for both stations. The graph shows the effect of the nominal variable ID (representing station)

```
> par(mar = c(2, 2, 2, 2))
> vis.gam(M4, theta = 120, color = "heat")
```

The `par` command adjusts the white space around the figure and the `vis.gam` function visualises what the gam is doing; it produces two parallel smoothing curves with different intercepts. The `theta` option changes the angle the window is viewed from.

Another useful tool is the `gam.check` (`M4`) command. It produces graphical diagnostics for the model; see Fig. 3.15. These can be used to (i) assess normality (the QQ-plot and the histogram), (ii) homogeneity (residuals versus fitted values, also called the linear predictor for the Gaussian distribution with identity link – see Chapter 7)), and (iii) model fit (fitted values versus observed values). In this case, the results in this graph do not look very promising. There are clear patterns in the right two panels. Before concluding that more complicated models (e.g. additive mixed modelling) are needed, we should first try to extend the current model with more covariates, or as in this case, add an interaction term between depth and station.

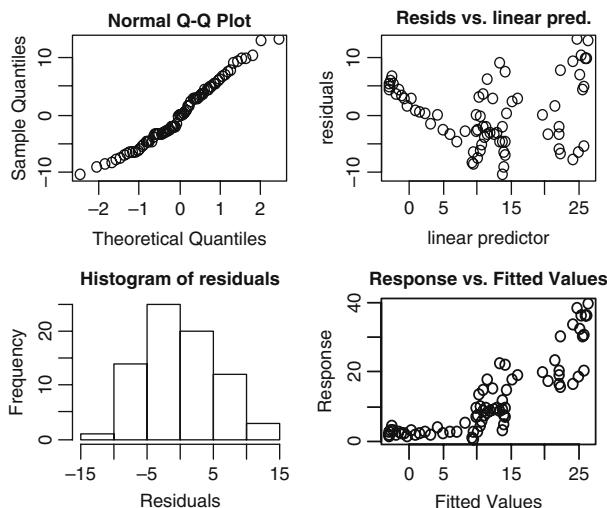


Fig. 3.15 Validation tools for the GAM model that contains one smoother for depth and a nominal variable ID (= station). The QQ-plot and the histogram are used to assess normality and the residuals versus fitted values homogeneity. The response against fitted values should ideally show a straight line

3.4.1 Interaction Between a Continuous and Nominal Variable

Recall that the model in Equation (3.14) assumes that both stations have the same depth-source relationship. However, the scatterplots in Fig. 3.12 clearly indicate that there are differences in this relationship per station. The term `factor` (`ID`) in the

GAM is only adding or subtracting a constant value to the smoother; it does not allow for a *change* in the source-depth relationship. In a GAM, interaction is not the same interaction we know from linear regression. To understand this, we first write down the R code required for the ‘interaction’ in the mgcv package. There are two ways of doing this. The first option is as follows.

```
> M5<-gam(So ~ s(De) +
            s(De, by = as.numeric(ID == 13)) +
            factor(ID), subset = I1)
> anova(M5)

Parametric Terms:
      df      F p-value
factor(ID) 1 573.6 <2e-16

Approximate significance of smooth terms:
              edf Est.rank      F p-value
s(De)           8.073   9.000 98.88 <2e-16
s(De):as.numeric(ID == 13) 6.696   9.000 48.39 <2e-16
```

The `s(De)` part applies a smoother along depth for all data points. Hence, it represents the overall depth effect at both stations. The second smoother along depth contains a `by` argument. The `as.numeric` is used to convert the vector with TRUE (an observation is from station 13) and FALSE (it is not from station 13) into a vector with ones (station 13) and zeros (not from station 13). This smoother is then only using the observations for which there is a 1 in the `by` command. In this case, the second depth smoother represents the deviation at station 13 from the overall source-depth relationship. The output from the `anova` indicates that the second depth smoother is highly significant. Hence, there are different depth patterns at each station. To see what the model is doing, both smoothers are plotted in Fig. 3.16. The smoother in the left panel represents the overall source-depth relationship, and the right panel the deviation from this at station 13. Hence, for station 13, you need to add up both the smoothers, and add the contribution from the factor to get the fitted values. Comparing Figs. 3.12 and 3.16 should give some clues what the second smoother (obtained with the `by` command) is doing; it is adjusting the pattern along the depth gradient for the second station.

The graphical results obtained by the `gam.check(M5)` command are given in Fig. 3.17 and looks considerably better than in Fig. 3.15, especially the right two panels. However, the variation for larger fitted values (nearer the surface) seems to be larger than for the smaller fitted values (at deeper depths). Biologically, this makes sense, but it does violate the homogeneity assumption. Tools to solve this for these data are discussed in case study 17.

But which of the models is better? Although this question can be readily answered in favour of the second model based on the model validation plots (compare Figs. 3.15 and 3.17), it would be nice if we could also compare the models using a test or selection criteria as the graphical evidence may not always be as clear as it

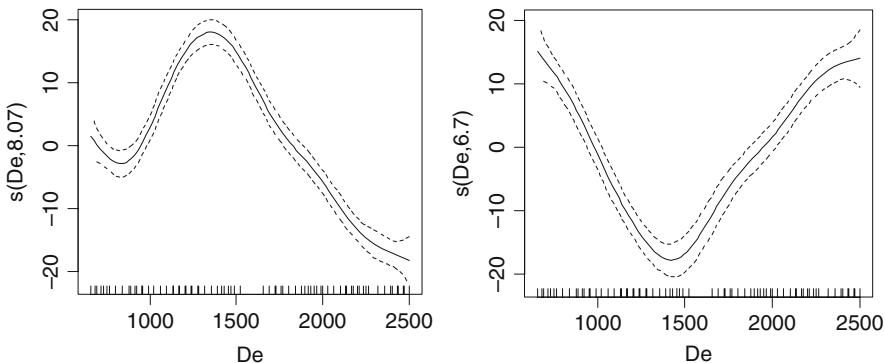
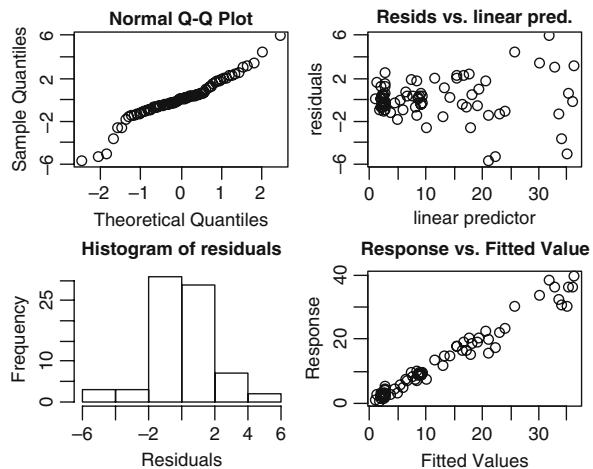


Fig. 3.16 Estimated smoothing curves for the model that contains one depth smoother for both stations (left panel) and a depth smoother using the `by` command, representing the deviation at station 13 (right panel)

Fig. 3.17 Validation tools for the GAM model that contains one smoother for depth, a smoother for depth that represents the deviation at station 13, and a nominal variable ID. The QQ-plot and the histogram are used to assess normality, and the residuals versus fitted values homogeneity. The response versus fitted values should ideally show a *straight line*



is in this case. There are many ways of doing this. The first option is with the Akaike Information Criteria (AIC), which measures goodness of fit and model complexity. The lower the AIC, the better the model fits the data. The AIC can be obtained by the commands `AIC(M4)` and `AIC(M5)` and gives the values 488.56 and 345.26, respectively, clearly favouring the second model. A nearly identical option is to use the generalised cross validation score, indicated by GCV in the output and obtained by the `summary` command. For the first model, it is 38.80 and for the second model (not presented above) it is 6.03, giving the same conclusion as using the AIC. The third option is to use the *F*-statistic and associated *p*-value for the smoother using the `by` option, obtained by `summary(M5)`. These values are not presented here, but

give $p < 0.001$. Yet, another option is to compare the model with and without the second smoother and apply an F -test (Wood, 2006). This is done with the command

```
> anova(M4, M5, test = "F")

Analysis of Deviance Table
Model 1: So ~ s(De) + factor(ID)
Model 2: So ~ s(De) + s(De, by = as.numeric(ID == 13))
          + factor(ID)

Resid. Df Resid. Dev      Df Deviance      F    Pr(>F)
1     68.1507    2402.90
2     58.2309    272.94    9.91   2129.96  45.80 < 0.001
```

The results of this test confirm earlier results. Interpretation of all these tests and resulting p -values should be done with care as these p -values are ‘approximate’. Don’t be too confident with a p -value of 0.04. We will discuss this aspect further in Section 3.6.

Instead of fitting a model that contains one smoother for depth for all stations, and one that represents the deviation at station 13, it is also possible to fit a GAM with one smoother for the data of station 8, and one smoother for the data of station 13; both smoothers would use the `by` option in this case. The R code is as follows.

```
> M6 <- gam(So ~ s(De, by = as.numeric(ID == 8)) +
             s(De, by = as.numeric(ID == 13)) +
             factor(ID), subset = I1)
```

Application of this type of models is discussed in various case study chapters. Models constructed with the `by` command are also called variable coefficient models (Hastie and Tibshirani, 1990). Their general statistical model formulation is given by

$$Y_i = \alpha + f_1(X_{1i}) + f_2(X_{2i}) \times X_{3i} + \varepsilon_i$$

Y_i is the response variable, α the intercept, $f_1(X_{1i})$ is a smoother, and the values of the smoother $f_2(X_{2i})$ are multiplied with the values of X_{3i} . Note that we are not multiplying X_{2i} itself with X_{3i} , but the smoother. Hence, the equations for the models in M5 and M6 are, respectively,

$$\begin{aligned} M5 : Sources_i &= \alpha + f_1(Depth_i) + f_2(Depth_i) \times ID_{13,i} + \varepsilon_i \\ M6 : Sources_i &= \alpha + f_1(Depth_i) \times ID_{8,i} + f_2(Depth_i) \times ID_{13,i} + \varepsilon_i \end{aligned}$$

The variable $ID_{8,i}$ is 1 if an observation is from station 8 and 0 else. The variable $ID_{13,i}$ is 1 if an observation is from station 13 and 0 else.

3.5 GAM Example 2: Dealing with Collinearity

The previous example was relatively simple as there were only two explanatory variables. This makes the model selection process relatively simple; just compare the model with and without the interaction term (obtained by the `by` option). Now we consider an example that contains considerably more explanatory variables. As explanatory variables increase, it becomes important to avoid using collinear explanatory variables in GAM. If you do use explanatory variables that are highly correlated with each other, using GAMs becomes a rather confusing experience. Data exploration tools to identify collinearity are discussed in Appendix A and can also be found in Zuur et al. (2007).

In this section, we show how confusing GAM becomes if you ignore this step of avoiding correlated explanatory variables. We use a plant vegetation data set for illustration. Sikkink et al. (2007) analysed grassland data from a monitoring programme from two temperate communities in Montana, USA: Yellowstone National Park and National Bison Range. The aim of the study was to determine whether the biodiversity of these bunchgrass communities changed over time and if they did, whether the changes in biodiversity relate to specific environmental factors. Here, we use the Yellowstone National Park data. Sikkink et al. (2007) quantified biodiversity using species richness to summarise the large number of species: ninety species were identified in the study. Richness is defined as the *different number* of species per site. The data were measured in eight different transects and each transect was measured repeatedly over time with time intervals of about four to ten years. For the moment, we ignore the temporal aspects of the data. And, instead of using all 20 or so explanatory variables, we use only those explanatory variables that Sikkink et al. (2007) identified as important. Figure 3.18 shows a scatterplot of all the variables used in this section. The response variable is species richness for the 64 observations, and the explanatory variables are rock content (ROCK), litter content (LITTER), bare soil (BARESOIL), rainfall in the fall (FallPrec), and maximum temperature in the spring (SprTmax). The correlation between ROCK and LITTER is reasonably high with a Pearson correlation of -0.7.

The following R code was used to run a GAM on these data. We ignore the fact that a Poisson distribution might be more appropriate, and that there are temporal aspects in this data set.

```
> library(AED); data(Vegetation)
> library(mgcv)
> M7 <- gam(Richness ~ s(ROCK, bs = "cs") +
+             s(LITTER, bs = "cs") + s(BARESOIL, bs = "cs") +
+             s(FallPrec, bs = "cs") + s(SprTmax, bs = "cs"),
+             data = Vegetation)
```

The only new bit, compared to the previous section, is the `bs = "cs"` part. It ensures that a regression spline with shrinkage is applied. Shrinkage means that a smoother can have 0 degrees of freedom. All smoothers that have 0 degrees of

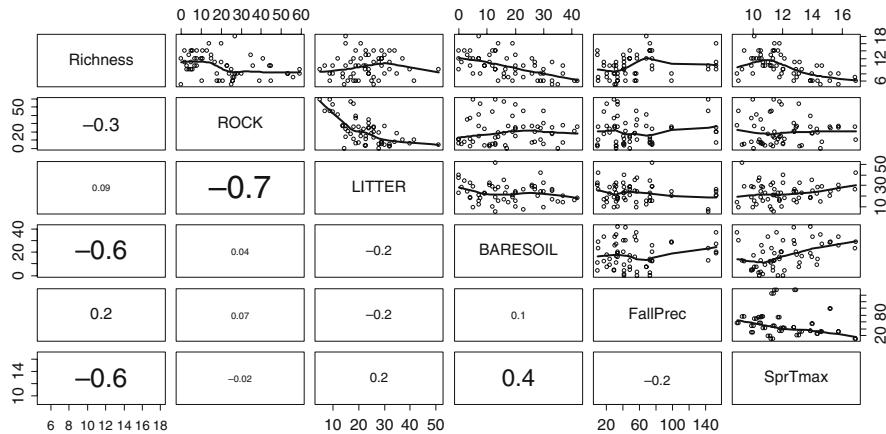


Fig. 3.18 Scatterplot of richness (response variable), rock content, litter content, baresoil, rainfall in the fall, and maximum temperature in the spring for the vegetation data set. The upper panels show scatterplots with a smoother added to visualise the patterns, and the lower panels contain the Pearson correlation coefficients. The font of the correlation coefficient is proportional to its estimated value. The code to produce this graph is given on the book website and is based on code presented in the help file of the `pairs` function

freedom can be dropped simultaneously from the model. The relevant output from the `anova` (M7) command is presented below, and shows that this is not happening.

Approximate significance of smooth terms:

	edf	Est.rank	F	p-value
s(ROCK)	1.750	4.000	4.634	0.00367
s(LITTER)	1.865	4.000	2.320	0.07376
s(BARESOIL)	4.167	9.000	2.991	0.00820
s(FallPrec)	4.699	9.000	2.216	0.04150
s(SprTmax)	5.103	9.000	3.508	0.00286

Not all terms are significant at the 5% level, and the estimated smoothing curves, with the wide confidence bands for some of the terms confirm this (Fig. 3.19). We can now do two things: either leave the model as it is or apply a backwards selection approach to find the optimal model. We prefer to use a model with only significant terms; so we go for the second option. We can either use a selection criterion like the AIC or CGV in a stepwise backwards selection process, like in linear regression or use hypothesis testing procedures. The first approach is better than the second, but the second approach takes less time; it drops the least significant term, refits the model, and continues this process until all terms are significant. If you do this, you end up with a GAM that only contains smoothing terms of ROCK, BARESOIL, and SprTmax: the estimated smoothing curves are presented in Fig. 3.20. The shape of the smoothers suggest that the higher the rock content, the lower the species richness, and the same relationship applies to bare soil.

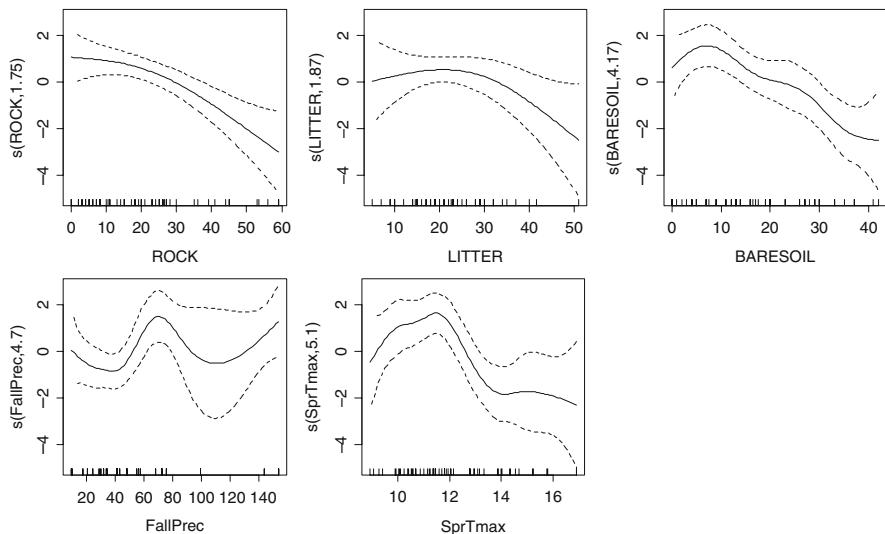


Fig. 3.19 Estimated smoothing curves for the GAM model containing all explanatory variables

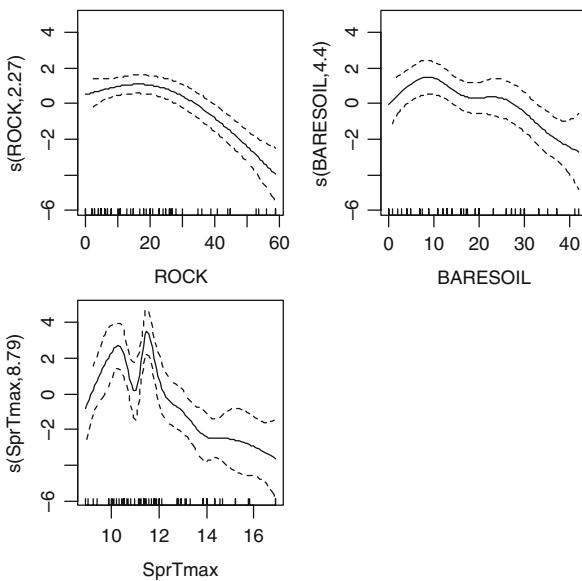


Fig. 3.20 Estimated smoothing curves for the optimal GAM model

The smoother for maximum temperature is rather difficult to interpret. During the model selection process it took various different shapes. This may indicate that the smoother for SprTmax may represent patterns that are actually due to other variables. Draper and Smith (1998) discuss various tools in linear regression analysis to identify the presence of collinearity. One of their recommendations is to see whether slopes change radically if a term is omitted as this is an indication for the presence of collinearity. Although we do not have slopes here, the amount of smoothing (or better: the associated degrees of freedom) for SprTmax did show considerable changes during the selection process (from 4 to 8 edf). This is probably caused by collinearity.

3.6 Inference*

In the previous section, the software gave us p -values for smoothers (these were derived from F -tests). There is a little catch here; these p -values are approximate and should be used with care.

The principle behind confidence intervals in GAM is relatively simple, and follows linear regression. A multiple linear regression model is given by

$$Y_i = \alpha + \beta_1 \times X_{1i} + \cdots + \beta_q \times X_{qi} + \varepsilon_i$$

Using matrix algebra, this can also be written as

$$\mathbf{Y} = \mathbf{X} \times \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The vector \mathbf{Y} contains all observed data, the matrix \mathbf{X} all q explanatory variables and its first column contains only ones (for the intercept), $\boldsymbol{\beta}$ contains all regression parameters (including the intercept α), and $\boldsymbol{\varepsilon}$ all the residuals ε_i . Using this notation, the ordinary least square estimator for $\boldsymbol{\beta}$ can be found using

$$\hat{\boldsymbol{\beta}} = (\mathbf{X} \times \mathbf{X})^{-1} \times \mathbf{X}' \times \mathbf{Y}$$

If you are not familiar with matrix algebra, then this may look intimidating, but this solution can be found in most statistical textbooks. Once you have the estimated regression parameters, the fitted values are given

$$\hat{\mathbf{Y}} = \mathbf{X} \times \hat{\boldsymbol{\beta}} = \mathbf{X} \times (\mathbf{X} \times \mathbf{X})^{-1} \times \mathbf{X}' \times \mathbf{Y} = \mathbf{H} \times \mathbf{Y}$$

The reason that we show all this matrix algebra is because of $\mathbf{H} \times \mathbf{Y}$. \mathbf{H} is also called the hat matrix, and it plays an important role as it goes straight into equations for standard errors of predicted values, and it is also used to determine degrees of freedom.

The LOESS smoother and the polynomial and cubic regression splines are all local regression models, and in fact, can all be written in exactly the same form as the linear regression model:

*Means that it is difficult and that it can be skipped upon first reading.

$$\hat{\mathbf{Y}} = \mathbf{S} \times \mathbf{Y}$$

Instead of using an \mathbf{H} , we used a matrix \mathbf{S} . Depending on the type of smoother, the computer software will fill in the elements of \mathbf{S} . An example can be found in Section 7.4 in Zuur et al. (2007) for a moving average smoother. Because polynomial and cubic regression splines are linear regression models, we can follow exactly the same theory as in linear regression. Hence, the variance of the fitted values are given by

$$\text{var}(\hat{\mathbf{Y}}) = \sigma^2 \times \mathbf{S} \times \mathbf{S}'$$

where σ^2 is the variance of the Y_i . This all follows immediately from linear regression theory. The matrix \mathbf{S} and its trace are then used to calculate degrees of freedom and residual degrees of freedom.

Recall that we decided to work with the smoothing splines (or penalised splines). Broadly speaking, the same approach can be followed, although the matrix \mathbf{S} is now more complex, see Keele (pg. 75, 2008). Basically, the solution of Equation (3.13) is again of the form

$$\hat{f} = \mathbf{S} \times \mathbf{Y}$$

but this \mathbf{S} is more complicated compared to regression splines as it is also a function of λ . The way \mathbf{S} is used in the calculation of confidence bands, degrees of freedom, etc., is nearly the same, though. The ‘nearly’ bit refers to the fact the p -values behave reasonable well for smoothing splines with known degrees of freedom, but if these are estimated (e.g. using cross-validation), they can be misleading. This is because the uncertainty due to estimating the λ s is neglected. Wood (2006) mentioned that based on limited simulation experience, the p -values close to 0.05 can be around half of their correct value when the null hypothesis is true. This means that smoothers with p -values smaller than 0.001 can be trusted, and we can also trust the p -value if it is 0.2, 0.5, or 0.8. It is the smoother with a p -value of 0.02, 0.03, 0.04, or 0.05 that gives trouble. The same holds for F -tests comparing nested GAMs. If this is a serious issue for your data, then you could do bootstrapping to get better p -values. Chapter 8 in Keele (2008) gives a nice introduction, and a detailed algorithm is presented in Davison and Hinkley (1997). When writing this chapter, we were tempted to add a bootstrapping example; however, it would only duplicate Chapter 8 from Keele (2008).

3.7 Summary and Where to Go from Here?

In this chapter, we started with LOESS smoothers. Recall that a weighted linear (or quadric) regression model is applied on all data in a window around the target value. The amount of smoothing is determined by the size of the window, also called the

span width. We then introduced splines; simple linear splines, quadratic and cubic splines. The gradient is divided in segments using a certain number of knots, and a linear, quadratic, or cubic polynomial model is fitted on the data in each segment. Certain conditions are imposed ensuring a smooth connection at the edges. Finally, smoothing splines are introduced as the best option. It minimises the penalised least squares criterion. Table 3.1 summarises all smoothers.

In Section 3.6, we discussed that the p -values thrown at you by the software are approximate for smoothing splines. You can safely make biological statements based on smoothers with a p -value of 0.001 (or smaller) or a p -value of 0.1 (or larger), but be very careful with smoothers for which p is just below the magical 0.05 level. If you really want to say something about such smoothers, apply bootstrapping.

Some of the problems encountered for linear regression can also cause trouble in additive modelling, e.g. violation of independence, heterogeneity, and nested data. In fact, we have been cheating in most GAM examples presented in this chapter. For example, we used vegetation data that were measured repeatedly over time. The analysis carried out assumes (implicitly) that there is no correlation between observations made at the same transect. The lags between two observations at the same transect is approximately 4–10 years.

The same holds for the bioluminescent data; measurements at the surface may be related to measurements at deeper depths, simply because particles tend to move down from the surface to the bottom of the ocean. It is like rain; if it rains at 100 m, it

Table 3.1 Summary of all smoothers discussed in Chapter 3

Name of smoother	What is it?	Relevant options for smoothing
LOESS	Weighted linear regression on a window around the target value. Move target value.	Size of the span
Simple linear regression spline	Gradient is divided in segments using knots. Fit bivariate linear regression model on each segment.	Number and location of knots
Quadratic and cubic regression splines	Gradient is divided in segments using knots. Fit a quadratic or cubic polynomial on each segment, and ensure a smooth connection at the knots.	Number of knots, location of knots, and degree of polynomial
Smoothing splines (alias penalised splines)	Gradient is divided in a large number of segments. Fit a cubic polynomial model on each segment, and ensure a smooth connection at the knots. Minimise the penalised sum of squares in Equation (3.9).	Use large number of knots. Find optimal value of λ

will probably also rain at 50 m. Hence, there is also a correlation issue here. On top of this, we noticed that there is violation of homogeneity along the depth gradient (more variation towards the surface). Then there is another issue if we analyse data of all 19 stations; these are nested data. Data from the same station may be more similar than data from different stations.

All these issues (heterogeneity, nested data, and correlation) are addressed in Chapters 4, 5, 6, and 7.

Chapter 4

Dealing with Heterogeneity

This chapter, and the following three chapters, discuss solutions to the problems introduced in Chapters 2 and 3: heterogeneity, nested data, temporal correlation, and spatial correlation. We use both the linear regression model and the additive model as starting points. Figure 4.1 shows an overview of the methods we discuss in Chapters 4, 5, 6, and 7. In all these chapters, the model consists of a fixed term and a random term. The fixed term describes the response variable Y as a function of the explanatory variables via $\alpha + \beta_1 \times X_1 + \dots + \beta_q \times X_q$ in linear regression or $\alpha + f_1(X_1) + \dots + f_q(X_q)$ in additive modelling. This part of the model is described in Appendix A and Chapter 3. The random part contains components that allow for heterogeneity, nested data (random effects), temporal correlation, spatial correlation, and a real random term. It is also possible to have a combination of these components.

If the random part only contains the real random term, we are back to linear regression or additive modelling. If it allows for nested data, the resulting model is called a mixed effects model. If it only allows for heterogeneity, we call it a generalised least squares (GLS) model. This is essentially a weighted linear regression. GLS is the subject of this chapter. It is tempting to call the whole equation in Fig. 4.1 mixed effects modelling (or just mixed modelling), even if it only contains the heterogeneity bit, but strictly speaking this is wrong. However, as software routines for GLS, auto-correlation and nested data can all use the same R package, and sometimes the same routines, then it is easy to get confused about names.

We closely follow Chapter 5 in Pinheiro and Bates (2000), and the first 5 chapters of Verbeke and Molenberghs (2000). We also made extensive use of Diggle et al. (2002). We strongly recommend these books, as they provide a good technical explanation and a more unified overview of mixed modelling techniques than we have provided, albeit at a much higher mathematical level. Another good ecological source for the linear mixed model is Schabenberg and Pierce (2002), but it does not contain R code.

For the additive mixed modelling, Ruppert et al. (2003) and Wood (2006) are some of the few available books. But again, these are rather technical.

If you are willing to read non-ecological textbooks, we strongly recommend West et al. (2006), as it contains a series of case studies. However, a basic familiarity

Y = fixed part	+ random part
$\alpha + \beta_1 X_1 + \dots + \beta_q X_q$	Heterogeneity
$\alpha + f_1(X_1) + \dots + f_q(X_q)$	Nested data (random effects)
	Temporal correlation
	Spatial correlation
	Random noise

Fig. 4.1 Outline of the different methodologies discussed in Chapters 4, 5, 6, and 7. The fixed part consists of the explanatory variables as we know from linear regression or additive modelling. The random part consists of a real random term and terms that allow for heterogeneity, nested data (random effects), temporal correlation, or spatial correlation. The subject of this chapter is heterogeneity

with linear mixed modelling is recommended as their first chapter summarises the underlying theory rather quickly. Other useful books, but mainly focussed on economics and social science are Goldstein (2003), Raudenbush and Bryk (2002), Snijders and Bosker (1999), and at a higher mathematical level, Jiang (2007).

The confusing aspects of most of these books are the wide range of different names and underlying mathematical notation. Mixed modelling, multilevel analysis, hierarchical linear models, and repeated measurements are just a few of the names that all refer to the same set of models.

4.1 Dealing with Heterogeneity

4.1.1 Linear Regression Applied on Squid

Several examples in Chapters 2 and 3 showed residual spread varying per stratum (level) of a nominal variable, or increasing or decreasing along an explanatory variable. For example, the spread in pelagic bioluminescent data (Chapter 2) decreased at deeper depths, and both the *Hediste diversicolor* and wedge clam data sets (Chapter 2) showed different residual spread per stratum for some of the variables (month, biomass, nutrient). This violates the homogeneity of variance assumption, one of the most important assumptions of linear regression and additive modelling. Ignoring this problem may result in regression parameters with incorrect standard errors, and an F statistic no longer F distributed and the t statistic not following a t distribution. This invalidates the statistics used in Chapters 2 and 3 for assessing statistical significance (Wooldridge, 2006). In this section, we provide several solutions to heterogeneity. The easiest solution is a data transformation, but we try to avoid this for as long as possible. In our view, heterogeneity is interesting ecological information that you should not throw away, just because it is statistically inconvenient. With a ‘little’ bit of extra mathematical effort, heterogeneity can be incorporated into the models and can provide extra biological information.

To illustrate the methods, we use data published by Smith et al. (2005), who looked at seasonal patterns in reproductive and somatic tissues in the squid *Loligo*

forbesi. They used several variables on female and male squid, but in this chapter, we only use the dorsal mantle length (in mm) and testis weight from 768 male squid. The aim is to model the testis weight as a function of the dorsal mantle length (DML) and the month recorded. The idea behind the original analysis was to investigate the role of endogenous and exogenous factors affecting sexual maturation, more specifically to determine the extent to which maturation is size-related and seasonal. Further biological information can be found in Smith et al. (2005). Our starting point is a linear regression model of the form (in words):

$$\text{Testisweight}_i = \text{intercept} + \text{DML}_i + \text{Month}_i + \text{DML}_i : \text{Month}_i + \text{residuals}_i \quad (4.1)$$

Month is used as a nominal variable (with 12 levels) and is DML fitted as a continuous variable. The notation ‘:’ is used for the interaction between DML and Month. Previous work on the related species *Loligo vulgaris* showed graphically that maturity was a function of both size and season, and that size-at-maturity differed between seasons (Raya et al., 1999). The index i runs from 1 to 768. The crucial assumption in Equation (4.1) is that the residuals are normally distributed with a mean of 0 and the variance is σ^2 . In mathematical notation

$$\varepsilon_i \sim N(0, \sigma^2)$$

where ε_i are the residuals. The important thing is that $\text{var}(\varepsilon_i) = \sigma^2$. The following R code loads the data, applies linear regression, and produces the validation graphs in Fig. 4.2. Note that there is a clear violation of homogeneity.

```
> library(AED); data(Squid)
> Squid$fMONTH <- factor(Squid$MONTH)
> M1 <- lm(Testisweight ~ DML * fMONTH, data = Squid)
> op <- par(mfrow = c(2, 2), mar = c(4, 4, 2, 2))
> plot(M1, which = c(1), col = 1, add.smooth = FALSE,
       caption = "")
> plot(Squid$fMONTH, resid(M1), xlab = "Month",
       ylab = "Residuals")
> plot(Squid$DML, resid(M1), xlab = "DML",
       ylab = "Residuals")
> par(op)
```

The $\text{DML} * \text{fMONTH}$ fits the main terms DML and MONTH (as a factor) and the interaction between these two variables (‘*’ replaces the ‘:’ from the word equation to denote interaction). Alternatively, code that does the same is $\text{DML} + \text{fMONTH} + \text{DML} : \text{fMONTH}$. This keeps the notation similar to the one we used in Equation (4.1). By default, the `plot` command produces four graphs (see Chapter 2), but the `which = c(1)` ensures that only the residuals versus fitted values are plotted. We decided not to add a smoothing curve (`add.smooth = FALSE`)

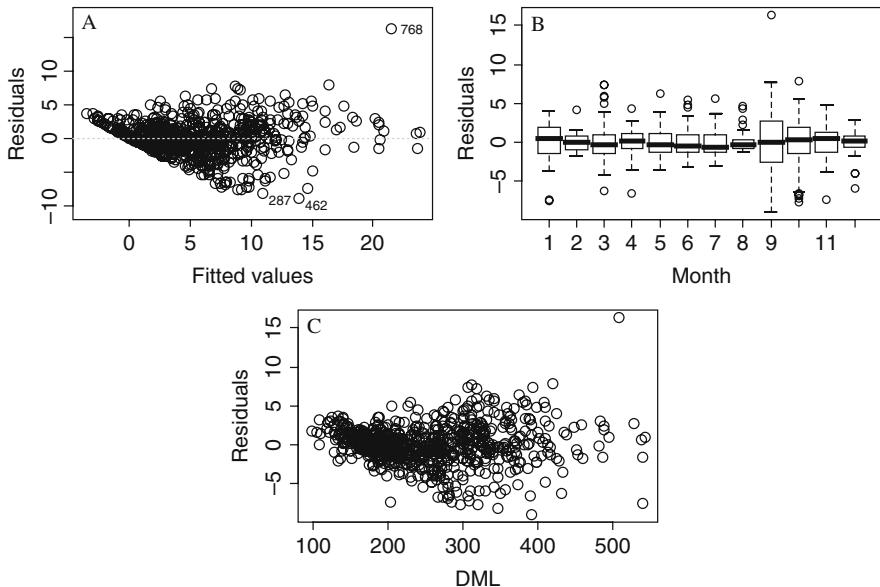


Fig. 4.2 **A:** Residuals versus fitted values. **B:** Residuals versus month. Because month is a nominal variable, boxplots are produced. **C:** Residuals versus DML. Panel A shows that there is clear violation of heterogeneity. Panels B and C were made to detect why there is heterogeneity

and omit the caption (`caption = " "`). All other commands are discussed in Chapters 2 and 3.

The numerical output (not shown here) shows that all regression parameters are significantly different from 0 at the 5% level. The problem is that we cannot trust these results as we are clearly violating the homogeneity assumption (note the cone shape pattern of the residuals in Fig. 4.2A). This means that the assumption that the residuals are normally distributed with mean 0 and variance σ^2 is wrong. However, in this case, the homogeneity clearly has an identifiable structure; the larger the length (DML), the larger the variation (Fig. 4.2C). So, instead of assuming that the residuals have variance $\text{var}(\varepsilon_i) = \sigma^2$, it might make more sense to assume that $\text{var}(\varepsilon_i)$ increases when DML_i increases. We can implement this in various mathematical parameterisations, and we discuss these next.

4.1.2 The Fixed Variance Structure

The first option is called the *fixed variance*, it assumes that $\text{var}(\varepsilon_i) = \sigma^2 \times DML_i$, and as a result we have

$$\varepsilon_i \sim N(0, \sigma^2 \times DML_i) \quad i = 1, \dots, 768 \quad (4.2)$$

Such a variance structure allows for larger residual spread if DML increases. And the good news is that there are no extra parameters involved! Technically, this model is fitted using the generalised least squares (GLS) method, and the technical aspects of this method are discussed later in this chapter. To fit a GLS in R, the function `gls` from the `nlme` package can be used. The variance structure (and any of the others we discuss later) can be selected by specifying the `weights` argument in the `gls` function. In fact, running the `gls` code without a `weights` option, gives you the same linear regression model already seen in Equation (4.1). The following R code applies the linear regression model in (4.1) and also the GLS with the fixed variance structure in Equation (4.2). The reason we refitted the linear regression model in Equation (4.1) with the `gls` function was to avoid a warning message in the `anova` comparison.

```
> library(nlme)
> M.lm <- gls(Testisweight ~ DML * fMONTH, data=Squid)
> vflFixed <- varFixed(~DML)
> M.gls1 <- gls(Testisweight ~ DML * fMONTH,
+                 weights = vflFixed, data = Squid)
> anova(M.lm, M.gls1)
```

The command `varFixed (~DML)` ensures a variance that is proportional to DML, and it needs to be specified via the `weights` argument in the `gls` function. Finally, the `anova` command gives

	Model	df	AIC	BIC	logLik
M.lm	1	25	3752.084	3867.385	-1851.042
M.gls1	2	25	3620.898	3736.199	-1785.449

The models are not nested; so no log-likelihood ratio test statistic is given, but the AIC clearly favours the model with the fixed variance in Equation (4.2). Note that both models have the same number of parameters! You can also use the command `AIC(M.lm, M.gls1)`.

4.1.3 The VarIdent Variance Structure

Now, just for a moment, we will forget about the residual spread increasing for larger DML values. So instead of recognising from Fig. 4.2C that the spread increases for larger DML values, we now realise from Fig. 4.2B that the spread also differs per month. To incorporate this pattern into the model, it is better to slightly change the indices used in the model notation:

$$\text{Testisweight}_{ij} = \text{intercept} + DML_{ij} + \text{Month}_j + DML_{ij}:\text{Month}_j + \text{residuals}_{ij} \quad (4.3)$$

Testisweight_{ij} is the testis weight of the i th observation in month j . This is exactly the same model as in Equation (4.1); we have only changed notation of the indices.

However, the new notation makes it easier to formulate the variance structure with different spread per stratum:

$$\varepsilon_{ij} \sim N(0, \sigma_j^2) \quad j = 1, \dots, 12 \quad (4.4)$$

So, we now have $\text{var}(\varepsilon_{ij}) = \sigma_j^2$, and each month is allowed to have a different variance. The following code implements different variances per stratum for month and applies the anova comparison.

```
> vf2 <- varIdent(form= ~ 1 | fMONTH)
> M.gls2 <- gls(Testisweight ~ DML*fMONTH, data =Squid,
+ weights = vf2)
> anova(M.lm, M.gls1, M.gls2)
```

The output of the anova command is given by:

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
M.lm		1 25	3752.084	3867.385	-1851.042			
M.gls1		2 25	3620.898	3736.199	-1785.449			
M.gls2		3 36	3614.436	3780.469	-1771.218	2 vs 3	28.46161	0.0027

We have decreased the font size of the numerical output to ensure it fits the page. The first two lines in the output are the same as above. The AIC of the model using the different variances per month is lower. You can also use the command `AIC(M.lm, M.gls1, M.gls2)`.

Notice that due to the variance structure in Equation (4.4), we now have to estimate 11 more parameters. We discuss below why it is not 12. We also get a log likelihood ratio comparing the variance structures in Equations (4.2) and (4.4). However, as these models are not nested, it is better not to use the log likelihood ratio. However, comparing models (4.1) and (4.4) does make sense as they are both nested. The null-hypothesis is

$$H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \dots = \sigma_{12}^2$$

with the alternative that they are not equal to each other. The R code to carry out this test and the resulting output is given below.

```
> anova(M.lm, M.gls2)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M.lm       1 25 3752.084 3867.385 -1851.042
M.gls2     2 36 3614.436 3780.469 -1771.218 1 vs 2 159.6479 <.0001
```

You can see the log likelihood ratio test indicates that the model with different variances per month is better, allowing us to reject the null hypothesis that all variances are the same. The `summary(M.gls2)` command gives the different variances (along with lots of other information).

```
> summary(M.gls2)
...
Variance function:
Structure: Different standard deviations per stratum
Formula: ~1 | fMONTH
Parameter estimates:
 2      9      12     11      8      10      5      7      6      4
1.00  2.99  1.27  1.50  0.98  2.21  1.63  1.37  1.64  1.42
 1      3
1.95  1.97
...
Residual standard error: 1.27
```

The numbers under the months (2, 9, 12, etc.) are multiplication factors. They show the ratio with the estimated residual standard error (1.27), the estimator for σ . Let us call this estimator s ; hence, $s = 1.27$. One multiplication factor is set to 1 (in this case month 2). In month 9, the variance is $2.99 \times s$, in month 12 it is $1.27 \times s$, etc. You can also change the nominal variable fMONTH and set January to the baseline. Note that months 9 and 10, and 3 have the highest ratios indicating that in these months there is more residual variation.

If you have two nominal explanatory variables, say month and location, and the spread differs for all stratum, then you can use varIdent (form = ~1|fMONTH * factor(LOCATION)). But we don't have location information for the squid data.

So, which option is better: different spread per month or different spread along DML? If in Fig. 4.2A, the smaller fitted values are from months with less spread and the larger fitted values are from months with higher spread, then using different variances per month makes more sense. The following code produces a graph like Fig. 4.2A and colours observations of the same month:

```
> plot(M.lm, which = c(1), col = Squid$MONTH,
       add.smooth = FALSE, caption = "")
```

The col = Squid\$MONTH part ensures that observations of the same month have the same colour. This approach works here because MONTH is coded with values 1–12. If you coded it as 'January', 'February', etc. then you would need to make a new vector with values 1, 2, 3, etc.; see, for example, Dalgaard (2002) on how to do this. Although not presented here, the graph does not show any clear grouping.

Let us try to understand what is really going on. The R code below makes a coplot (explained in Chapter 2) of the residuals versus DML, conditional on month for the linear regression model in Equation (4.1). The resulting coplot is given in Fig. 4.3. The residual variation differs per month, but in some months (e.g. 3, 9, and 10) the residual spread also increases for larger DML values. So, both are influential: residual spread is influenced by both month and length!

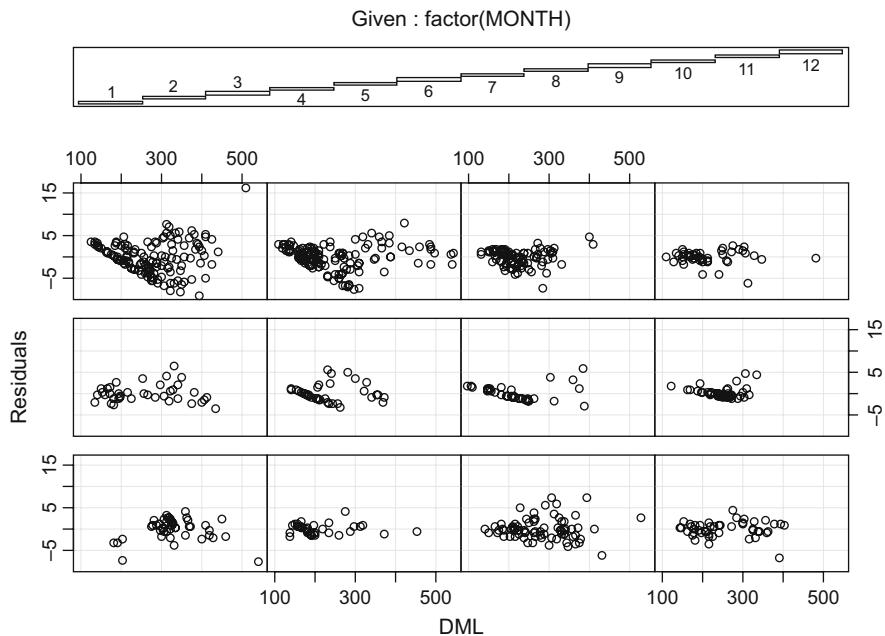


Fig. 4.3 Coplot of residuals obtained by the linear regression model in Equation (4.1) versus DML conditional on month. The lower left panel corresponds to month 1, the lower right to month 4, and the upper right to month 12. Note that some months show clear heterogeneity, and others do not. Sample size may also be an issue here!

```
> E <- resid(M.lm)
> coplot(E ~ DML | fMONTH, data = Squid)
```

Before discussing how to combine both types of variation (variation linked with DML and variation linked with Month), we introduce a few more variance structures. In all these structures, the variance of the residuals is not necessarily equal to σ^2 , but is a function of DML and/or month.

An explanatory variable that is used in the variance of the residuals is called a *variance covariate*. The trick is to find the appropriate structure for the variance of ε_{ij} . The easiest approach to choosing the best variance structure is to apply the various available structures in R and compare them using the AIC or to use biological knowledge combined with some informative graphs like the coplot. Some of the variance functions are nested, and a likelihood ratio test can be applied to judge which one performs better for your data.

4.1.4 The varPower Variance Structure

So far, we have looked at the varFixed and varIdent variance structures. Next we look at the ‘power of the covariate’ variance structure. It uses the R

function `varPower`. For the squid data, a potential power of the covariate variance structure is

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times |DML_{ij}|^{2\delta}) \quad (4.5)$$

Hence, $\text{var}(\varepsilon_{ij}) = \sigma^2 \times |DML_{ij}|^{2\delta}$. The variance of the residuals is modelled as σ^2 , multiplied with the power of the absolute value of the variance covariate DML. The parameter δ is unknown and needs to be estimated. If $\delta = 0$, we obtain the linear regression model in Equation (4.1), meaning (4.1) and (4.5) are nested, and therefore the likelihood ratio test can be applied to judge which one is better. For $\delta = 0.5$ and a variance covariate with positive values, we get the same variance structure as specified in Equation (4.2). But if the variance covariate has values equal to 0, the variance of the residuals is 0 as well. This causes problems in the numerical estimation process, and if the variance covariate has values equal to zero, the `varPower` should not be used. For the squid data, all DML values are larger than 0 (DML is length); so it is not a problem with this example. The following R code implements the `varPower` function.

```
> vf3 <- varPower(form =~ DML)
> M.gls3 <- gls(Testisweight ~ DML * fMONTH,
  weights = vf3, data = Squid)
```

The AIC of this model is 3473.019, which is the lowest value so far (the lower the AIC the better the model). The `summary` command gives the value of $\delta = 1.75$. It is also possible to allow multiple variables in the `form` argument. This extension makes it possible to model an increase in spread for larger DML values, but only in certain months! The structure for the residuals is now

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times |DML_{ij}|^{2\delta}) \quad (4.6)$$

Hence, $\text{var}(\varepsilon_{ij}) = \sigma^2 \times |DML_{ij}|^{2\delta_j}$. The following R code implements this variance structure.

```
> vf4 <- varPower(form =~ DML | fMONTH)
> M.gls4 <- gls(Testisweight ~ DML * fMONTH,
  data = Squid, weights = vf4)
```

The `anova` command gives an AIC of 3407.51, now making it the best model so far. The parameters δ_j can be obtained using the `summary` command, and are

```
Variance function:
Structure: Power of variance covariate, different strata
Formula: ~DML | factor(MONTH)
Parameter estimates:
 2      9     12     11      8     10      5      7      6
1.73  1.79  1.73  1.75  1.62  1.79  1.75  1.67  1.75
 4      1      3
1.71  1.70  1.72
```

So, instead of having one δ , we now have twelve of them ($\delta_j, j = 1, \dots, 12$). There is little variation between the estimated values of δ_j , but keep in mind they are multiplied by two, before being used to take the power. It is also possible to set the δ_j for some months equal to an a priori chosen value and keep it fixed. This is handy if you know or want to test whether the spread along DML in some months is constant (e.g. in month 4, as suggested by the coplot in Fig. 4.3). This can be done with the `fixed` option in `varPower` (see page 210 in Pinheiro and Bates (2000) and the help file of `varPower`). The AIC can be used to judge whether fixing or not fixing is better.

4.1.5 The `varExp` Variance Structure

If the variance covariate can take the value of zero, the exponential variance structure is a better option. It uses the `varExp` function in R, and for the squid data, a possible exponential variance structure is

$$\text{var}(\varepsilon_{ij}) = \sigma^2 \times e^{2\delta \times \text{DML}_i} \quad (4.7)$$

This structure models the variance of the residuals as σ^2 multiplied by an exponential function of the variance covariate DML and an unknown parameter δ . If $\delta = 0$, this gives the variance structure of model (4.1). There are no restrictions on δ or DML. This structure also allows a decrease of spread for DML values if δ is negative. As before, we can allow for different δ per month. The R code to implement the exponential variance structure is

```
> vf5 <- varExp(form =~ DML)
> M.gls5 <- gls(Testisweight ~ DML * fMONTH,
+ weights = vf5, data = Squid)
```

The AIC of this model is 3478.15, which is slightly higher than for model `M.gls3`. Using `varExp(form =~ DML | fMONTH)` does the same trick as for model `M.gls4`, and allows the spread in DML to differ per month. Again, it is possible to fix some of the δ s.

4.1.6 The `varConstPower` Variance Structure

Another variance structure is the *constant plus power of the variance covariate* function, and it is implemented in the function `varConstPower`. It is defined by

$$\text{var}(\varepsilon_{ij}) = \sigma^2 \times (\delta_1 + |\text{DML}_{ij}|^{\delta_2})^2 \quad (4.8)$$

This function looks rather complicated. If $\delta_1 = 1$ and $\delta_2 = 0$, we are back to the linear regression model in Equation (4.1). If not, then the variance is proportional to a constant plus the power of the variance covariate DML. According to Pinheiro and Bates (2000), this variance structure works better than the `varExp` if the variance covariate has values close to zero. To use this variance structure in R, use

```
> vf6 <- varConstPower(form =~ DML)
> M.gls6 <- gls(Testisweight ~ DML * fMONTH,
                    weights = vf6, data = Squid)
```

Its AIC is 3475.02. Again, we can allow for different δ_1 s and δ_2 s per stratum of a nominal variable (e.g. MONTH). Such a model is fitted in R by

```
> vf7 <- varConstPower(form =~ DML | fMONTH)
> M.gls7 <- gls(Testisweight ~ DML * fMONTH,
                    weights = vf7, data = Squid)
```

The AIC of this model is 3431.51. The associated variance structure is given by

$$\text{var}(\varepsilon_{ij}) = \sigma^2 \times (\delta_{1j} + |DML_{ij}|^{\delta_{2j}})^2 \quad (4.9)$$

The only difference with the variance in Equation (4.8) is the index j ($j = 1, \dots, 12$) from δ_1 and δ_2 . Again, it is possible to set the δ_1 s and δ_2 s to a preset value for particular months and keep it fixed during the estimation process.

4.1.7 The `varComb` Variance Structure

The last variance structure we discuss is the *combination of variance structures* using the `varComb` function. With this variance structure, we can allow for both an increase in residual spread for larger DML values as well as a different spread per month. This variance structure is of the form:

$$\text{var}(\varepsilon_{ij}) = \sigma_j^2 \times e^{2\delta \times DML_{ij}} \quad (4.10)$$

Note that σ has an index j running from 1 to 12, allowing for different spreads per month. Additionally, the variance increases for larger DML values. This is a combination of `varIdent` and `varExp`. The following R code applies this variance structure and gives the AIC of all models applied so far.

```
> vf8 <- varComb(varIdent(form =~ 1 | fMONTH) ,
                    varExp(form =~ DML) )
> M.gls8 <- gls(Testisweight ~ DML * fMONTH,
                    weights = vf8, data = Squid)
```

```
> anova(M.lm, M.gls1, M.gls2, M.gls3, M.gls4,
         M.gls5, M.gls6, M.gls7, M.gls8)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
M.lm	1	25	3752.084	3867.385	-1851.042			
M.gls1	2	25	3620.898	3736.199	-1785.449			
M.gls2	3	36	3614.436	3780.469	-1771.218	2 vs 3	28.461	0.0027
M.gls3	4	26	3473.019	3592.932	-1710.509	3 vs 4	121.417	<.0001
M.gls4	5	37	3407.511	3578.156	-1666.755	4 vs 5	87.507	<.0001
M.gls5	6	26	3478.152	3598.066	-1713.076	5 vs 6	92.641	<.0001
M.gls6	7	27	3475.019	3599.544	-1710.509	6 vs 7	5.133	0.0235
M.gls7	8	49	3431.511	3657.501	-1666.755	7 vs 8	87.507	<.0001
M.gls8	9	37	3414.817	3585.463	-1670.409	8 vs 9	7.306	0.8367

The model allowing for an increase in spread for larger DML values (which is allowed to differ per month), M.gls4, has the lowest AIC and is therefore selected as the optimal model. Note that the tests above depend on the order in the anova command. If you are only after the AIC, you better use the command:

```
> AIC(M.lm, M.gls1, M.gls2, M.gls3, M.gls4,
      M.gls5, M.gls6, M.gls7, M.gls8)
```

This command only gives the AICs of the models. The anova (M.gls4) command shows that the interaction is highly significant. Testing fixed terms in the model is further discussed in Section 4.2.

4.1.8 Overview of All Variance Structures

Table 4.1 shows all the applied variance structures and their names. As well as these functions, you can also specify your own variance structure; see pg. 214 in Pinheiro and Bates (2000). Instead of using a covariate in the variance structure, we can use the fitted values of the model, which allows the spread in residuals to increase (or decrease) for larger fitted values.

Table 4.1 Various variance structures used in this section. The table follows Pinheiro and Bates (2000)

Name of the function in R	What does it do?
VarFixed	Fixed variance
VarIdent	Different variances per stratum
VarPower	Power of the variance covariate
VarExp	Exponential of the variance covariate
VarConstPower	Constant plus power of the variance covariate
VarComb	A combination of variance functions

If the variance covariate has large values (e.g. larger than 100), numerical instabilities may occur; $\exp(100)$ is rather large! In such cases, it is better to rescale the variance covariate before using it in any of the variance structures. For example, we could have used DML/max(DML) or express it in meters instead of millimetres in the variance functions. The unscaled DML can still be used in the fixed part of the model.

Remember from Appendix A, that two models are called nested if one model can be obtained from the other model by setting specific parameters equal to zero. The same definition also applies to variance structures. For example, the variance structure of the linear regression model in Equation (4.1) is nested within most of the other models. However, one of the exceptions is the linear regression model and the varFixed structure.

In this case, we cannot obtain the homogeneous residual variance from the linear regression model by setting a specific parameter in the varFixed model equal to zero. To see this, compare the following two variance structures:

$$\varepsilon_i \sim N(0, \sigma^2) \quad \varepsilon_i \sim N(0, \sigma^2 \times DML_i)$$

The first variance structure is from the linear regression model and the second one from the varFixed. We cannot obtain the variance structure on the left from the right one, unless DML is equal to 1 for all observations. Compare this with the linear regression model and the varPower structure:

$$\varepsilon_i \sim N(0, \sigma^2) \quad \varepsilon_{ij} \sim N(0, \sigma^2 \times |DML_{ij}|^{2\delta_j})$$

By setting all δ_j s equal to zero in the right variance structure, we obtain the left variance structure; hence, these are nested variance structures. Note that the varIdent is nested in the varPower structure! And nested models mean that we can apply the likelihood ratio test.

To test certain types of heterogeneity, we can apply the log likelihood ratio test. For example, for model (4.1) and the optimal variance structure in (4.6), we can type anova (M.lm, M.gls4), which gives:

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
	M.lm	1	25	3752.084	3867.385	-1851.042		
	M.gls4	2	37	3407.511	3578.156	-1666.755	1 vs 2	368.5728 <.0001

The log likelihood ratio statistic is 368.57, indicating that the variance structure in (4.6) is considerably better than the constant variance in the linear regression model (4.1). Hence, the varPower option provides a significantly better variance structure than the one used for the linear regression model in (4.1). In a paper, you would write this as $L = 368.57$ ($df = 12, p < 0.001$). This model comparison provides a better testing procedure for homogeneity than those presented in Chapter 2.

4.1.8.1 Which One to Choose?

So, which variance structure should you choose, and how do you decide which one is best? If the variance covariate is a nominal variable, the choice is simple; use `varIdent`. In our example, it allowed modelling different residual variation for the testis weight per month.

The underlying variance structure imposed by `varIdent` is relatively easy to understand, but the difference between the variance structured modelled by the `varFixed`, `varPower`, `varExp`, and `varConstPower` functions are more difficult to explain. All four variance structures allow for an increase (or decrease) in residual variation for the testis weight data along a continuous variance covariate like DML (an explanatory variable in this case).

But, the `varFixed` is rather limited, as it assumes that the variance of the residuals is linearly related to a variance covariate. This causes problems if the variance covariate takes non-positive values or where the linear relationship requirements between variation and the variance covariate is too stringent.

In practise, it may be better to use the `varPower`, `varExp`, or `varConstPower` functions, which allow for more flexibility than the `varFixed`. So, how to choose between these three? The difference between them is the mathematical parameterisation of the variance function. The `varPower` should not be used if the variance covariate takes the value of zero. In this case, this is not an issue as DML (length) is always larger than zero. But it may be an issue with variance covariates like temperature or height compared to a baseline, etc.

However, finding the right variance structure for a variance covariate like DML, which is always non-zero, is more a matter of trial and error, and the best choice is judged through using tools like the AIC. Another important aspect is biological knowledge. If you know a priori that there is a certain type of heterogeneity in your data, then you can greatly speed up the selection process by including this information!

4.1.9 Graphical Validation of the Optimal Model

For graphical model validation, we can use two types of residuals: (i) residuals calculated as observed minus fitted values (also called ordinary residuals) and (ii) normalised residuals. We start with the first one. The following R code extracts the residuals and plots them in a coplot (Fig. 4.4). Note that these residuals still show heterogeneity, but this is now allowed (because the residual variation differs depending on the chosen variance structure and values of the variance covariate). Hence, these residuals are less useful for the model validation process.

```
> E1 <- resid(M.gls4)
> coplot(E1 ~ DML | fMONTH,
  ylab = "Ordinary residuals", data = Squid)
```

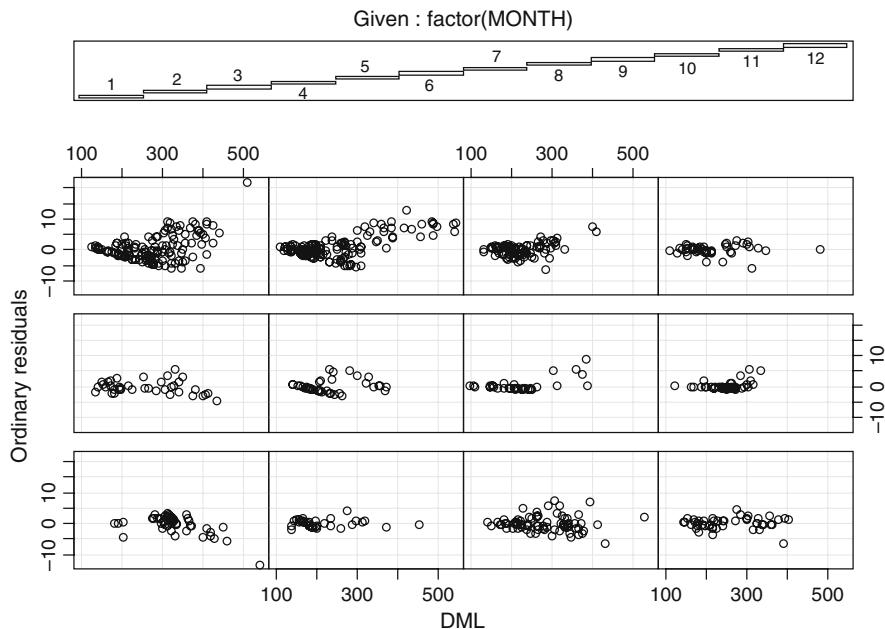


Fig. 4.4 Ordinary residuals (observed minus fitted values) versus DML conditional on month for the optimal model. These residuals are allowed to have a cone effect

You should use standardised residuals instead of the ordinary residuals for the model validation. These are obtained by calculating the observed minus the fitted values and then dividing by the square root of the variance. These residuals are therefore obtained from

$$\varepsilon_{ij} = \frac{\text{Testisweight}_{ij} - \text{Fitted values}_{ij}}{\sqrt{\sigma^2 \times |DML_{ij}|^{2\delta_j}}} \quad (4.11)$$

Plotting these residuals should not show any heterogeneity. If there is any heterogeneity, then further model improvement is required. Luckily, we don't have to program Equation (4.11) as the standardised residuals can be obtained using an R function.

The following R code extracts the standardised residuals, and makes a coplot, (Fig. 4.5) where there is no clear evidence of heterogeneity.

```
> E2 <- resid(M.gls4, type = "normalized")
> coplot(E2 ~ DML | fMONTH, data = Squid,
  ylab = "Normalised residuals")
```

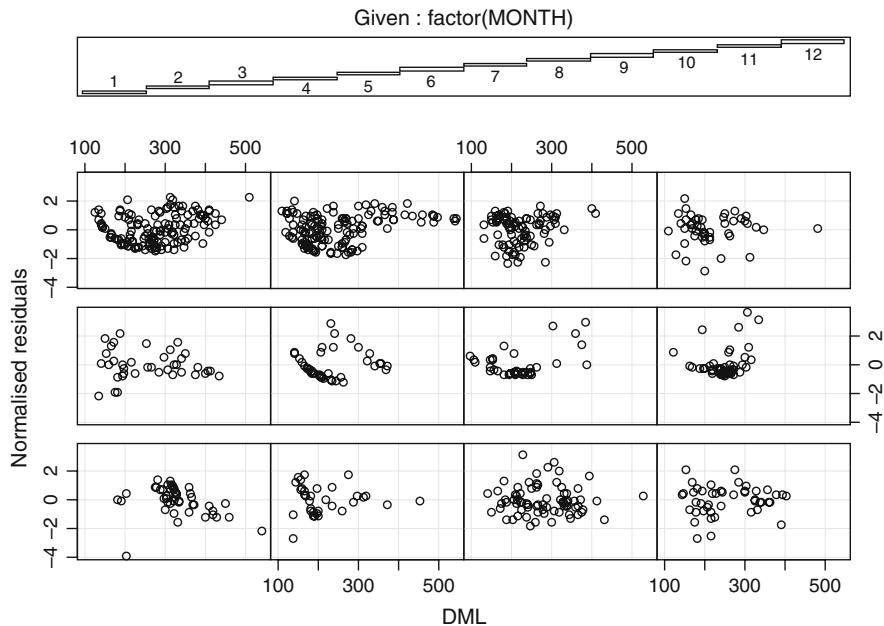


Fig. 4.5 Coplot of standardised residuals versus DML conditional on month for the optimal model. There is no evidence of heterogeneity

The option `type = "normalized"` ensures that E2 contains the standardised residuals.

4.2 Benthic Biodiversity Experiment

4.2.1 Linear Regression Applied on the Benthic Biodiversity Data

In this section, we provide another example of a linear regression model for statistically heterogeneous data. Based on experimental protocols developed in Emmerson and Raffaelli (2000), Emmerson et al. (2001), Solan and Ford (2003), and Ieno et al. (2006), among others, replicate mesocosm experiments (using plastic ice containers) were carried out. Benthic macrofaunal single and/or multiple species (biodiversity) were manipulated in a multi-patch environment, and the release of ammonium ($\text{NH}_4\text{-N}$), nitrate ($\text{NO}_x\text{-N}$) and phosphate ($\text{PO}_4\text{-P}$) concentrates were recorded from the sediment (ecosystem processes).

The data used for the specific example shown below relies on both published data (Ieno et al., 2006) and unpublished data (Oceanlab, University of Aberdeen). The experiment examines the effect of macrofauna density (*Hediste diversicolor*; Poly-chaeta), and habitat heterogeneity on sediment nutrient release. Figure 4.6 shows the experimental set up.



Fig. 4.6 Photograph showing the experimental set up. One nutrient is measured per container

At the start of the experiment, each container was filled with homogenised sediment from mudflats on the Ythan estuary (Scotland, UK). The macrofaunal biomass (*H. diversicolor*) was fixed across the following levels (0, 0.5, 1, 1.5, and 2 g), and replicated within each biomass level ($n = 3$). The response variable is the concentration of a particular nutrient.

To study the effect of habitat heterogeneity, the previous procedure was repeated for algae-enriched sediment. This gave 36 observations per nutrient, 18 enriched, and 18 non-enriched. Because there are three nutrients, the data set contains 108 samples (containers).

We can either analyse the data for each nutrient separately or combine all the data and analyse it all at the same time. The latter option is applied here as it allows us to test for interactions between nutrients and treatment levels (note that the nutrients were not measured in the same container; so there are no pseudo-replication problems).

To analyse the concentration data from all three nutrients, we need to concatenate the 36 observations from each nutrient, resulting in a response variable of length 108 (36×3), one continuous explanatory variable (biomass), and two nominal explanatory variables: enrichment (with or without algae), and a variable identifying the nutrient with the levels NH₄-N, NO₃-N, and PO₃-P.

There is, however, a major problem with the statistical analysis of the combined data. Due to the nature of the variables, we expect massive differences in variation in concentrations per nutrient and enrichment combination. This is illustrated in Fig. 4.7, which shows a boxplot for each nutrient–enrichment combination. Note

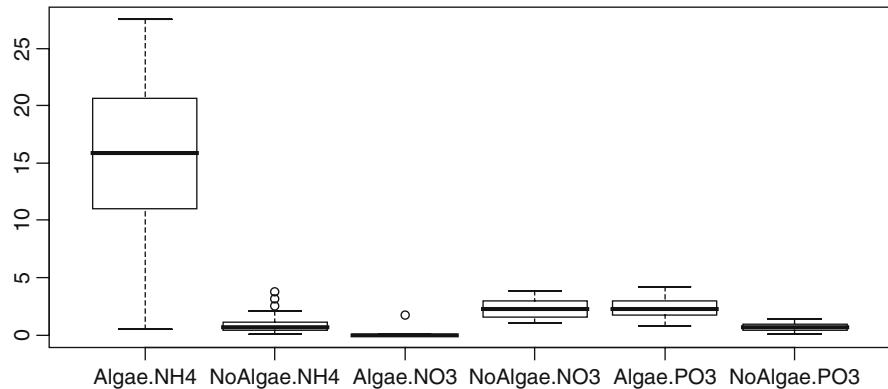


Fig. 4.7 Boxplot of concentrations for NH_4 , NO_3 and PO_3 . The first two boxplots on the left hand side are for enriched and non-enriched NH_4 concentrations

that the samples enriched with algae and with NH_4 , have higher concentrations and show more variation.

An initial linear regression analysis, using biomass, enrichment and nutrient, with all the two-way interactions, and the three-way interaction as explanatory variables clearly showed serious violation of homogeneity, as can be seen from Fig. 4.8.

A $\log_{10}(\text{Concentration} + 0.5)$ transformation was applied, but the enrichment \times NO_3 combination still had lower variation than the other combinations. We, therefore, cannot easily obtain homogeneity with a data transformation. And, as we mentioned in Chapter 2, we want to avoid data transformations whenever possible. So, instead of transforming the data, we will allow for different variances by using GLS.

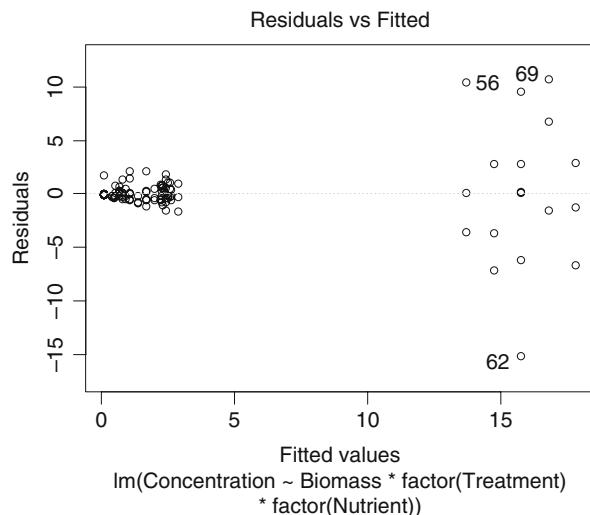


Fig. 4.8 Residuals versus fitted values for the linear regression model. Note the difference in spread
 $\text{Im}(\text{Concentration} \sim \text{Biomass} * \text{factor(Treatment)} * \text{factor(Nutrient)})$

The following R code was used to make Figs. 4.7 and 4.8.

```
> library(AED); data(Biodiversity);
> Biodiv <- Biodiversity #Saves some space
> Biodiv$fTreatment <- factor(Biodiv$Treatment)
> Biodiv$fNutrient <- factor(Biodiv$Nutrient)
> boxplot(Concentration ~
+   fTreatment * fNutrient, data = Biodiv)
> M0 <- lm(Concentration ~
+   Biomass * fTreatment * fNutrient,
+   data = Biodiv)
> plot(M0, which = c(1), add.smooth = FALSE)
```

The library and data commands are used to load the data. The variables Treatment and Nutrient are converted into factors, and the rest is basic code for a boxplot (Chapter 2) and linear regression (Appendix A).

4.2.2 GLS Applied on the Benthic Biodiversity Data

As with the squid data, we have to investigate why there is heterogeneity in these benthos data. Biological knowledge suggests that treatment and nutrient levels, possibly both, may be driving the heterogeneity. A scatterplot of biomass versus concentration did not show any clear increase or decrease in spread. This indicates that the potential variance covariates are nutrient and/or enrichment. The following R code assumes you have already loaded the data. It first applies the linear regression model again with the gls command, and then the three GLS models with different variance covariates are fitted.

```
> library(nlme)
> f1 <- formula(Concentration ~ Biomass * fTreatment *
+   fNutrient)
> M0 <- gls(f1, data = Biodiv)
> M1A <- gls(f1, data = Biodiv, weights = varIdent(
+   form =~ 1 | fTreatment * fNutrient))
> M1B <- gls(f1, data = Biodiv,
+   weights = varIdent(form =~ 1 | fNutrient))
> M1C <- gls(f1, data = Biodiv,
+   weights = varIdent(form =~ 1 | fTreatment))
```

The first model M0 is the linear regression model without any variance covariates. The second model M1A uses one variance term per nutrient-enrichment combination. And the third and fourth models use as variance covariates, nutrient and enrichment, respectively. The models have all main terms, two-way interactions, and the three-way interaction term as a fixed component. The anova command can be used to compare the models.

```
> anova(M0, M1A, M1B, M1C)
```

Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
M0	1	13	534.5203	567.8569	-254.2602		
M1A	2	18	330.1298	376.2881	-147.0649	1 vs 2	214.39054 <.0001
M1B	3	15	380.0830	418.5482	-175.0415	2 vs 3	55.95320 <.0001
M1C	4	14	439.7639	475.6647	-205.8819	3 vs 4	61.68087 <.0001

The AIC of the model with both nutrient and enrichment as variance covariates (M1A) is by far the best model, as judged by the AIC and BIC. Note that not all the likelihood ratio tests make sense (not all comparisons are from nested models). The `plot(M1A, col = 1)` command plots the standardised residuals versus fitted values. The graph is not shown here, but there is no sign of heterogeneity.

The commands `anova(M1A)` and `summary(M1A)` give information of the significance of the fixed explanatory variables (the three-way interaction, etc.). Results are not given here, but both functions show that the three-way interaction is not significant.

4.2.3 A Protocol

The problem is that we have still not discussed all aspects of model selection. This requires knowledge of things like maximum likelihood (ML) estimation and restricted maximum likelihood estimation (REML), and we discuss these in more detail in the next chapter. For the moment, we present them in a rather abstract manner and justify them later in Chapter 5. So to fully understand the differences between ML and REML, you need to read Chapter 5. In Chapter 5, the protocol for model selection in mixed modelling is explained (and justified) in detail, but the same protocol applies for GLS and is introduced in less detail below.

1. Start with a linear regression model that contains as many explanatory variables and their interactions as possible. The residuals of this model are assumed to be normally distributed with mean 0 and variance σ^2 . Investigate whether the homogeneity assumptions are valid by plotting the standardised residuals versus fitted values and by plotting the standardised residuals versus each individual explanatory variable. Any sign of variation in residual patterns is an indication of heterogeneity and means you have to go on to step 2. If you do not see any clear violation of homogeneity, there is no need to continue to step 2; just continue with a model selection on the explanatory variables (Appendix A). It should be noted that the graphical assessment of heterogeneity is difficult for small data sets.
2. For formal model comparison, repeat step 1 using the `gls` function from the `nlme` package. Do not specify any special variance structure yet and ensure that REML estimation is used (the default estimation method). You will get exactly the same estimated values, *t*-values and *p*-values as in step 1. The reason for

this step is that the `anova` command cannot compare objects obtained by the functions `lm` and `gls`. A call to the `gls` function without any extra options is a linear regression.

3. Depending on the graphical model validation in step 1, choose an appropriate variance structure. It helps to plot residuals versus fitted values and use different colours and symbols for different nutrients and/or enrichment levels (for our particular example). In the previous section, a wide range of residual variance structures was introduced.
4. Fit a new `gls` model with the selected variance covariance structure selected in step 3. Ensure that REML estimation is used, which is done with `gls(..., method = "REML")`, and that you use the same selection of explanatory variables. This is now called the fixed part of the model, and the residuals are called the random part. We will first try to find the optimal random structure using as many explanatory variables in the fixed part as possible.
5. Compare the new GLS model with the earlier results using the AIC, BIC, or likelihood ratio test. If the new model is better, extract the normalised residuals, and inspect these for homogeneity (using the same tools as in step 1). If the homogeneity assumption is not valid for the normalised residual of the model obtained in step 4, then go to step 6. If it is valid, then go to step 7.
6. If the residuals still show heterogeneity, go to step 4, and choose another residual variance structure. If you keep iterating between steps 4, 5, and 6, either try improving the fixed component (using for example additive modelling), try a different distribution (e.g. Poisson or negative binomial), consider a transformation on the response variable as a last resort, or conclude that your residual spread is not related to any of the measured covariates.
7. You are now half way. You have found the optimal residual variance structure using REML estimation. Now it is time to find the optimal fixed component. Or stated differently, which explanatory variables are significant, and which are not. You have three tools to find the optimal fixed component: the *t*-statistic, the *F*-statistic, and the likelihood ratio test. The *t*-statistics are obtained with the `summary` command, and the *F*-statistic with the `anova` command. Both functions are applied on one model, e.g. by typing `summary(M1A)` or `anova(M1A)`. Ensure that REML estimation is used in the `gls` command. Remember the `anova` command is doing sequential testing. This is useful for testing the significance of the highest interaction term, but not for the other terms in the model. It is also of less use if you only have main terms as the order of the variables is of importance in sequential testing. The problem with the *t*-statistic is that it should not be used to assess the significance of a nominal variable with more than two levels (e.g. nutrient). The third option is the likelihood ratio test. You need to specify a full model and a nested model (Appendix A). Both models need ML estimation (and the same random structure, but you already selected these in step 5). This approach is conceptually probably the easiest to work with, but it can be time consuming.

8. Apply any of the model selection tools described in step 7, and stop once all terms are significant.
9. Reapply the model that was found in step 8, and refit it with REML estimation. Apply a graphical model validation, checking for homogeneity (see step 1), normality, and independence. If no problems are highlighted, go to step 10. If problems are identified, return to step 8, and consider adding non-significant terms to see if this improves the model validation graphs.
10. Present the results in a table and try to understand what it all means in terms of ecology.

We demonstrated steps 1–7 for the benthic biodiversity data earlier in this chapter and now continue with this example for the remaining steps in the protocol just described.

4.2.4 Application of the Protocol on the Benthic Biodiversity Data

The `anova(M1A)` command gives the following output.

Denom. DF: 96

	numDF	F-value	p-value
(Intercept)	1	205.73781	<.0001
Biomass	1	1.22179	0.2718
fTreatment	1	14.62895	0.0002
fNutrient	2	1.57754	0.2118
Biomass:fTreatment	1	0.26657	0.6068
Biomass:fNutrient	2	4.17802	0.0182
fTreatment:fNutrient	2	121.57149	<.0001
Biomass:fTreatment:fNutrient	2	1.09043	0.3402

An explanation of the nominator and denominator degrees of freedom is delayed until Chapter 5. Here, we focus on the value of the *F*-statistic and its *p*-value. The `anova` function applies sequential testing. This means that the *p*-values will change if you change the order of the main terms or the order of the two-way interactions. In this example, it is only the last term that is of real interest as it shows the significance of the three-way interaction term (you can't change the order of this term). In this case, it is not significant at the 5% level. This means that we can drop the three-way term and refit the model.

Refitting the model with the main terms and all three two-way terms gives exactly the same `anova` table as above, except for the last line. The problem is that we cannot assess the significance of the Biomass × Treatment term, and the biomass × Nutrient term, due to the order how they were put in. Obviously, we could apply three models, ensure each time that a different two-way term is the last, and deselect the least significant two-way interaction.

The second model selection approach (using hypothesis testing) is based on the *t*-statistic, but we do not want to use this option as nutrient has three levels. One level will be used as baseline, and the *p*-values from the *t*-statistic will only tell us whether the second and third nutrients are different from the baseline nutrient.

The third model selection approach (using hypothesis testing) is based on comparing nested models. Let us go back a step and test the significance of the three-way interaction term again. We compare the full model (with the three-way interaction term) with a model that does not contain the three-way interaction term using the likelihood ratio test. Both models need ML estimation. The R code for this is as follows.

```
> M2A1 <- gls(Concentration ~ Biomass + fTreatment +
  fNutrient +
  Biomass:fTreatment +
  Biomass:fNutrient +
  fTreatment:fNutrient +
  Biomass:fTreatment:fNutrient,
  weights = varIdent(form =~ 1 |
    fTreatment * fNutrient),
  method = "ML", data = Biodiv)

> M2A2 <- gls(Concentration ~ Biomass + fTreatment +
  Nutrient +
  Biomass:fTreatment +
  Biomass:fNutrient +
  fTreatment:fNutrient,
  weights=varIdent(form =~ 1 |
    fTreatment * fNutrient),
  method = "ML", data = Biodiv)
```

The output of the anova (M2A1, M2A2) command is given below.

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
M2A1		1	18	321.0648	369.3432	-142.5324		
M2A2		2	16	319.4653	362.3794	-143.7327	1 vs 2	2.400507 0.3011

The anova command also indicates that the three-way interaction can be dropped. In the next step of the model selection, we have to find a *p*-value for each two-way interaction. This is done as follows. Use model M2A2 as the starting point and drop each of the two-way interactions in turn, and use the anova command to obtain a *p*-value. Also consider whether any of the main terms can be dropped. The rule is that if an interaction term is included, then all the associated main terms should be included as well, and are not a candidate for dropping. However, if you have the main terms A, B, C, and the interaction A × B, then the two terms that can be potentially dropped are A × B and also C!

This whole process is rather time consuming and you will want to think twice before adding four-way interactions! It was our intention to put the code for this example online, but all our book reviewers asked us to include it in the text of the

book. Perhaps they are right, and you should see this at least once in your life. So, take a deep breath, and read on!

4.2.4.1 Round 1 of the Backwards Selection

The following code drops each two-way interaction and applies a likelihood ratio test.

```
> vfOptim <- varIdent(form =~ 1 | fTreatment*fNutrient)
> #Assess significance of all 3 2-way interactions
> #Full model
> M3.Full <- gls(Concentration ~
+ Biomass + fTreatment + fNutrient +
+ Biomass:fTreatment +
+ Biomass:fNutrient +
+ fTreatment:fNutrient,
+ weights = vfOptim,
+ method = "ML", data = Biodiv)
> #Drop Biomass:fTreatment
> M3.Drop1 <- gls(Concentration ~
+ Biomass + fTreatment + fNutrient +
+ Biomass:fNutrient +
+ fTreatment:fNutrient,
+ weights = vfOptim,
+ method = "ML", data = Biodiv)
> anova(M3.Full, M3.Drop1)

      Model df      AIC      BIC  logLik   Test L.Ratio p-value
M3.Full     1 16 319.4653 362.3794 -143.7327
M3.Drop1    2 15 319.3730 359.6050 -144.6865 1 vs 2 1.907680 0.1672

>
> #Drop Biomass:fNutrient
> M3.Drop2 <- gls(Concentration ~
+ Biomass + fTreatment + fNutrient +
+ Biomass:fTreatment +
+ fTreatment:fNutrient,
+ weights = vfOptim,
+ method = "ML", data = Biodiv)
> anova(M3.Full, M3.Drop2)

      Model df      AIC      BIC  logLik   Test L.Ratio p-value
M3.Full     1 16 319.4653 362.3794 -143.7327
M3.Drop2    2 14 323.2165 360.7664 -147.6083 1 vs 2 7.751179 0.0207

>
> #Drop fTreatment:fNutrient
> M3.Drop3 <- gls(Concentration ~
+ Biomass + fTreatment + fNutrient +
+ Biomass:fTreatment +
+ Biomass:fNutrient,
+ weights = vfOptim,
+ method = "ML", data = Biodiv)
```

```
> anova(M3.Full, M3.Drop3)
      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full     1 16 319.4653 362.3794 -143.7327
M3.Drop3    2 14 403.3288 440.8786 -187.6644 1 vs 2 87.86346 <.0001
```

So, we dropped each two-way interaction term in turn, applied the likelihood ratio test, and obtained *p*-values. Clearly, the two way interaction term Biomass:fTreatment is not significant at the 5% level and should be dropped. You can make the code above a bit friendlier using the update command. The following code produces exactly the same results.

```
> #Alternative coding with same results
> fFull <- formula(Concentration~
+ Biomass + fTreatment + fNutrient +
+ Biomass:fTreatment +
+ Biomass:fNutrient + fTreatment:fNutrient)
> M3.Full <- gls(fFull, weights = vfOptim,
+ method = "ML", data = Biodiv)

> #Drop Biomass:fTreatment
> M3.Drop1<-update(M3.Full, .~. - Biomass:fTreatment)
> anova(M3.Full, M3.Drop1)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full     1 16 319.4653 362.3794 -143.7327
M3.Drop1    2 15 319.3730 359.6050 -144.6865 1 vs 2 1.907680  0.1672

> #Drop Biomass:fNutrient
> M3.Drop2 <- update(M3.Full, .~. - Biomass:fNutrient)
> anova(M3.Full, M3.Drop2)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full     1 16 319.4653 362.3794 -143.7327
M3.Drop2    2 14 323.2165 360.7664 -147.6083 1 vs 2 7.751179  0.0207

> #Drop fTreatment:fNutrient
> M3.Drop3<-update(M3.Full, .~. - fTreatment:fNutrient)
> anova(M3.Full,M3.Drop3)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full     1 16 319.4653 362.3794 -143.7327
M3.Drop3    2 14 403.3288 440.8786 -187.6644 1 vs 2 87.86346 <.0001
```

As you can see, this gives the same results. The advantage of the update command is that the code is shorter, but you it also makes it easier to lose track what exactly you are fitting.

4.2.4.2 Round 2 of the Backwards Selection

Whichever coding you use, we need to drop the term Biomass:fTreatment. This means that the new full model is

```
> #New full model
> M4.Full <- gls(Concentration~
```

```
Biomass + fTreatment + fNutrient +
Biomass:fNutrient + fTreatment:fNutrient,
weights = vfOptim,
method = "ML", data = Biodiv)
```

From this model, you can drop two of the two-way interaction terms. No main terms can be dropped yet. We will use the `update` command again and try to avoid turning this chapter into something that looks like a telephone book.

```
> #Drop Biomass:fNutrient
> M4.Drop1 <- update(M4.Full, .~. -Biomass:fNutrient)
> anova(M4.Full, M4.Drop1)

      Model df     AIC     BIC   logLik   Test L.Ratio p-value
M4.Full     1 15 319.3730 359.6050 -144.6865
M4.Drop1    2 13 321.7872 356.6549 -147.8936 1 vs 2 6.414148  0.0405

> #Drop fTreatment:fNutrient
> M4.Drop2<-update(M4.Full, .~. -fTreatment:fNutrient)
> anova(M4.Full, M4.Drop2)

      Model df     AIC     BIC   logLik   Test L.Ratio p-value
M4.Full     1 15 319.3730 359.6050 -144.6865
M4.Drop2    2 13 404.8657 439.7335 -189.4329 1 vs 2 89.49272 <.0001
```

A *p*-value of 0.04 for the `Biomass:fNutrient` interaction is not impressive, especially not with a series of hypothesis tests. So, we decided to drop it as well and continue with the following full model.

4.2.4.3 Round 3 of the Backwards Selection

The new full model is

```
> #New full model
> M5.Full <- gls(Concentration ~
+ Biomass + fTreatment + fNutrient +
+ fTreatment:fNutrient,
+ weights = vfOptim, method = "ML",
+ data = Biodiv)
```

We can drop the `fTreatment:fNutrient` interaction term, but also the main term `Biomass`.

```
> #Drop fTreatment:fNutrient
> M5.Drop1 <- update(M5.Full, .~.-fTreatment:fNutrient)
> anova(M5.Full, M5.Drop1)

      Model df     AIC     BIC   logLik   Test L.Ratio p-value
M5.Full     1 13 321.7872 356.6549 -147.8936
M5.Drop1    2 11 406.7950 436.2985 -192.3975 1 vs 2 89.00786 <.0001
> #Drop Biomass
```

```
> M5.Drop2 <- update(M5.Full, .~. -Biomass)
> anova(M5.Full, M5.Drop2)

Model df      AIC      BIC    logLik   Test L.Ratio p-value
M5.Full     1 13 321.7872 356.6549 -147.8936
M5.Drop2    2 12 321.2595 353.4450 -148.6297 1 vs 2 1.472279  0.225
```

The biomass term is not significant and can be dropped.

4.2.4.4 Round 4 of the Backwards Selection

The new full model is

```
> M6.Full<-gls(Concentration ~ fTreatment + fNutrient+
+ fTreatment:fNutrient,
+ weights = vfOptim, method = "ML",
+ data = Biodiv)
```

The only term that can be dropped is the interaction term.

```
> M6.Drop1<-update(M6.Full, .~. -fTreatment:fNutrient)
> anova(M6.Full, M6.Drop1)

Model df      AIC      BIC    logLik   Test L.Ratio p-value
M6.Full     1 12 321.2595 353.4450 -148.6297
M6.Drop1    2 10 406.0323 432.8536 -193.0161 1 vs 2 88.77283 <.0001
```

The interaction term `fTreatment:fNutrient` is highly significant, so no further terms can be dropped.

4.2.4.5 The Aftermath

We applied the process of comparing nested models several times, and ended up with a model containing Nutrient, Enrichment, and their interaction. The two-way interaction term was significant. We reapplied this model with REML estimation (step 9). Normality and homogeneity can safely be assumed (see Fig. 4.9). Figure 4.9 was created with the following R code.

```
> MFinal <- gls(Concentration ~ fTreatment * fNutrient,
+                 weights = vfOptim, method = "REML",
+                 data = Biodiv)
> E <- resid(MFinal, type = "normalized")
> Fit <- fitted(MFinal)
> op <- par(mfrow = c(1, 2))
> plot(x = Fit, y = E,
+       xlab = "Fitted values", ylab = "Residuals",
+       main = "Residuals versus fitted values")
> identify(Fit, E)
> hist(E, nclass = 15)
> par(op)
```

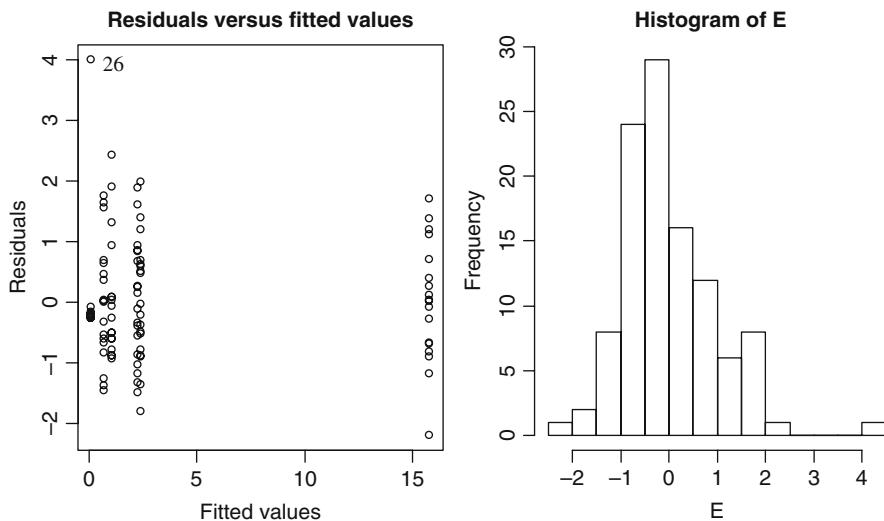


Fig. 4.9 Residuals versus fitted values and a histogram of the residuals (denoted by E) for the optimal GLS model that contains Nutrient, Enrichment, and their interaction

The `gls` command refits the model with REML, the `resid` command extracts the normalised residuals, the object `Fit` are the fitted values, the `plot` command plots the fitted values versus the residuals, and the `hist` command makes a histogram with 15 bars. The `identify` command allows us to identify the observation with the large residual (observation 26). We will return to this observation in a moment.

Assuming that everything is ok, we can now proceed to step 10 and present the relevant output of the final model using the `summary(MFinal)` command.

```
fTreatmentNoAlgae:fNutrientNO3 16.86929 1.663956 10.138067      0
fTreatmentNoAlgae:fNutrientPO3 12.95293 1.666324  7.773353      0

Residual standard error: 0.8195605
Degrees of freedom: 108 total; 102 residual
```

The AIC and BIC are model selection tools, and there is little to say about them at this point as we have passed the model selection stage. The information on the different standard deviations (multiplication factors of σ) is interesting, as it shows the different variances (or better: the ratio with the standard error) per treatment–nutrient combination. The estimated value for σ is 0.819. Note that the combination enrichment with algae and NH₄ has the largest variance, namely $(8.43 \times 0.819)^2$.

The estimated regression parameters, standard errors, *t*-values, *p*-values, and other relevant information are given as well. Note that all terms are significantly different from 0 at the 5% level. To understand what the model is trying to tell us, it can be helpful to consider a couple of scenarios and obtain the equations for the fitted values or just graph the fit of the model. The easiest way of doing this is

```
> boxplot(predict(MFinal) ~ fTreatment * fNutrient,
           data = Biodiv)
```

This only works because all the explanatory variables are nominal. The resulting graph is shown in Fig. 4.10 and clearly shows that the observations exposed to algae treatment and NH₄ enrichment have the highest values. This explains why the interaction term is significant. Unfortunately, at the time of writing, the *predict.gls* function (which is the one used to obtain the predicted values) does not give standard errors for predicted values. To obtain the 95% confidence bands around the fitted values, you need to use equations similar to those used for linear regression

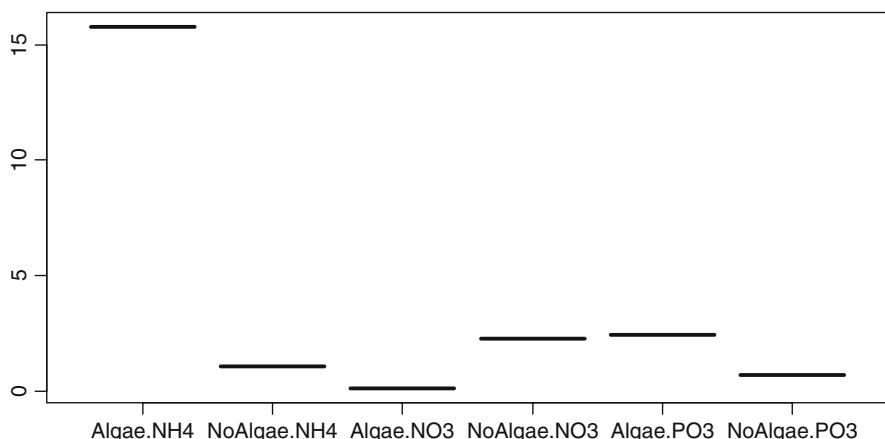


Fig. 4.10 Fitted values for the optimal model. Note the high values for the algae–NH₄ combination

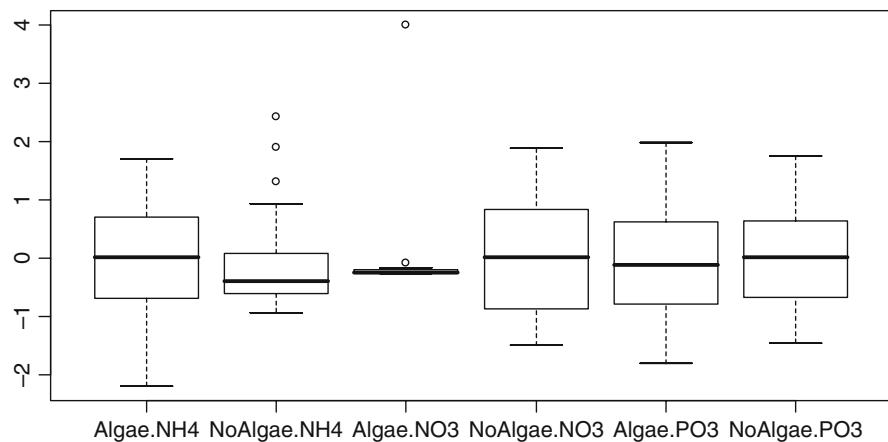


Fig. 4.11 Normalised residuals versus treatment–nutrient combination. Note the effect of the outlier for the algae–NO₃ combination. This is observation 26

(Appendix A), but this requires some ugly R programming. Alternatively, you can do some bootstrapping.

Before you happily write your paper using these results, there is one final point you should know. Figure 4.11 shows a boxplot of normalised residuals versus the treatment–nutrient combination. Note the effect of observation 26! We suggest that you repeat the entire analysis without this observation. If this was an email, we would now add a ☺ as this obviously means a lot of extra work!. You will need to remove row 26 from the data, or add `subset = -26` to each `gls` command. The first option is a bit clumsy, but avoids any potential error messages in the validation graphs (due to different data sizes).

Chapter 5

Mixed Effects Modelling for Nested Data

In this chapter, we continue with Gaussian linear and additive mixed modelling methods and discuss their application on nested data. Nested data is also referred to as hierarchical data or multilevel data in other scientific fields (Snijders and Boskers, 1999; Raudenbush and Bryk, 2002).

In the first section of this chapter, we give an outline to mixed effects models for nested data before moving on to a formal introduction in the second section. Several different types of mixed effects models are presented, followed by a section discussing the induced correlation structure between observations. Maximum likelihood and restricted maximum likelihood estimation methods are discussed in Section 5.6. The material presented in Section 5.6 is more technical, and you need only skim through it if you are not interested in the mathematical details. Model selection and model validation tools are presented in Sections 5.7, 5.8, and 5.9. A detailed example is presented in Section 5.10.

5.1 Introduction

Zuur et al. (2007) used marine benthic data from nine inter-tidal areas along the Dutch coast. The data were collected by the Dutch institute RIKZ in the summer of 2002. In each inter-tidal area (denoted by ‘beach’), five samples were taken, and the macro-fauna and abiotic variables were measured. Zuur et al. (2007) used species richness (the number of different species) and NAP (the height of a sampling station compared to mean tidal level) from these data to illustrate statistical methods like linear regression and mixed effects modelling. Here, we use the same data, but from a slightly different pedagogical angle. Mixed modelling may not be the optimal statistical technique to analyse these data, but it is a useful data set for our purposes. It is relatively small, and it shows all the characteristics of a data set that needs a mixed effects modelling approach.

The underlying question for these data is whether there is a relationship between species richness, exposure, and NAP. Exposure is an index composed of the following elements: wave action, length of the surf zone, slope, grain size, and the depth of the anaerobic layer.

As species richness is a count (number of different species), a generalised linear model (GLM) with a Poisson distribution may be appropriate. However, we want to keep things simple for now; so we begin with a linear regression model with the Gaussian distribution and leave using Poisson GLMs until later. A first candidate model for the data is

$$R_{ij} = \alpha + \beta_1 \times NAP_{ij} + \beta_2 \times Exposure_i + \varepsilon_{ij} \quad \varepsilon_{ij} \sim N(0, \sigma^2) \quad (5.1)$$

R_{ij} is the species richness at site j on beach i , NAP_{ij} the corresponding NAP value, $Exposure_i$ the exposure on beach i , and ε_{ij} the unexplained information. Indeed, this is the familiar linear regression model. The explanatory variable *Exposure* is nominal and has two¹ classes. However, as we have five sites per beach, the richness values at these five sites are likely to be more related to each other than to the richness values from sites on different beaches. The linear regression model does not take this relatedness into account. The nested structure of the data is visualised in Fig. 5.1.

Many books introduce mixed effects modelling by first presenting an easy to understand technique called 2-stage analysis, conclude that it is not optimal, and then present the underlying model for mixed effects modelling by combining the 2 stages into a single model (e.g. Fitzmaurice et al., 2004). This is a useful way to introduce mixed effects modelling, and we also start with the 2-stage analysis method before moving onto mixed effects modelling.

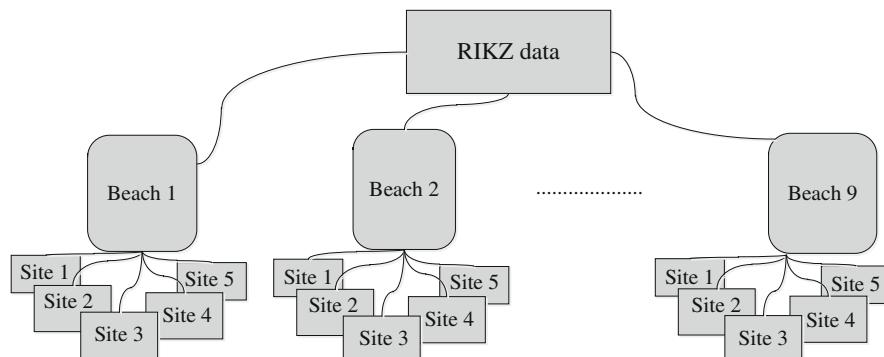


Fig. 5.1 Set up of the RIKZ data. Measurements were taken on 9 beaches, and on each beach 5 sites were sampled. Richness values at sites on the same beach are likely to be more similar to each other than to values from different beaches

¹Originally, this variable had three classes, but because the lowest level was only observed on one beach, we relabeled, and grouped the two lowest levels into one level called ‘a’. The highest level is labeled ‘b’.

5.2 2-Stage Analysis Method

In the first step of the 2-stage analysis method, a linear regression model is applied on data of one beach. It models the relationship between species richness and NAP on each beach using

$$R_{ij} = \alpha + \beta_i \times NAP_{ij} + \varepsilon_{ij} \quad j = 1, \dots, 5 \quad (5.2)$$

This process is then carried out for data of each beach in turn. In a more abstract matrix notation, we can write the model for the data of beach i as

$$\begin{pmatrix} R_{i1} \\ R_{i2} \\ R_{i3} \\ R_{i4} \\ R_{i5} \end{pmatrix} = \begin{pmatrix} 1 & NAP_{i1} \\ 1 & NAP_{i1} \\ 1 & NAP_{i1} \\ 1 & NAP_{i1} \\ 1 & NAP_{i1} \end{pmatrix} \times \begin{pmatrix} \alpha \\ \beta_i \end{pmatrix} + \begin{pmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \varepsilon_{i3} \\ \varepsilon_{i4} \\ \varepsilon_{i5} \end{pmatrix} \Leftrightarrow \mathbf{R}_i = \mathbf{Z}_i \times \boldsymbol{\beta}_i + \boldsymbol{\varepsilon}_i \quad (5.3)$$

\mathbf{R}_i is now a vector of length 5 containing the species richness values of the 5 sites on beach i : R_{i1} to R_{i5} . The first column of \mathbf{Z}_i contains ones and models the intercept, and the second column contains the five NAP values on beach i . The unknown vector $\boldsymbol{\beta}_i$ contains the regression parameters (intercept and slope) for beach i . This general matrix notation allows for different numbers of observations per beach as the dimension of \mathbf{R}_i , \mathbf{Z}_i , and $\boldsymbol{\varepsilon}_i$ can easily be adjusted. For example, if beach $i = 2$ has 4 observations instead of 5, \mathbf{Z}_2 contains 4 rows and 2 columns, but we still obtain an estimate for the intercept and slope. In this case, Equation (5.3) takes the form

$$\begin{pmatrix} R_{i1} \\ R_{i2} \\ R_{i3} \\ R_{i5} \end{pmatrix} = \begin{pmatrix} 1 & NAP_{i1} \\ 1 & NAP_{i2} \\ 1 & NAP_{i3} \\ 1 & NAP_{i5} \end{pmatrix} \times \begin{pmatrix} \alpha \\ \beta_i \end{pmatrix} + \begin{pmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \varepsilon_{i3} \\ \varepsilon_{i5} \end{pmatrix}$$

The model in Equation (5.3) is applied on data of each beach, resulting in nine estimated values for the slope and intercept. The following loop gives the results in the R software.

```
> library(AED); data(RIKZ)
> Beta <- vector(length = 9)
> for (i in 1:9){
  Mi <- summary(lm(Richness ~ NAP,
                     subset = (Beach==i), data=RIKZ))
  Beta[i] <- Mi$coefficients[2, 1]}
```

The subset option in the linear regression function `lm` ensures that data from each beach are analysed in a particular iteration of the loop. The last line in the loop

extracts and stores the slope for NAP for each regression analysis. The estimated betas can be obtained by typing `Beta` in R:

```
-0.37 -4.17 -1.75 -1.24 -8.90 -1.38 -1.51 -1.89 -2.96
```

Note that there are considerable differences in the nine estimated slopes for NAP. Instead of the loop in the code above, you can also use the `lmList` command from the `nlme` package to produce the same results. This option also gives a nice graphical presentation of estimated intercepts and slopes (Pinheiro and Bates, 2000).

In the second step, the estimated regression coefficients are modelled as a function of exposure.

$$\hat{\beta}_i = \eta + \tau \times \text{Exposure}_i + b_i \quad i = 1, \dots, 9 \quad (5.4)$$

This is ‘just’ a one-way ANOVA. The response variable is the estimated slopes from step 1, Exposure is the (nominal) explanatory variable, τ is the corresponding regression parameter, η is the intercept, and b_i is random noise. The matrix notation for this is below. It looks intimidating, but this is only because exposure is a factor with levels 0 and 1. Level 0 is used as the baseline. The model in Equation (5.4) is written in matrix notation as

$$\begin{pmatrix} -0.37 \\ -4.17 \\ -1.75 \\ -1.24 \\ -8.90 \\ -1.38 \\ -1.51 \\ -1.89 \\ -2.96 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} \eta \\ \tau \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \end{pmatrix} \Leftrightarrow \hat{\beta}_i = \mathbf{K}_i \times \boldsymbol{\gamma} + \mathbf{b}_i \quad i = 1, \dots, 9 \quad (5.5)$$

The vector $\boldsymbol{\gamma}$ contains the intercept η and slope τ and is not the same thing as $\boldsymbol{\beta}_i$. The following R code was used to apply this model.

```
> fExposure9 <- factor(c(0, 0, 1, 1, 0, 1, 1, 0, 0))
> tmp2 <- lm(Beta ~ fExposure9)
```

As we already mentioned, this linear regression model is also called a one-way analysis of variance (ANOVA). The results of the `anova` command are not presented here, but it shows that the p -value for exposure is 0.22, indicating that there is no significant exposure effect on the nine slopes.

The two formulae of the 2-stage approach are repeated in Equation (5.6).

$$\begin{aligned} \mathbf{R}_i &= \mathbf{Z}_i \times \boldsymbol{\beta}_i + \boldsymbol{\varepsilon}_i \\ \hat{\boldsymbol{\beta}}_i &= \mathbf{K}_i \times \boldsymbol{\gamma} + \mathbf{b}_i \end{aligned} \quad (5.6)$$

It is common to assume that the residuals \mathbf{b}_i are normally distributed with mean 0 and variance \mathbf{D} . The second step of the two-stage analysis can be seen as an analysis of a summary statistic; in this case, it is the slope representing the strength of the relationship between species richness and NAP on a beach. The two-stage analysis has various disadvantages. Firstly, we summarise all the data from a beach with one parameter. Secondly, in the second step, we analyse regression parameters, not the observed data. Hence, we are not modelling the variable of interest directly. Finally, the number of observations used to calculate the summary statistic is not used in the second step. In this case, we had five observations for each beach. But if you have 5, 50, or 50,000 observations, you still end up with only one summary statistic.

5.3 The Linear Mixed Effects Model

5.3.1 Introduction

The linear mixed effects model combines both the earlier steps into a single model.

$$\mathbf{R}_i = \mathbf{X}_i \times \boldsymbol{\beta} + \mathbf{Z}_i \times \mathbf{b}_i + \boldsymbol{\varepsilon}_i \quad (5.7)$$

As before, \mathbf{R}_i contains the richness values for beach i , $i = 1, \dots, 9$. There are two components in this model that contain explanatory variables; the fixed $\mathbf{X}_i \times \boldsymbol{\beta}$ term and the random $\mathbf{Z}_i \times \mathbf{b}_i$ term. Because we have a fixed and a random component, we call the model a *mixed* effects model. We discuss later how to fill in the \mathbf{X}_i and \mathbf{Z}_i . In this case, the $\mathbf{Z}_i \times \mathbf{b}_i$ component represents the Richness–NAP effect for each beach; each beach is allowed to have a different Richness–NAP relationship because there is an index i attached to \mathbf{b} . There is no index attached to the parameter $\boldsymbol{\beta}$; hence, it is for all beaches. \mathbf{X}_i and \mathbf{Z}_i are design matrices of dimension $n_i \times p$ and $n_i \times q$, respectively, where n_i is the number of observations in \mathbf{R}_i (the number of observations per beach), p the number of explanatory variables in \mathbf{X}_i , and q the number of explanatory variables in \mathbf{Z}_i .

Many textbooks on linear mixed effects modelling are orientated towards medical science, where i is typically denoted as ‘subject’ because it represents a patient or person. The component $\mathbf{Z}_i \times \mathbf{b}_i$ is then the subject specific or random effect and $\mathbf{X}_i \times \boldsymbol{\beta}$ the overall or fixed component. The matrices \mathbf{X}_i and \mathbf{Z}_i may, or may not, contain the same explanatory variables. This depends on what type of model is fitted. Because the model in Equation (5.7) forms the basis of much of the material to come, we present it one more time, but now with all the assumptions.

$$\begin{aligned} \mathbf{Y}_i &= \mathbf{X}_i \times \boldsymbol{\beta} + \mathbf{Z}_i \times \mathbf{b}_i + \boldsymbol{\varepsilon}_i \\ \mathbf{b}_i &\sim N(\mathbf{0}, \mathbf{D}) \\ \boldsymbol{\varepsilon}_i &\sim N(\mathbf{0}, \boldsymbol{\Sigma}_i) \\ \mathbf{b}_1, \dots, \mathbf{b}_N, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_N &\text{ independent} \end{aligned} \quad (5.8)$$

This model is also called the Laird and Ware model formulation after a paper by these two authors in 1982. It is fundamentally important that you understand the model formulation, and therefore we give three examples before continuing with more details. These are the random intercept model, the random intercept and slope model, and the random effects model.

5.3.2 The Random Intercept Model

Suppose we model species richness as a linear function of NAP where the intercept is allowed to change per beach. Within linear regression, we can model this as

$$R_{ij} = \alpha + \beta_1 \times Beach_i + \beta_2 \times NAP_{ij} + \varepsilon_{ij} \quad (5.9)$$

$Beach_i$ is a factor with nine levels, and the first level is used as baseline. The price we pay for including this term is eight regression parameters (which will cost 8 degrees of freedom). However, perhaps we are not interested in knowing the exact nature of the beach effect. In that case, eight regression parameters is a high price! One option is to use beach as a random effect. This means that we include a beach effect in the model, but we assume that the variation around the intercept, for each beach, is normally distributed with a certain variance. A small variance means that differences per beach (in terms of the intercept) are small, whereas a large variance allows for more variation. Such a mixed effects model is defined as follows.

$$\begin{pmatrix} R_{i1} \\ R_{i2} \\ R_{i3} \\ R_{i4} \\ R_{i5} \end{pmatrix} = \begin{pmatrix} 1 & NAP_{i1} \\ 1 & NAP_{i2} \\ 1 & NAP_{i3} \\ 1 & NAP_{i4} \\ 1 & NAP_{i5} \end{pmatrix} \times \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \times b_i + \begin{pmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \varepsilon_{i3} \\ \varepsilon_{i4} \\ \varepsilon_{i5} \end{pmatrix} \Leftrightarrow \mathbf{R}_i = \mathbf{X}_i \times \boldsymbol{\beta} + \mathbf{Z}_i \times \mathbf{b}_i \quad (5.10)$$

In this example, five observations are taken on each beach, hence $n_i = 5$ for all i . Therefore, \mathbf{Z}_i is a matrix of dimension 5×1 containing only ones. Now let us have a look at the assumptions. The first assumption is that the random effects b_i are normally distributed: $N(0, d^2)$. The second assumption is that the errors ε_i (containing the five errors ε_{i1} to ε_{i5}) are normally distributed with covariance matrix Σ_i . The easiest option is to assume

$$\Sigma_i = \sigma^2 \times \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & \vdots \\ \vdots & 0 & 1 & 0 & \vdots \\ \vdots & \vdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

In general, the elements of Σ_i do not depend on i , but this may not always be the case (Verbeke and Molenberghs, 2000; Pinheiro and Bates, 2000). In Chapter 4, we discussed various methods to incorporate heterogeneity into the model, and these will influence the structure of Σ_i . But for the moment, we ignore these methods. To apply the random intercept model in R, we need the following code.

```
> library(nlme)
> RIKZ$fBeach <- factor(RIKZ$Beach)
> Mlme1 <- lme(Richness ~ NAP, random = ~1 | fBeach,
+                 data = RIKZ)
> summary(Mlme1)
```

The mixed effects model is applied using the function `lme`, which stands for linear mixed effects model. The difference with the `lm` command for linear regression is that in the `lme` function, we need to specify the random component. The `~1 | fBeach` bit specifies a random intercept model. The argument on the right hand side of the ‘|’ sign is a nominal variable. The relevant output from the `summary` command is given below.

```
Linear mixed-effects model fit by REML
  AIC      BIC      logLik
 247.48  254.52 -119.74

Random effects:
 Formula: ~1 | fBeach
            (Intercept) Residual
 StdDev:     2.944      3.059

Fixed effects: Richness ~ NAP
    Value Std.Error DF   t-value p-value
(Intercept)  6.58   1.09   35   6.00   <0.001
NAP         -2.56   0.49   35  -5.19   <0.001
```

The first part of the output gives the AIC and BIC. Their definitions and examples on how to use them are given later in this chapter. For the moment, it is sufficient to know that we are using them just as in linear regression to help with model selection. The remaining part of the output is split up in random effects and fixed effects. The residual variance σ^2 is estimated as $3.05^2 = 9.30$, and the variance for the random intercept d^2 is estimated as $2.94^2 = 8.64$. We should not say that $d = 2.94$, as d is a population parameter, and the value of 2.94 is an estimator for it. It is better to put a hat on d , and say that the estimated value for d is

$$\hat{d} = 2.94.$$

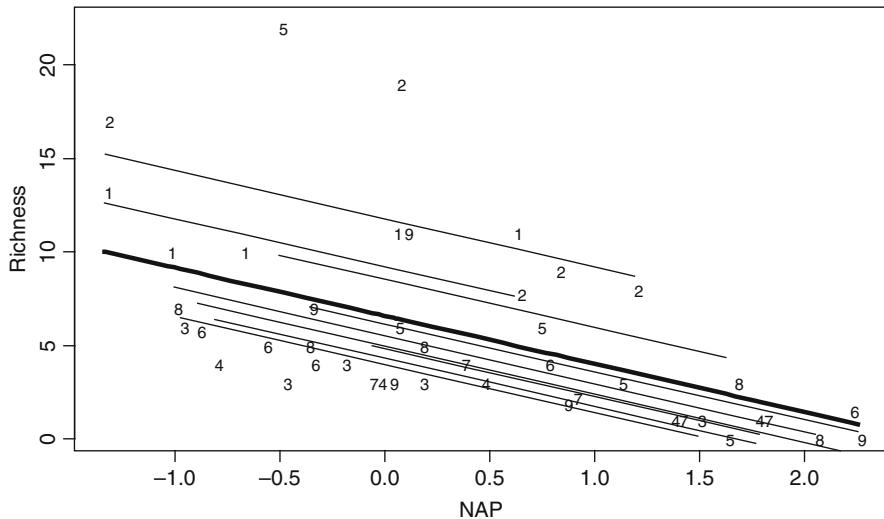


Fig. 5.2 Fitted values obtained by mixed effects modelling. The thick line represents the fitted values for the population and is specified by $6.58 - 2.56 \times NAP_i$, whereas the other lines are obtained by $\mathbf{X}_i \times \boldsymbol{\beta} + \mathbf{Z}_i \times \mathbf{b}_i$. Numbers represent the beaches

The fixed effects part shows that the intercept α is 6.58 and the slope β is -2.56 (again, we should put hats on parameters as both are estimators). Both parameters are significantly different from 0 at the 5% level. We discuss later how degrees of freedom are obtained.

All this information may look wonderful but what does it mean? The best way to answer this is to plot the fitted values. This raises the question: What are the fitted values? There are two options. We can either consider $\mathbf{X}_i \times \boldsymbol{\beta}$ as the fitted values (again, we should put a hat on the $\boldsymbol{\beta}$), which is $6.58 - 2.56 \times NAP_i$ or use $\mathbf{X}_i \times \boldsymbol{\beta} + \mathbf{Z}_i \times \mathbf{b}_i$ as the fitted values. Both types of fitted values are presented in Fig. 5.2.

The thick line represents the fitted line obtained by the fixed component $6.58 - 2.56 NAP_i$, also called the population model. The other lines are obtained by adding the contribution of \mathbf{b}_i for each beach i to the population fitted curve. Hence, the random intercept model implies one average curve (the thick line) that is allowed to be shifted up, or down, for each beach by something that is normally distributed with a certain variance d^2 . If d^2 is large, the vertical shifts will be relative large. If $d^2 = 0$, all shifts are zero and coincide with the thick line. The following R code was used to generate Fig. 5.2.

```
> F0 <- fitted(Mlme1, level = 0)
> F1 <- fitted(Mlme1, level = 1)
> I <- order(RIKZ$NAP); NAPs <- sort(RIKZ$NAP)
> plot(NAPs, F0[I], lwd = 4, type = "l",
       ylim = c(0, 22), ylab = "Richness", xlab = "NAP")
```

```

> for (i in 1:9){
  x1 <- RIKZ$NAP[RIKZ$Beach == i]
  y1 <- F1[RIKZ$Beach == i]
  K <- order(x1)
  lines(sort(x1), y1[K])
}
> text(RIKZ$NAP, RIKZ$Richness, RIKZ$Beach, cex = 0.9)

```

The `fitted` command takes as argument the object from the function `lme` plus a `level` argument. The `level = 0` option means that we take the fitted values obtained by the population model, whereas `level = 1` gives the *within-beach* fitted values. The `order` and `sort` commands avoid spaghetti plots, and the loop draws the nine lines in the same plot as the population curve.

5.3.3 The Random Intercept and Slope Model

The model in Equation (5.10) allows for a random shift around the intercept resulting in fitted lines parallel to the population fitted line (Fig. 5.2). This immediately raises the question whether we can use the same trick for the slope. The answer is yes, but before showing the model and the R code, we first discuss why we want to do this.

Suppose that the relationship between species richness and NAP is different on each beach. This implies that we need to include a NAP–Beach interaction term to the model. Such a model is specified by: $R_i = \text{factor(Beach)} + \text{NAP} \times \text{factor(Beach)}$. This is a linear regression model with one nominal variable, one continuous variable, and an interaction between them. A different name is an analysis of covariance (ANCOVA). Because beach has nine levels and one level is used as the baseline, the number of parameters used by this model is excessively high, at 17. And we are not even interested in beach effects! But if there is any between beach variation and a NAP–Beach interaction, then we cannot ignore these terms. If we do, this systematic variation ends up in the residuals, leading to potentially biased inference. To estimate model degrees of freedom more efficiently, we can apply the mixed effects model with a random intercept (as before) *and* a random slope. The required R code is a simple extension of the code we used for the random intercept model.

```

> Mlme2 <- lme(Richness ~ NAP,
                  random = ~1 + NAP | fBeach, data = RIKZ)
> summary(Mlme2)

Linear mixed-effects model fit by REML
AIC      BIC      logLik
244.38   254.95   -116.19

```

Random effects:

Formula:	$\sim 1 + \text{NAP} \text{fBeach}$	StdDev	Corr
(Intercept)	3.549	(Intr)	
NAP	1.714	-0.99	
Residual	2.702		

Fixed effects: Richness ~ NAP

	Value	Std.Error	DF	t-value	p-value
(Intercept)	6.58	1.26	35	5.20	<0.001
NAP	-2.83	0.72	35	-3.91	<0.001

Later we discuss how to compare the two models (random intercept and random intercept and slope models) and how to judge which one is better. For the moment, it is sufficient to note that the random intercept and slope model has a lower AIC than the earlier models (the lower the AIC, the better). Later in this chapter, we also dedicate an entire section to the phrase ‘Linear mixed-effects model fit by REML’.

The random effects part now has three standard errors and one correlation term. The model that we are fitting is of the form

$$\begin{pmatrix} R_{i1} \\ R_{i2} \\ R_{i3} \\ R_{i4} \\ R_{i5} \end{pmatrix} = \begin{pmatrix} 1 & \text{NAP}_{i1} \\ 1 & \text{NAP}_{i2} \\ 1 & \text{NAP}_{i3} \\ 1 & \text{NAP}_{i4} \\ 1 & \text{NAP}_{i5} \end{pmatrix} \times \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} 1 & \text{NAP}_{i1} \\ 1 & \text{NAP}_{i2} \\ 1 & \text{NAP}_{i3} \\ 1 & \text{NAP}_{i4} \\ 1 & \text{NAP}_{i5} \end{pmatrix} \times \begin{pmatrix} b_{i1} \\ b_{i2} \end{pmatrix} + \begin{pmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \varepsilon_{i3} \\ \varepsilon_{i4} \\ \varepsilon_{i5} \end{pmatrix} \quad (5.11)$$

The only difference with this model, compared to the one in Equation (5.10), is the modification of the matrix \mathbf{Z}_i ; NAP values for beach i have been included. As a result, \mathbf{b}_i is now of dimension 2×1 , and its distribution is given by

$$\begin{pmatrix} b_{i1} \\ b_{i2} \end{pmatrix} \sim N(\mathbf{0}, \mathbf{D}) \quad \text{where} \quad \mathbf{D} = \begin{pmatrix} d_{11}^2 & d_{12} \\ d_{12} & d_{22}^2 \end{pmatrix} \quad (5.12)$$

The variance d_{11}^2 plays the same role as d^2 in the random intercept model; it determines the amount of variation around the population intercept α . The numerical output shows that its estimated value is $3.54^2 = 12.5$. The model also allows for random variation around the population slope in a similar way as it does for the intercept. The variance d_{22}^2 determines the variation in slopes at the nine beaches. The estimated value of $1.71^2 = 2.92$ shows that there is considerably more variation in intercepts than in slopes at the nine beaches. Finally, there is a correlation between the random intercepts and slopes. Its value of -0.99 is rather high (causing potential numerical problems), but indicates that beaches with a high positive intercept also have a high negative slope. This can also be seen from the fitted values in Fig. 5.3.

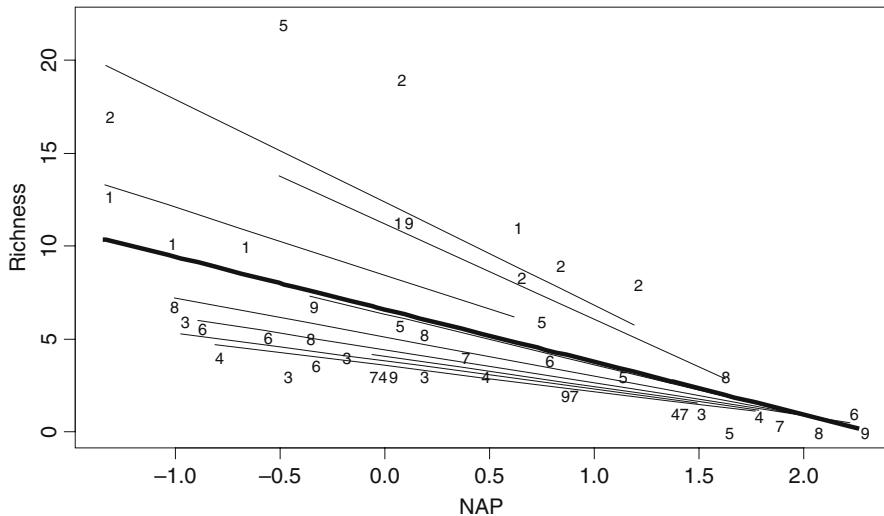


Fig. 5.3 Fitted values obtained by the random intercept and slope model. The thick line represents the fitted values for the population, and the other lines represent the so-called within-group fitted curves. Numbers represent beaches

The thick line is the fitted population curve, and the other lines the within-beach fitted curves. Note the difference with Fig. 5.2.

5.3.4 Random Effects Model

A linear mixed effects model that does not contain any β , except for an intercept is called a random effects model. By dropping the NAP variable in Equation (5.9), we obtain the following random effects model.

$$R_i = \alpha + b_i + \varepsilon_i$$

The term b_i is normally distributed with mean 0 and variance d^2 ; ε_i is normally distributed with mean 0 and variance σ^2 . The index i runs from 1 to 9. The model implies that richness is modelled as an intercept plus a random term b_i that is allowed to differ per beach. The R code to run this model is

```
> Mlme3 <- lme(Richness ~ 1, random = ~1 | fBeach,
  data = RIKZ)
```

The output of the `summary(Mlme3)` command is given below. The estimated values for d and σ are 3.23 and 3.93, respectively.

```

Linear mixed-effects model fit by REML
      AIC          BIC          logLik
  267.11        272.46       -130.55

Random effects:
      (Intercept)   Residual
StdDev:      3.23       3.93

Fixed effects:
      Value Std.Error DF t-value p-value
(Intercept) 5.68    1.22   36  4.63    <0.001

```

Later in this chapter, we discuss how to choose between a random effects model, random intercept model, and random intercept plus slope model. There are also several other issues that we need to discuss such as: What is the correlation between richness values measured at the same beach and measured at different beaches? How do we estimate the parameters? How do we find the optimal model? Finally, once an optimal model has been identified, how do we then validate it? Each of these points is discussed next.

5.4 Induced Correlations

Returning to the RIKZ data discussed earlier in this chapter, we modelled species richness as a function of NAP and a random intercept. The question we now address is: What is the correlation between two observations from the same beach, and from different beaches? To answer this question, we first need to find an expression for the covariance matrix of \mathbf{Y}_i . The mathematical notation that we have used so far for the model was $\mathbf{Y}_i = \mathbf{X}_i \times \boldsymbol{\beta} + \mathbf{Z}_i \times \mathbf{b}_i + \boldsymbol{\epsilon}_i$. This is also called the hierarchical model. The underlying assumptions of this model were given in Equation (5.8). We now derive an expression for the covariance matrix of the \mathbf{Y}_i . It is relatively easy to show that \mathbf{V}_i is normally distributed with mean $\mathbf{X}_i \times \boldsymbol{\beta}$ and variance \mathbf{V}_i in mathematical notation:

$$\mathbf{Y}_i \sim N(\mathbf{X}_i \times \boldsymbol{\beta}, \mathbf{V}_i) \quad \text{where } \mathbf{V}_i = \mathbf{Z}_i \times \mathbf{D} \times \mathbf{Z}'_i + \boldsymbol{\Sigma}_i \quad (5.13)$$

Recall that \mathbf{D} was the covariance matrix of the random effects. So, including random effects has an effect on the structure of the covariance matrix \mathbf{V}_i . To illustrate this, we discuss the random intercept model for the RIKZ data we presented in the previous section.

For the random intercept model, \mathbf{Z}_i is a vector of length five containing ones and $\boldsymbol{\Sigma}_i = \sigma^2 \times \mathbf{I}_{5 \times 5}$ is a diagonal matrix of dimension 5×5 . $\mathbf{I}_{5 \times 5}$ is an identity matrix with 5 rows and 5 columns; it has ones on the diagonal and zeros elsewhere. As a result we have

$$\mathbf{V}_i = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \times d^2 \times (1 \ 1 \ 1 \ 1 \ 1) + \sigma^2 \times \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & 1 & & \vdots \\ \vdots & & & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \sigma^2 + d^2 & d^2 & d^2 & d^2 & d^2 \\ d^2 & \sigma^2 + d^2 & d^2 & d^2 & d^2 \\ d^2 & d^2 & \sigma^2 + d^2 & d^2 & d^2 \\ d^2 & d^2 & d^2 & \sigma^2 + d^2 & d^2 \\ d^2 & d^2 & d^2 & d^2 & \sigma^2 + d^2 \end{pmatrix}$$

Showing that, the covariance between any two sites on the same beach is d^2 , and the variance is $d^2 + \sigma^2$. By definition, the correlation between two observations from the same beach is $d^2/(d^2 + \sigma^2)$. This is irrespective of the identity of the beach (all the \mathbf{V}_i s are the same). This is called an induced correlation (or covariance) structure as we did not explicitly specify it. It is the consequence of the random effects structure. The results presented in Section 5.3 show that the estimated value for d is 2.944 and for σ it is 3.06. Giving an induced correlation of $2.94^2/(2.94^2 + 3.06^2) = 0.48$, which is relatively high. This correlation is also called the intraclass correlation and is further discussed at the end of this section.

As to the second question, the model implies that observations from different beaches are uncorrelated.

We can make things a bit more complicated by using the random intercept and slope model. In this case, we get

$$\mathbf{V}_i = \begin{pmatrix} 1 & NAP_{i1} \\ 1 & NAP_{i2} \\ 1 & NAP_{i3} \\ 1 & NAP_{i4} \\ 1 & NAP_{i5} \end{pmatrix} \times \begin{pmatrix} d_{11}^2 & d_{21} \\ d_{12} & d_{22}^2 \end{pmatrix} \times \begin{pmatrix} 1 & NAP_{i1} \\ 1 & NAP_{i2} \\ 1 & NAP_{i3} \\ 1 & NAP_{i4} \\ 1 & NAP_{i5} \end{pmatrix} + \sigma^2 \times \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & 1 & & \vdots \\ \vdots & & & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}$$

This is a bit more challenging, but it turns out that the variance of Y_{ij} and covariance of two observations from the same beach, Y_{ij} and Y_{ik} , are given by (Fitzmaurice et al., 2004)

$$\text{var}(Y_{ij}) = d_{11}^2 + 2 \times NAP_{ij} \times d_{12} + NAP_{ij}^2 \times d_{22}^2 + \sigma^2$$

$$\text{cov}(Y_{ij}, Y_{ik}) = d_{11}^2 + (NAP_{ij} + NAP_{ik}) \times d_{12} + NAP_{ij} \times NAP_{ik} \times d_{22}^2$$

This looks complicated, but it tells us that the variance and covariance of Y_{ij} depend not only on the variances and covariances of the random terms, but also on NAP. Fitzmaurice et al. (2004) used time instead of NAP. In that case, the variance and covariance depend on time.

5.4.1 Intraclass Correlation Coefficient

Although outside the scope of the underlying questions raised at the start of this section, it is useful to take some time interpreting the intraclass correlation as it can be used to determine appropriate sample sizes. It is also called the intraclass correlation (Snijders and Bosker, 1999). Recall that we have nine beaches, five observations per beach, and an intraclass correlation of 0.48. If we take a sample of a certain size, the standard error of the mean is given by

$$\text{standard error} = \frac{\text{standard deviation}}{\sqrt{\text{sample size}}}$$

Obviously, we want a small standard error and a large sample size may help achieve this as it is the denominator. In this case, we have a sample size of 45. However, these data are nested (hierarchical) and this should be taken somehow into account, especially of the correlation between observations on a beach is relative high. The design effect indicates how much the denominator should be adjusted. For a more formal definition, see Snijders and Bosker (1999). For a two-stage design with equal number of samples per beach ($n = 5$) and intraclass correlation ρ , the design effect is defined as

$$\text{design effect} = 1 + (n - 1) \times \rho = 1 + 4 \times 0.48 = 2.92$$

If this number is larger than 1, and in this case it is 2.92, we should not use 45 in the denominator for the standard error, but an adjusted sample size, also called the effective sample size, should be used. It is given by

$$N_{\text{effective}} = \frac{N \times n}{\text{design effect}} = \frac{9 \times 5}{2.92} = 15.41$$

A high intraclass correlation means that the corrected sample size is considerably lower, and this means less precise standard errors! At the end of the day, this makes sense; if observations on a beach are highly correlated, we cannot treat them as independent observations. Why then bother taking many observations per beach? Perhaps we should sample more beaches with fewer observations per beach? Further examples are given in Chapter 3 in Snijder and Bosker (2000).

5.5 The Marginal Model

In the previous section, we saw how including random effects induces a correlation structure between observations from the same beach. With the random intercept model, the induced correlation structure was fairly simple with the correlation between any two observations from the same beach given as $d^2/(d^2 + \sigma^2)$. Surprisingly, we can get the same correlation structure and estimated parameters in a

different way, and it does not contain any random effects. The model we use is the linear regression model $\mathbf{Y}_i = \mathbf{X}_i \times \boldsymbol{\beta} + \boldsymbol{\varepsilon}_i$; but instead of assuming that the five residuals of the same beach, $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \varepsilon_{i2}, \varepsilon_{i3}, \varepsilon_{i4}, \varepsilon_{i5})$ are independent of each other, we allow for dependence between them. This is done as follows. We start again with an expression for the covariance matrix of the \mathbf{Y}_i . Using the standard linear regression theory, it is easy to show that \mathbf{Y}_i is normally distributed with mean $\mathbf{X}_i \times \boldsymbol{\beta}$ and variance \mathbf{V}_i ; in mathematical notation,

$$\mathbf{Y}_i \sim N(\mathbf{X}_i \times \boldsymbol{\beta}, \mathbf{V}_i) \quad \text{where } \mathbf{V}_i = \boldsymbol{\Sigma}_i$$

Note that there is no covariance matrix \mathbf{D} in \mathbf{V}_i as there are no random effects in the model. In linear regression, we use $\boldsymbol{\Sigma}_i = \sigma^2 \times \mathbf{I}_{5 \times 5}$. $\mathbf{I}_{5 \times 5}$ is an identity matrix with 5 rows and 5 columns, implying independence between residuals (or observations) of the same beach i . The dependence structure is built in by allowing for non-zero off-diagonal elements in the covariance matrix. One option is the so-called *general correlation matrix*

$$\mathbf{V}_i = \boldsymbol{\Sigma}_i = \begin{pmatrix} \sigma^2 & c_{21} & c_{31} & c_{41} & c_{51} \\ c_{21} & \sigma^2 & c_{32} & c_{42} & c_{52} \\ c_{31} & c_{32} & \sigma^2 & c_{43} & c_{53} \\ c_{41} & c_{42} & c_{43} & \sigma^2 & c_{54} \\ c_{51} & c_{52} & c_{53} & c_{54} & \sigma^2 \end{pmatrix}$$

Because the covariance between observations Y_{i1} and Y_{i2} is the same as that between observations Y_{i2} and Y_{i1} , the covariance matrix \mathbf{V}_i is symmetric. So, in this example, we have to estimate 10 parameters (all the elements in the upper or lower diagonal). But for data sets with larger number of observations per beach, this number increases dramatically. Therefore, we can use more restrictive covariance matrices. The most restrictive correlation structure is the so-called *compound symmetric* structure defined by

$$\mathbf{V}_i = \boldsymbol{\Sigma}_i = \begin{pmatrix} \sigma^2 & \varphi & \varphi & \varphi & \varphi \\ \varphi & \sigma^2 & \varphi & \varphi & \varphi \\ \varphi & \varphi & \sigma^2 & \varphi & \varphi \\ \varphi & \varphi & \varphi & \sigma^2 & \varphi \\ \varphi & \varphi & \varphi & \varphi & \sigma^2 \end{pmatrix}$$

In this case, there is only one unknown parameter, namely φ . So, the covariance between any two observations on the same beach i is given by φ . If it is estimated as 0, then we can assume independence. General correlation and compound symmetry correlation are the two most extreme correlation structures, and there are various intermediate structures that we will see later, which can be applied to spatial and temporal data.

The R code for the marginal model is given below. We also give the command for the equivalent random intercept mixed effects model.

```
> M.mixed <- lme(Richness ~ NAP, random = ~1 | fBeach,
                    method = "REML", data = RIKZ)
> M.gls <- gls(Richness ~ NAP, method = "REML",
                  correlation = corCompSymm(form =~ 1 | fBeach),
                  data = RIKZ)
```

The argument `corCompSymm(form =~ 1 | fBeach)` for the correlation option in the `gls` function tells R that all observations from the same beach are correlated. The `summary(M.mixed)` and `summary(M.gls)` commands give identical estimated parameters, standard errors, *t*-values, and *p*-values, and these are not shown here (see Section 5.3). We only show the relevant output of the GLS model.

```
Correlation Structure: Compound symmetry
Formula: ~1 | factor(Beach)
Parameter estimate(s):
  Rho
0.4807353
...
Residual standard error: 4.246141
```

The estimated Rho is φ divided by the estimated value of σ^2 ($= 4.25^2$) as we expressed \mathbf{V}_i as a covariance matrix and not a correlation matrix.

There are also subtle differences between the hierarchical model and the marginal model with respect to the numerical estimation process (West et al., 2006).

5.6 Maximum Likelihood and REML Estimation*

When applying mixed effects modelling, you will see phrases like ‘REML’ and ‘maximum likelihood’ estimation. Unlike linear regression models, where you can get away with not knowing the underlying mathematics, there is no escaping some maths when using REML and maximum likelihood (ML) in mixed effects modelling. So, what does REML mean, and what does it do? The first question is easy; REML stands for restricted maximum likelihood estimation. As to the second question, most books at this point get rather technical or avoid the detail and only present REML as a mystical way to ‘correct the degrees of freedom’. We have chosen to try and explain it in more detail and for this we need to use matrix algebra. But, to understand REML you need to first understand the principle of maximum likelihood estimation, and this is where we will begin. If you are not familiar with matrix algebra, or if the mathematical level in this section is too high, we still advise you to skim through this section before reading on.

We start by revising maximum likelihood for linear regression, and then show how REML is used to correct the estimator for the variance.

Assume we have a linear regression model $Y_i = \alpha + \beta \times X_i + \varepsilon_i$, where ε_i is normally distributed with mean 0 and variance σ^2 . The unknown parameters in this model are α , β , and σ . Instead of writing these three variables all the time, we can refer to them as $\boldsymbol{\theta}$, where $\boldsymbol{\theta} = (\alpha, \beta, \sigma)$. One option to estimate $\boldsymbol{\theta}$ is ordinary least squares. It gives an expression for each element of $\boldsymbol{\theta}$, see, for example, Montgomery and Peck (1992), among many other books on linear regression. The expression for the estimated variance obtained by linear regression is

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum_{i=1}^n (Y_i - \hat{\alpha} - \hat{\beta} \times X_i)^2 \quad (5.14)$$

We have put a $\hat{\cdot}$ on the parameters to indicate that these are the estimated values, and n is the number of observations. It can be shown that $\hat{\sigma}$ is an unbiased estimator of σ ; this means that $E[\hat{\sigma}] = \sigma$. Now let us have a look at the maximum likelihood estimation approach. We have used results from Section 2.10 in Montgomery and Peck (1992), which assume that Y_i is normally distributed and its density function is given by

$$f_i(Y_i, X_i, \alpha, \beta, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(Y_i - \alpha - \beta \times X_i)^2}{2\sigma^2}} \quad (5.15)$$

Because we also assume that the Y_i are independent, we can write the joint density function for Y_1, Y_2, \dots, Y_n as a product of the individual density curves f_1, f_2, \dots, f_n . This is called the likelihood function L . It is a function of the data and $\boldsymbol{\theta}$. The question is how to choose $\boldsymbol{\theta}$ such that L is the highest. To simplify the mathematics, the natural log is taken of L (The log converts the product of the density functions to a sum of log-density functions; it is easier to work with a sum than a product.), resulting in the following log-likelihood equation.

$$\ln L(Y_i, X_i, \alpha, \beta, \sigma) = -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (Y_i - \alpha - \beta \times X_i)^2 \quad (5.16)$$

We need to maximise this function with respect to α , β , and σ . This is a matter of taking partial derivatives of L with respect to each of these parameters, setting them to zero, and solving the equations. It turns out that these equations give simple expressions for the estimators of α and β . Because we can easily calculate them, these equations are called *closed form* solutions. In the generalized linear mixed model chapters, we will see open form solutions, which means there is no direct solution for the parameters.

The formulae for the estimators of α and β are not given here, but for the variance we get

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{\alpha} - \hat{\beta} \times X_i)^2 \quad (5.17)$$

Note this is nearly the same expression as we found with ordinary least squares in Equation (5.14). In fact, the estimator for the variance obtained by maximum likelihood is biased by a factor $(n - 2)/n$. If the linear regression model contains p explanatory variables, the bias is $(n - p)/n$. The reason that the maximum likelihood estimator is biased is because it ignores the fact that the intercept and slope are estimated as well (as opposed to being known for certain). So, we need a mechanism that gives better ML estimators, and indeed this is what restricted maximum likelihood (REML) does.

REML works as follows. The linear regression model $Y_i = \alpha + \beta \times X_i + \varepsilon_i$ can be written as $Y_i = \mathbf{X}_i \times \boldsymbol{\beta} + \varepsilon_i$. This is based on simple matrix notation using $\mathbf{X}_i = (1 \ X_i)$, and the first element of $\boldsymbol{\beta}$ is the intercept and the second element is the original β . The normality assumption implies that

$$Y_i \sim N(\mathbf{X}_i \times \boldsymbol{\beta}, \sigma^2) \quad (5.18)$$

The problem with the ML estimator is that we have to estimate the intercept and the slope, which are in $\boldsymbol{\beta}$ in Equation (5.18). Obviously, the problem is solved if there is no $\boldsymbol{\beta}$. All that REML does is apply a little trick to avoid having any $\boldsymbol{\beta}$ in Equation (5.18). It does this by finding a special matrix \mathbf{A} of dimension $n \times (n - 2)$, and special means ‘orthogonal to (or independent of) \mathbf{X}' , multiplies $\mathbf{Y} = (Y_1, \dots, Y_n)'$ with this matrix and continues with ML estimation. Orthogonal means that if \mathbf{A} and \mathbf{X} are multiplied, the result is 0. Hence, we get $\mathbf{A}' \times \mathbf{Y} = \mathbf{A}' \times \mathbf{X} \times \boldsymbol{\beta} + \mathbf{A}' \times \boldsymbol{\varepsilon} = \mathbf{0} + \mathbf{A}' \times \boldsymbol{\varepsilon} = \mathbf{A}' \times \boldsymbol{\varepsilon}$. The distribution for $\mathbf{A}' \times \mathbf{Y}$ is now given by

$$\mathbf{A}' \times \mathbf{Y} \sim N(\mathbf{0}, \sigma^2 \times \mathbf{A}' \times \mathbf{A}) \quad (5.19)$$

which no longer depends on $\boldsymbol{\beta}$. Applying ML on $\mathbf{A}' \times \mathbf{Y}$ gives an unbiased estimator for σ^2 (same expression as in Equation (5.14)). Now we discuss how REML can be used for the mixed effects model. Our starting point is the marginal model

$$\mathbf{Y}_i \sim N(\mathbf{X}_i \times \boldsymbol{\beta}, \mathbf{V}_i) \quad \mathbf{V}_i = \mathbf{Z}_i \times \mathbf{D} \times \mathbf{Z}_i' + \boldsymbol{\Sigma}_i \quad (5.20)$$

The story now starts all over again. As before, we can formulate a slightly different log-likelihood criteria. The unknown parameters are $\boldsymbol{\beta}$ and the elements of \mathbf{D} and $\boldsymbol{\Sigma}_i$. Again, we denote them all by $\boldsymbol{\theta}$. The log-likelihood function is given by

$$\ln L(\mathbf{Y}_i, \mathbf{X}_i, \boldsymbol{\theta}) = -\frac{n}{2} \ln 2\pi - \frac{1}{2} \sum_{i=1}^n \ln |\mathbf{V}_i| - \frac{1}{2} \sum_{i=1}^n (\mathbf{Y}_i - \mathbf{X}_i \times \boldsymbol{\beta})' \times \mathbf{V}_i^{-1} \times (\mathbf{Y}_i - \mathbf{X}_i \times \boldsymbol{\beta})$$

The notation $|\mathbf{V}_i|$ stands for the determinant of \mathbf{V}_i . This looks intimidating, but can be found in many introductory statistical textbooks. Just as before, an expression for $\boldsymbol{\beta}$ is obtained by setting the partial derivative of L with respect to $\boldsymbol{\beta}$ equal to zero and solving the equation. Just as in the example discussed on the previous page,

doing the same for the elements of the covariance matrix \mathbf{V}_i gives biased estimates, and therefore we need REML.

For the RIKZ data, we had 9 beaches; hence, $i = 1, \dots, 9$. In general, the index i runs from 1 to n . We can stack all the vectors $\mathbf{Y}_i = (Y_{i1}, \dots, Y_{i5})$ into one long vector of dimension 45×1 (nine beaches, five observations per beach). Let us denote the stacked column by \mathbf{Y} . We can also stack all the \mathbf{X}_i into one matrix of dimension $45 \times p$, where p is the number of fixed covariates. Denote it by \mathbf{X} . We have to do something slightly different for the covariance matrix. Instead of stacking them, we create a new matrix \mathbf{V} with diagonal blocks \mathbf{V}_1 to \mathbf{V}_9 . The other elements of \mathbf{V} are equal to 0. A similar approach is followed for the \mathbf{Z}_i s. Using this new notation, we can write Equation (5.20) as $\mathbf{Y} \sim N(\mathbf{X} \times \boldsymbol{\beta}, \mathbf{V})$. Just as before, the \mathbf{Y} vector is multiplied with a special matrix \mathbf{A} , such that $\mathbf{A}' \times \mathbf{Y} = \mathbf{A}' \times \mathbf{X} \times \boldsymbol{\beta} + \mathbf{A}' \times \mathbf{V} = \mathbf{0} + \mathbf{A}' \times \mathbf{V}$. We can write $\mathbf{A}' \times \mathbf{Y} \sim N(\mathbf{0}, \mathbf{A}' \times \mathbf{V} \times \mathbf{A})$, and maximum likelihood is used to obtain unbiased estimates for the elements of \mathbf{V} . The good news is that the estimators for the variance terms are independent of (not related to) the choice for \mathbf{A} . Summarising, REML applies a special matrix multiplication on \mathbf{Y} in such a way that the $\mathbf{X} \times \boldsymbol{\beta}$ -bit disappears. It then continues with maximum likelihood estimation and the resulting parameter estimators are unbiased and not related to the specific matrix multiplication. As a consequence, the REML estimators for the $\boldsymbol{\beta}$ s are not identical to the maximum likelihood estimators. If the number of fixed covariates is small relative to the number of observations, there are not many differences, but for models with many fixed terms, this may not be the case.

5.6.1 Illustration of Difference Between ML and REML

To illustrate this, we applied two models on the RIKZ data. Both models are random intercept models estimated with ML and REML. In the first model, we used only NAP as fixed covariate and in the second model NAP and exposure. All numerical outputs are given in Table 5.1. For the model that only contains NAP as the fixed term, differences in estimated parameters and variances between REML and ML are relatively small. Adding the nominal variable exposure increases the number of regression parameters by 1. The ML estimated slope for NAP is -2.60 with the REML now -2.58. The R code for the two models is as follows. The `method = "ML"` or `method = "REML"` specifies which estimation method is used. The first three lines define the nominal variable exposure with two levels (instead of 3). The output was obtained with the `summary` command.

```
> RIKZ$fExp <- RIKZ$Exposure
> RIKZ$fExp[RIKZ$fExp == 8] <- 10
> RIKZ$fExp <- factor(RIKZ$fExp, levels = c(10, 11))
> M0.ML <- lme(Richness ~ NAP, data = RIKZ,
+ random = ~1 | fBeach, method = "ML")
```

```
> M0.REML <- lme(Richness ~ NAP, random = ~1 | fBeach,
                     method = "REML", data = RIKZ)
> M1.ML <- lme(Richness ~ NAP + fExp, data = RIKZ,
                  random = ~1 | fBeach, method = "ML")
> M1.REML <- lme(Richness ~NAP + fExp, data = RIKZ,
                  random = ~1 | fBeach, method = "REML")
```

Table 5.1 Results for two models using ML (middle column) and REML (right column) estimation. Numbers between brackets are standard errors. The first model (upper part of the table) uses an intercept and NAP as fixed covariates and a random intercept. The second model (lower part of the table) used the same terms, except that the nominal variable exposure is used as a fixed term as well

Mixed model with NAP as fixed covariate and random intercept		
Parameter	Estimate using ML	Estimate using REML
Fixed intercept	6.58 (1.05)	6.58 (1.09)
Fixed slope NAP	-2.57 (0.49)	-2.56 (0.49)
Variance random intercept	7.50	8.66
Residual variance	9.11	9.36
AIC	249.82	247.48
BIC	257.05	254.52
Mixed model with NAP and exposure as fixed covariate and random intercept		
Parameter	Estimate using ML	Estimate using REML
Fixed intercept	8.60 (0.96)	8.60 (1.05)
Fixed slope NAP	-2.60 (0.49)	-2.58 (0.48)
Fixed Exposure level	-4.53 (1.43)	-4.53 (1.57)
Variance random intercept	2.41	3.63
Residual variance	9.11	9.35
AIC	244.75	240.55
BIC	253.79	249.24

5.7 Model Selection in (Additive) Mixed Effects Modelling

In the earlier sections, we applied a series of models on the species richness for the RIKZ data. Although the original data set contained 10–15 explanatory variables, we have only used NAP and exposure as explanatory variables because our prime aim here is to explain methodology and not to provide the best model for these data. The case studies can be consulted for examples of best possible models. We now use the RIKZ data to explain model selection in mixed effects modelling.

Just as in linear regression, there are two main options for model selection. One option is based on selection tools like the Akaike Information Criteria (AIC), or the Bayesian Information Criteria (BIC). Both the AIC and BIC contain two terms that measure the fit of the model and the complexity of the model. The likelihood value is used in defining the measure of fit, and the number of parameters measures the complexity.

As a measure of fit, we can use the log likelihood function. But there are two likelihood functions: the REML and the ML one. It can be shown (Verbeke and Molenberghs, 2000) that

$$L_{REML}(\boldsymbol{\theta}) = \left| \sum_{i=1}^n \mathbf{X}_i' \times V_i^{-1} \times \mathbf{X}_i \right|^{-0.5} \times L_{ML}(\boldsymbol{\theta})$$

The AIC is defined as twice the difference between the value of the likelihood L (measure of fit) and the number of parameters (penalty for model complexity) in $\boldsymbol{\theta}$. For the BIC, the number of observations is also taken into account, which means that more severe increases in the likelihood are required for larger data sets to label a model as better. In the formulae below, p is the number of parameters in $\boldsymbol{\theta}$, L is either the ML or REML likelihood, and for ML, we have $n^* = n$, but for REML, $n^* = n - p$.

$$\begin{aligned} AIC &= -2 \times L(\boldsymbol{\theta}) + 2 \times p \\ BIC &= -2 \times L(\boldsymbol{\theta}) + 2 \times p \times \ln(n^*) \end{aligned}$$

This means that an AIC based on REML is not comparable with an AIC obtained by ML. The same holds for the BIC.

The second approach to find the optimal model is via hypothesis testing. There are three options here: (i) the t -statistic, the F -statistic, or the likelihood ratio test. In Chapter 1, we discussed how to compare nested linear models using the maximum likelihood ratio test. The problem is that the mixed effects model contains two components: a fixed effect (the explanatory variables) and the random effects. So, we need to select not only an optimal fixed effects structure but also an optimal random effects structure. In most cases, we are interested in the fixed effects. But if the random effects are poorly chosen, then this affects the values (biased) and quality of the fixed effects as the random effects work their way into the standard errors of the slopes for the fixed effects. On the other hand, variation in the response variable not modelled in terms of fixed effects ends up in the random effects. There are two strategies to work your way through the model selection process: the top-down strategy and the step-up strategy (West et al., 2006). The first one is recommended by Diggle et al. (2002) and is the only one discussed here. The protocol for the top-down strategy contains the following steps:

1. Start with a model where the fixed component contains all explanatory variables and as many interactions as possible. This is called the *beyond optimal* model. If this is impractical, e.g. due to a large number of explanatory variables, interactions, or numerical problems, use a selection of explanatory variables that you think are most likely to contribute to the optimal model.
2. Using the beyond optimal model, find the optimal structure of the random component. Because we have as many explanatory variables as possible in the fixed component, the random component (hopefully) does not contain any information that we would like to have in the fixed component. The problem is that comparing

two models with nested random structures cannot be done with ML because the estimators for the variance terms are biased. Therefore, we must use REML estimators to compare these (nested) models. Obtaining valid p -values for such tests is another non-trivial issue due to something called *testing on the boundary*, which we will discuss later in this section. As well as using the (REML) likelihood ratio test, we can also use the AIC or BIC, but again we need to use REML. Using AIC or BIC does not avoid boundary problems.

3. Once the optimal random structure has been found, it is time to find the optimal fixed structure. As mentioned above, we can either use the F -statistic or the t -statistic obtained with REML estimation or compare nested models. To compare models with nested fixed effects (but with the same random structure), ML estimation must be used and not REML. We discuss the details of these tests later in this chapter.
4. Present the final model using REML estimation.

These steps should only be used as a general guidance, and sometimes common sense is required to derive a slightly different approach. For example, sometimes, it is impractical to apply a model with as many explanatory variables as possible, especially in generalised additive modelling.

5.8 RIKZ Data: Good Versus Bad Model Selection

5.8.1 The Wrong Approach

We start with an illustration how not to do a mixed effects model selection. In particular, we show the danger of not starting with a full model. To illustrate this, we take NAP as the only fixed explanatory variable for the fixed component and ignore exposure for the moment. As to the random structure, there are three options: (i) no random term, except for the ordinary residuals; (ii) a random intercept model using beach; and (iii) a random intercept and slope model.

A requirement for the `n1me` function in R is the specification of a random term, and to avoid an error message, the `gls` function can be used instead. The R code for these three models is:

```
> Wrong1 <- gls(Richness ~ 1 + NAP, method = "REML",
+                  data = RIKZ)
> Wrong2 <- lme(Richness ~ 1 + NAP, random = ~1|fBeach,
+                  method = "REML", data = RIKZ)
> Wrong3 <- lme(Richness ~ 1 + NAP, method = "REML",
+                  random = ~1 + NAP | fBeach, data = RIKZ)
```

All models have the same fixed effect structure, but different random components.

5.8.1.1 Step 2 of the Protocol

The second step of the protocol dictates that we judge which of these models is optimal. Note that the only difference is the random structure. Because REML estimation was used, we can compare AICs or BICs. These are obtained with the `AIC` or `BIC` commands:

```
> AIC(Wrong1, Wrong2, Wrong3)
```

	df	AIC
Wrong1	3	258.2010
Wrong2	4	247.4802
Wrong3	6	244.3839

This suggests the model with the random intercept and slope is the best. Note that both the second and third models are considerably better than the model without a random effect. The BIC for the second and third models are similar and both are lower than the BIC of the first model. Instead of the AIC (or BIC), we can also use the likelihood ratio test via the `anova` command as the models are nested.

```
> anova(Wrong1, Wrong2, Wrong3)
```

Model	df	AIC	BIC	logLik	Test	L.Ratio	p-val.
Wrong1	1	3	258.20	263.48	-126.10		
Wrong2	2	4	247.48	254.52	-119.74	1 vs 2	12.72 <0.001
Wrong3	3	6	244.38	254.95	-116.19	2 vs 3	7.09 0.03

The second line compares a model without any random effect versus a model with a random intercept. These models are nested with respect to the variances. Unfortunately, there is a little problem here, which is the ‘testing on the boundary’ mentioned earlier. The null hypothesis of this test is $H_0: \sigma^2 = 0$ versus the alternative $H_1: \sigma^2 > 0$. This is different from how you normally use this test to see whether a regression parameter is equal to zero or not. In that case, we use $H_0: \beta = 0$ versus the alternative $H_1: \beta \neq 0$. Note the subtle difference with respect to the $>$ and \neq symbols. This is called testing on the boundary for the obvious reason that if there is no evidence to reject the null-hypothesis, then $\sigma^2 = 0$ is the lowest possible value as a variance is always non-negative. The *p*-value provided by the `anova` function is incorrect as this function assumes that twice the differences between the two log-likelihood values, $L = -2 \times (-126.10 + 119.74) = 12.72$, follows a Chi-square distribution with p degrees of freedom; p is the number of extra parameters in the full model (here $p = 1$). The mathematical notation for such a distribution is χ_p^2 . However, when testing on the boundary, L does not follow this distribution, and therefore the *p*-value from the table is incorrect. Verbeke and Molenberghs (2000) showed that L follows a $0.5 \times (\chi_0^2 + \chi_1^2) = 0.5 \times \chi_1^2$ distribution. This means that the *p*-value in the table should be divided by 2. In R, you can get the correct *p*-value by typing

```
> 0.5 * (1 - pchisq(12.720753, 1))
```

The resulting p -value is still smaller than 0.001. This means that adding a random effect beach to the model is a significant improvement. Note that this correction only applies for comparing a model without and with a random intercept! If we want to compare the model with the random intercept and the model with random intercept and slope, then $L = 7.09$ follows a $0.5 \times (\chi_1^2 + \chi_2^2)$ distribution. The resulting p -value of 0.018 is calculated by

```
> 0.5 * ((1 - pchisq(7.09, 1)) + (1 - pchisq(7.09, 2)))
```

So, the random structure that contains both the random intercept and slope is significantly better (at least at the 5% level) than the random intercept model. The conclusion of step 2 is that you should proceed to step 3 with the random intercept and slope model.

5.8.1.2 Step 3 of the Protocol

In step 3, we search for the optimal fixed structure for a given random structure. Typing `summary(Wrong3)` gives a slope of -2.83 for NAP, and the associated standard error and t -value are 0.72 and -3.91 , respectively. The p -value of the t -statistic is smaller than 0.001, indicating that the slope for NAP is significant. Hence, dropping NAP from the model is not an option. The only thing we can try is adding exposure or adding exposure *and* the interaction between exposure and NAP. We can test the significance of these tests in three ways: either with an F -test or t -test obtained with REML or by comparing nested models using ML estimation. The first approach is carried out in R as follows. In case you skipped the previous section, the first three lines redefine the nominal variable exposure such that it only has two levels instead of three.

```
> RIKEZ$fExp <- RIKEZ$Exposure
> RIKEZ$fExp[RIKEZ$fExp == 8] <- 10
> RIKEZ$fExp <- factor(RIKEZ$fExp, levels = c(10, 11))
> lmc <- lmeControl(niterEM = 2200, msMaxIter = 2200)
> Wrong4 <- lme(Richness ~1 + NAP * fExp,
+                 random = ~1 + NAP | fBeach,
+                 method = "REML", data = RIKEZ)
> anova(Wrong4)

numDF  denDF   F-value p-value
(Intercept)    1      34  34.87139  <.0001
NAP            1      34  18.65502  0.0001
fExp           1       7   5.65495  0.0490
NAP:fExp       1      34   3.32296  0.0771
```

The `anova` command applies sequential testing; the interaction term is the last term to be added, but the order of NAP and exposure depends on how we specified the model. Change the order of NAP and exposure and we may get different p -values

for these two terms. The useful bit of this table is the last line, where it is testing whether the $\text{NAP} \times \text{exposure}$ interaction term is significant. The F -statistic is 3.32, and the p -value suggests it is not significant at the 5% level.

We can use the t -statistic as an alternative to the F -statistic. They are calculated in the same way as in linear regression, namely, the estimated value divided by its standard error. They are obtained with the `summary(Wrong4)` command, and the relevant output is given below.

Fixed effects: Richness ~ 1 + NAP * fExp					
	Value	Std.Error	DF	t-value	p-value
(Intercept)	9.118945	1.2242357	34	7.448684	0.0000
NAP	-3.879203	0.8816476	34	-4.399947	0.0001
fExp11	-5.534743	1.8510032	7	-2.990132	0.0202
NAP:fExp11	2.429496	1.3327641	34	1.822900	0.0771

The t -statistic also shows that we can drop the interaction term. Rerunning the model without the interaction term gives a t -statistic of $t = -2.44$ ($p = 0.04$), which is not convincing either. The new output is given below.

Fixed effects: Richness ~ 1 + NAP + fExp					
	Value	Std.Error	DF	t-value	p-value
(Intercept)	8.407714	1.183419	35	7.104595	0.0000
NAP	-2.808422	0.759642	35	-3.697034	0.0007
fExp11	-3.704917	1.517669	7	-2.441189	0.0447

Both the F -statistic and the t -statistic indicate a strong NAP effect, but a weak exposure effect and no significant interaction. Both these test are approximate. This means that we should not take them too literally. Hence, $p = 0.04$ is not convincing evidence of an exposure effect.

Before moving on to the ML testing procedure, we first need to address the issue of degrees of freedom. Within the mixed effects modelling literature, explanatory variables are divided into level 1 and level 2 variables. An explanatory variable that has the same value for all observations within the levels of the random effect is called a level 2 variable. An example is exposure; it has the same value for all observations on a beach. NAP, on the other hand, has a different value for each observation within a beach; it is called a level 1 variable. The degrees of freedom for a level 1 variable (NAP) in R is calculated as the number of level 1 observations (= 45) minus the number of level 2 clusters (= 9 levels in the random variable beach) minus the number of level 1 fixed effects (1, namely NAP). This explains the 35 degrees of freedom for NAP. For a level 2 variable, the equation is slightly different. It is calculated as the number of level 2 clusters (= 9 levels in the random variable beach) minus the number of level 2 fixed variables (In case only exposure) minus 1 if there is an intercept. This explains why the degrees of freedom are equal to 7 for exposure. Further details can be found on page 111 in West et al. (2006), Verbeke and Molenberghs (2000), or Pinheiro and Bates (2000).

So far, we only discussed the use of the approximate F -statistic and t -statistic. Estimation was done with REML. We now show the third hypothesis testing approach: the likelihood ratio test using ML estimation. In this approach, we fit two models with the same random effects structure using ML estimation and compare the likelihood criteria. The code below compares the model with a fixed structure containing NAP versus NAP + exposure. It also compares the model with NAP + exposure versus the model that also contains the interaction between both explanatory variables.

```
> lmc <- lmeControl(niterEM = 5200, msMaxIter = 5200)
> Wrong4A <- lme(Richness ~1 + NAP, method="ML",
+ control = lmc, data = RIKZ,
+ random = ~1 + NAP | fBeach)
> Wrong4B <- lme(Richness ~ 1 + NAP + fExposure,
+ random = ~1 + NAP | fBeach, method="ML",
+ data = RIKZ)
> Wrong4C <- lme(Richness ~1 + NAP * fExposure,
+ random = ~1 + NAP | fBeach, data = RIKZ,
+ method = "ML", control = lmc)
> anova(Wrong4A, Wrong4B, Wrong4C)
```

To avoid an error message related to convergence, we used the `control` option, which basically tells R to use more iterations. The output from the `anova` command is given below.

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
Wrong4A		1	6	246.6578	257.4977	-117.3289		
Wrong4B		2	7	245.3353	257.9820	-115.6677	1 vs 2	3.322437 0.0683
Wrong4C		3	8	243.2228	257.6761	-113.6114	2 vs 3	4.112574 0.0426

The comparison of model 1 versus 2 (Wrong4A versus Wrong4B) shows that exposure is not significant at the 5% level. Adding the interaction to a model that already contains exposure gives a log ratio statistic of $L = 4.11$, which is borderline significant ($p = 0.04$). So the ML testing procedure also indicates the interaction and exposure effects may be dropped from the model. This means that the optimal model, according to our model selection strategy, contains NAP as a fixed effect with a random slope and intercept. This means that the NAP effect is changing per beach (but in a random fashion).

5.8.1.3 Step 4 of the Protocol

As a last step, we need to present the numerical output of the optimal model using REML estimation. The code for the optimal model is

```
> Wrong5 <- lme(Richness~1+NAP,
                  random = ~1 + NAP | fBeach,
                  method = "REML", data = RIKZ)
> summary(Wrong5)

Random effects:
 Formula: ~1 + NAP | fBeach
          StdDev   Corr
(Intercept) 3.55     (Intr)
NAP         1.71    -0.988
Residual    2.69

Fixed effects: Richness ~ 1 + NAP
                Value Std.Error DF  t-value p-value
(Intercept)      6.59      1.26 35   5.20 <0.001
NAP            -2.83      0.72 35  -3.90 <0.001
```

5.8.2 *The Good Approach*

5.8.2.1 Step 1 of the Protocol

The top-down strategy specified earlier in this chapter indicated that we should start with as many explanatory variables as possible in the fixed component. So, we should start with a model that contains as fixed effects NAP, exposure, and their interaction. The starting point, therefore, is

```
> B1 <- gls(Richness ~ 1 + NAP * fExp,
              method = "REML", data = RIKZ)
> B2 <- lme(Richness ~1 + NAP * fExp, data = RIKZ,
              random = ~1 | fBeach, method = "REML")
> B3 <- lme(Richness ~ 1 + NAP * fExp, data = RIKZ,
              random = ~1 + NAP | fBeach, method="REML")
```

5.8.2.2 Step 2 of the Protocol

The AIC values of these three models are 238.53, 236.49, and 237.13. The random intercept model is therefore the preferred option.

5.8.2.3 Step 3 of the Protocol

The `summary(B2)` command indicates that all parameters in this model are significant as can be seen from the table below.

```
Fixed effects: Richness ~ 1 + NAP * fExp
              Value Std.Error DF   t-value p-value
(Intercept)  8.861084 1.0208449 34  8.680147  0.0000
NAP          -3.4463651 0.6278583 34 -5.516613  0.0000
fExp11       -5.255617 1.5452292  7 -3.401190  0.0114
NAP:fExp11   2.000464 0.9461260 34  2.114374  0.0419
```

If we use the same argument as above that a p -value of 0.04 is unconvincing, we could drop the interaction and refit the model. In that case, the p -value for exposure is 0.01, which is probably small enough to keep it in.

5.8.2.4 Step 4 of the Protocol

The results of the optimal model are given below.

```
Linear mixed-effects model fit by REML
Data: RIKZ
      AIC      BIC      logLik
 240.5538 249.2422 -115.2769

Random effects:
Formula: ~1 | fBeach
             (Intercept) Residual
StdDev:     1.907175 3.059089

Fixed effects: Richness ~ 1 + NAP + fExp
              Value Std.Error DF   t-value p-value
(Intercept)  8.601088 1.0594876 35  8.118158  0.0000
NAP          -2.581708 0.4883901 35 -5.286160  0.0000
fExp11       -4.532777 1.5755610  7 -2.876929  0.0238
```

Note that we end up with a fundamentally different model compared to our first approach above. The biological conclusion is also very different as this model suggests there is a strong NAP effect, a weak exposure effect, and absolute values differ per beach in a random way (as modelled by the random intercept).

The reason we ended up with a different model is because in the previous example, part of the information that we want to have in the fixed effects ended up in the random effects. This is due to starting with a fixed component that only contained NAP.

5.9 Model Validation

As with linear regression and additive modelling, the prime tool to validate the model is the normalised residuals based on the REML fit in step 4 of the protocol. These were defined in Chapter 4. Residuals should be plotted against fitted val-

ues to identify violation of homogeneity, indicated by differences in spread. If you do see an increase in spread for larger fitted values, then there are several options: (i) apply a transformation, (ii) check whether the increase in spread is due to a covariate, and (iii) apply generalised linear mixed modelling with a Poisson distribution (if the data are counts). If the increase in spread is due to a covariate, use the methods described in Chapter 4. These can easily be combined with a random effect.

You should also plot the residuals against each explanatory variable. Again, you do not want to see any patterns in the spread. Nor do you want to see a pattern in the residuals as it indicates the wrong model was applied. If this happens, consider adding more explanatory variables, interactions, quadratic terms, and if this does not help, use additive mixed modelling.

To verify normality, make histograms of the residuals. We recommend assessing normality (and homogeneity) using graphical tools. However, some software packages provide normality tests like the Shapiro-Wilks test, and these offer an alternative approach.

Examples of the model validation are given in the case studies and in the next section.

5.10 Begging Behaviour of Nestling Barn Owls

For those readers who enjoy television shows with many people in a house and cameras all over the place, here is the ecological version of it. Roulin and Bersier (2007) analysed the begging behaviour of nestling barn owls.

They looked at how nestlings responded to the presence of the father and of the mother. Using microphones inside and a video outside the nests, they sampled 27 nests and studied vocal begging behaviour when the parents brought prey. The number of nestlings was between 2 and 7 per nest.

Different response variables were defined in the paper: the amount of time spent on the perch by a parent, the amount of time in the nestbox, sibling negotiation, and begging. Here, we analyse sibling negotiation², which is defined as follows. Using the recorded footage, the number of calls made by all offspring in the absence of the parents was counted during 30-s time intervals every 15 min. To allocate a number of calls to a visit from a parent, the counted number of calls from the preceding 15 min of the arrival was used. This number was then divided by the number of nestlings. You may need to read this last sentence more than once, but in summary, the sibling

²When the need for food varies between the young owls, the calls used in the absence of parent birds have been shown to communicate the different levels of hunger between the chicks. This pre-parental arrival behaviour then seems to influence competitive behaviour between chicks when the parent bird arrives. Using information from this sibling communication, the least hungry chick avoids competing for food against the hungriest chick, which is the more likely to succeed in winning the food from the parent bird. Thus saving energy to only compete for food when there is the highest probability of successfully winning it.

negotiation is just the number of counted calls in the nearest 30-s interval before the arrival of a parent divided by the number of nestlings. In Chapters 12 and 13, we return to these data and analyse the number of calls using a Poisson distribution. We also use the data in Chapter 6 to model a more detailed auto-correlation structure.

The explanatory variables are sex of the parent, treatment of food, and arrival time of the parent. Half of the nests were given extra prey, and from the other half, prey (remaining) were removed. These were called ‘food-satiated’ and ‘food-deprived’, respectively. Measurements took place on two nights, and the food treatment was swapped on the second night. Note that the original paper contains an ethical note stating that food treatment did not have an effect on survival of the chicks. Measurements took place between 21.30 h and 05.30 h and the variable `ArrivalTime` reflects the time when a parent arrived at the perch with a prey. Further biological information and a description of the fascinating behaviour of barn owl nestlings can be found in the Roulin and Bersier (2007).

How should we analyse these data? Ok, given the fact that this is a section in a mixed effects modelling chapter, it should not be difficult to guess that nest will be used as a random effect. The reasons for this are as follows. Firstly, there were multiple observations from the same nests so these observations will be correlated. Secondly, there are 27 nests and using nest as a fixed effect would be rather expensive in terms of degrees of freedom. Furthermore, we would like to make a statement on relationships for barn owl nests in general and not just on these 27. If we use nest as a random effect, we allow for correlation between multiple observations from the same nest, and we only need to estimate one variance, and our statements will hold for all similar nests. Instead of starting immediately with a model that contains nest as a random effect, we will follow one of the protocols described earlier. We can either use the four-step protocol presented in Section 5.7 or the ten-step protocol discussed in Chapter 4. The later one has more intermediate steps, but basically does the same thing. Because the protocol from Chapter 4 is easier to follow (more detail, less chance to make mistakes), we use it here.

5.10.1 Step 1 of the Protocol: Linear Regression

We start with a linear regression model. Nestling negotiation is modelled as a function of sex of the parents, arrival time, and food treatment. Because one of the prime aims of the analysis is to find a sex effect, we also include the interaction between sex and each of the other variables. See also Appendix A for a discussion on interactions. The following R code imports the data, applies the linear regression model, and produces the graph in Fig. 5.4.

```
> library(AED) ; data(Owls)
> M.lm <- lm(NegPerChick ~ SexParent * FoodTreatment +
+               SexParent * ArrivalTime, data = Owls)
> plot(M.lm, select = c(1))
```

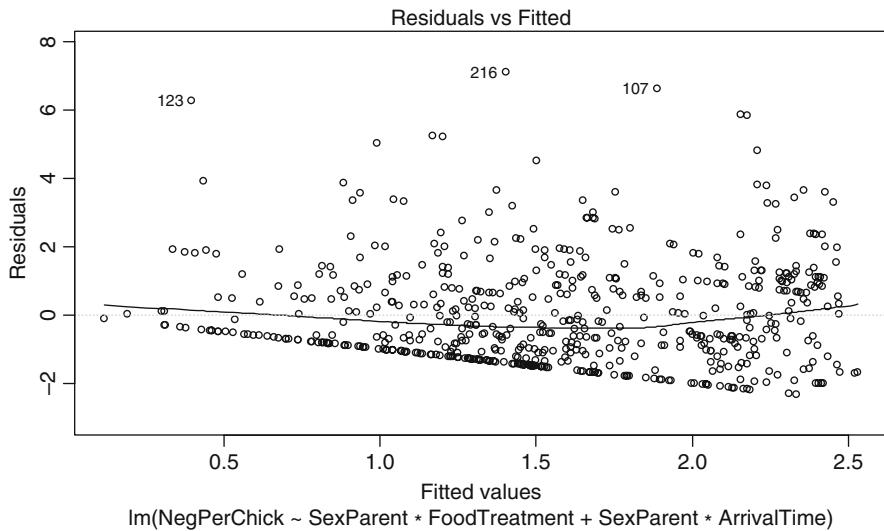


Fig. 5.4 Residuals versus fitted values for the linear regression model. Note that the residual spread increases for larger fitted values, indicating heterogeneity

The graph indicates heterogeneity because the residual spread increases along the horizontal axis. To understand why we have heterogeneity, we plotted residuals versus sex of the parents, food treatment, and arrival time.

However, as there is no clear pattern in any of these graphs, we cannot easily model the heterogeneity the way we did in Chapter 4. For this reason, we went for plan B and applied a $\log_{10}(Y + 1)$ transformation on the sibling negotiation data. This transformation was also used in Roulin and Bersier (2007). The code below applies the \log_{10} transformation, refits the model, and plots the residuals versus the nominal variable nest (Fig. 5.5).

```
> Owls$LogNeg <- log10(Owls$NegPerChick + 1)
> M2.1m <- lm(LogNeg ~ SexParent * FoodTreatment +
+                 SexParent * ArrivalTime, data = Owls)
> E <- rstandard(M2.1m)
> boxplot(E ~ Nest, data = Owls, axes = FALSE,
+           ylim = c(-3, 3))
> abline(0, 0); axis(2)
> text(1:27, -2.5, levels(Owls$Nest), cex=0.75, srt=65)
```

The `abline(0, 0)` command adds a horizontal line at $y = 0$. The `axes = FALSE` and `text` commands are used to add fancy labels along the horizontal axis. In a perfect world, the residuals should lie in a cloud around this line without any patterns. However, for some nests, all residuals are above or below the zero line, indicating that the term 'nest' has to be included in the model. We can do this as

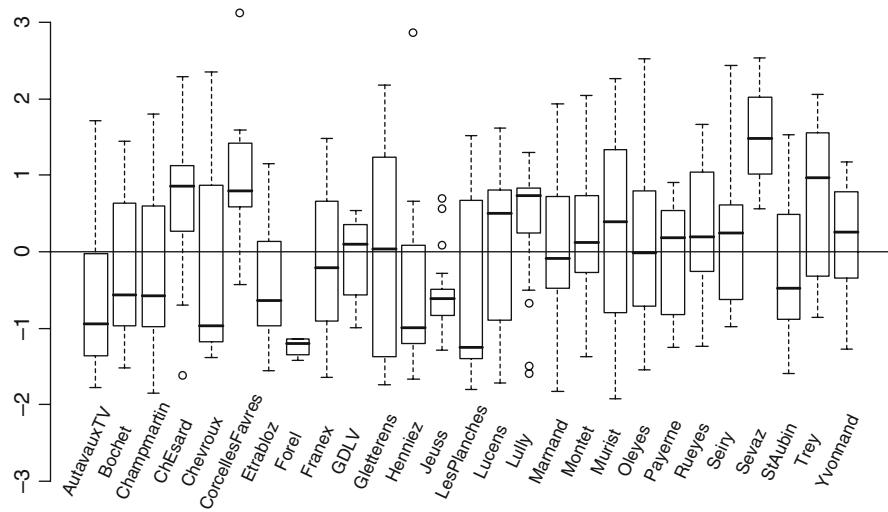


Fig. 5.5 Boxplot of standardised residuals obtained by a linear regression model applied on the log-transformed sibling negotiation data. The y-axis shows the values of the residuals and the horizontal axis the nests. Note that some nests have residuals that are above or below the zero line, indicating the need for a random effect

a fixed term or as a random term, but we already discussed that this has to be as a random term.

5.10.2 Step 2 of the Protocol: Fit the Model with GLS

In this step we fit the model using the `gls` function. It allows us to compare the linear regression model with the mixed effects model that we will calculate using the `lme` function in a moment.

```
> library(nlme)
> Form <- formula(LogNeg ~ SexParent * FoodTreatment +
+                     SexParent * ArrivalTime)
> M.gls <- gls(Form, data = Owls)
```

To reduce the code, we have used the formula expression. The numerical output in the object `M.gls` is identical to that of the `lm` function.

5.10.3 Step 3 of the Protocol: Choose a Variance Structure

In Chapter 4, this step consisted of finding the optimal variance structure in terms of heterogeneity. We can still do that here, but adding the random component nest is our first priority. Note that the random intercept is also part of the ‘choose a

variance structure' process. This means that the following random intercept mixed effects model is fitted.

$$\begin{aligned} \text{LogNeg}_{ij} = & \alpha + \beta_1 \times \text{SexParent}_{ij} + \beta_2 \times \text{Foodtreatment}_{ij} \\ & + \beta_3 \times \text{ArrivalTime}_{ij} + \beta_4 \times \text{SexParent}_{ij} \times \text{FoodTreatment}_{ij} \\ & + \beta_5 \times \text{SexParent}_{ij} \times \text{ArrivalTime}_{ij} + a_i + \varepsilon_{ij} \end{aligned}$$

LogNeg_{ij} is the log-10 transformed sibling negotiation for observation j at nest i . SexParent_{ij} and $\text{FoodTreatment}_{ij}$ are nominal variables with two levels, and ArrivalTime_{ij} is a continuous variable. The second line contains interactions. The term a_i is a random intercept and is assumed to be normally distributed with mean 0 and variance d^2 . The residual ε_{ij} is assumed to be normally distributed with mean 0 and variance σ^2 . Both random terms are assumed to be independent of each other.

5.10.4 Step 4: Fit the Model

The linear mixed effects model is applied in R with the following code.

```
> M1.lme <- lme(Form, random = ~ 1 | Nest,
                    method = "REML", data = Owls)
```

5.10.5 Step 5 of the Protocol: Compare New Model with Old Model

We use the `anova` command to compare the models `M.gls` and `M1.lme`. Note that the models were estimated with REML, which allows us to apply the likelihood ratio test to see whether we need the random intercept.

```
> anova(M.gls, M1.lme)
      Model df     AIC     BIC   logLik   Test  L.Ratio p-value
M.gls     1 7 64.37422 95.07058 -25.18711
M1.lme    2 8 37.71547 72.79702 -10.85773 1 vs 2 28.65875 <.0001
```

The likelihood ratio test indicates that the model with the random intercept is considerably better. You would quote this statistic in a paper as $L = 28.65$ ($df = 1, p < 0.001$). Recall from Section 5.8 that we are testing on the boundary here. If we did the correction for testing on the boundary, the p -value would get even smaller. Because the random intercept is highly significant, testing on the boundary is not a problem here.

The AIC of the model with the random intercept is also considerably smaller, confirming the results of the likelihood ratio test. As well as the random intercept, it is also an option to use a random intercept and random slope model. In this case, you assume that the strength of the relationship between sibling negotiation and arrival time changes randomly between the nests. We leave this as an exercise to the reader.

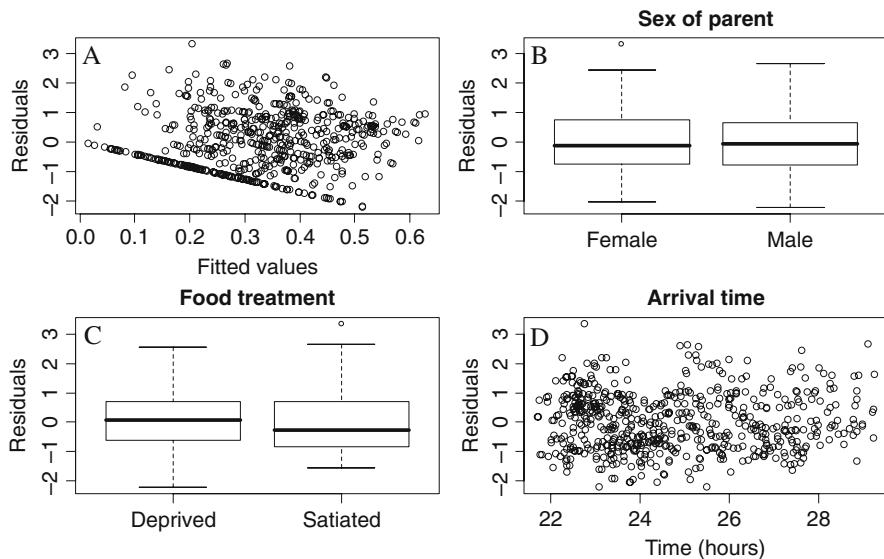


Fig. 5.6 Model validation graphs for the random intercept mixed effects model. Residuals are plotted versus fitted values (A), sex of the parent (B), food treatment (C), and arrival time (D)

5.10.6 Step 6 of the Protocol: Everything Ok?

The next thing we should think of is whether we have homogeneity of variance in the model and independence. Before doing anything, ask yourself whether you expect different residual spread per sex or per treatment (or over time). We have a large data set and blindly following some test statistics may not be wise. The large number of observations means that even small differences in spread may cause a significant variance covariate and we prefer to judge homogeneity by eye. Figure 5.6 shows residuals versus fitted values, sex, food treatment, and arrival time. These graphs do not show any clear violation of heterogeneity. There may be a violation of independence along arrival time, but Fig. 5.6D is not very clear. For the moment, we ignore any potential independence problems, and return to this issue later in this section. The R code to make Fig. 5.6 is as follows. Residuals and fitted values are extracted, a graph with four panels is set up, and the rest is a matter of trivial boxplot and plot commands.

```
> E2 <- resid(M1.lme, type = "normalized")
> F2 <- fitted(M1.lme)
> op <- par(mfrow = c(2, 2), mar = c(4, 4, 3, 2))
> MyYlab <- "Residuals"
> plot(x = F2, y = E2, xlab = "Fitted values", ylab = MyYlab)
> boxplot(E2 ~ SexParent, data = Owls,
           main = "Sex of parent", ylab = MyYlab)
```

```
> boxplot(E2 ~ FoodTreatment, data = Owls,
           main = "Food treatment", ylab = MyYlab)
> plot(x = Owls$ArrivalTime, y = E, ylab = MyYlab,
       main = "Arrival time", xlab = "Time (hours)")
> par(op)
```

5.10.7 Steps 7 and 8 of the Protocol: The Optimal Fixed Structure

In this step, we look at the optimal model in terms of the explanatory variables sex, food treatment, arrival time, and the selected interaction terms. The first thing we should do is to type `summary(M1.lme)` and inspect the significance of the regression parameters.

	Value	Std.Error	DF	t-value	p-value
(Intercept)	1.1236414	0.19522087	567	5.755744	0.0000
SexParentMale	0.1082138	0.25456854	567	0.425087	0.6709
FoodTreatmentSatiated	-0.1818952	0.03062840	567	-5.938776	0.0000
ArrivalTime	-0.0290079	0.00781832	567	-3.710251	0.0002
SexParMale:FoodTSatiated	0.0140178	0.03971071	567	0.352998	0.7242
SexParMale:ArrivalTime	-0.0038358	0.01019764	567	-0.376144	0.7070

Note, the interaction terms have been edited, to let the R printout fit on the page. Neither interaction term is significant. We could drop the least significant term, and reapply the model. Note that you should not use the `anova (M1.lme)` command as it applies sequential testing (which depends on the order of the two-way interaction terms). Its output is given below.

	numDF	denDF	F-value	p-value
(Intercept)	1	567	252.64611	<.0001
SexParent	1	567	1.52859	0.2168
FoodTreatment	1	567	71.43972	<.0001
ArrivalTime	1	567	37.13833	<.0001
SeParent:FoodTreatment	1	567	0.13472	0.7137
SexParent:ArrivalTime	1	567	0.14148	0.7070

The *p*-value of the last interaction term is the same as that obtained by the `summary` command. The third option, and our preferred one, is the likelihood ratio test. We need to fit the same model again, but now with ML. Both interaction terms can be dropped from the model. Using the likelihood ratio test, the significance of the dropped term is determined.

```
> M1.Full <- lme(Form, random =~ 1 | Nest,
                     method = "ML", data = Owls)
> M1.A <- update(M1.Full, .~. -SexParent:FoodTreatment)
> M1.B <- update(M1.Full, .~. -SexParent:ArrivalTime)
```

```
> anova(M1.Full, M1.A)

      Model df      AIC      BIC  logLik   Test L.Ratio p-value
M1.Full     1  8 -0.7484292 34.41366 8.374215
M1.A       2  7 -2.6246932 28.14214 8.312347 1 vs 2 0.123736  0.725
> anova(M1.Full, M1.B)

      Model df      AIC      BIC  logLik   Test L.Ratio p-value
M1.Full     1  8 -0.7484292 34.41366 8.374215
M1.B       2  7 -2.6103305 28.15650 8.305165 1 vs 2 0.1380986  0.7102
```

Recall that the update command takes all settings from the original lme command, and -SexParent:FoodTreatment means that this term is dropped from the model. We decided to omit the sex–food treatment interaction as it is the least significant. In the second round, we have a model that contains sex, food treatment, arrival time, and the interaction between sex and arrival time. There are two more terms that can be dropped from this model, the interaction term and food treatment.

```
> Form2 <- formula(LogNeg ~ SexParent + FoodTreatment +
                     SexParent * ArrivalTime)
> M2.Full <- lme(Form2, random= ~1| Nest, method= "ML",
                    data = Owls)
> M2.A <- update(M2.Full, .~. -FoodTreatment)
> M2.B <- update(M2.Full, .~. -SexParent:ArrivalTime)
> anova(M2.Full, M2.A)

      Model df      AIC      BIC  logLik   Test L.Ratio p-value
M2.Full     1  7 -2.62469 28.14214   8.312347
M2.A       2  6 65.52071 91.89228 -26.760355 1 vs 2 70.1454 <.0001

> anova(M2.Full, M2.B)

      Model df      AIC      BIC  logLik   Test L.Ratio p-value
M2.Full     1  7 -2.624693 28.14214 8.312347
M2.B       2  6 -4.476920 21.89465 8.238460 1 vs 2 0.1477732  0.7007
```

The interaction term sex–arrival time is not significant so this was also omitted. The new model contains the main terms sex, food treatment, and arrival time. We dropped them each in turn and applied the likelihood ratio test.

```
> Form3 <- formula(LogNeg ~ Sex-Parent + FoodTreatment +
                     ArrivalTime)
> M3.Full <- lme(Form3, random= ~1 | Nest,
                    method = "ML", data = Owls)
> M3.A <- update(M3.Full, .~. -FoodTreatment)
> M3.B <- update(M3.Full, .~. -SexParent)
> M3.C <- update(M3.Full, .~. -ArrivalTime)
```

```
> anova(M3.Full, M3.A)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full     1 6 -4.47692 21.89465  8.23846
M3.A       2 5 63.56865 85.54496 -26.78433 1 vs 2 70.04557 <.0001

> anova(M3.Full, M3.B)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full     1 6 -4.476920 21.89465 8.238460
M3.B       2 5 -5.545145 16.43116 7.772572 1 vs 2 0.9317755 0.3344

> anova(M3.Full, M3.C)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full     1 6 -4.47692 21.89465 8.23846
M3.C       2 5 29.71756 51.69387 -9.85878 1 vs 2 36.19448 <.0001
```

The term sex of the parent is not significant, and we omitted it from the model. In the next round, the model has the terms food treatment and arrival time. It is fitted with the following code. Each term is dropped in turn.

```
> Form4 <- formula(LogNeg ~ FoodTreatment + ArrivalTime)
> M4.Full <- lme(Form4, random= ~1 | Nest,
                     method = "ML", data = Owls)
> M4.A <- update(M4.Full, .~. -FoodTreatment)
> M4.B <- update(M4.Full, .~. -ArrivalTime)

> anova(M4.Full, M4.A)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M4.Full     1 5 -5.54514 16.43116  7.772572
M4.A       2 4 64.03857 81.61962 -28.019286 1 vs 2 71.58372 <.0001

> anova(M4.Full, M4.B)

      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M4.Full     1 5 -5.545145 16.43116 7.772572
M4.B       2 4 28.177833 45.75888 -10.088917 1 vs 2 35.72298 <.0001
```

Both food treatment and arrival time are significant at the 5% level and we have reached the end of the model selection process.

5.10.8 Step 9 of the Protocol: Refit with REML and Validate the Model

The model that we have selected above is of the form

$$\text{LogNeg}_{ij} = \alpha + \beta_2 \times \text{FoodTreatment}_{ij} + \beta_3 \times \text{ArrivalTime}_{ij} + a_i + \varepsilon_{ij}$$

The estimated parameters are obtained by the following R code.

```
> M5 <- lme(LogNeg ~ FoodTreatment + ArrivalTime,
    random= ~1 | Nest, method = "REML", data = Owls)
> summary(M5)

Linear mixed-effects model fit by REML
  AIC      BIC      logLik
15.07383 37.02503 -2.536915

Random effects:
Formula: ~1 | Nest
          (Intercept) Residual
StdDev:   0.0946877 0.2316398

Fixed effects: LogNeg ~ FoodTreatment + ArrivalTime
                Value Std.Error DF t-val p-val
(Intercept) 1.1821386 0.12897491 570 9.165648 0
FoodTrSatiated -0.1750754 0.01996606 570 -8.768650 0
ArrivalTime -0.0310214 0.00511232 570 -6.067954 0

Correlation:
          (Intr) FdTrts
FoodTreatmentSatiated -0.112
ArrivalTime            -0.984  0.039

Number of Observations: 599. Number of Groups: 27
```

The slope for food treatment is -0.175 . This means that sibling negotiation for an observation from an owl that was food satiated is -0.175 lower (on the log-10 scale) than a food deprived sibling. Indicating that siblings are quieter if they have more food. The slope for arrival time is -0.03 , which means that the later in the night the parents arrive, the lower the level of sibling negotiation.

As to the random effects, the random intercept a_i is normally distributed with mean 0 and variance 0.09^2 . The residual term ε_{ij} is normally distributed with mean 0 and variance 0.23^2 . These two variances can be used to calculate the correlation between observations from the same nest: $0.09^2/(0.09^2 + 0.23^2) = 0.13$. This is relatively low, but significant (as shown by the likelihood ratio test above).

Note that there is a high correlation between the intercept and the slope for arrival. This is because all arrival values are between 22 and 30 (recall that 30 is 06.00 AM). The intercept is the value of the response if all explanatory variables are 0 (or have the baseline value for a nominal variable), which is obviously far outside the range of the sampled arrival time values. A small change in the slope can therefore have a large change on the intercept, hence the high correlation. It would be better to centre arrival time around 0 and refit all models. Something like

```
> Owls$CArrivalTime <- Owls$ArrivalTime -
  mean(Owls$ArrivalTime)
```

will do the job. You can also use the `scale` function (with `center = TRUE` and `scale = FALSE`). In all the analyses presented in this section, you should then use `CArrivalTime`. We leave this as an exercise for the reader.

To validate the model, you should make a graph like Fig. 5.6. It is not presented here, but homogeneity seems a fair assumption. Independence will be discussed in a moment.

5.10.9 Step 10 of the Protocol

A biological discussion can be found in Roulin and Bersier (2007).

5.10.10 Sorry, We are Not Done Yet

Our optimal model contained food treatment as a nominal variable and arrival time as a continuous variable. We assumed independence because we cannot see a clear pattern if residuals are plotted versus arrival time; see also Fig. 5.6D. In Fig. 5.7, we made a multipanel plot with the `xyplot` from the `lattice` package. It shows the residuals of the optimal mixed effects model versus arrival time for each sex–food treatment combination. A LOESS smoother was added. This smoother should not show any pattern. Unfortunately, it raises some suspicion about a possible pattern. So, how do we know for sure there is no pattern in the residuals? The answer is to fit an additive mixed model. However, before we do this, we present the R code to make Fig. 5.7.

```
> library(lattice)
> xyplot(E2 ~ ArrivalTime | SexParent * FoodTreatment,
  data = Owls, ylab = "Residuals",
  xlab = "Arrival time (hours)",
  panel = function(x,y) {
    panel.grid(h = -1, v = 2)
    panel.points(x, y, col = 1)
    panel.loess(x, y, span = 0.5, col = 1, lwd=2) })
```

The R code to make multiple panel graphs with smoothers is discussed in various case studies, e.g. Chapters 13, 14, 15, 16, 17, and 18. Note that the argument(s) on the right hand side of the ‘|’ symbol are nominal variables. Due to the way we coded them in the data files, they are indeed nominal variables. If you coded them as numbers, use the `factor` command.

Before fitting the additive mixed model, we give the underlying equation.

$$\text{LogNeg}_{ij} = \alpha + \beta_2 \times \text{FoodTreatment}_{ij} + f(\text{ArrivalTime}_{ij}) + a_i + \varepsilon_{ij}$$

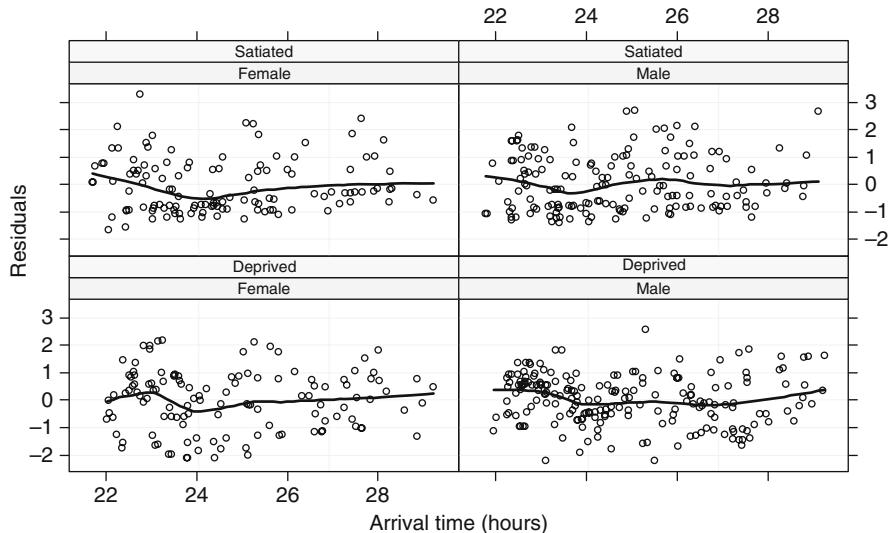


Fig. 5.7 Residuals versus arrival time for each sex–food treatment combination. A LOESS smoother with a span of 0.5 was fitted to aid visual interpretation

The term $\beta_3 \times \text{ArrivalTime}_{ij}$ has been replaced by $f(\text{ArrivalTime}_{ij})$, which is now a smoother (smoothing spline); see also Chapter 3. If the resulting shape of the smoother is a straight line, we know that in the model presented in step 9 of the protocol, arrival time has indeed a linear effect. However, if the smoother is not a straight line, the linear mixed effects model is wrong!

The following R code fits the additive mixed model.

```
> library(mgcv)
> M6 <- gamm(LogNeg ~ FoodTreatment + s(ArrivalTime),
  random = list(Nest =~ 1), data = Owls)
```

Formulation of the random intercept is slightly different and uses the list argument. Just do it, it's better not to ask why at this stage. Because no family argument is specified, the gamm function uses the Gaussian distribution. Other options are the Poisson, binomial, negative binomial, etc., and these will be discussed in Chapter 9. The output from gamm is slightly confusing. If you type `summary(M6)`, R gives:

	Length	Class	Mode
lme	18	lme	list
gam	25	gam	list

The object M6 has an lme component and a gam component. You can use the following commands:

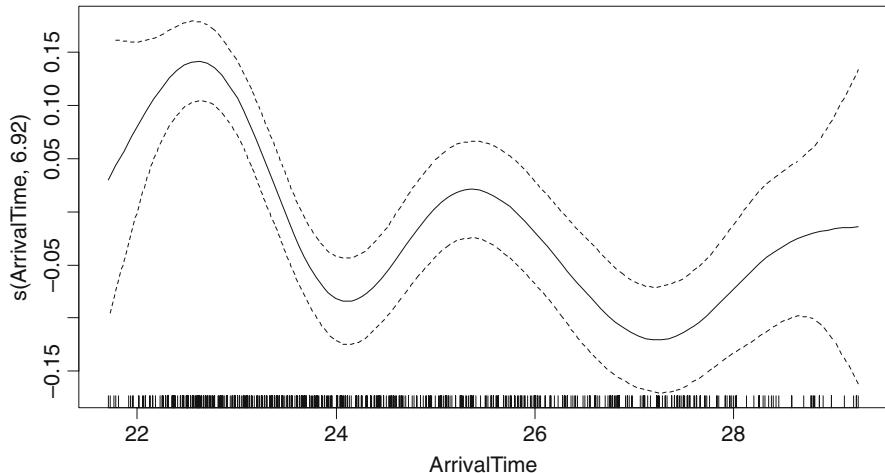


Fig. 5.8 Estimated smoother for the additive mixed model. The solid line is the estimated smoother and the dotted lines are 95% point-wise confidence bands. The horizontal axis shows the arrival time in hours (25 is 01.00 AM) and the vertical axis the contribution of the smoother to the fitted values. The smoother is centred around 0

- `summary(M6$gam)`. This gives detailed output on the smoothers and parametric terms in the models.
- `anova(M6$gam)`. This command gives a more compact presentation of the results as compared to the `summary(M6$gam)` command. The anova table is not doing sequential testing!
- `plot(M6$gam)`. This command plots the smoothers.
- `plot(M6$lme)`. This command plots the normalised residuals versus fitted values and can be used to assess homogeneity.
- `summary(M6$lme)`. Detailed output on the estimated variances. Not everything is relevant.

Good, let us now have a look at the shape of the smoother and see whether it is a straight line or not. The command `plot(M6$gam)` produces the smoother in Fig. 5.8 and indicates that it is bad news for the linear mixed effects model; the effect of arrival time is non-linear. The `summary(M6$gam)` gives the following output.

```

Family: Gaussian. Link function: identity
Formula: LogNeg ~ FoodTreatment + s(ArrivalTime)

Parametric coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.41379 0.02222 18.622 <2e-16
FoodTreaSatiated -0.17496 0.01937 -9.035 <2e-16

```

Approximate significance of smooth terms:

	edf	Est.rank	F	p-value
s(ArrivalTime)	6.921		9	10.26 8.93e-15
R-sq.(adj) =	0.184	Scale est.	= 0.049715	n = 599

The estimated regression parameter for food treatment is the similar to the one obtained by the linear mixed effects model. The smoother is significant and has nearly seven degrees of freedom! A straight line would have had one degree of freedom.

We also tried models with two smoothers using the `by` command (one smoother per sex or one smoother per treatment), but the AIC indicated that the model with one smoother was the best.

So, it seems that there is a lot of sibling negotiation at around 23.00 hours and a second (though smaller) peak at about 01.00–02.00 hours.

Chapter 6

Violation of Independence – Part I

This chapter explains how correlation structures can be added to the linear regression and additive model. The mixed effects models from Chapters 4 and 5 can also be extended with a temporal correlation structure. The title of this chapter contains ‘Part I’, suggesting that there is also a Part II. Indeed, that is the next chapter. In part I, we use regularly spaced time series, whereas in the next chapter, irregular spaced time series, spatial data, and data along an age gradient are analysed. We use a bird time series data set previously analysed in Reed et al. (2007). In the first section, we start with only one species and show how the linear regression model can be extended with a residual temporal correlation structure. In the second section, we use the same approach for a multivariate time series. In Section 6.3, the owl data are used again.

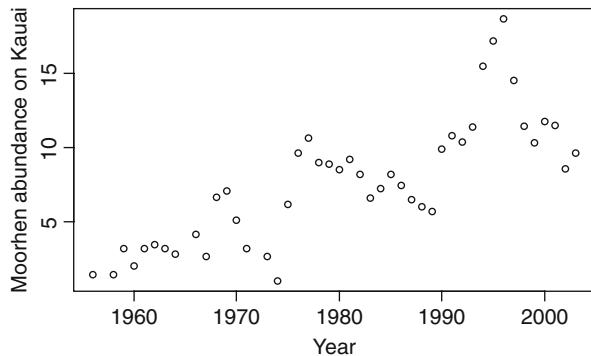
6.1 Temporal Correlation and Linear Regression

Reed et al. (2007) analysed abundances of three bird species measured at three islands in Hawaii. The data were annual abundances from 1956 to 2003. Here, we use one of these time series, moorhen abundance on the island of Kauai, to illustrate how to deal with violation of independence. A time series plot is given in Fig. 6.1. We applied a square root transformation to stabilise the variance, but strictly speaking, this is unnecessary as methods discussed earlier (Chapter 4) can be used to model the heterogeneity present in the original series. However, we do not want to over-complicate matters at this stage by mixing different concepts in the same model. The following R code imports the data and makes a plot of square-root-transformed moorhen numbers.

```
> library(AED); data(Hawaii)
> Hawaii$Birds <- sqrt(Hawaii$Moorhen.Kauai)
> plot(Hawaii$Year, Hawaii$Birds, xlab = "Year",
      ylab = "Moorhen abundance on Kauai")
```

Note that there is a general increase since the mid 1970s. Reed et al. (2007) used a dummy variable to test the effects of the implementation of new management

Fig. 6.1 Time series plot of square-root-transformed moorhen abundance measured on the island of Kauai



activities in 1974 on multiple bird time series, but to keep things simple, we will not do this here. The (transformed) abundance of birds is modelled as a function of annual rainfall and the variable Year (representing a long-term trend) using linear regression.

This gives a model of the form

$$\text{Birds}_s = \alpha + \beta_1 \times \text{Rainfall}_s + \beta_2 \times \text{Year}_s + \varepsilon_s \quad (6.1)$$

An alternative option is to use an additive model (Chapter 3) of the form:

$$\text{Birds}_s = \alpha + f_1(\text{Rainfall}_s) + f_2(\text{Year}_s) + \varepsilon_s$$

The advantage of the smoothers is that they allow for a non-linear trend over time and non-linear rainfall effects. Whichever model we use, the underlying assumption is that the residuals are independently normally distributed with mean 0 and variance σ^2 . In formula we have

$$\begin{aligned} \varepsilon_s &\sim N(0, \sigma^2) \\ \text{cov}(\varepsilon_s, \varepsilon_t) &= \begin{cases} \sigma^2 & \text{if } s = t \\ 0 & \text{else} \end{cases} \end{aligned} \quad (6.2)$$

The second line is due to the independence assumption; residuals from different time points are not allowed to covariate. We already discussed how to incorporate heterogeneity using variance covariates in Chapter 4. Now, we focus on the independence assumption. The underlying principle is rather simple; instead of using the ‘0 else’ in Equation (6.2), we model the auto-correlation between residuals of different time points by introducing a function $h(.)$:

$$\text{cor}(\varepsilon_s, \varepsilon_t) = \begin{cases} 1 & \text{if } s = t \\ h(\varepsilon_s, \varepsilon_t, \rho) & \text{else} \end{cases}$$

The function $h(\cdot)$ is called the correlation function, and it takes values between -1 and 1 . Just as Pinheiro and Bates (2000), we assume stationarity. This means we assume that the correlation between the residuals ε_s and ε_t only depends on their time difference $s - t$. Hence, the correlation between ε_s and ε_t is assumed to be the same as that between ε_{s+1} and ε_{t+1} , between ε_{s+2} and ε_{t+2} , etc. The task of the analyst is to find the optimal parameterisation of the function $h(\cdot)$, and we discuss several options in this and the next chapter. We assume the reader is familiar with the definition of the auto-correlation function, and how to estimate it from sample data; see for example Chatfield (2003), Diggle (1990), and Zuur et al. (2007), among others.

Before applying any model with a residual auto-correlation structure, we first apply the linear model without auto-correlation so that we have a reference point. In a preliminary analysis (not presented here), the cross-validation in the additive model gave one degree of freedom for each smoother, indicating that parametric models are preferred over smoothing models for this time series.

```
> library(nlme)
> M0 <- gls(Birds ~ Rainfall + Year,
+             na.action = na.omit, data = Hawaii)
> summary(M0)
```

We used the `gls` function without any correlation or weights option, and as a result it fits an ordinary linear regression model. The `na.action` option is required as the time series contains missing value. The relevant output produced by the `summary` command is given below:

```
Generalized least squares fit by REML
Model: Birds ~ Rainfall + Year
Data: Hawaii
      AIC      BIC    logLik
228.4798 235.4305 -110.2399

Coefficients:
            Value Std.Error  t-value p-value
(Intercept) -477.6634   56.41907 -8.466346 0.0000
Rainfall       0.0009    0.04989   0.017245 0.9863
Year          0.2450    0.02847   8.604858 0.0000

Residual standard error: 2.608391
Degrees of freedom: 45 total; 42 residual
```

The summary table shows that the effect of rainfall is not significant, but there is a significant increase in birds over time. The problem is that we cannot trust these p -values as we may be violating the independence assumption. The first choice to test this is to extract the standardised residuals and plot them against time (Fig. 6.2). Note that there is a clear pattern in the residuals.

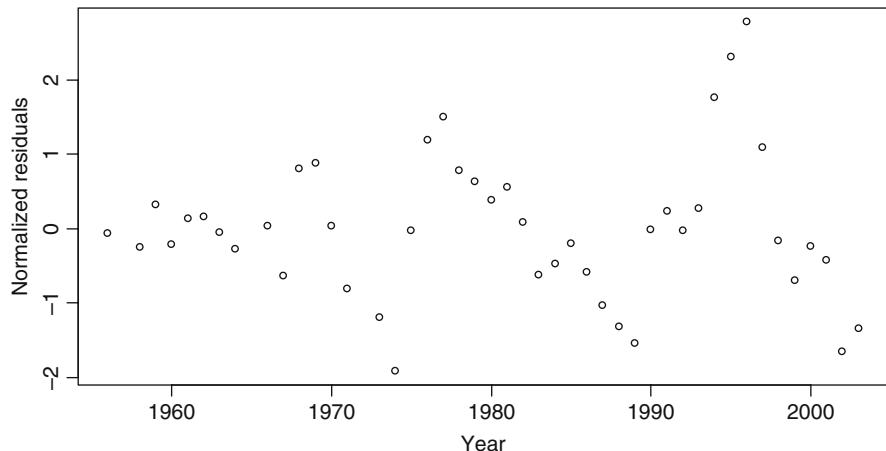


Fig. 6.2 Normalised residuals plotted versus time. Note the pattern in the residuals

A more formal visualisation tool to detect patterns is the auto-correlation function (ACF). The value of the ACF at different time lags gives an indication whether there is any auto-correlation in the data. The required R code for an ACF and the resulting graph are presented below. Note that the auto-correlation plot in Fig. 6.3 shows a clear violation of the independence assumption; various time lags have a significant

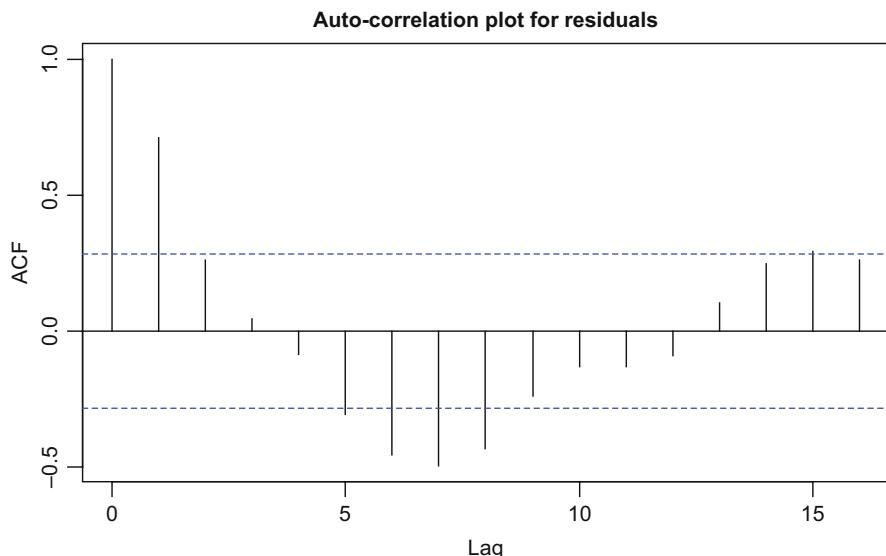


Fig. 6.3 Auto-correlation plot for the residuals obtained by applying linear regression on the Bird time series. Note that there is a clear indication of violation of independence

correlation! The ACF plot has a general pattern of decreasing values for the first 5 years, something we will use later in this section.

The R code for the ACF is given below.

```
> E <- residuals(M0, type = "normalized")
> I1 <- !is.na(Hawaii$Birds)
> Efull <- vector(length = length(Hawaii$Birds))
> Efull <- NA
> Efull[I1] <- E
> acf(Efull, na.action = na.pass,
      main = "Auto-correlation plot for residuals")
```

The function `residuals` extracts the normalised residuals. If there are no missing values, then you can just continue with `acf` (`E`), but it is not that easy here. The time series has two missing values and to ensure that the correlation function is correctly calculated, we need to insert the two missing values in the right place. This is because the `gls` function is removing the missing values, whereas the `acf` function assumes that the points are at the right time position. Once this is done, we can calculate the auto-correlation function and the resulting graph is presented in Fig. 6.3.

Figure 6.3 shows the type of pattern you do not want to see if you were hoping for a quick analysis; these data clearly contain residual correlation. As a result, we cannot assume that the F -statistic follows an F -distribution and the t -statistic a t -distribution.

An alternative approach to judge whether auto-correlation is present and one that does not depend on a visual judgement of the auto-correlation plot is to include an auto-correlation structure into the model. Then compare the models with and without an auto-correlation structure using the AIC, BIC, or if the models are nested, a likelihood ratio test. However, you should not spend too much time trying to find the optimal residual auto-correlation structure. Citing from Schabenberger and Pierce (2002): ‘In our experience it is more important to model the correlation structure in a reasonable and meaningful way rather than to model the correlation structure perfectly’. Similar statements can be found in Diggle et al. (2002), and Verbeke and Molenberghs (2000). We agree with this statement as differences in p -values for the F - and t -statistics obtained by using similar correlation structures tend to differ only marginally.

In Chapter 5, we used a slightly different mathematical notation compared to Equation (6.1); but if we use it here, the time series model for the birds in Equation (6.1) can be written as

$$\mathbf{Birds} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

The vector **Birds** contains all 58 bird observations, **X** is a matrix of dimension 58×3 , where the first columns consists of only ones, the second column the rainfall data, and the third column the years. The vector $\boldsymbol{\beta}$ is of dimension 3×1 , and contains α , β_1 , and β_2 . Finally, $\boldsymbol{\varepsilon}$ is equal to a vector of length 58 with the elements $(\varepsilon_{1958}, \dots, \varepsilon_{2003})$. Just as in Chapter 5, we can write $\text{Birds} \sim N(\mathbf{X} \times \boldsymbol{\beta}, \mathbf{V})$, where **V** is the covariance matrix of $\boldsymbol{\varepsilon}$. It is of the form

$$\mathbf{V} = \text{cov}(\boldsymbol{\varepsilon}) = \begin{pmatrix} \text{var}(\varepsilon_{1958}) & & & & \\ \text{cov}(\varepsilon_{1959}, \varepsilon_{1958}) & \text{var}(\varepsilon_{1959}) & & & \\ \text{cov}(\varepsilon_{1960}, \varepsilon_{1958}) & \text{cov}(\varepsilon_{1960}, \varepsilon_{1959}) & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ \text{cov}(\varepsilon_{2003}, \varepsilon_{1958}) & \text{cov}(\varepsilon_{2003}, \varepsilon_{1959}) & \cdots & \text{cov}(\varepsilon_{2003}, \varepsilon_{2002}) & \text{var}(\varepsilon_{2003}) \end{pmatrix}$$

Under the independence assumption, \mathbf{V} is a diagonal matrix of the form $\sigma^2 \times \mathbf{I}$, where \mathbf{I} is a 58×58 identity matrix. The easiest auto-correlation structure is the so-called compound symmetry structure. We have already met this correlation structure in Chapter 5. It assumes that whatever the distance in time between two observations, their residual correlation is the same. This can be modelled as

$$\text{cor}(\varepsilon_s, \varepsilon_t) = \begin{cases} 1 & \text{if } s = t \\ \rho & \text{else} \end{cases} \quad (6.3)$$

Hence, the correlation structure in Equation (6.3) is implying the following correlation matrix for $\boldsymbol{\varepsilon}$.

$$\text{cor}(\boldsymbol{\varepsilon}) = \begin{pmatrix} 1 & \rho & \rho & \cdots & \rho \\ \rho & 1 & \rho & \cdots & \rho \\ \rho & \rho & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \rho \\ \rho & \rho & \cdots & \rho & 1 \end{pmatrix}$$

This corresponds to the following covariance matrix \mathbf{V} , where $\rho = \theta/(\theta + \sigma^2)$.

$$\mathbf{V} = \text{cov}(\boldsymbol{\varepsilon}) = \begin{pmatrix} \theta + \sigma^2 & \theta & \theta & \cdots & \theta \\ \theta & \theta + \sigma^2 & \theta & \cdots & \theta \\ \theta & \theta & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \theta \\ \theta & \theta & \cdots & \theta & \theta + \sigma^2 \end{pmatrix}$$

Pinheiro and Bates (2000) mention that this correlation structure is often too simplistic for time series, but may still be useful for short time series. It can be implemented in R using the following code.

```
> M1 <- gls(Birds ~ Rainfall + Year,
  na.action = na.omit, data = Hawaii ,
  correlation = corCompSymm(form =~ Year))
```

The residual correlation structure is implemented using the `correlation` option in the `gls` function. The argument `corCompSymm` is the compound symmetry auto-correlation structure. The `form` argument within this argument is used to tell R that the order of the data is determined by the variable `Year`. However, due to the nature of the correlation structure, the `form` option is not needed (yet). Results of the `summary` command are not presented here, but give $AIC = 230.47$, $BIC = 239.16$, $\rho = 0$, and the estimated regression parameters and p -values are the same as for the ordinary linear regression model. So, we have made no improvements in the model.

The next structure we discuss is the AR-1 auto-correlation. This cryptic notation stands for an auto-regressive model of order 1. It models the residual at time s as a function of the residual of time $s - 1$ along with noise:

$$\varepsilon_s = \rho \varepsilon_{s-1} + \eta_s \quad (6.4)$$

The parameter ρ is unknown, and needs to be estimated from the data. It is relatively easy to show that this error structure results in the following correlation structure:

$$\text{cor}(\varepsilon_s, \varepsilon_t) = \begin{cases} 1 & \text{if } s = t \\ \rho^{|t-s|} & \text{else} \end{cases} \quad (6.5)$$

Suppose $\rho = 0.5$ and $t = s + 1$. The correlation between residuals separated by one unit in time is then 0.5. If the separation is two units in time, the correlation is $0.5^2 = 0.25$. Hence, the further away two residuals are separated in time, the lower their correlation. For many ecological examples, this makes sense. To emphasise the imposed correlation structure, we show the correlation matrix for $\boldsymbol{\varepsilon}$ again.

$$\text{cor}(\boldsymbol{\varepsilon}) = \begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \cdots & \rho^{57} \\ \rho & 1 & \rho & \ddots & \ddots & \vdots \\ \rho^2 & \rho & 1 & \ddots & \rho^2 & \rho^3 \\ \rho^3 & \rho^2 & \rho & \ddots & \rho & \rho^2 \\ \vdots & \ddots & \ddots & \ddots & 1 & \rho \\ \rho^{57} & \cdots & \rho^3 & \rho^2 & \rho & 1 \end{pmatrix}$$

The following code implements the AR-1 correlation structure.

```
> M2 <- gls(Birds ~ Rainfall + Year,
    na.action = na.omit, data = Hawaii,
    correlation = corAR1(form =~ Year))
> summary(M2)
```

The only thing that has changed compared to the compound symmetry structure is the correlation argument `corAR1`. The `form` argument is now essential as R needs to know position of the observations over time. The `na.action` option is also required due to the missing values. The relevant output obtained by the `summary` command is

```
Generalized least squares fit by REML
Model: Birds ~ Rainfall + Year. Data: Hawaii
      AIC      BIC    logLik
 199.1394 207.8277 -94.5697

Correlation Structure: ARMA(1,0)
  Formula: ~Year
Parameter estimate(s):
  Phil1
0.7734303

Coefficients:
            Value Std.Error t-value p-value
(Intercept) -436.4326 138.74948 -3.145472 0.0030
Rainfall      -0.0098   0.03268 -0.300964 0.7649
Year          0.2241   0.07009  3.197828 0.0026

Residual standard error: 2.928588
Degrees of freedom: 45 total; 42 residual
```

The parameter ρ is equal to 0.77. This means that residuals separated by one year have a correlation of 0.77; by two years it is $0.77^2 = 0.59$. This is rather high, but seems to be in line with the pattern for the first few years in the auto-correlation function in Fig. 6.3. The AIC indicates that the AR-1 correlation structure is a considerable model improvement compared to the linear regression model. In general, you would expect ρ to be positive as values at any particular point in time are positively related to preceding time points. Occasionally, you find a negative ρ . Plausible explanations are either the model is missing an important explanatory variable or the abundances go from high values in one year to low values in the next year.

6.1.1 ARMA Error Structures

The AR-1 structure can easily be extended to a more complex structure using an auto-regressive moving average (ARMA) model for the residuals. The ARMA model has two parameters defining its order: the number of auto-regressive parameters (p) and the number of moving average parameters (q). The notation $\text{ARMA}(1, 0)$ refers to the AR-1 model described above. The $\text{ARMA}(p, 0)$ structure is given by

$$\varepsilon_s = \phi_1 \varepsilon_{s-1} + \phi_2 \varepsilon_{s-2} + \phi_3 \varepsilon_{s-3} + \cdots + \phi_p \varepsilon_{s-p} + \eta_s \quad (6.6)$$

The residuals at time s are modelled as a function of the residuals of the p previous time points and white noise. In this case, the function $h(\cdot)$ does not have an easy formulation, see Equation (6.27) in Pinheiro and Bates (2000). The $\text{ARM}(0,q)$ is specified by

$$\varepsilon_s = \theta_1 \eta_{s-1} + \theta_2 \eta_{s-2} + \theta_3 \eta_{s-3} + \cdots + \theta_q \eta_{s-q} + \eta_s \quad (6.7)$$

And the $\text{ARMA}(p, q)$ is a combination of the two. You should realise that all these p and q parameters have to be estimated from the data, and in our experience, using values of p or q larger than 2 or 3 tend to give error messages related to convergence problems. Even for $p = q = 3$, it already becomes an art to find starting values so that the algorithm converges. Obviously, this also depends on the data, and how good the model is in terms of fixed covariates (year and rainfall in this case). The $\text{ARMA}(p, q)$ can be seen as a black box to fix residual correlation problems.

The implementation of the $\text{ARMA}(p, q)$ error structure in R is as follows.

```
> cs1 <- corARMA(c(0.2), p = 1, q = 0)
> cs2 <- corARMA(c(0.3, -0.3), p = 2, q = 0)
> M3arma1 <- gls(Birds ~ Rainfall + Year,
+                   na.action = na.omit,
+                   correlation = cs1, data = Hawaii)
> M3arma2 <- gls(Birds ~ Rainfall + Year,
+                   na.action = na.omit,
+                   correlation = cs2, data = Hawaii)
> AIC(M3arma1, M3arma2)
```

This code applies the $\text{ARMA}(1,0)$ and $\text{ARMA}(2,0)$ error structure. We chose arbitrary starting values. For larger values of p and q , you may need to change these starting values a little.

Finding the optimal model in terms of the residual correlation structure is then a matter of applying the model with different values of p and q . But remember the citation from Schabenberger and Pierce (2002) given at the start of this section; there is not much to be gained from finding the perfect correlation structure compared to finding one that is adequate. We tried each combination of $p = 0, 1, 2, 3$ and $q = 0, 1, 2, 3$, and each time we wrote down the AIC. Because not all the models are nested, we cannot apply a likelihood ratio test and have therefore based our model selection on the AIC. The lowest AICs were obtained by the $\text{ARMA}(2,0)$ and $\text{ARMA}(2,3)$ models and were 194.5 and 194.1, respectively. Both AICs differed only in the first decimal, and we selected the $\text{ARMA}(2,0)$ model as it is considerably less complex than the $\text{ARMA}(2,3)$ model. Recall that the linear regression model without a residual auto-correlation structure had $\text{AIC} = 228.47$, and the AR-1 structure gave $\text{AIC} = 199.13$. So, going from no residual correlation to an AR-1 structure gave a large improvement, while the more complicated structures gave only a marginal

improvement. The estimated auto-regressive parameters of the ARMA(2,0) model were $\varphi_1 = 0.99$ and $\varphi_2 = -0.35$. The value for φ_1 close to 1 may indicate a more serious problem of the residuals being non-stationary (non-constant mean or variance). Note that the auto-correlation function in Fig. 6.3 becomes positive again for larger time lags. This suggests that an error structure that allows for a sinusoidal pattern may be more appropriate.

The correlation structure can also be used for generalised additive models, and it is also possible to have a model with residual correlation and/or heterogeneity structures.

6.2 Linear Regression Model and Multivariate Time Series

Figure 6.4 shows the untransformed abundances of two bird species (stilts and coots) measured on the islands Maui and Oahu. These time series form part of a larger data set analysed in Reed et al. (2007), but these four series are the most complete. Again, we use annual rainfall and year as explanatory variables to model bird abundances. Preliminary analyses suggested a linear rainfall effect that was the same for all four time series and a non-linear trend over time. Hence, a good starting model is

$$\text{Birds}_{is} = \alpha_i + \beta \times \text{Rainfall}_{is} + f_i(\text{Year}_s) + \varepsilon_{is} \quad (6.8)$$

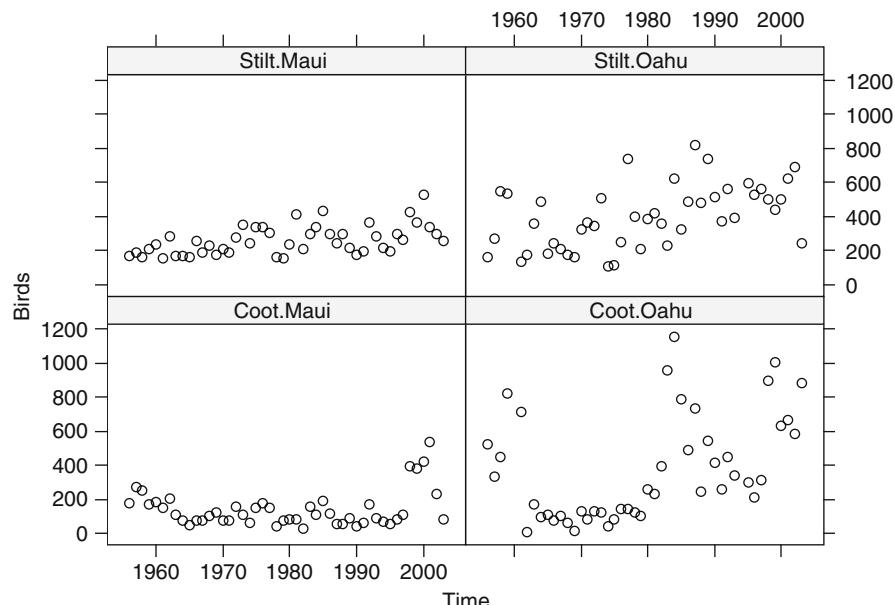


Fig. 6.4 Time series of (untransformed) silt and coot abundances on the islands of Maui and Oahu

$Bird_{is}$ is the value of time series i ($i = 1, \dots, 4$) in year s ($s = 1, \dots, 48$). For the moment, we treat the time series for the two species and two islands as different time series. The intercept α_i allows for a different mean value per time series. An extra motivation to use no rainfall–species or rainfall–island interaction is that some intermediate models had numerical problems with the interaction term. $Year_s$ is the year and $f_i(Year_s)$ is a smoother for each species–island combination. If we remove the index i , then all four time series are assumed to follow the same trend.

The range of the y -axes in the lattice plot immediately indicates that some species have considerably more variation, indicating violation of homogeneity. The solution is to allow for different spread per time series.

The following code (i) imports the data into R, (ii) creates the lattice graph in Fig. 6.4, and (iii) applies the model in Equation (6.8).

```
> library(AED); data(Hawaii)
> Birds <- c(Hawaii$Stilt.Oahu, Hawaii$Stilt.Maui,
   Hawaii$Coot.Oahu, Hawaii$Coot.Maui)
> Time <- rep(Hawaii$Year, 4)
> Rain <- rep(Hawaii$Rainfall, 4)
> ID <- factor(rep(c("Stilt.Oahu", "Stilt.Maui",
   "Coot.Oahu", "Coot.Maui"),
   each = length(Hawaii$Year)))
> library(lattice)
> xyplot(Birds ~ Time | ID, col = 1)
> library(mgcv)
> BM1<-gamm(Birds ~ Rain + ID +
  s(Time, by = as.numeric(ID == "Stilt.Oahu")) +
  s(Time, by = as.numeric(ID == "Stilt.Maui")) +
  s(Time, by = as.numeric(ID == "Coot.Oahu")) +
  s(Time, by = as.numeric(ID == "Coot.Maui")),
  weights = varIdent(form =~ 1 | ID))
```

The first line imports the data. The next line stacks all four time series and calls it ‘Birds’. Obviously, we also have to stack the variables Year and Rainfall, and the `rep` command is a useful tool for this. Finally, we need to make sure we know which observation belongs to which time series, and this is done using the variable ‘ID’. The familiar `xyplot` command from the lattice package draws Fig. 6.4. The interested reader can find information on how to add gridlines, connect the dots, etc., in other parts of this book. The model in Equation (6.8) is an additive model with Gaussian distribution. The `weights` option with the `varIdent` argument was discussed in Chapter 4. Recall that it implements the following variance structure:

$$\varepsilon_s \sim N(0, \sigma_i^2) \quad i = 1, \dots, 4 \quad (6.9)$$

Each time series is allowed to have a different residual spread. The `by = as.numeric(.)` command ensures that each smoother is only applied on one

time series. The same model could have been fitted with the `gam` command instead of the `gamm`, but our choice allows for a comparison with what is to come.

The numerical output for the smoothing model is given below.

```
> summary(BM1$gam)

Family: gaussian. Link function: identity
Formula: Birds ~ Rain + ID +
           s(Time, by = as.numeric(ID == "Stilt.Oahu")) +
           s(Time, by = as.numeric(ID == "Stilt.Maui")) +
           s(Time, by = as.numeric(ID == "Coot.Oahu")) +
           s(Time, by = as.numeric(ID == "Coot.Maui"))

Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 225.3761    20.0596 11.235 < 2e-16
Rain        -4.5017     0.8867 -5.077 9.93e-07
IDCoot.Oahu 237.7378    30.3910  7.823 5.06e-13
IDStilt.Maui 117.1357    14.9378  7.842 4.53e-13
IDStilt.Oahu 257.4746    27.1512  9.483 < 2e-16

Approximate significance of smooth terms:
                                         edf Est.rank      F p-value
s(Time):as.numeric(ID == "Stilt.Oahu") 1.000          1 13.283 0.000355
s(Time):as.numeric(ID == "Stilt.Maui") 1.000          1 20.447 1.14e-05
s(Time):as.numeric(ID == "Coot.Oahu")   6.660          9  8.998 4.43e-11
s(Time):as.numeric(ID == "Coot.Maui")   2.847          6  3.593 0.002216

R-sq.(adj) =  0.813  Scale est. = 26218      n = 188
```

The problem here is that the p -values assume independence and because the data are time series, these assumptions may be violated. However, just as for the univariate time series, we can easily implement a residual auto-correlation structure, for example, the AR-1:

$$\varepsilon_{is} = \rho \varepsilon_{i,s-1} + \eta_{is} \quad (6.10)$$

As before, this implies the following correlation structure:

$$\text{cor}(\varepsilon_{is}, \varepsilon_{it}) = \begin{cases} 1 & \text{if } s = t \\ \rho^{|t-s|} & \text{else} \end{cases} \quad (6.11)$$

The correlation between residuals of different time series is assumed to be 0. Note that the correlation is applied at the deepest level: Observations of the same time series. This means that all time series have the same ρ . The following R code implements the additive model with a residual AR-1 correlation structure.

```
> BM2 <- gamm(Birds ~ Rain + ID +
               s(Time, by = as.numeric(ID == "Stilt.Oahu")) +
               s(Time, by = as.numeric(ID == "Stilt.Maui")) +
```

```

s(Time, by = as.numeric(ID == "Coot.Oahu")) +
s(Time, by = as.numeric(ID == "Coot.Maui")),
correlation = corAR1(form =~ Time | ID),
weights = varIdent(form = ~1 | ID))
> AIC(BM1$lme, BM2$lme)

```

The only new piece of is the `correlation = corAR1` (`form = ~Time | ID`). The `form` option specifies that the temporal order of the data is specified by the variable `Time`, and the time series are nested. The auto-correlation is therefore applied at the deepest level (on each individual time series), and we get one ρ for all four time series. The AIC for the model without auto-correlation is 2362.14 and with auto-correlation it is 2351.59, which is a worthwhile reduction. The `anova(BM2$gam)` command gives the following numerical output for the model with AR-1 auto-correlation.

```

Parametric Terms:
  df      F p-value
Rain   1 18.69 2.60e-05
ID     3 20.50 2.08e-11

```

```
Approximate significance of smooth terms:
```

	edf	Est.rank	F	p-value
<code>s(Time) : as.numeric(ID == "Stilt.Oahu")</code>	1.000	1.000	27.892	3.82e-07
<code>s(Time) : as.numeric(ID == "Stilt.Maui")</code>	1.000	1.000	1.756	0.187
<code>s(Time) : as.numeric(ID == "Coot.Oahu")</code>	6.850	9.000	22.605	< 2e-16
<code>s(Time) : as.numeric(ID == "Coot.Maui")</code>	1.588	4.000	1.791	0.133

The Oahu time series have a significant long-term trend and rainfall effect, whereas the Maui time series are only affected by rainfall. The `plot(BM2$gam, scale = FALSE)` command produces the four panels in Fig. 6.5. Note that the smoothers in panels B and D are not significant. Further model improvements can be obtained by dropping these two smoothers from the model.

The long-term trend for stilts on Oahu (panel A) is linear, but the coots on Oahu show a non-linear trend over time. Abundances are increasing from the early 1970s onwards. The results from the `summary(BM2$gam)` command are not shown, but indicate that the rainfall effect is negative and highly significant ($p < 0.001$). The adjusted R^2 is 0.721. The `summary(BM2$lme)` results are not shown either, but give $\rho = 0.32$, large enough to keep it in the model.

The normalised residuals are plotted versus time in Fig. 6.6. The stilt residuals at Maui show some evidence of heterogeneity over time. It may be an option to use the `varComb` option to allow for heterogeneity per time series (as we have done here) but also along time, see Chapter 4. We leave this as an exercise for the reader. If you do attempt to apply such a model, it would make sense to remove the square root transformation. Figure 6.5 was created using the following R code.

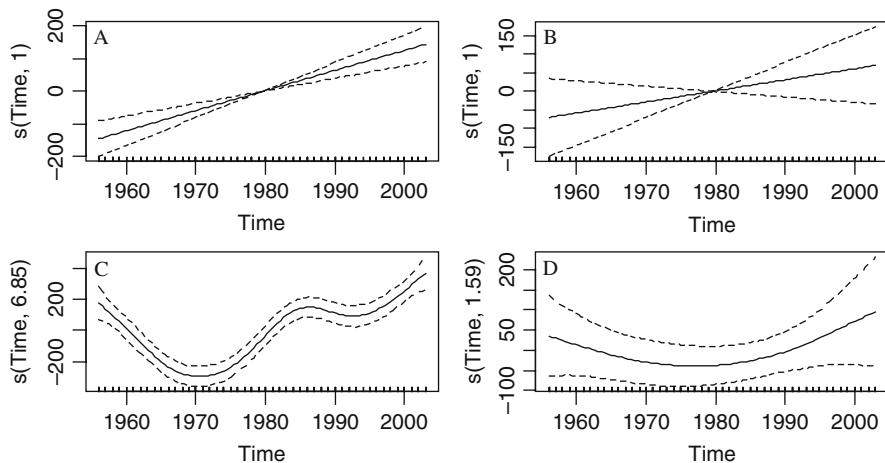


Fig. 6.5 A: Significant smoother for stilts in Oahu showing a linear increase over time. B: Non-significant smoother for stilts on Maui. C: Significant smoother for coots on Oahu. D: Non-significant smoother for coots on Maui. The four panels were created with the `par (mfrow = c (2 , 2))` command before the plot command

```
> E2 <- resid(BM2$lme, type = "normalized")
> EAll <- vector(length = length(Birds))
> EAll[] <- NA
> I1 <- !is.na(Birds)
> EAll[I1] <- E2
> library(lattice)
> xyplot(EAll ~ Time | ID, col = 1, ylab = "Residuals")
```

The only difficult aspect of the R code is dealing with missing values. Our approach is to create a vector `EAll` of length 192, fill in missing values, and fill in the matching values of the residuals `E2` at the right places (i.e. where we do not have missing values).

We need to investigate one last aspect. The model we applied above assumes that residuals are normally distributed with a variance that differs per time series and allows for auto-correlation within a time series. But, we also assume there is no correlation of residuals for different time series. This assumption could be violated if birds on one island are affecting those on other islands. Or there may be other biological reasons why the residual patterns of different time series are correlated. Whatever the biological reason, we need to verify this assumption. This is done by calculating the correlation coefficients between the four residual time series. If these correlation coefficients are reasonably small, we can assume independence between residuals of different time series. The following code extracts the residuals per time series, calculates an auto-correlation function, and a 4-by-4 correlation matrix.

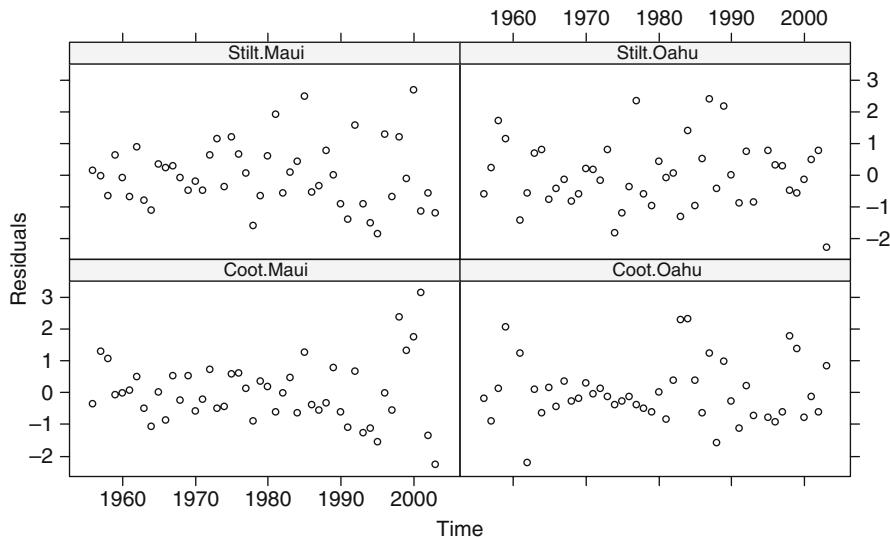


Fig. 6.6 Normalised residuals obtained by the additive model that allows for heterogeneity and an AR-1 residual error structure. The residual spread for the stilt at Oahu are perfect, but the residual spread for the stilts at Maui show a clear increase. One can argue about the interpretation of coot residual patterns

```

> E1 <- EA11[ID == "Stilt.Oahu"]
> E2 <- EA11[ID == "Stilt.Maui"]
> E3 <- EA11[ID == "Coot.Oahu"]
> E4 <- EA11[ID == "Coot.Maui"]
> par(mfrow = c(2, 2))
> acf(E1, na.action = na.pass)
> acf(E2, na.action = na.pass)
> acf(E3, na.action = na.pass)
> acf(E4, na.action = na.pass)
> D <- cbind(E1, E2, E3, E4)
> cor(D, use = "pairwise.complete.obs")

```

Results are not presented here, but all correlation coefficients are smaller than 0.2, except for the correlation coefficient between stilts and coots on Maui ($r = 0.46$). This may indicate that the model is missing an important covariate for the Maui time series. The three options are (i) find the missing covariate and put it into the model, (ii) extend the residual correlation structure by allowing for the correlation, and (iii) ignore the problem because it is only one out of the six correlations, and all p -values in the model were rather small (so it may have little influence on the conclusions). If more than one correlation has a high values, option (iii) should not be considered. You could try programming your own correlation structure allowing for spatial *and* temporal correlation.

6.3 Owl Sibling Negotiation Data

In Section 5.10, we analysed the owl sibling negotiation data. The starting point was a model of the form:

$$\begin{aligned} \text{LogNeg}_{is} = & \alpha + \beta_1 \times \text{SexParent}_{is} + \beta_2 \times \text{FoodTreatment}_{is} + \\ & \beta_3 \times \text{ArrivalTime}_{is} + \beta_4 \times \text{SexParent}_{is} \times \text{FoodTreatment}_{is} + \\ & \beta_5 \times \text{SexParent}_{is} \times \text{ArrivalTime}_{is} + \varepsilon_{is} \end{aligned}$$

LogNeg_{is} is the log transformed sibling negotiation at time s in nest i . Recall that we used nest as a random intercept, and therefore, the compound correlation structure was imposed on the observations from the same nest. We can get the same correlation structure (and estimated parameters) by specifying this correlation structure explicitly with the R code:

```
> library(AED) ; data(Owls)
> library(nlme)
> Owls$LogNeg <- log10(Owls$NegPerChick + 1)
> Form <- formula(LogNeg ~ SexParent * FoodTreatment +
+                     SexParent * ArrivalTime)
> M2.gls <- gls(Form, method = "REML", data = Owls,
+                  correlation = corCompSymm(form =~ 1 | Nest))
```

You will see that the `summary(M2.gls)` command produces exactly the same estimated parameters and correlation structure compared to the random intercept model presented in Section 5.10. The `summary` command gives an estimated correlation of 0.138. Hence, the correlation between any two observations from the same nest i is given by

$$\text{cor}(\varepsilon_{is}, \varepsilon_{it}) = 0.138$$

It is important to realise that both random intercept and compound correlation models assume that the correlation coefficient between any two observations from the same nest are equal, whether the time difference is 5 minutes or 5 hours. Based on the biological knowledge of these owls, it is more natural to assume that observations made close to each other in time are more similar than those separated further in time. This sounds like the auto-regressive correlation structure of order 1, which was introduced in Section 6.1, and is given again below.

$$\text{cor}(\varepsilon_{is}, \varepsilon_{it}) = \rho^{|t-s|}$$

There are two ‘little’ problems. The numbers below are the first 12 lines of the data file and were obtained by typing

```
> Owls[Owls$Nest=="AutavauxTV",1:5]
```

	Nest	FoodTreatment	SexParent	ArrivalTime	SiblingNegotiation
1	AutavauxTV	Deprived	Male	22.25	4
2	AutavauxTV	Satiated	Male	22.38	0
3	AutavauxTV	Deprived	Male	22.53	2
4	AutavauxTV	Deprived	Male	22.56	2
5	AutavauxTV	Deprived	Male	22.61	2
6	AutavauxTV	Deprived	Male	22.65	2
7	AutavauxTV	Deprived	Male	22.76	18
8	AutavauxTV	Satiated	Female	22.90	4
9	AutavauxTV	Deprived	Male	22.98	18
10	AutavauxTV	Satiated	Female	23.07	0
11	AutavauxTV	Satiated	Female	23.18	0
12	AutavauxTV	Deprived	Female	23.28	3

The experiment was carried out on two nights, and the food treatment changed. Observations 1 and 2 were made at 22.25 and 22.38 hours, but the time difference between them is not 13 minutes, but 24 hours and 13 minutes! So, we have to be very careful where we place the auto-regressive correlation structure. It should be within a nest on a certain night. The random intercept and the compound correlation models place the correlation within the same nest, irrespective of the night.

The second problem is that the observations are not regularly spaced, at least not from our point of view; see Fig. 6.7. However, from the owl parent's point of view, time between visits may be regularly spaced. With this we mean that it may well be possible that the parents chose the nest visiting times. Obviously, if there is not enough food, and the parents need a lot of effort or time to catch prey, this is not a valid assumption. But if there is a surplus of food, this may well be a valid assumption. For the sake of the example, let us assume the owls indeed chose the times, and therefore, we consider the longitudinal data as regularly spaced. This basically means that we assume that distances (along the time axis) between the vertical lines in Fig. 6.7 are all the same. A similar approach was followed in Ellis-Iversen et al. (2008). Note that this is a biological assumption.

In this scenario, we can consider the visits at a nest on a particular night as regular spaced and apply the models with an auto-regressive correlation structure, e.g. the corAR1 structure. The following R code does the job (the first few lines are used for Fig. 6.7):

```
> library(lattice)
> xyplot(LogNeg ~ ArrivalTime | Nest, data = Owls,
+ type = "h", col = 1, main = "Deprived",
+ subset = (FoodTreatment == "Deprived"))
> M3.gls <- gls(Form, method = "REML", data = Owls,
+ correlation = corAR1(form =~ 1 |
+ FoodTreatment))
```

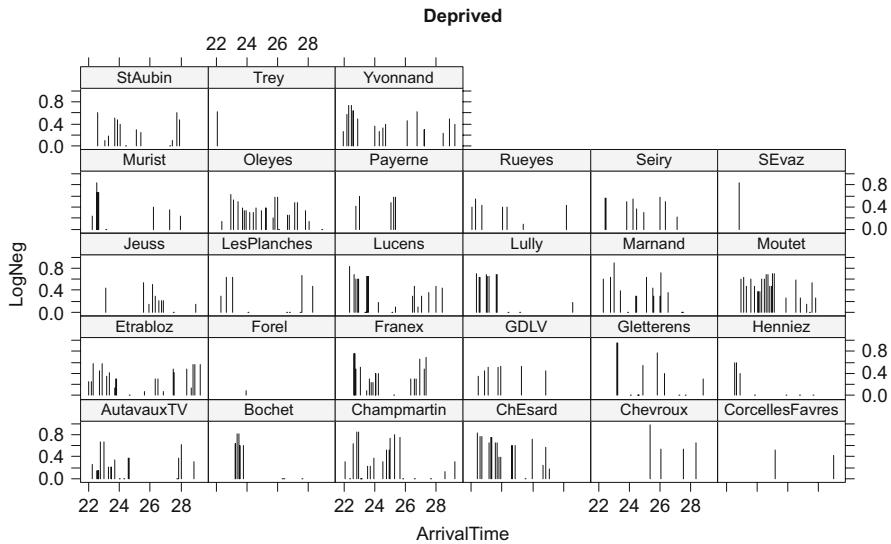


Fig. 6.7 Log-transformed sibling negotiation data versus arrival time. Each panel shows the data from one nest on a particular night. A similar graph can be made for the satiated data. R code to make this graph is given in the text

The variables `FoodTreatment` and `Nest` identify the group of observations from the same night, and the correlation is applied within this group. As a result, the index i in the model does not represent nest, but night in the nest. The `summary(M3.gls)` command shows that the estimated auto-correlation is 0.418, which is relatively high. The whole 10-step protocol approach can now be applied again: first chose the optimal random structure and then the optimal fixed structure. You can also choose to model arrival time as a smoother, just as we did in Section 5.8. This gives a GAM with auto-correlation.

The model with the auto-regressive correlation structure assumes that observations from different nests are independent and also that the observations from the same nest on two different nights are independent. It may be an option to extend the model with the AR1 correlation structure with a random intercept nest. Such a model allows for the compound correlation between all observations from the same nest, and temporal correlation between observations from the same nest *and* night. But the danger is that the random intercept and auto-correlation will fight with each other for the same information. These types of models are also applied in Chapter 17, where station is used as a random intercept and a correlation structure is applied along depth, but *within* the station.

Chapter 7

Violation of Independence – Part II

In the previous chapter, we discussed violation of independence for measurements taken repeatedly over time and how temporal correlation structures can be added to linear regression and additive models. We used a regular spaced data set. In this chapter, we consider data measured at multiple spatial locations, and we show how similar correlation structures can be used. The ‘Part II’ in the title refers to irregular spaced data, either in space, time, or along an age or depth gradient. The general principle with spatial data is that things that are close to each other are likely to be more similar than things that are further apart (Tobler, 1979).

In this chapter, we use various examples. The first example uses data obtained at multiple spatial locations. We then revisit the Hawaiian bird data and show how to add spatial correlation to time series models. We also present an example where a correlation structure along an age gradient is required. In Section 7.5, we discuss the possibility that spatial correlation may be due to missing covariates. In the final section, we analyse data from a bird behavioural study, but this time with short longitudinal (temporal) measurements.

7.1 Tools to Detect Violation of Independence

In this section, we use data from Chapter 37 in Zuur et al. (2007). The case study in that chapter illustrates the application of spatial analysis methods on a boreal forest in Tatarstan, Russia. Using remotely sensed data and spatial statistical methods, they explored the influence of relief, soil, and climatic factors on the forests of the Raifa section of Volzhsko-Kamsky State Nature Biosphere. The response variable is a boreal forest index and is defined as the number of species that belong to a set of boreal species divided by the total number of species at a site. Several remotely sensed variables derived from the LANDSAT 5 satellite images were used as explanatory variables: (i) the normalised difference vegetation index, (ii) temperature, (iii) an index of wetness, and (iv) an index of greenness. A data exploration indicated high collinearity between these variables, and we therefore only used wetness. In addition to these variables, we also know the latitude (X) and longitude (Y) of each site. Boreality was transformed using the following transformation:

$$z_i = (1000 \times (S_i + 1)/n_i)^{1/2}$$

where z_i is transformed boreality, S_i is the number of species that belong to boreal coenosis species, and n_i is the number of all species at the site i . See Cressie (p. 395, 1993) for a discussion of this transformation.

In Chapter 6, we started by applying a model without a temporal correlation structure and used graphical tools to asses violation of independence. As a second step, we made an auto-correlation (ACF) of the residuals, and finally we added a temporal correlation structure to the regression and GAM models. The same can be done for spatial data. We first fit the following linear regression model.

$$z_i = \alpha + \beta \times \text{Wetness}_i + \varepsilon_i$$

where z_i is the transformed boreality, Wetness_i is the wetness at site i , and the index $i = 1, \dots, 533$. The following R code imports the data and applies the transformation and linear regression.

```
> library(AED); data(Boreality)
> Boreality$Bor <- sqrt(1000 * (Boreality$nBor + 1) /
+                           (Boreality$nTot))
> B.lm <- lm(Bor ~ Wet, data = Boreality)
> summary(B.lm)
```

The results from the `summary` command are not given here, but the explanatory variable `Wetness` is highly significant ($t = 15.64$, $df = 532$, $p < 0.001$). Based on residual graphs (not shown here), homogeneity is a reasonable assumption. As a first step to verify independence, we plot the residuals versus their spatial coordinates. The package `gstat` (Pebesma, 2004) has a nice tool for this called a bubble plot, see Fig. 7.1. This package is not part of the base installation and you will need to install it from the R website. The size of the dots is proportional to the value of the residuals. This graph should not show any spatial pattern (e.g. groups of negative or positive residuals close to each other). If it does, then their may be a missing covariate or spatial correlation. In this case, there seems to be some spatial pattern as most of the positive residuals as well as the negative residuals are showing some clustering. The following R code was used to create the graph:

```
> E <- rstandard(B.lm)
> library(gstat)
> mydata <- data.frame(E, Boreality$x, Boreality$y)
> coordinates(mydata) <- c("Boreality.x", "Boreality.y")
> bubble(mydata, "E", col = c("black", "grey"),
+         main = "Residuals", xlab = "X-coordinates",
+         ylab = "Y-coordinates")
```

The first part of the R code extracts the standardised residuals, loads the `gstat` package, and creates a data frame containing the residuals and the coordinates.

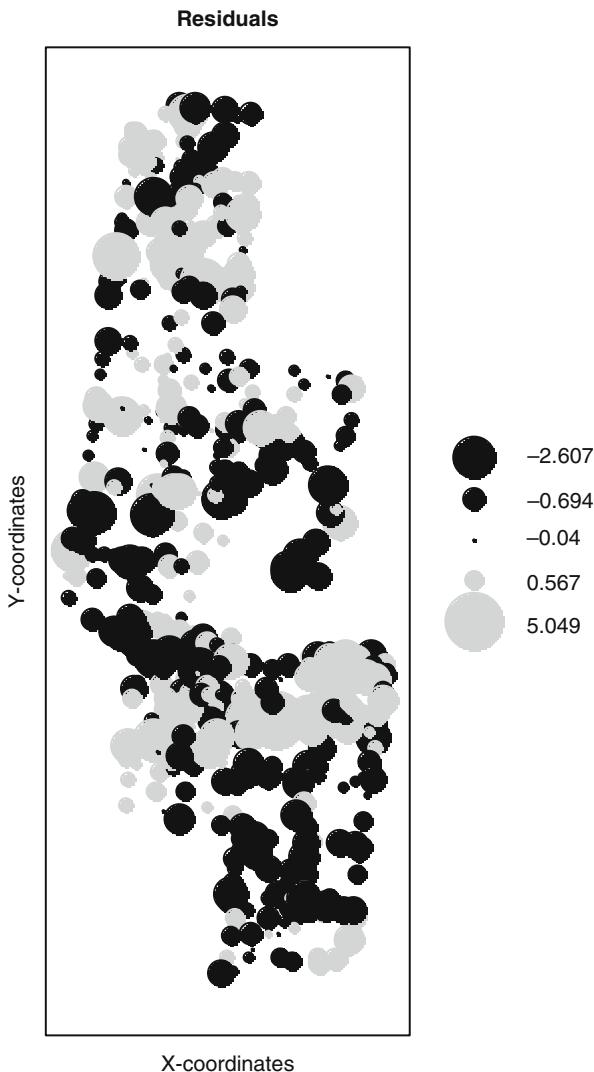


Fig. 7.1 Standardised residuals obtained by the linear regression model plotted versus their spatial coordinates. Black dots are negative residuals, and grey dots are positive residuals

The `coordinates` command is from the `gstat` package and ensures that the `bubble` functions know that `x` and `y` are spatial coordinates. The names of the `x` and `y` columns in the `coordinates` command must match the ones from the data frame, hence the ‘Boreality’ in ‘Boreality.x’.

As an alternative to the informal approach of making a bubble plot of residuals and judging whether there is spatial dependence, you can make a variogram of the

residuals. In Chapter 6, we used the ACF to judge whether there was dependence over time. For this, we assumed stationarity of the residuals and calculated the correlation between ε_s and ε_{s+k} , where k is the time lag. So residuals that are separated by k time units are aggregated to calculate the ACF.

In the forest data example, we do not have residuals at time s and t , but we have residuals at sites i and j and instead of using the ACF, we use the variogram. It is defined by:

$$\gamma(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2} E[(Z(\mathbf{x}_1) - Z(\mathbf{x}_2))^2]$$

This is a function that measures the spatial dependence between two sites with coordinates \mathbf{x}_1 and \mathbf{x}_2 . If these two sites are located close to each other, then you would expect the values of the variables of interest (residuals in this case) are similar. A low value of $\gamma(\mathbf{x}_1, \mathbf{x}_2)$ indicates that this is indeed the case (dependence), whereas a large value indicates independence. Spatial statistics tends to be rather complicated and intimidating. Zuur et al. (2007) discussed various aspects like ergodicity, stationarity, and weak stationarity. Without going into detail here, weak stationarity leads to the following variogram.

$$\gamma(\mathbf{h}) = \frac{1}{2} \text{Var}[(Z(\mathbf{x} + \mathbf{h}) - Z(\mathbf{x}))].$$

In the same way as the ACF measures the temporal dependence by comparing the value of Z at times t and $t + k$, so does the variogram in space. Instead of comparing all time points that are separated by k units, it takes all points that are separated by a vector \mathbf{h} , and it uses these to calculate the sample (or experimental) variogram:

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2 N(\mathbf{h})} \sum_{i=1}^{N(\mathbf{h})} [z(\mathbf{x}_i + \mathbf{h}) - z(\mathbf{x}_i)]^2$$

The hat notation $\hat{\gamma}$ is used because it is an estimator based on sample data. If there is spatial dependence, points close to each other tend to have similar values and the experimental variogram will be small. Large values for the experimental variogram indicate spatial independence. There are all kinds of ‘little’ details that play a role here, for example, the number of points that are exactly separated by a distance \mathbf{h} . This is the $N(\mathbf{h})$. In reality only a few points, if any, are separated exactly by a distance \mathbf{h} . The software code used to estimate the variogram puts a small range around \mathbf{h} so that enough points are available for analysis. Another important issue is that we assume isotropy. This means that the spatial dependence of the residuals is the same in any direction. If this is not the case, we cannot calculate the variogram using sites that are separated by a distance \mathbf{h} in any direction. If you do not have isotropy in the residuals, you may try to add more covariates and model the anisotropy.

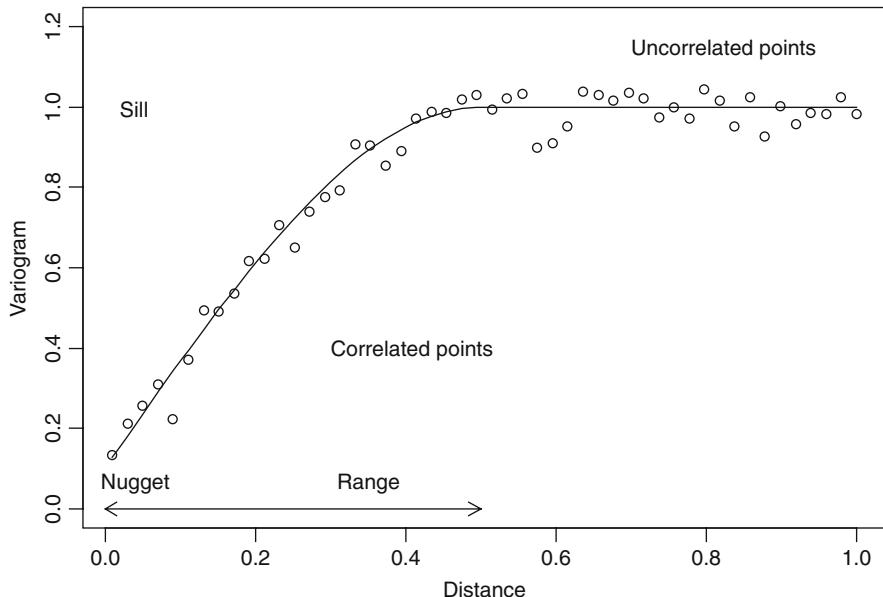


Fig. 7.2 Variogram with fitted line. The sill is the asymptotic value and the range is the distance where this value occurs. Pairs of points that have a distance larger than the range are uncorrelated. The nugget effect occurs if $\hat{\gamma}(\mathbf{h})$ is far from 0 for small \mathbf{h}

Figure 7.2 shows a theoretical variogram (line) plus some simulated data (dots). Along the x -axis, the distances between the sites are plotted, and along the y -axis, the estimated values of the variogram are plotted. Spatial dependence shows itself as an increasing band of points, which then levels off at a certain distance. The point along the x -axis at which this pattern levels off is called the range, and the y -value at the range is the sill. The nugget is the y -value when the distance is 0. It represents the discontinuity of the variable caused by spatial structures at distances less than the minimum distance between points.

Figure 7.3A shows the experimental variogram for the residuals of the linear regression model applied on the forest boreality data. Note that there is a clear spatial correlation up to about 1000 m. There is also a nugget effect of approximately 0.75. The following R code was used to create the experimental variogram.

```
> Vario1 <- variogram(E ~ 1, mydata)
> plot(Vario1)
```

Note that this variogram assumes isotropy; the strength of the spatial correlation is the same in each direction. We can verify this by making experimental variograms in multiple directions; see Fig. 7.3B. It seems that isotropy is a reasonable assumption as the strength, and pattern, of the spatial correlation seems to be broadly the

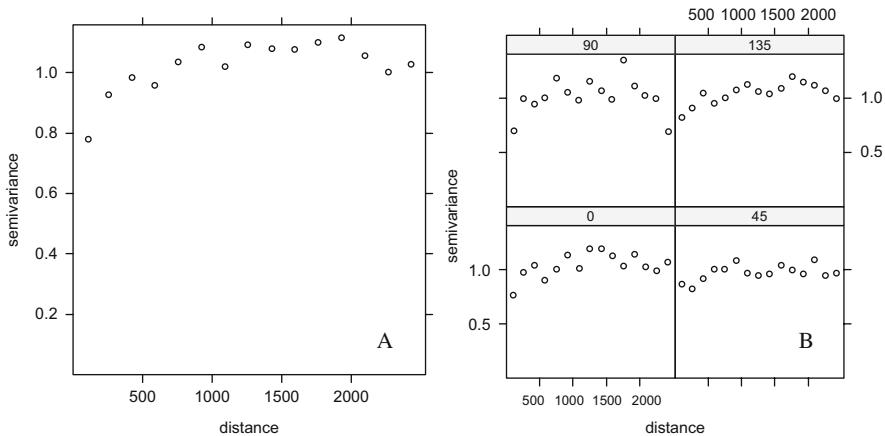


Fig. 7.3 **A:** Semi-variogram of the standardised residuals obtained by the linear regression model. The semi-variogram assumes isotropy. Note that there is spatial correlation up to 1000 m. **B:** Experimental variograms for four different directions

same in all four directions. The code to produce this graph is similar as above, except that the argument `alpha = c(0, 45, 90, 135)` is added to the `variogram` function.

```
> Vario2 <- variogram(E ~ 1, mydata,
                        alpha = c(0, 45, 90, 135))
> plot(Vario2)
```

7.2 Adding Spatial Correlation Structures to the Model

Both the bubble plot and the experimental variogram indicate that there is spatial correlation in the residuals, and Fig. 7.3 seems to suggest that isotropy is a reasonable assumption. We are now going from an informal assessment (looking at the bubble plot or experimental variogram) to a more formal approach. Now we include the correlation structure and use the AIC, BIC, or likelihood ratio test to judge the best model, the one with or without spatial correlation. This process works in a similar way as in the previous chapter. The only conceptual difference is that time goes in only one direction and space goes in multiple directions.

The question now is how do we include a spatial residual correlation structure in a linear regression, additive model, or (additive) mixed model? In Chapter 6, temporal dependences were included using the AR-1 or ARMA structures. Recall that these were used to parameterise the correlation function $h()$ in

$$\text{cor}(\varepsilon_s, \varepsilon_t) = \begin{cases} 1 & \text{if } s = t \\ h(\varepsilon_s, \varepsilon_t, \rho) & \text{else} \end{cases} \quad (7.1)$$

We now need to do the same trick we used with the time series, but this time, based on the shape of the variogram we need to choose a parameterisation for the correlation function $h()$. Options available in the R package `nlme` are as follows:

- Exponential correlation using the function `corExp`.
- Gaussian correlation using the function `corGaus`.
- Linear correlation using the function `corLin`.
- Rational quadratic correlation using the function `corRatio`.
- Spherical correlation using the function `corSpher`.

Each of these options implies a specific mathematical structure for the function $h()$, and a good overview is given in Schabenberger and Pierce (2002). Instead of diving straight into these formulae, it is perhaps more useful to first look at a couple of typical shapes implied by these spatial correlation structures. In Fig. 7.4, we show several theoretical variograms; the Gaussian, linear, rational quadratic, exponential, and the spherical correlation. Lines in the same panel were obtained by using a different range and sill. Some of these curves look similar and selecting the right one is a matter of expertise and pre-knowledge.

The R code for Fig. 7.4 is not given in full here. Instead, we show how to make one particular variogram, which should provide the background required to build the others.

```
> library(nlme)
> D <- seq(from = 0, to = 1, by = 0.1)
> Mydata2 <- data.frame(D = D)
> cs1C <- corSpher(c(0.8, 0.1), form = ~ D, nugget = TRUE)
> cs1C <- Initialize(cs1C, data = Mydata2)
> v1C <- Variogram(cs1C)
> plot(v1C, smooth = FALSE, type = "l", col = 1)
```

The first line creates an artificial distance gradient from 0 to 1, which we store in the data frame `mydata`. It is used in the function `corSpher`, which takes as arguments the range and the sill (optional) and the `form` option that specifies the gradient. Note that the sill is scaled to 1 by this particular `Variogram` function from the `nlme` package. The function uses the specified range and sill, substitutes these in the formulae for the spherical correlation (Schabenberger and Pierce, 2002), and calculates the corresponding variogram values. The multipanel plot in Fig. 7.4 is then a matter of repeating this with different ranges and sills and correlation functions, and then, with a bit of R magic, using the `rep` function and `xyplot`.

Some of the underlying formulae for the variogram are intimidating and some are surprisingly simple. For example, the exponential correlation structure is given by

$$\gamma(s, \rho) = 1 - e^{\frac{s}{\rho}}$$

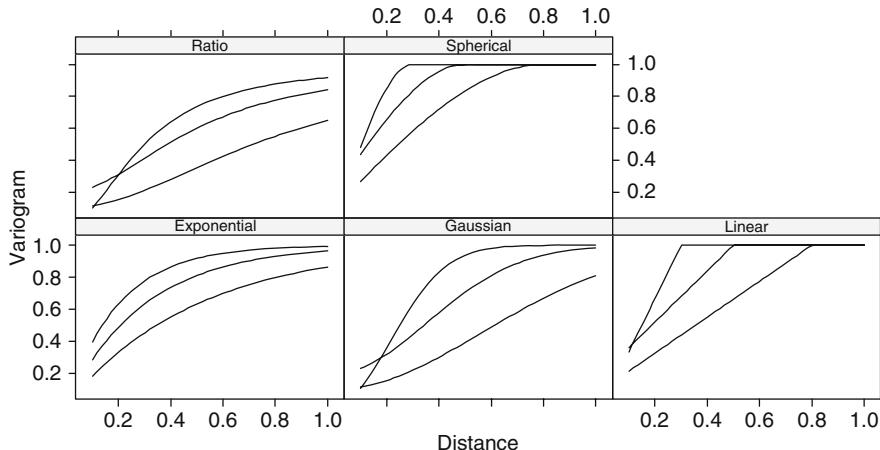


Fig. 7.4 Different variogram patterns. The three lines in the same panel were obtained using different values for the range and nugget

where ρ is the range and s the distance. If you decide to add a nugget effect c_0 , the formulation changes to

$$\gamma(s, \rho) = \begin{cases} c_0 + (1 - c_0)(1 - e^{\frac{s}{\rho}}) & \text{if } s > 0 \\ 0 & \text{if } s = 0 \end{cases}$$

All this does is specify that the variogram is 0 for $s = 0$, shifts up the curve with c_0 , and ensures it is not larger than 1. The Gaussian model is similar, but it squares the s/ρ term. The linear, rational quadratic, and spherical correlations are slightly more complicated and are not given here, but the principle is the same. The function $\gamma(s, \rho)$ is actually called the correlogram; you need to multiply it with the variance to get the variogram.

So, our task is to extract the (standardised; observed, minus fitted, and potentially corrected for heterogeneity) residuals from the linear regression or GLS model, make an experimental variogram of the residuals, and judge which correlation structure is the most appropriate. We look at this process using the boreality forest data. Instead of the function `variogram` from the `gstat` package, we use the `Variogram` function from the `nlme` package as it takes as input the object from a `gls`, `lme`, or `gamm` command. The following R code produces a similar experimental variogram for the residuals of the linear regression model as in Fig. 7.3A.

```
> library(nlme)
> f1 <- formula(Bor ~ Wet)
> B1.gls <- gls(f1, data = Boreality)
```

```
> Vario.gls <- Variogram(B1.gls, form =~ x + y,
                           robust = TRUE, maxDist = 2000,
                           resType = "pearson")
> plot(Vario.gls, smooth = TRUE)
```

The first three lines apply the same linear regression model as above (transformed boreality as a function of wetness), but now with the `gls` command. The function `Variogram` takes the object from the `gls` function and extracts the standardised residuals (because we specified this type of residuals with the `resType` option). It then calculates the experimental variogram. The *x* and *y*-coordinates are used to calculate distances (using Pythagoras theorem) between points. To aid visual interpretation, a LOESS smoother was added, but can be suppressed using `smooth = FALSE`. Sometimes it is handy to add it, and sometimes it is not. The `robust` and `maxDist` are further parameters for calculating the experimental variogram (Cressie, 1993). Spatial independence is a likely assumption if the experimental variogram shows a band of horizontal points, but this is not the case here.

Instead of judging from the experimental variogram whether residual independence can be assumed, we can add a spatial correlation structure to the GLS model and compare it with the model without the spatial correlation. The following R code adds the various correlation structures to the GLS model.

```
> B1A <- gls(f1, correlation = corSpher(form =~ x + y,
                                         nugget = TRUE), data = Boreality)
> B1B <- gls(f1, correlation = corLin(form =~ x + y,
                                         nugget = TRUE), data = Boreality)
> B1C <- gls(f1, correlation = corRatio(form =~ x + y,
                                         nugget = TRUE), data = Boreality)
> B1D <- gls(f1, correlation = corGaus(form =~ x + y,
                                         nugget = TRUE), data = Boreality)
> B1E <- gls(f1, correlation = corExp(form =~ x + y,
                                         nugget = TRUE), data = Boreality)
> AIC(B1, B1A, B1B, B1C, B1D, B1E)
```

We could have used the `update` command, but in this case, it does not shorten the code. If there are convergence problems (and this can happen quite often), then it may help to modify the `lmeControl` settings (see its help file). The `anova` or `AIC` command can be used to obtain the AIC values, and these are given in Table 7.1. The AIC of the model with no correlation is 2844.54, but the models with the `corLin`, `corGaus`, and `corExp` correlation structures have considerable lower AIC values, making them all candidate models. So adding a spatial correlation structure improves the model, as judged by the AIC.

We can also apply a hypothesis test with the `anova(B1, B1E)` command (we could have used any of the other candidate models). It gives $L = 116.31$, ($df = 2$, $p < 0.001$), indicating that adding a spatial correlation structure gives a significantly better model.

Table 7.1 AIC values obtained by adding various correlation structures to the linear regression model. The first column shows which correlation structure is added, the second column the object name, all models with spatial correlation use two extra parameters, and the last column gives the AIC

Model	Object	df	AIC
No correlation	B1	3	2844.54
corSpher	B1A	5	2737.01
corLin	B1B	5	2848.51
corRatio	B1C	5	2732.93
corGaus	B1D	5	2736.29
corExp	B1E	5	2732.22

From the AICs and likelihood ratio test, we can conclude that we are violating the independence assumption in the linear regression model. So the remaining question is now whether adding any of these spatial correlation structures can solve the independence problem. The commands

```
> Vario1E <- Variogram(B1E, form =~ x + y,
                         robust = TRUE, maxDist = 2000,
                         resType = "pearson")
> plot(Vario1E, smooth = FALSE)
```

will show the experimental variogram with the fitted spatial correlation (results are not shown here), and the following code

```
> Vario2E <- Variogram(B1E, form =~ x + y,
                         robust = TRUE, maxDist = 2000,
                         resType = "normalized")
> plot(Vario2E, smooth = FALSE)
```

does the same for the normalised residuals. The later ones should no longer show a spatial correlation (you should see a horizontal band of points). Results are not presented here, but the experimental variogram of the normalised residuals indeed form a horizontal band of points, indicating spatial independence.

Note that we should apply the same 10-step protocol we used in Chapters 4 and 5. First determine the optimal random structure using REML estimation, using as many fixed covariates as possible. (However, here all covariates are highly collinear; so there is effectively only one variable.) Once the optimal random structure has been found, the optimal fixed structure can be found using the tools described in Chapters 4 and 5. So, the whole REML and ML process used earlier also applies here.

For this chapter, we used the GLS model. If a random effects model is used, the spatial correlation structure is applied within the deepest level of the data. See also Chapters 16 and 17 where we impose a correlation structure on nested data.

7.3 Revisiting the Hawaiian Birds

Now we return to the Hawaiian bird data set, which we left with an AR1 auto-correlation structure. In the previous section, we used the `form =~ x + y` argument in the correlation option. If included in the `gls`, `lme`, or `gamm` function, it ensures that R calculates distances between the sampling points with coordinates given by `x` and `y`. The default option to calculate distances is Euclidean distances (using Pythagoras) and alternatives are Manhattan and maximum distances (Pinheiro and Bates, 2000). In the Hawaiian data, we used `form =~ Time | ID` in the `corAR1` function. Nothing stops us using for example a spatial correlation function like `corSpher` for time series. It can cope better with missing values and irregularly spaced data. In fact, the `corExp` structure is closely related to the `corAR1` (Diggle et al., 2002). The following code applies the model with the `corAR1` structure and all four spatial correlation functions. We copied and pasted the code from Chapter 6 to access the data.

```
> library(AED); data(Hawaii)
> Birds <- c(Hawaii$Stilt.Oahu, Hawaii$Stilt.Maui,
+              Hawaii$Coot.Oahu, Hawaii$Coot.Maui)
> Time <- rep(Hawaii$Year, 4)
> Rain <- rep(Hawaii$Rainfall, 4)
> ID <- factor(rep(c("Stilt.Oahu", "Stilt.Maui",
+                     "Coot.Oahu", "Coot.Maui"),
+                     each = length(Hawaii$Year)))
> library(mgcv); library(nlme)
> #Define the fixed part of the model
> f1 <- formula(Birds ~ Rain + ID +
+                 s(Time, by = as.numeric(ID == "Stilt.Oahu"))+
+                 s(Time, by = as.numeric(ID == "Stilt.Maui"))+
+                 s(Time, by = as.numeric(ID == "Coot.Oahu"))+
+                 s(Time, by = as.numeric(ID == "Coot.Maui")))
> #Fit the gamms
> HawA <- gamm(f1, method = "REML", correlation =
+               corAR1(form =~ Time | ID),
+               weights = varIdent(form =~ 1 | ID))
> HawB <- gamm(f1, method = "REML", correlation =
+               corLin(form =~ Time | ID, nugget = TRUE),
+               weights = varIdent(form =~ 1 | ID))
> HawC <- gamm(f1, method = "REML", correlation =
+               corGaus(form =~ Time | ID, nugget = TRUE),
+               weights = varIdent(form =~ 1 | ID))
> HawD <- gamm(f1, method = "REML", correlation =
+               corExp(form =~ Time | ID, nugget = TRUE),
+               weights = varIdent(form =~ 1 | ID))
```

```
> HawE <- gamm(f1, method = "REML", correlation =
+   corSpher(form =~ Time | ID, nugget = TRUE),
+   weights = varIdent(form =~ 1| ID))
> #Compare the models
> AIC(HawA$lme, HawB$lme, HawC$lme, HawD$lme, HawE$lme)

      df      AIC
HawA$lme 18 2277.677
HawB$lme 19 2281.336
HawC$lme 19 2279.182
HawD$lme 19 2279.677
HawE$lme 19 2278.898
```

The results of the `AIC` command indicate that the model with the `corAR1` structure should be chosen.

7.4 Nitrogen Isotope Ratios in Whales

In this section, we analyse the nitrogen isotopic data of teeth growth layers of 11 whales. We start with one whale and then analyse the data from all whales.

7.4.1 Moby

In Chapter 2, we applied linear regression on the nitrogen isotope values of a whale nicknamed Moby, and we discussed two potential sources of violating the independence assumption. The first was a potential improper model specification (a linear relationship when the real relationship may be non-linear). The second one was due to the nature of the data; nitrogen concentrations at a certain age s may depend on the concentrations at age $s - 1$, $s - 2$, $s - 3$, etc. To deal with the first problem, we applied a Gaussian additive model on the data for Moby:

$$y_s = \alpha + f(age_s) + \varepsilon_s \quad \boldsymbol{\varepsilon} \sim N(0, \sigma^2 \times \mathbf{V}), \text{ where } \boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2 \dots, \varepsilon_T)'$$

The index s represents year and runs from 3 to 44 for Moby. The variable y_s contains the isotopic value in year s , α is the intercept, age_s is the age in year s , $f(age_s)$ is the smoothing function of age, and ε_s are the residuals. In an ordinary Gaussian additive model (or linear regression model), we assume that the residuals are independent and normally distributed with mean 0 and variance σ^2 . This means that \mathbf{V} is a 42-by-42 identity matrix. (This is matrix full of zeros, except for the diagonal; these are all equal to 1.)

To allow for a dependence structure between the observations, we can use any of the correlation structures discussed earlier in Chapter 6 or in this chapter. Instead of

temporal or geographical coordinates, age is now the variable that we use to set up the variogram. As a consequence, \mathbf{V} is no longer a diagonal matrix. Its off-diagonal elements give the residual covariance at different ages. The key question is now, how we should parameterise this matrix. Clearly, using a completely unspecified matrix results in too many unknown parameters. We can use the variogram or the AR1 residual correlation structures. These will specify that observations that are separated by an age of k years have a correlation as specified by, for example, the linear, spherical, exponential, or Gaussian variogram structure. All we have to do is to apply models with different covariance structures and assess which one is the most appropriate using, for example, the AIC.

The model selection process is identical to mixed modelling; (i) start with a model that contains as many explanatory variables as possible, (ii) find the optimal random structure, and (iii) find the optimal fixed structure. If we have data on only one whale, the first step is rather simple: use age. The following code imports the data, extracts the data from Moby, and applies the models.

```
> library(AED); data(TeethNitrogen)
> TN <- TeethNitrogen
> N.Moby <- TN$X15N[TN$Tooth == "Moby"]
> Age.Moby <- TN$Age[TN$Tooth == "Moby"]
> library(mgcv); library(nlme)
> f <- formula(N.Moby ~ s(Age.Moby))
> #Apply gamm models
> Mob0 <- gamm(f, method = "REML")
> Mob1 <- gamm(f, method = "REML", cor =
+     corSpher(form =~ Age.Moby, nugget = TRUE))
> Mob2 <- gamm(f, method = "REML", cor =
+     corLin(form =~ Age.Moby, nugget = TRUE))
> Mob3 <- gamm(f, method = "REML", cor =
+     corGaus(form =~ Age.Moby, nugget = TRUE))
> Mob4 <- gamm(f, method = "REML", cor =
+     corExp(form =~ Age.Moby, nugget = TRUE))
> Mob5 <- gamm(f, method = "REML", cor =
+     corRatio(form =~ Age.Moby, nugget = TRUE))
> Mob6 <- gamm(f, method = "REML", cor =
+     corAR1(form =~ Age.Moby))
> AIC(Mob0$lme, Mob1$lme, Mob2$lme, Mob4$lme, Mob5$lme,
+     Mob6$lme)

Mob0$lme 4 64.52995
Mob1$lme 6 67.02795
Mob2$lme 6 67.02795
Mob4$lme 6 63.38405
Mob5$lme 6 63.09320
Mob6$lme 5 63.60480
```

The model with the `corGaus` correlation structure did not converge and is therefore not used in the AIC command. Except for the `corSpher` correlation structure, all AICs are similar; hence, we might as well choose the simplest model, which is the one without a correlation structure (the linear regression model, `Mob0`). Comparing model `Mob0` with `Mob5` (`Mob0` is the model without a correlation structure and `Mob5` is the best *potential* model with respect to spatial correlation) using a likelihood ratio test gave a *p*-value of 0.06 (just type: `anova(Mob0$1me, Mob5$1me)`). Hence, there is no convincing evidence to use a correlation structure for the data of this whale. Furthermore, the estimated smoother in model `Mob5` is a straight line. This indicates that for the Moby data, the linear regression model that was presented in Chapter 2 suffices. This is rather confusing as the model did have residual patterns!

7.4.2 All Whales

What about the other whales? Instead of applying the above method on each individual whale data, we apply one additive model on the data of all animals and estimate one underlying ‘spatial’ correlation structure. This is the same approach we applied on the Hawaiian time series in Chapter 6. The following model was applied:

$$N_{is} = \alpha_i + f_i(\text{Age}_{is}) + \varepsilon_{is}$$

The subscript i refers to whale ($i = 1, \dots, 11$) and s to year. Here, we assume that each whale i has a potentially different age-effect on isotopic nitrogen values, hence the subscript i for the smoothing function f . Later, in the case studies, we show how we can test whether multiple smoothers can be replaced by one or a few smoothers using a deep sea research data set.

In a standard application of this model, the residuals ε_{is} are assumed to be independent and normally distributed with mean 0 and covariance matrix $\sigma^2 \mathbf{V}_i$, where \mathbf{V}_i is an identity matrix. The size of this matrix depends on the number of observations for whale i . This is perhaps clearer if we switch to a vector notation.

$$\mathbf{N}_i = \boldsymbol{\alpha} + \mathbf{f}(\mathbf{Age}_i) + \boldsymbol{\varepsilon}_i$$

Each vector contains all the age data for one whale. A dependence structure between residuals of different ages can be introduced by using a non-diagonal matrix \mathbf{V}_i , just as we did for the Moby data earlier in this section. We use the data from all the 11 whales and apply the correlation structure at the deepest level within a time series for each whale. All whales are assumed to have the same spatial correlation structure.

A model that contains as many explanatory variables as possible is a model that has one smoother per whale. This means that we have to use 11 smoothers, and this could potentially take considerable computing power (even with modern

computers). We therefore use cubic regression splines as these are less time consuming to calculate than the default thin plate regression spline smoother (Wood, 2006).

The following R code applies the model in R. Note the use of the `by` command in the smoother; it ensures we have one smoother per whale.

```
> lmc <- lmeControl(niterEM = 5200, msMaxIter = 5200)
> AllWhales.corGaus <- gamm( X15N ~
  s(Age, by=as.numeric(Tooth=="M2679/93"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="M2683/93"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="M2583/94(1)"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="M2583/94(7)"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="M2583/94(10)"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="M546/95"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="M143/96E"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="Moby"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="M447/98"), bs="cr") +
  s(Age, by=as.numeric(Tooth=="I1/98"), bs="cr") +
  factor(Tooth), control = lmc, method = "REML",
  correlation = corGaus(form =~ Age|Tooth, nugget=T),
  data = TN)
```

Besides the `corGaus` correlation structure, we also applied all the other correlation structures we discussed earlier in this chapter. Using no correlation structure gave AIC = 529.16. The lowest AIC value was obtained by the `corGaus` structure with a value of 478.82, closely followed by the `corRatio`. Other correlation structures were all slightly higher (around 485). This shows that a correlation structure improves the model considerably! The estimated range by the `corGaus` structure was 2.9 years. This means that after removing the age effect, the nitrogen isotopic values are correlated up to 2.9 years.

An interesting question is then what are the differences between the models with and without the `corGaus` correlation structure? The results of the model without the correlation structure are presented below. The object `AllWhales.0$gam` was fitted with the code below, except that the correlation option was removed.

```
> anova(AllWhales.0$gam)
```

Approximate significance of smooth terms:

	edf	F	p-value
s(Age):as.numeric(Tooth=="M2679/93")	6.055	58.440	< 2e-16
s(Age):as.numeric(Tooth=="M2683/93")	1.000	39.421	1.42e-09
s(Age):as.numeric(Tooth=="M2583/94(1)")	1.000	175.088	< 2e-16
s(Age):as.numeric(Tooth=="M2583/94(7)")	4.215	12.742	1.30e-12
s(Age):as.numeric(Tooth=="M2583/94(10)")	3.839	6.103	5.32e-06
s(Age):as.numeric(Tooth=="M546/95")	4.039	18.847	< 2e-16
s(Age):as.numeric(Tooth=="M143/96E")	1.000	32.316	3.50e-08
s(Age):as.numeric(Tooth=="Moby")	4.272	44.760	< 2e-16

```
s(Age):as.numeric(Tooth=="M447/98")      4.408      21.910    < 2e-16
s(Age):as.numeric(Tooth=="I1/98")        5.244      14.361    < 2e-16
```

All the smoothers are highly significant at the 5% level. However, this model ignores the potential dependence. The following results were obtained by the model with the corGaus correlation structure.

```
> anova(AllWhales.corGaus$gam)

Approximate significance of smooth terms:
                                         edf      F   p-value
s(Age):as.numeric(Tooth=="M2679/93")     1.000 86.928    < 2e-16
s(Age):as.numeric(Tooth=="M2683/93")     1.000 10.746  0.001178
s(Age):as.numeric(Tooth=="M2583/94(1)")  1.000 48.341  2.56e-11
s(Age):as.numeric(Tooth=="M2583/94(7)")  2.414  4.983  0.000218
s(Age):as.numeric(Tooth=="M2583/94(10)") 3.290  4.715  0.000137
s(Age):as.numeric(Tooth=="M546/95")      3.371  5.071  5.89e-05
s(Age):as.numeric(Tooth=="M143/96E")     1.000  7.896  0.005307
s(Age):as.numeric(Tooth=="Moby")          1.000 73.198  8.08e-16
s(Age):as.numeric(Tooth=="M447/98")      1.000 32.954  2.47e-08
s(Age):as.numeric(Tooth=="I1/98")         3.336  3.035  0.004317
```

Note that there are considerable differences in the *p*-values, and the model without the correlation structure giving a rosier but misleading picture in terms of significance levels!

These models assume the same residual spread per whale and over time. A model validation did not reveal any immediate problems with homogeneity, but the analysis may be extended by allowing for different spread per whale, which means the use of the weights and varIdent functions. The reason that we mention this is that most examples used in this book contain some form of heterogeneity. It would be a small miracle if this is not also the case here.

To save some parameters, it is also possible to use Tooth as random effect instead of a fixed nominal variable with 11 levels. It is also interesting to compare the compound symmetric correlation structure (by using a random intercept) versus the spatial correlation model. Or perhaps, use both correlation structures: a spatial correlation within the random effect tooth. We leave this an exercise for the reader.

Mendes et al. (2007) analysed the same data and looked at sudden changes in nitrogen isotopic values. Multivariate time series techniques like chronological clustering were used (Legendre and Legendre, 1998). Such an analysis can also be carried out within the additive mixed modelling framework. A dummy variable (also called intervention variable in this context) is an explanatory variable of the form 0 0 0 0 0 ... 1 1 1 1 1 (Harvey, 1989). These can be used to test for sudden changes using a model of the form

$$N_{is} = \alpha_i + f_i(\text{Age}_{is}) + \beta_i \times D_{is} + \varepsilon_{is}$$

where D_{is} is a vector of zeros and ones. A sudden increase in nitrogen isotope ratios for a particular whale can be tested by looking at the significance of the regression parameter β_i . The only problem is at which age the dummy variable D_{is} should switch from a zero to a one. Adding an optimisation routine that tries different switching points per whale and compares them using the AIC may be an option. This is also called intervention analysis (Harvey, 1989; Zuur et al., 2007).

Something we have ignored so far is the assumption of a fixed X. Recall that this means that before sampling, we know the value of the explanatory variables. For the whale data, this assumption is clearly violated as there may be an error of 1–2 years on an age reading. Bootstrapping (Efron and Tibshirani, 1993) may be a tool to deal with this. There are many ways to carry out a bootstrap, and one of these, to apply an ordinary bootstrap, is as follows.

1. Apply the smoothing model for the given data, and estimate the smoothers, etc. Obtain the fitted values and the residuals for the original data.
2. Permute the residuals from step 1, or apply a parametric bootstrap on the residuals. Add the permuted residuals to the fitted values from step 1. This gives bootstrapped data (response variable).
3. Apply the smoothing model on the new data.
4. Repeat steps 2 and 3 1000 times.
5. Use the 1000 estimated smoothers to create confidence bands.

More details can be found in Davison and Hinkley (1997). To take account of the age of 20 years being anything between 18 and 22, we can add an extra permutation step to the algorithm described above that will slightly modify the age in each bootstrap iteration. Note that this 5-step scheme is not a full recipe. Details on bootstrapping GAMs can be found in Davison and Hinkley (1997) and Keele (2008).

7.5 Spatial Correlation due to a Missing Covariate

In this section, we show how a missing covariate may cause spatial correlation. The data used are a subset of the data analysed in Cruikshanks et al. (2006), a technical report by the Environmental Protection Agency, Wexford, Ireland). We only use the 2003 data, and several recordings were dropped. So, our results may be different from those presented in the original report.

The original research sampled 257 rivers in Ireland during 2002 and 2003. One of the aims was to find a different tool for identifying acid-sensitive waters, which currently uses measures of pH. The problem with pH is that it is extremely variable within a catchment and depends on both flow conditions and underlying geology. As an alternative measure, the Sodium Dominance Index (SDI) is proposed as an indicator of the acid sensitivity of rivers. SDI is defined as the contribution of sodium (Na^+) to the sum of the major cations. The motivation for this research is

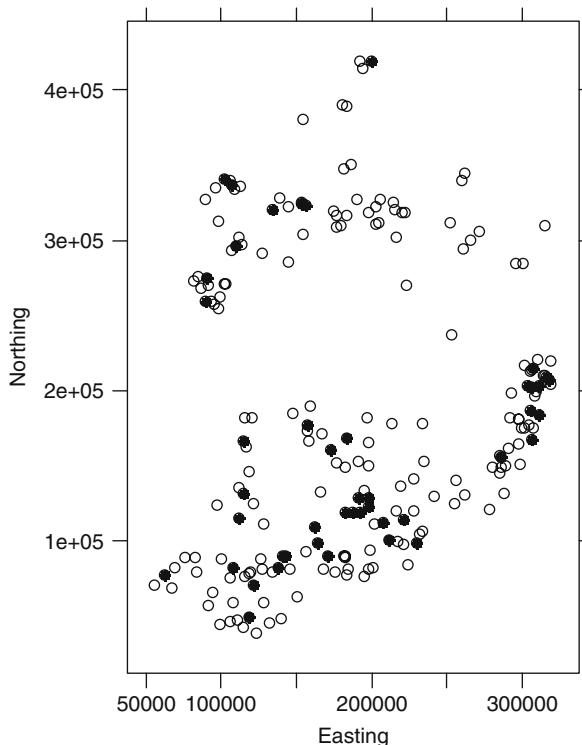


Fig. 7.5 Positions of the sites in Ireland that were sampled in 2003. Filled circles are forested sites and open circles the non-forested sites

the increase in plantation forestry cover in Irish landscapes and its potential impacts on aquatic resources. Of the 257 sites, 192 were non-forested and 65 were forested.

In this section, we model pH as a function of SDI, whether a site is forested or not, and altitude. Figure 7.5 shows the geographical position of the sites in Ireland that were sampled in 2003. The following code accesses the data and makes the graph.

```
> library(AED); data(SDI2003);
> library(lattice)
> MyPch <- vector(length = dim(SDI2003)[1])
> MyPch[SDI2003$Forested == 1] <- 16
> MyPch[SDI2003$Forested == 2] <- 1
> xyplot(Northing ~ Easting, aspect = "iso", col = 1,
  pch = MyPch, data = SDI2003)
```

The `xyplot` from the `lattice` package is used to ensure that the units along the vertical and horizontal axes are the same; see also Chapter 16 for other ways of doing this. The variable `MyPch` is used to plot different types of dots for forested and non-forested sites.

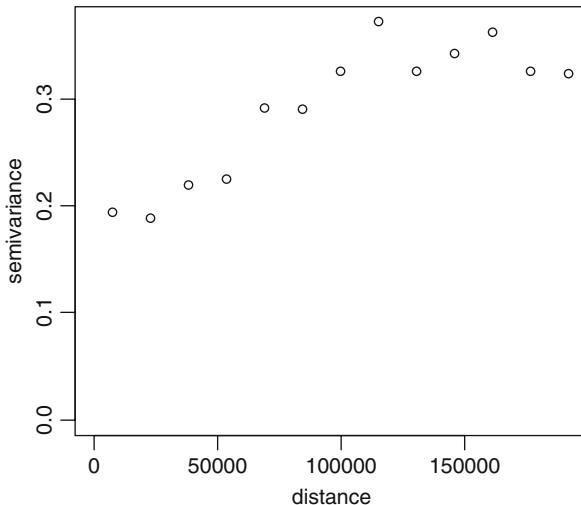


Fig. 7.6 Experimental variogram for the pH data sampled in Ireland in 2003. Note that there is spatial dependence because there is an increase in the experimental variogram

We first show that there is spatial correlation in the pH data with help of an experimental variogram (Fig. 7.6). Results clearly indicate that there is spatial dependence as the pattern slowly increases and then levels off. Earlier in this chapter, we used the variogram function from the `gstat` package. Here, we use yet another package to make variograms, namely `geoR`. In practise, these packages tend to give similar results, but it is useful to know (and be able to use) that there are multiple packages for the same thing.

The code to make Fig. 7.6 is given below.

```
> library(geoR)
> cords <- matrix(0, length(SDI2003$pH), 2)
> coords[, 1] <- SDI2003$Easting;
> coords[, 2] <- SDI2003$Northing
> gb <- list(data = SDI2003$pH, cords = coords)
> plot(variog(gb, max.dist = 200000))
```

Before adding spatial correlation structures, we should first apply a model without spatial correlation structures, extract its residuals, and see whether these residuals show spatial dependence. After all, we may be able to explain the spatial patterns in pH with SDI or altitude. The following linear regression model (in words) is applied.

$$\text{pH}_i = \alpha + \text{SDI}_i \times \text{Altitude}_i \times \text{factor(Forested}_i\text{)} + \varepsilon_i$$

Actually, we used the log-transformed altitude. The model contains 3 main terms, all 2-way interactions, and one 3-way interaction term, and the residuals

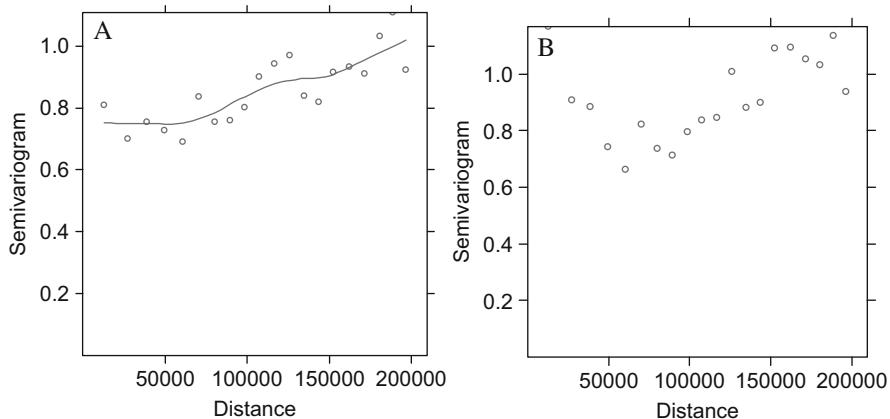


Fig. 7.7 **A:** Experimental variogram of the residuals obtained by applying a linear regression model on pH. Note that there is evidence that the independence assumption is violated. The line is a smoother and can be suppressed by adding `smooth = FALSE` to the `plot(Vario1)` command. **B:** Experimental variogram of the normalised residual using the `corRatio` structure. The `corExp` is equally bad

are assumed to be independent and normally distributed with mean 0 and variance σ^2 . Homogeneity and normality are valid assumptions, and the numerical output indicates that we may expect a significant SDI effect and a significant altitude \times Forested effect. The following R code applies the linear regression model and draws an experimental variogram of the residuals (Fig. 7.7A). A smoother was added to aid visual interpretation.

```
> library(nlme)
> SDI2003$fForested <- factor(SDI2003$Forested)
> SDI2003$LAltitude <- log(SDI2003$Altitude)
> M1 <- gls(pH ~ SDI * fForested * LAltitude,
  data = SDI2003)
> Vario1 <- Variogram(M1, form =~ Easting + Northing,
  data = SDI2003, nugget = TRUE, maxDist = 200000)
> plot(Vario1)
```

The AIC of the GLS model without auto-correlation is 248.34. Just as in previous sections, we can add any of the five available correlation structures to the GLS and the `corRatio` and `corExp` structures give considerable lower AICs: 205.95 and 208.57, respectively. These models are implemented with the following code:

```
> M1C <- gls(pH ~ SDI * fForested * LAltitude,
  correlation = corRatio(form =~ Easting +
  Northing, nugget = TRUE), data = SDI2003)
```

```
> M1E <- gls(pH ~ SDI * fForested * LAltitude,
  correlation = corExp(form =~ Easting +
  Northing, nugget = TRUE), data = SDI2003)
```

However, neither of these correlation structures produces a fitted line that matches the experimental variogram. Figure 7.7B shows the experimental variogram of the normalised residuals and shows a clear pattern. It was made with the following R code. If you remove the `resType` option, the plot function shows the fitted experimental variogram.

```
> Vario1C <- Variogram(M1C, form =~ Easting + Northing,
  data = SDI2003, nugget = TRUE, maxDist = 200000,
  resType = "normalized")
> plot(Vario1C, smooth = FALSE)
```

We could try and choose fixed values for the nugget and range, but the real problem is that we are missing a covariate. This can be seen from a bubble plot of the normalised residuals of the linear regression model (Fig. 7.8). The negative residuals are mainly clustered along the south and south-east coastline, and the western coastline mainly contains positive residuals. So there is a clear pattern in these residuals. To solve this problem, we need to think very carefully about which missing covariate could be causing this type of pattern and hope that it can (still) be quantified

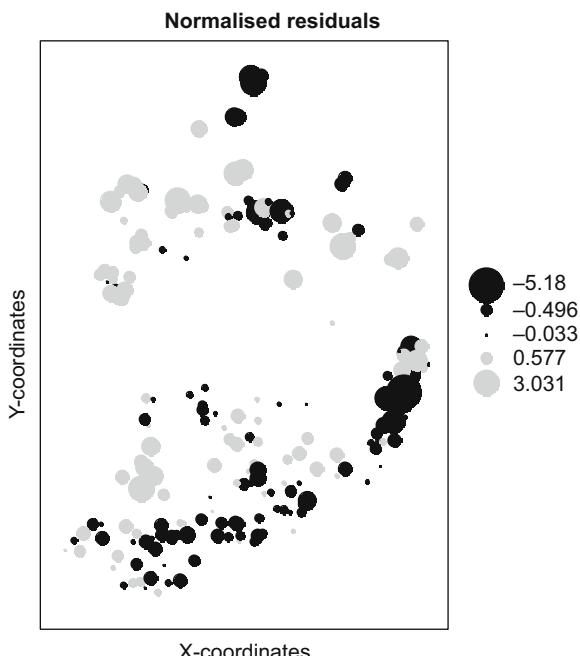


Fig. 7.8 Normalised residuals of the linear regression model plotted against spatial coordinates. The size of the dot is proportional to its value, and the colour refers to the sign. Note that most negative residuals are clustered along the south-east coast, and the west coast mainly contains positive residuals

and included in the model. In the meantime, we should refrain from making any inferential conclusions from these models, and we cannot say (yet) whether there is an altitude \times Forested interaction or an SDI effect on pH.

The following R code was used to create the bubble plot.

```
> library(gstat)
> E <- resid(M1, type = "normalized")
> mydata <- data.frame(E, SDI2003$Easting,
+ SDI2003$Northing)
> coordinates(mydata) <- c("SDI2003.Easting",
+ "SDI2003.Northing")
> bubble(mydata, "E", col = c("black", "grey"),
+ main = "Normalised residuals",
+ xlab = "X-coordinates", ylab = "Y-coordinates")
```

7.6 Short Godwits Time Series

In the previous chapter, we showed how to include a temporal correlation structure using relatively long and regularly spaced time series with the `corAR1` and `corARMA` functions. In earlier sections in this chapter, we had spatial data and data along an age gradient. In all cases, the length of the gradient was long. We now use an example that consists of rather short and irregularly spaced time series of feeding behaviour patterns in the godwits (*Limosa haemastica*) data (Ieno, unpublished data).

7.6.1 Description of the Data

Food intake rates of migrating godwits were observed at a tidal channel, on a section of a South Atlantic mudflat system in Argentina (Samborombón Bay). Sampling took place on 20 (non-sequential) days, divided over three consecutive periods. On the basis of plumage and time of the year, birds were classified as ‘birds preparing for migration’ (southern late summer/fall) and ‘birds not preparing for migration’. The second group can be further divided in southern spring/early summer, and southern winter. Measurements took place during the low water period on at least two days per month during 15 consecutive months.

On each sampling day, between 7 and 19 observations were taken, which gives us short longitudinal time series per day.

The observations consist of the food intake rates, which is the mg of Ash free dried prey (nereid worm) weight per second of feeding (mg AFDW/s). The time when the godwits took food was also recorded. Because time itself has no ecological meaning for the birds, it is expressed in hours before and after the low tide.

The underlying question is whether intake rate depends on period of migration, time with respect to low tide (does food consumption depend on the tide), and sex. What we have in mind is a model of the form:

$$\text{IntakeRate}_{ij} = \text{function}(\text{Time}_{ij}, \text{Sex}_{ij}, \text{Period}_{ij}) + \varepsilon_{ij}$$

IntakeRate_{ij} is the intake rate of observation j on day i . Time_{ij} is the corresponding time. It tells you how many minutes before or after low tide an observation was made. Sex has the values unknown, female or male. Period is a nominal variable with three levels; 0 if an observation was made in January, September–December; 1 if an observation was made during February, March, or April; and 3 for May–August. These three periods represent the migration ‘status’ of godwits as explained above.

The potential complicating factor is that the intake rate at a particular time on a particular day may depend on the intake rate at an earlier time on the same day. Your alarm bells for violation of independence should now make a lot of noise!

7.6.2 Data Exploration

As always, we started the statistical analysis with a detailed graphical data exploration. Results are not presented here, but none of the data exploration tools (boxplots, Cleveland dotplots, and pairplots) indicated any outliers. The coplot in Fig. 7.9

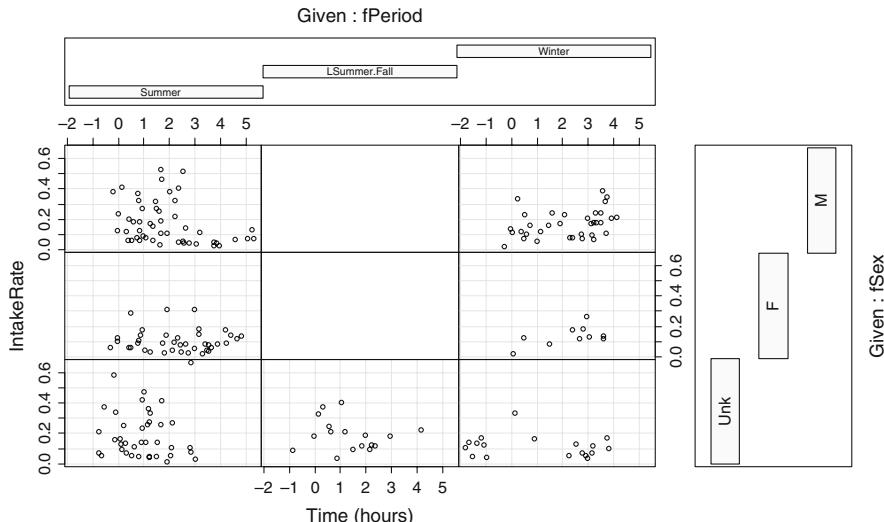


Fig. 7.9 Coplot of intake rate versus time (time since low tide in hours), conditional on sex and period. Note that in late summer and fall, not all sexes were measured

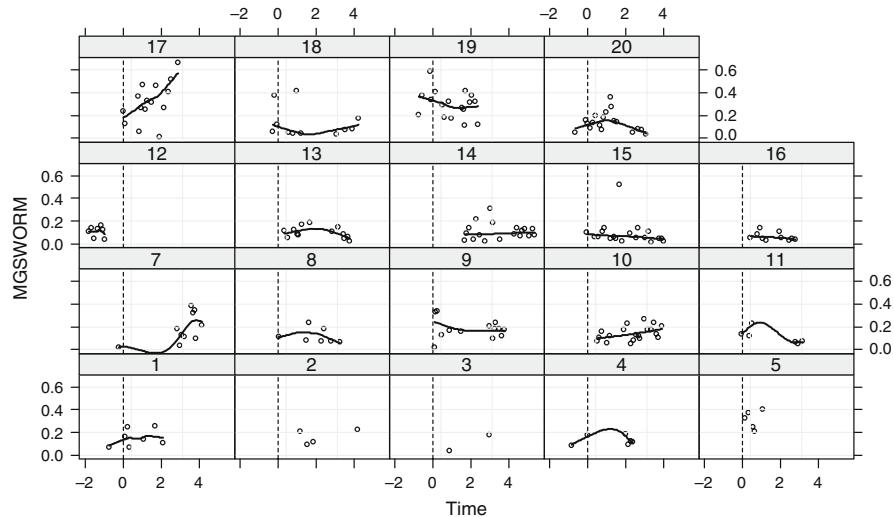


Fig. 7.10 An *xyplot* from the lattice package. The y-axis shows the intake rate (mg AFDW/s) of godwits, and the x-axis the time since low tide in hours. The numbers 1–20 refer to the sampling day. The vertical dotted line is the moment of low tide

shows that in some periods (late summer and fall), not all sexes are measured. Hence, we cannot include a sex-period interaction term.

The coplot accumulates the data from all sampling days. To show how intake rate changes on each day, we made an *xyplot* from the lattice package (Fig. 7.10). We added a LOESS smoother to aid visual interpretation. At some days, there seems to be a non-linear time effect; hence, we should perhaps model time as a quadratic function.

7.6.3 Linear Regression

Based on the data exploration, we think that a reasonably starting model is

$$\text{IntakeRate}_{ij} = \alpha + \beta_1 \times \text{Time}_{ij} + \beta_2 \times \text{Time}_{ij}^2 + \beta_3 \times \text{Sex}_{ij} + \beta_4 \times \text{Period}_{ij} + \varepsilon_{ij}$$

where the residuals are independently and normally distributed with mean 0 and variance σ^2 . The R code to import the data, make the two graphs, and apply the linear regression model is given below.

```
> library(AED); data(Limosa)
> Limosa$FID <- factor(Limosa$ID)
> Limosa$fPeriod <- factor(Limosa$Period,
  levels = c(0, 1, 2),
```

```

        labels = c("Summer", "LSummer.Fall",
                  "Winter"))
> Limosa$fSex <- factor(Limosa$Sex, levels = c(0, 1, 2),
                           Labels = c("Unk", "F", "M"))
> coplot(IntakeRate ~ Time | fPeriod * fSex,
          data = Limosa, xlab = c("Time (hours)"))
> library(lattice)
> xyplot(IntakeRate ~ Time | fID, data = Limosa,
          panel=function(x, y){
            panel.xyplot(x, y, col = 1, cex = 0.5, pch = 1)
            panel.grid(h = -1, v = 2)
            panel.abline(v = 0, lty = 2)
            if (length(x) > 5) panel.loess(x, y, span = 0.9,
                                              col = 1, lwd = 2)
          })

```

The first line accesses the data from our package. Because the nominal variables Sex and Period were coded as 0, 1, and 2, we relabelled them; this will make the numerical output of the models easier to understand. The `coplot` command is straightforward and the `xyplot` has some fancy commands in the panel function to draw the LOESS smoother (a smoother is only added if there are at least 5 observations on a particular day). With so few data points, we choose a large span width. The linear regression is applied with the following code. We also produce some numerical output.

```

> Limosa$Time2 <- Limosa$Time^2 - mean(Limosa$Time^2)
> M.lm <- lm(IntakeRate ~ Time + Time2 + fPeriod +
               fSex, data = Limosa)
> drop1(M.lm, test = "F")

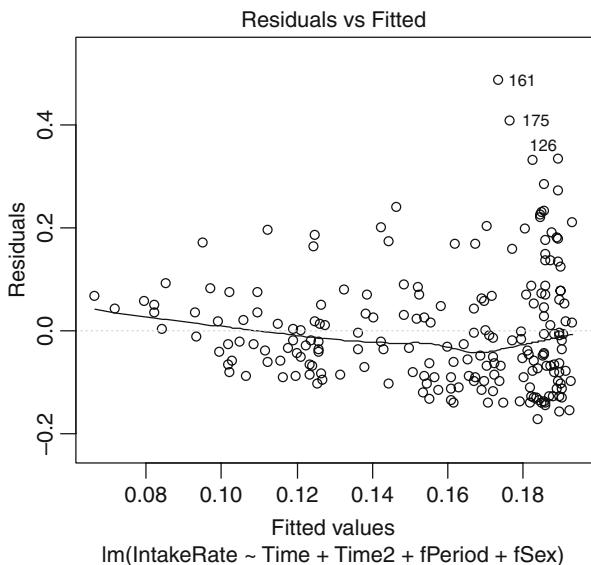
Single term deletions
Model: IntakeRate ~ Time + Time2 + fPeriod + fSex

      Df Sum of Sq    RSS      AIC F value    Pr(F)
<none>              2.74 -881.37
Time     1     0.01  2.75 -882.51  0.8330  0.362515
Time2    1     0.03  2.77 -881.10  2.2055  0.139095
fPeriod  2     0.01  2.75 -884.25  0.5460  0.580142
fSex     2     0.13  2.87 -875.73  4.7675  0.009491

```

We centred the quadratic time component to reduce the collinearity. Note that there is a significant sex effect; the F statistic is 4.76 with a corresponding p -value of 0.009. Good enough to start thinking about writing a paper! But to spoil the fun, let us plot the residuals versus the fitted values (Fig. 7.11) with the command `plot(M.lm, which = c(1))`. Note that there is clear violation of homogeneity. It is now time to go back to the protocols from Chapters 4 and 5.

Fig. 7.11 Residuals versus fitted values for the linear regression model. Note that there is heterogeneity



7.6.4 Protocol Time

In the previous subsection, we detected heterogeneity in the residuals of the linear regression model (which is step 1 of the protocol). We can now do two things. We can either mess around with variance covariates and then discover that there is still misery (in terms of correlation) or be clever and do everything at once. Assuming that you read this book from A to Z (and are therefore familiar with the material in Chapters 4 and 5), we follow the second approach. We will use the 10-step protocol from Chapter 4.

7.6.4.1 Step 2 of the Protocol: Refit with gls

In this step, we refit the linear regression with the `gls` function (so that we have a base model) and make some fancy graphical validation graphs; see Fig. 7.12. The R code does not contain any new aspects.

```
> library(nlme)
> M1.gls <- gls(IntakeRate ~ Time + Time2 + fPeriod +
+   fSex, data = Limosa)
> E <- resid(M1.gls)
> op <- par(mfrow = c(2, 2))
> boxplot(E ~ Limosa$fPeriod, main = "Period")
> abline(0, 0)
> boxplot(E ~ Limosa$fSex, main = "Sex")
```

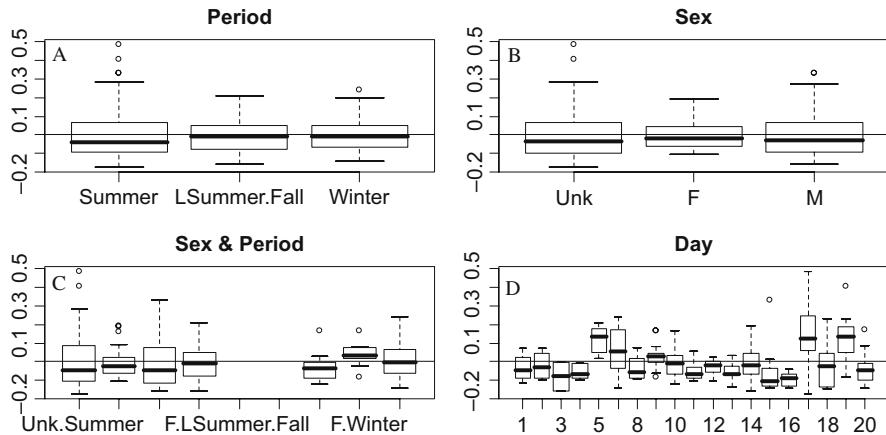


Fig. 7.12 Graphical validation plots for the linear regression model fitted with the `gls` function. **A:** Residuals versus period. **B:** Residuals versus sex. **C:** Residuals versus sex and period. **D:** Residuals versus day (coded by the variable ID). Due to lack of space, not all labels are presented on panels C and D

```
> abline(0, 0)
> boxplot(E ~ Limosa$fSex * Limosa$fPeriod,
           main = "Sex & Period")
> abline(0, 0)
> boxplot(E ~ Limosa$ID, main = "Day")
> abline(0, 0)
> par(op)
```

Note that the variation in residual spread is larger for the unknown sex, and it is also larger for the summer period. This means that in step 3 of the protocol, we could do with a `varIdent` variance structure with the variance covariates `Period` and `Sex`. Figure 7.12D shows that we need the term `ID` (sampling day) as an explanatory variable; at some days, all the residuals are above or below zero. We can either use `ID` as a fixed effect or as a random effect. In this example, it is obvious to use it as a random effect (it allows for correlation between observations from the same day; it avoids estimating lots of parameters and it allows us to generalise the conclusions); see also Chapter 5.

7.6.4.2 Step 3 of the Protocol: Choose an Appropriate Variance Structure

We already discussed in the previous step that we need a `varIdent` variance structure and `ID` as random effect. Such a model is given by

```
> M1.lme <- lme(IntakeRate ~ Time + Time2 + fPeriod +
  fSex, data = Limosa,
```

```
weights = varIdent(form =~ 1 | fSex * fPeriod),
random =~ 1 | fID, method = "REML")
```

Perhaps it is useful to give the corresponding equation for this, just in case you find it difficult to see this from the R code.

$$\begin{aligned} \text{IntakeRate}_{ij} &= \alpha + \beta_1 \times \text{Time}_{ij} + \beta_2 \times \text{Time}_{ij}^2 + \beta_3 \times \text{Period}_{ij} + \beta_4 \times \text{Sex}_{ij} + a_i + \varepsilon_{ij} \\ a_i &\sim N(0, d^2) \\ \varepsilon_{ij} &\sim N(0, \sigma_{\text{Sex}, \text{Period}}^2) \end{aligned}$$

We have seen most of this equation already in Section 7.6.1. The term a_i is the random intercept (Chapter 5). The subscripts for the σ are there because we allow for different residual variances depending on sex and period.

7.6.4.3 Steps 4–6 of the Protocol: Find the Optimal Random Structure

We are going to save some space by summarising a couple of model selection steps. The model that was fitted in step 3 is the optimal one in terms of the random structure. Leave out the random effect, refit the model, and compare both models with the likelihood ratio test, and you will get p -values smaller than 0.001. The same holds if you drop the `varIdent` variance structure if you use the `varIdent` with only sex or only with period. The R code to do these analyses was given in Chapters 4 and 5.

7.6.4.4 Steps 7–8 of the Protocol: Find the Optimal Fixed Structure

It is now time to find the optimal model in terms of the explanatory variables time, period, and sex. We use the likelihood ratio test with ML estimation. The starting model contains Time, Time², Period, and Sex. The last three can be dropped (Time is nested in Time² and cannot be dropped). The R code to do this is as follows.

```
> M1.lme <- lme(IntakeRate ~ Time + Time2 + fPeriod +
+ fSex, data = Limosa,
+ weights = varIdent(form =~ 1 | fSex * fPeriod),
+ random =~ 1 | fID, method = "ML")
> M1.lmeA <- update(M1.lme, . ~. -Time2)
> M1.lmeB <- update(M1.lme, . ~. -fPeriod)
> M1.lmeC <- update(M1.lme, . ~. -fSex)
> anova(M1.lme, M1.lmeA)
> anova(M1.lme, M1.lmeB)
> anova(M1.lme, M1.lmeC)
```

The output is not shown here, but the least significant term is Period ($L = 1.28$, $df = 2$, $p = 0.52$); hence, it can be dropped. In the next round, Time² is

dropped, followed by Time in the third round. In the fourth and last round, we have a model that only contains Sex. The following code gives us one *p*-value for the nominal variable Sex (the update command fits a model with only the intercept):

```
> M4.lme <- lme(IntakeRate ~ fSex, data = Limosa,
+   weights = varIdent(form =~ 1 | fSex * fPeriod),
+   random =~ 1 | fID, method = "ML")
> M4.lmeA <- update(M4.lme, .~. -fSex)
> anova(M4.lme, M4.lmeA)

Model df      AIC      BIC    logLik  Test L.Ratio p-value
M4.lme     1 11 -359.3379 -322.6779 190.6689
M4.lmeA    2  9 -355.4784 -325.4839 186.7392 1 vs 2 7.85945  0.0196
```

Hence, the optimal model contains only Sex in the fixed part of the model. If we have to quote a *p*-value for this term, it will be 0.0196, which is not very impressive. A model validation shows that everything is now ok (no heterogeneity patterns in the normalised residuals).

7.6.4.5 Step 9 of the Protocol: Refit with REML

We now discuss the numerical output of the model. First we have to refit it with REML.

```
> M4.lme <- lme(IntakeRate ~ fSex, data = Limosa,
+   weights = varIdent(form =~ 1 | fSex * fPeriod),
+   random =~ 1 | fID, method = "REML")
> summary(M4.lme)

Linear mixed-effects model fit by REML. Data: Limosa
      AIC      BIC    logLik
-340.1566 -303.6573 181.0783

Random effects:
Formula: ~1 | fID
        (Intercept) Residual
StdDev:  0.06425989 0.1369959

Variance function:
Structure: Different standard deviations per stratum
Formula: ~1 | fSex * fPeriod

Parameter estimates:
Unk*Summer  Unk*LSummer.Fall  M*Winter  F*Winter  Unk*Winter
          1.0000      0.4938      0.6249      0.5566      0.5035
M*Summer          F*Summer
          0.7971      0.4366
```

```

Fixed effects: IntakeRate ~ fSex
                Value Std.Error DF   t-value p-value
(Intercept)  0.15051634 0.01897585 186  7.931993 0.00000
fSexF        -0.02507688 0.01962955 186 -1.277506 0.20300
fSexM        0.01999006 0.01863430 186  1.072756 0.28480

Correlation:
  (Intr) fSexF
fSexF -0.491
fSexM -0.470  0.653

Number of Observations: 207. Number of Groups: 19

```

Let us discuss what this all means. Recall from Chapter 5 that in a mixed effects model with random intercept, the correlation between the observations from the same group (in this case: the same day), is given by

$$\frac{d^2}{d^2 + \sigma^2}$$

The problem is that in this case, we do not have one variance σ^2 , but we have a σ^2 that depends on Sex and Period. This means that the within-day correlation is given by

$$\frac{d^2}{d^2 + (s_{ij} \times \sigma^2)} = \frac{0.064^2}{0.064^2 + (s_{ij} \times 0.136)^2}$$

The s_{ij} 's are the multiplication factors denoted by ‘Different standard deviations per stratum’ in the numerical output. The largest value of s_{ij} is 1 for unknown sex in the summer, leading to a within-day correlation of 0.18. On the other extreme, for females in the summer, $s_{ij} = 0.436$, which leads to a within-day correlation of 0.54. Note that this correlation was called the intraclass correlation in Chapter 5.

As a final note, the p -values for the individual levels of sex (based on the t -statistic) are all larger than 0.05, but keep in mind that these p -values are with respect to the baseline level “Unknown”. The fact that the likelihood ratio test showed that sex was significant (though only weakly, the p -value was 0.0196), means that males and females are having a different effect. Just change the baseline of the variable fSex to verify this.

7.6.5 Why All the Fuss?

You may wonder what the benefit is of the mixed modelling approach. Let us compare the optimal mixed effects model with the other models. Recall that the linear regression model in Section 7.6.3 gave us a p -value of 0.009 for Sex. That is rather a different p -value compared to the 0.0196 from the mixed model. Ok, you can argue that the linear regression model contained various non-significant terms.

No problem; let us drop them and refit the linear regression model with only Sex as explanatory variable.

```
> M2.lm <- lm(IntakeRate ~ fSex, data = Limosa)
> drop1(M2.lm, test = "F")

Single term deletions
Model: IntakeRate ~ fSex
      Df Sum of Sq    RSS     AIC F value    Pr(F)
<none>          2.80 -884.38
fSex    2      0.15   2.96 -877.56   5.475 0.004829
```

Hence, in the linear regression model in which we only use Sex, this term has a *p*-value of 0.0048. You may argue that you should not compare the linear regression with the linear mixed model as the linear regression model ignores the heterogeneity. Ok, let us fit a model that allows for heterogeneity, but without the random effect and obtain a *p*-value for sex.

```
> M5A.gls <- gls(IntakeRate ~ fSex, data = Limosa,
+                   weights = varIdent(form = ~ 1 | fSex * fPeriod),
+                   method = "ML")
> M5B.gls <- gls(IntakeRate ~ 1, data = Limosa,
+                   weights = varIdent(form = ~ 1 | fSex * fPeriod),
+                   method = "ML")
> anova(M5A.gls, M5B.gls)

      Model df      AIC      BIC logLik   Test L.Ratio p-value
M5A.gls     1 10 -321.8643 -288.5371 170.9322
M5B.gls     2  8 -311.9607 -285.2989 163.9803 1 vs 2 13.90364 0.001
```

The analysis of variance compares a model with sex and without sex. Both have the `varIdent` variance structure, but not the random intercept. We are still let to believe that sex is highly significant. What this means is that as soon as we include the random intercept, we allow for correlation between observations on the same day. For some sex-period combinations, this correlation can be as high as 0.54. Ignoring this correlation means that we end up with a *p*-value of 0.001. Taking it into account gives a *p*-value of 0.0196. The difference is a factor of 20. This example shows the danger of ignoring temporal correlation, something which happens in many scientific papers on ecology.

In case you enjoyed this analysis, try fitting the correlation structure with the compound symmetry correlation directly as an exercise. With this we mean that you can also use the `correlation = corCompSymm()` instead of random effects. And a more complicated approach would be to use any of the spatial correlation functions.

Chapter 8

Meet the Exponential Family

8.1 Introduction

In Chapters 2 and 3 and in Appendix A, linear regression and additive modelling were discussed and various extensions allowing for different variances, nested data, temporal correlation, and spatial correlation were then discussed in Chapters 4, 5, 6, and 7. In Chapters 8, 9, and 10, we discuss generalised linear modelling (GLM) and generalised additive modelling (GAM) techniques. In linear regression and additive modelling, we use the Normal (or: Gaussian) distribution. It is important to realise that this distribution applies for the response variable. GLM and GAM are extensions of linear and additive modelling in the sense that a non-Gaussian distribution for the response variable is used and the relationship (or link) between the response variable and the explanatory variables may be different. In this chapter, we focus on the first point, the distribution.

There are many reasons for using GLM and GAM instead of linear regression and additive modelling. Absence–presence data are (generally) coded as 1 and 0, proportional data are always between 0 and 100%, and count data are always non-negative. The GLM and GAM models used for 0–1 and proportional data are typically based on the Bernoulli and binomial distributions and for count data the Poisson and negative binomial distributions are common options. For continuous data, the Gaussian distribution is the most used distribution, but you can also use the gamma distribution. So before using GLMs and GAMs, we should focus on the questions: What are these distributions, how do they look like, and when would you use them? These three questions form the basis of this chapter. We devote an entire chapter to this topic because in our experience few of our students have been familiar with Poisson, negative binomial or gamma distributions, and some level of familiarity is required before entering the world of GLMs and GAMs in the next chapter.

As we will see in the next chapter, a GLM (or GAM) consists of three steps: (i) choosing a distribution for the response variable, (ii) defining the systematic part in terms of covariates, and (iii) specifying the relationship (or: link) between the expected value of the response variable and the systematic part. This means that we have to stop for a moment and think about the nature of the response variable.

In most statistics textbooks and undergraduate statistics courses, only the Normal, Poisson, and binomial distributions are discussed in any detail. However, there are various other distributions that are equally interesting for ecological data, for example, the negative binomial distribution. These are useful if the ‘ordinary’ GLMs do not work, and in practise, this is quite often in ecological data analysis.

Useful references for distributions within the context of GLMs are McCullagh and Nelder (1989), Hilbe (2007), and Hardin and Hilbe (2007). It should be noted that most books on GLMs discuss distributions, but these three have detailed explanations.

8.2 The Normal Distribution

We start with some revision on the Normal distribution. Figure 8.1 A shows the histogram of the weight of 1280 sparrows (unpublished data from Chris Elphick, University of Connecticut, USA). The y-axis in panel A shows the number per class. It is also possible to rescale the y-axis so that the total surface under the histogram adds up to 1 (Fig. 8.1B). The reason for doing this is to give a better representation of the density curve that we are going to use in a moment. The shape of the histogram suggests that assuming normality may be reasonable, even though the histogram is

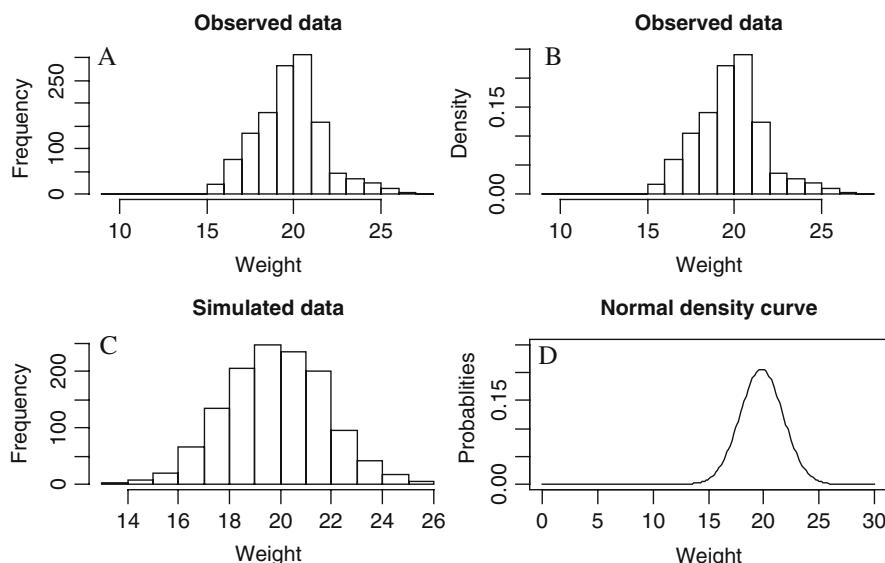


Fig. 8.1 **A:** Histogram of weight of 1281 sparrows. **B:** As panel A, but now scaled so that the total area in the histogram is equal to 1. **C:** Histogram of simulated data from a Normal distribution with mean and variance taken from the 1281 sparrows. **D:** Normal probability curve with values for the mean and the variance taken from the sample of 1281 sparrows. The surface under the Normal density curve adds up to 1

slightly skewed. Panel C shows simulated data (1280 observations) from a Normal distribution with the sample mean (18.9) and sample variance (3.7) from the 1280 sparrows. The shape of the histogram in panel C gives an impression of how a distribution looks if the data are really Normal distributed. Repeating this simulation ten times gives a good idea how much variation you can expect in the shape of the histogram.

Possible factors determining the weight of a sparrow are sex, age, time of the year, habitat, and diet, among others. But for the moment, we will not take these into account. The Normal distribution is given by the following formula:

$$f(y_i; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(y_i - \mu)^2}{2\sigma^2}} \quad (8.1)$$

The distribution function in Equation (8.1) gives the probability that bird i has a weight y_i , and μ and σ^2 are the population mean and variance, respectively, in the following formula:

$$E(Y) = \mu \quad \text{and} \quad \text{var}(Y) = \sigma^2 \quad (8.2)$$

The probability function is also called a density function. The notation $f(y_i; \mu, \sigma)$ means that the parameters are after the ‘;’ symbol. The variable y can take any value between $-\infty$ and ∞ . In general, we do not know the population mean μ and variance σ^2 , but if we just take the sample mean and variance and substitute these into the distribution function in Equation (8.1), we can calculate the probabilities for various values of y ; see Fig. 8.1D. Note that the y -axis in this panel represents probabilities of certain weight values. So, the probability that we measure a sparrow of weight 20 g is about 0.21, and for 5 g, the probability is very small. According to the Normal distribution, we can even measure a bird with weight -10 g, though with a very small probability.

In linear regression, we model the expected values μ_i (the index i refers to observations or cases) as a function of the explanatory variables, and this function contains unknown regression parameters (intercept and slopes).

The following R code was used to create Fig. 8.1.

```
> library(AED); data(Sparrows)
> op <- par(mfrow = c(2, 2))
> hist(Sparrows$wt, nclass = 15, xlab = "Weight",
       main = "Observed data")
> hist(Sparrows$wt, nclass = 15, xlab = "Weight",
       main = "Observed data", freq = FALSE)
> Y <- rnorm(1281, mean = mean(Sparrows$wt),
              sd = sd(Sparrows$wt))
> hist(Y, nclass = 15, main = "Simulated data",
       xlab = "Weight")
> X <- seq(from = 0, to = 30, length = 200)
```

```
> Y <- dnorm(X, mean = mean(Sparrows$wt),
               sd = sd(Sparrows$wt))
> plot(X, Y, type = "l", xlab = "Weight",
       ylab = "Probabilities", ylim = c(0, 0.25),
       xlim = c(0, 30), main = "Normal density curve")
> par(op)
```

The `freq = FALSE` option in the histogram scales it so that the area inside the histogram equals 1. The function `rnorm` takes random samples from a Normal distribution with a specified mean and standard deviation. The functions `mean` and `sd` calculate the mean and standard deviation of the weight variable `wt`. Similarly, the function `dnorm` calculates the Normal density curve for a given range of values `X` and for given mean and variance.

In this case, the histogram of the observed weight data (Fig. 8.1B) indicates that the Normal distribution may be a reasonable starting point. But what do you do if it is not (or if you do not agree with our statement)? The first option is to apply a data transformation, but this will also change the relationship between the response and explanatory variables. The second option is to do nothing yet and hope that the residuals of the model are normally distributed (and the explanatory variables cause the non-normality). Another option is to choose a different distribution and the type of data determines which distribution is the most appropriate. The best way to get some familiarity with different distributions for the response variable is to plot them. We have already seen the Normal distribution in Fig. 8.1, and also in Chapter 2. The second distribution we now discuss is the Poisson distribution.

8.3 The Poisson Distribution

The Poisson distribution function is given by

$$f(y; \mu) = \frac{\mu^y \times e^{-\mu}}{y!} \quad y \geq 0, \quad y \text{ integer} \quad (8.3)$$

This formula specifies the probability of Y with a mean μ . Note that Y has to be an integer value or else the $y! = y \times (y - 1) \times (y - 2) \times \dots \times 1$ is not defined. Once we know μ , we can calculate the probabilities for different y values. For example, if $\mu = 3$, the probability that $y = 1$ is given by $3 \times e^{-3} / (1!) = 0.149$. The same can be done for other values of y . Figure 8.2 shows four Poisson probability distributions, and to create these graphs, we used different values for the average μ . For small μ , the density curve is skewed, but for larger μ , it becomes symmetrical. Note that μ can be a non-integer, but the y s have to be non-negative and integers. Other characteristics of the Poisson distribution are that $P(Y < 0) = 0$ and the mean is the variance, in formula

$$E(Y) = \mu \quad \text{and} \quad \text{var}(Y) = \mu \quad (8.4)$$

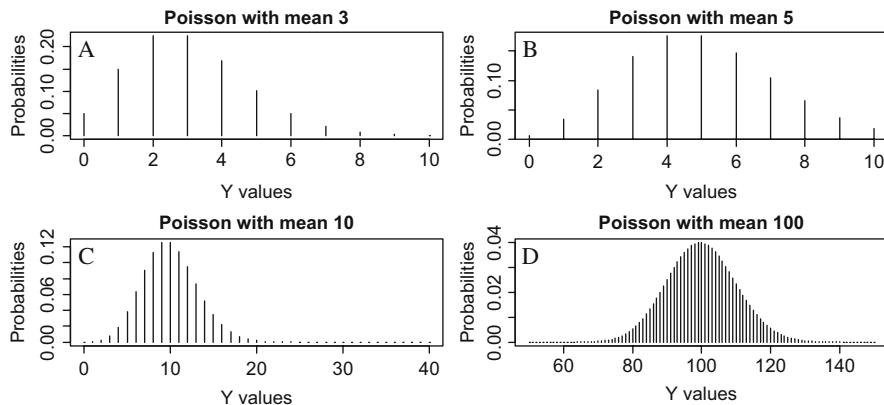


Fig. 8.2 Poisson probabilities for $\mu = 3$ (A), $\mu = 5$ (B), $\mu = 10$ (C), and $\mu = 100$ (D). Equation (8.3) is used to calculate the probabilities for certain values. Because the outcome variable y is a count, vertical lines are used instead of a line connecting all the points

This is also the reason that the probability distributions become wider and wider for larger mean values. Note that although the Poisson probability distribution in Fig. 8.2D looks like a normal distribution, it is not *equal* to a Normal distribution; a Normal distribution has two parameters (the mean μ and the variance σ^2), whereas a Poisson distribution only uses one parameter μ (which is the mean and the variance).

The following code was used to make Fig. 8.2.

```
> x1 <- 0:10;    Y1 <- dpois(x1, lambda = 3)
> x2 <- 0:10;    Y2 <- dpois(x2, lambda = 5)
> x3 <- 0:40;    Y3 <- dpois(x3, lambda = 10)
> x4 <- 50:150;  Y4 <- dpois(x4, lambda = 100)
> XLab <- "Y values"; YLab <- "Probabilities"
> op <- par(mfrow = c(2, 2))
> plot(x1, Y1, type = "h", xlab = XLab, ylab = YLab,
       main = "Poisson with mean 3")
> plot(x2, Y2, type = "h", xlab = XLab, ylab = YLab,
       main = "Poisson with mean 5")
> plot(x3, Y3, type = "h", xlab = XLab, ylab = YLab,
       main = "Poisson with mean 10")
> plot(x4, Y4, type = "h", xlab = XLab, ylab = YLab,
       main = "Poisson with mean 100")
> par(op)
```

The function `dpois` calculates the Poisson probabilities for a given μ , and it calculates the probability for certain Y -values using Equation (8.3). Note that we

use the symbol ‘;’ to print multiple R commands on one line; it saves space. The `type = "h"` part in the `plot` command ensures that vertical lines are used in the graph. The reason for using vertical lines is because the Poisson distribution is for discrete data.

In the graphs in Fig. 8.2, we pretended we knew the value of the mean μ , but in real life, we seldom know its value. A GLM models the value of μ as a function of the explanatory variables; see Chapter 9.

The Poisson distribution is typically used for count data, and its main advantages are that the probability for negative values is 0 and that the mean variance relationship allows for heterogeneity. However, in ecology, it is quite common to have data for which the variance is even larger than the mean, and this is called overdispersion. Depending how much larger the variance is compared to the mean, one option is to use the correction for overdispersion within the Poisson GLM, and this is discussed in Chapter 9. Alternatively, we may have to choose a different distribution, e.g. the negative binomial distribution, which is discussed in the next section.

8.3.1 Preparation for the Offset in GLM

The Poisson distribution in Equation (8.2) is written for only one observation, but in reality we have multiple observations. So, we need to add an index i to y and μ .

Penston et al. (2008) analysed the number of sea lice at sites around fish farms in the north-west of Scotland as a function of explanatory variables like time, depth, and station. The response variable was the number of sea lice at various sites i , denoted by N_i . However, samples were taken from a volume of water, denoted by V_i , that differed per site. One option is to use the density N_i/V_i as the response variable and work with a Gaussian distribution, but if the volumes differ considerably per site, then this is a poor approach as it ignores the differences in volumes.

Alternative scenarios are the number of arrivals Y_i per time unit t_i , numbers Y_i per area of size A_i , and number of bioluminescent flashes per depth range V_i . All these scenarios have in common that the volume V_i , time unit t_i , area of size A_i , may differ per observation i , making the ratio of Y_i and V_i a rate or density.

We can still use the Poisson distribution for this type of data. For example, for the sea lice data, we assume that Y_i is Poisson distributed with probability function:

$$f(y_i; \mu_i) = \frac{(V_i \times \mu_i)^{y_i} \times e^{-V_i \times \mu_i}}{y_i!} \quad (8.5)$$

The parameter μ_i is now the expected number of sea lice at site i for a 1-unit volume. If all the values V_i are the same, we may as well drop it (for the purpose of a GLM) and work with the Poisson distribution in Equation (8.3).

8.4 The Negative Binomial Distribution

We continue the trail of distribution functions with another discrete one: the negative binomial. There are various ways of presenting the negative binomial distribution and a detailed explanation can be found in Hilbe (2007). Because we are working towards a GLM, we present the negative binomial used in GLMs. It is presented in the literature as a combination of two distributions, giving a combined Poisson-gamma distribution. This means we first assume that the Y s are Poisson distributed with the mean μ assumed to follow a gamma distribution. With some mathematical manipulation, we end up with the negative binomial distribution for Y . Its density function looks rather more intimidating than that of the Poisson or Normal distributions and is given by

$$f(y; k, \mu) = \frac{\Gamma(y + k)}{\Gamma(k) \times \Gamma(y + 1)} \times \left(\frac{k}{\mu + k} \right)^k \times \left(1 - \frac{k}{\mu + k} \right)^y \quad (8.6)$$

Nowadays, the negative binomial distribution is considered a stand-alone distribution, and it is not necessary to dig into the Poisson-gamma mixture background. The distribution function has two parameters: μ and k . The symbol Γ is defined as: $\Gamma(y + 1) = (y + 1)!$. The mean and variance of Y are given by

$$E(Y) = \mu \quad \text{var}(Y) = \mu + \frac{\mu^2}{k} \quad (8.7)$$

We have overdispersion if the variance is larger than the mean. The second term in the variance of Y determines the amount of overdispersion. In fact, it is indirectly determined by k , where k is also called the dispersion parameter. If k is large (relative to μ^2), the term μ^2/k approximates 0, and the variance of Y is μ ; in such cases the negative binomial converges to the Poisson distribution. In this case, you might as well use the Poisson distribution. The smaller k , the larger the overdispersion.

Hilbe (2007) uses a different notation for the variance, namely,

$$\text{var}(Y) = \mu + \alpha \times \mu^2$$

This notation is slightly easier as $\alpha = 0$ means that the quadratic term disappears. However, the R code below uses the notation in Equation (8.7); so we will use it here.

It is important to realise that this distribution is for discrete (integers) and non-negative data. Memorising the complicated formulation of the density function is not needed; the computer can calculate the Γ terms. All you need to remember is that with this distribution, the mean of Y is equal to μ and the variance is $\mu + \mu^2/k$.

The probability function in Equation (8.6) looks complicated, but it is used in the same way as we used it in the previous section. We can specify a μ value and a k value, and calculate the probability for a certain y value. To get a feeling for the shape of the negative binomial probability curves, we drew a couple of density

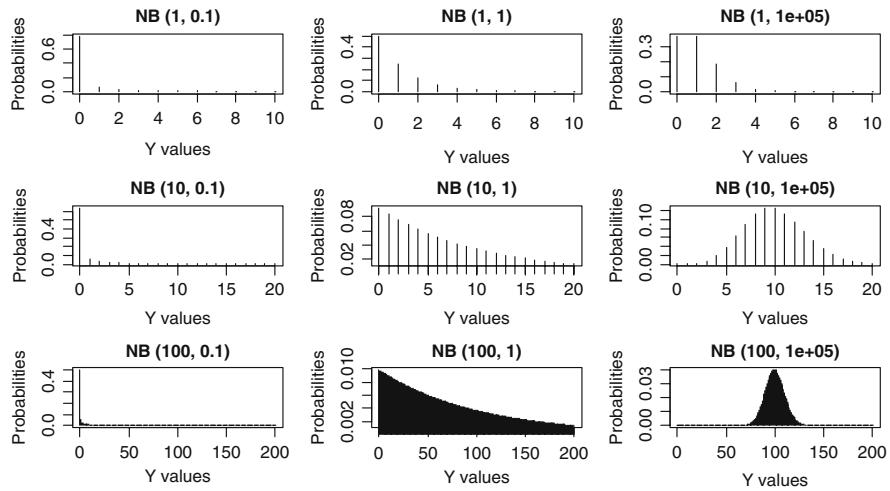


Fig. 8.3 Nine density curves from a negative binomial distribution $\text{NB}(\mu, k)$, where μ is the mean and k^{-1} is the dispersion parameter. The column of panels on the right have a large k , and these negative binomial curves approximate the Poisson distribution. R code to create this graph is given on the book website. If $k = 1$, the negative binomial distribution is also called the geometric distribution

curves for various values of μ and k , see Fig. 8.3. We arbitrarily choose three values for μ , of 1, 10, and 100. We also choose arbitrarily three values for k , of 0.1, 1, and 100,000. For $k = 100,000$, we expect to see a distribution function similar to the Poisson distribution with mean and variance μ , and this is indeed the case: see the panels in the right column. The three panels in the middle column have $E(Y) = \mu$ and $\text{var}(Y) = \mu + \mu^2$, because $k = 1$.

If we set $k = 1$ in the negative binomial distribution, then the resulting distribution is called the geometric distribution. Its mean and variance are defined by

$$E(Y) = \mu \quad \text{var}(Y) = \mu + \mu^2 \quad (8.8)$$

Hence, the variance increases as a quadratic function of the mean. As with the Poisson distribution, observations of the response variables with the value of zero are allowed in the negative binomial and the geometric distribution. Most software will not have a separate function for the geometric distribution; just set the parameter k in the software for a negative binomial equal to 1.

Returning to the negative binomial probability function, note that for a small mean μ and large overdispersion (small k), the value of 0 has by far the highest probability.

In Fig. 8.3 we know the values of μ and k . In reality we do not know these values, and in GLM models, the mean μ is a function of covariates. Estimation of k depends on the software, but can for example be done in a 2-stage iterative approach (Agresti, 2002).

8.5 The Gamma Distribution

The gamma distribution can be used for a continuous response variable Y that has positive values ($Y > 0$), and the distribution function has various forms. Within the context of a GLM, we use (Faraway, 2006)

$$f(y; \mu, v) = \frac{1}{\Gamma(v)} \times \left(\frac{v}{\mu}\right)^v \times y^{v-1} \times e^{\frac{-y}{\mu}} \quad y > 0 \quad (8.9)$$

Before starting to memorise the exact mathematical definition of this density function, let us first look at the mean and variance of a variable Y that is gamma distributed and sketch the density curve for various values of μ and v (which is the equivalent of the k in the negative binomial distribution). The mean and variance of Y are

$$E(Y) = \mu \quad \text{and} \quad \text{var}(Y) = \frac{\mu^2}{v} \quad (8.10)$$

The dispersion is determined by v^{-1} ; a small value of v (relative to μ^2) implies that the spread in the data is large. Density curves for different values of μ and v are given in Fig. 8.4. Note the wide range of shapes between these curves. For a large v , the gamma distribution becomes bell shaped and symmetric. In such cases, the Gaussian distribution can be used as well. Faraway (2006) gives an example of a linear regression model and a gamma GLM with a small (0.0045) dispersion parameter v^{-1} ; estimated parameters and standard errors obtained by both methods are nearly identical. However, for larger values of v^{-1} , this is not necessarily the case.

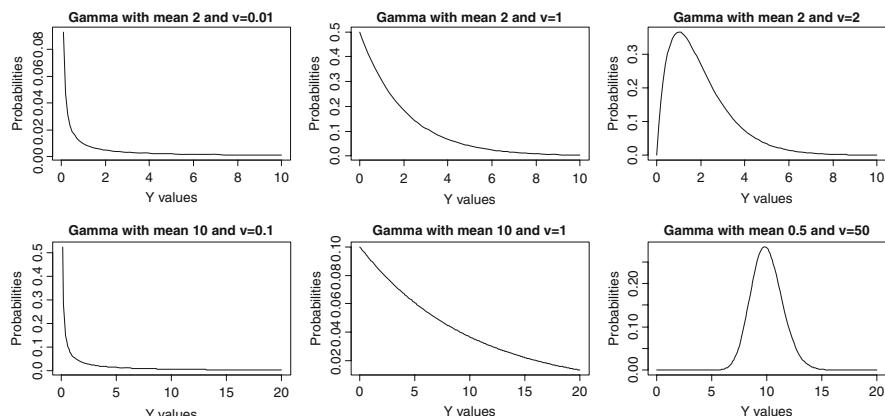


Fig. 8.4 Gamma distributions for different values of μ and v . The R function `dgamma` was applied, which uses a slightly different parameterisation: $E(Y) = a \times s$ and $\text{var}(Y) = a \times s^2$, where a is called the shape and s the scale. In our parameterisation, $v = a$ and $\mu = a \times s$

Note that the allowable range of Y values is larger than 0. So, you cannot use this distribution if your response variable takes negative values or has a value of zero.

8.6 The Bernoulli and Binomial Distributions

The last two distributions we review are the Bernoulli and binomial distributions, and we start with the latter. In a first year statistics course, it is often introduced as the distribution that is used for tossing a coin. Suppose you know that a coin is fair (no one has tampered with it and the probability of getting a head is the same as getting a tail), and you toss it 20 times. The question is how many heads do you expect? The possible values that you can get are from 0 to 20. Obviously, the most likely value is 10 heads. Using the binomial distribution, we can say how likely it is that you get 0, 1, 2, . . . , 19 or 20 heads.

A binomial distribution is defined as follows. We have N independent and identical trials, each with probability $P(Y_i = 1) = \pi$ of success, and probability $P(Y_i = 0) = 1 - \pi$ on failure. The labels ‘success’ and ‘failure’ are used for the outcomes of 1 and 0 of the experiment. The label ‘success’ can be thought of $P(Y_i = \text{head})$, and ‘failure’ can be $P(Y_i = \text{tail})$. The term independent means that all tosses are unrelated. Identical means that each toss has the same probability of success. Under these assumptions, the density function is given by

$$f(y; \pi) = \binom{N}{y} \times \pi^y \times (1 - \pi)^{N-y} \quad (8.11)$$

The probability for each value of y between 0 and 20 for the tossing example can be calculated with this probability function. For example, if $N = 20$ and $\pi = 0.5$, then the probability of measuring 9 heads is $(20!/(9! \times 11!)) \times 0.5^9 \times (1 - 0.5)^{11}$. The value can either be obtained from a calculator or you can read it from the panel in the middle of Fig. 8.5 ($N = 20$, $\pi = 0.5$). As expected, the value $y = 10$ has the highest probability, but 9 and 11 have very similar probabilities. The probability of getting 20 heads is close to zero; it is too small to read on the vertical axis (it is in fact something that starts with 7 zeros). For some arbitrarily chosen values of π and N , we drew more Binomial probability curves, just to get a feel for the shape of the density curves (Fig. 8.5).

The mean and variance of a Binomial distribution are given by

$$E(Y) = N \times \pi \quad \text{var}(Y) = N \times \pi \times (1 - \pi) \quad (8.12)$$

So, if you know that the probability of tossing a head is 0.5 and toss a coin 20 times, then the answer to the question that we started this section with is $20 \times 0.5 = 10$ heads.

In ecology, we are (hopefully) not tossing with coins, but instead we may go to a deer farm and sample N animals for the presence and absence of a particular disease. In such a research, you want to know the probability π that a particular animal is infected with the disease. Other examples are the presence or absence of koalas at

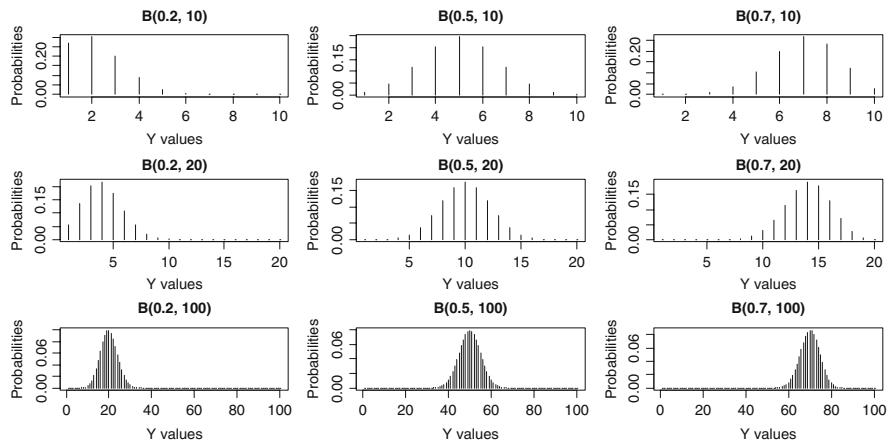


Fig. 8.5 Binomial density curves $B(\pi, N)$ for various values of π (namely 0.2, 0.5, and 0.7) and N (namely 10, 20, and 100). R code to create this graph is on the book website

particular sites (see Chapter 20 for a detailed example), badger activity (yes or no) around farms (Chapter 21), or the presence and absence of flat fish at 62 sites in an estuary (Zuur et al., 2007).

In the example of the N deer at the farm, we do not know the value of π and the GLM is used to model π as a function of covariates. In such a research problem, you can also question if your sample of 20 animals from the same farm is independent. But we leave this problem until Chapters 12 and 13.

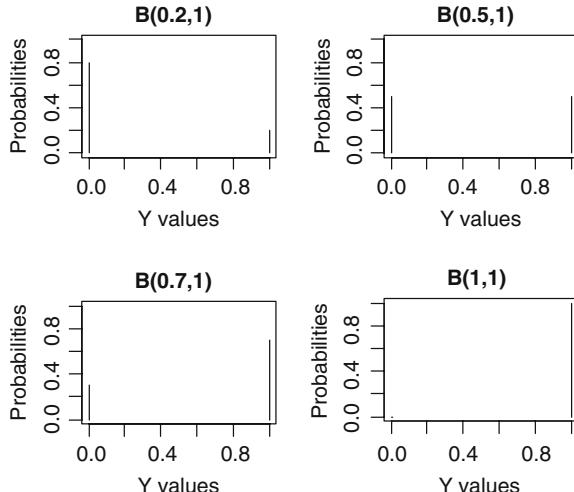


Fig. 8.6 Four Bernoulli distributions $B(\pi, 1)$ for different values of π . R code to create this graph is on the book website

A Bernoulli distribution is obtained if $N = 1$; hence, we only toss once or we only sample one animal on the farm. Four Bernoulli distributions with $\pi = 0.2$, $\pi = 0.5$, $\pi = 0.7$, and $\pi = 1$ are given in Fig. 8.6. Note that we only get a value of the probabilities at 0 (failure) and 1 (success).

In general, we do not make a distinction between a binomial and Bernoulli distribution and use the notation $B(\pi, N)$ for both, and $N = 1$ automatically implies the Bernoulli distribution.

8.7 The Natural Exponential Family

So far, we have discussed the Normal, Poisson, negative binomial, gamma, binomial, and Bernoulli distributions. There are, however, a lot more distributions around, for example, the multinomial distribution (useful for a response variable that is a categorical variable with more than two levels) and inverse Gaussian distribution (e.g. for lifetime distributions; these can be used for failure time of machines in production processes or lifetime of a product). It is relatively easy to show that all the distributions we have used so far can be written in a general formulation:

$$f(y; \theta, \phi) = e^{\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \theta)} \quad (8.13)$$

For example, if we use $\theta = \log(\mu)$, $\phi = 1$, $a(\phi) = 1$, $b(\theta) = \exp(\theta)$, $c(y, \phi) = -\log(y!)$, we get the Poisson distribution function. Similar definitions exist for the binomial, negative binomial, geometric, Normal, and gamma distributions; see McCullagh and Nelder (1989), Dobson (2002), Agresti (2002), or Hardin and Hilbe (2007). The advantage of this general notation is that when we build up a maximum likelihood criterion and optimise this to estimate the regression parameters, we can do this in terms of the general notation. This means that one set of equations can be used for all these distributions.

Using first- and second-order derivatives for the density function specified in Equation (8.13), we can easily derive an expression for the mean and variance of Y . These are as follows:

$$\begin{aligned} E(Y) &= b'(\theta) \\ \text{var}(Y) &= b''(\theta) \times a(\phi) \end{aligned} \quad (8.14)$$

The notation $b'(\theta)$ refers to the first-order derivative of the function b with respect to θ , and $b''(\theta)$ the second-order derivative. If you check this for the Poisson distribution, you will see that we get the familiar relationships $E(Y) = \mu$ and $\text{var}(Y) = \mu$.

The term $a(\phi)$ determines the dispersion. In the Gaussian linear regression model, we have $\theta = \mu$, $\phi = \sigma^2$, $a(\phi) = \phi$, $b(\theta) = \theta^2/2$, and $c(y, \phi) = -(y^2/\phi + \log(2\pi\phi))/2$, which gives us $E(Y) = \mu$ and $\text{var}(Y) = \sigma^2$.

Hence, the notation in Equation (8.13) allows us to summarise all distribution functions discussed so far in a general notation, and the mean and variance are specified by the set of equations in (8.14).

8.7.1 Which Distribution to Select?

We have discussed a large number of distributions for the response variable, but which one should we use? This choice should, in first instance, be made *a priori* based on the available knowledge on the response variable. For example, if you model the presence and absence of animals at M sites as a function of a couple of covariates, then your choice is simple: the binomial distribution should be used because your response variable contains zeros and ones. This is probably the only scenario where the choice is so obvious. Having said that, if we aggregate the response variable into groups, we (may) have a Poisson distribution.

If your data are counts (of animals, plants, etc.) without an upper limit, then the Poisson distribution is an option. This is because counts are always non-negative, and tend to be heterogeneous and both comply with the Poisson distribution. If there is high overdispersion, then the negative binomial distribution is an alternative to the Poisson for count data.

You can also use the Normal distribution for counts (potentially combined with a data transformation), but the Poisson or negative binomial may be more appropriate. However, the Normal distribution does not exclude negative realisations.

You can also have counts with an upper limit. For example, if you count the number of animals on a farm that are infected with a disease, out of a total of N animals. The maximum number of infected animals is then N . If you consider each individual animal as an independent trial and each animal has the same probability of being infected, then we are in the world of a binomial distribution.

But, what do you do with densities? Density is often defined as the numbers (which are counts!) per volume (or area, depth range, etc.). We will see in Chapter 9 that this can be modelled with the Poisson (or NB) distribution and an offset variable.

If the response variable is a continuous variable like weight of the animal, then the Normal distribution is your best option, but the gamma distribution may be an alternative choice.

The important thing to realise is that these distributions are for the response variable, not for explanatory variables. The choice of which distribution to use is an *a priori* choice. A list of all discussed distributions in this section is given in Table 8.1. If you are hesitating between two competing distributions, e.g. the Normal distribution and the gamma distribution, or the Poisson distribution and the negative binomial distribution, then you could plot the mean versus the variance of the response variable and see what type of mean–variance relationship you have and select a distribution function accordingly. In Chapter 9, we will see that the Poisson distribution is nested in the NB distribution, which opens the possibility for a likelihood ratio test.

Table 8.1 List of distributions for the response variable. Density means numbers per Area (or volume, range, etc), and in this case the offset option is needed in the Poisson or NB GLM

Distribution	Type of data	Mean – variance relationship
Normal	Continuous	Equation (8.2)
Poisson	Counts (integers) and density	Equation (8.4)
Negative binomial	Overdispersed counts and density	Equation (8.7)
Geometric	Overdispersed counts and density	Equation (8.8)
Gamma	Continuous	Equation (8.10)
Binomial	Proportional data	Equation (8.12)
Bernoulli	Presence absence data	Equation (8.12) with $N = 1$

8.8 Zero Truncated Distributions for Count Data

The discussion presented in this section applies to the Poisson, negative binomial, and the geometric distributions. All three distributions can be used for count data. Suppose we sample N sites, and at each site we count the number of birds, denoted by Y_i . The values that we can measure are 0, 1, 2, 3, ..., etc. For a given mean μ , the Poisson, negative binomial, and geometric distributions specify the probability of having a count of 0, 1, 2, etc. For example, if we use the Poisson distribution with $\mu = 3$, Fig. 8.2A shows that the probability of counting 0 animals is 13.5%. So, if you had a sample of size $N = 100$, you would expect to have a zero count approximately 14 times in your resulting data set. But what if you have a response variable that cannot take the value of 0? A typical example from the medical literature is the length of stay of a patient in a hospital. As soon as the patient enters the hospital, the length of stay is at least 1. In ecology, it is more difficult to envisage examples that structurally exclude zeros, but think of the number of plants in a transect and you know that it would be impossible to have transects with zero abundance due to the experimental design, the time that a whale stays at the surface before submerging (it has to breath) or the number of days per year with rain in Scotland. These are all variables that cannot have the value of 0. However, the Poisson, negative binomial, and geometric distributions do not exclude this value, and this can be a problem for small mean values μ .

The solution is to modify the distribution and exclude the possibility of a zero observation, and this is called a zero truncated distribution. We illustrate the process for a Poisson distribution, but the process is similar for the other two distributions. In fact, the same problem exists for continuous distributions. Think of the weight of an animal. The weight is always positive, and if the majority of the observations have small values, a Gaussian distribution may not be appropriate as it allows for negative values and realisations. In this chapter, we focus on discrete distributions (because we need them in Chapter 11), but the Tobit model can be used for the Gaussian distribution. Cameron and Trivedi (1998) is a good reference.

Recall that the Poisson distribution is given by

$$f(y_i; \mu) = \frac{\mu^{y_i} \times e^{-\mu}}{y_i!} \quad (8.15)$$

The probability that $y_i = 0$, is given by

$$f(0; \mu) = \frac{\mu^0 \times e^{-\mu}}{0!} = e^{-\mu}$$

The probability of not measuring a 0 is given by $1 - e^{-\mu}$. If we use $\mu = 2$, then the probability that $y_i = 0$, is 0.135 and the probability of not measuring a 0 is 0.864. In Fig. 8.7A, we have sketched the Poisson distribution with $\mu = 2$. In panel B, we put a cross through the line that represents the probability of sampling a 0 count. The cross is our pedagogical way of saying that we are changing the Poisson density and setting the probability that $y = 0$ equal to 0. However, this leaves us with the problem that by definition the sum of the probabilities of all outcome should be exactly 1. Removing the probability of $y = 0$ means that the remaining probabilities add up to 0.864. The solution is simple; divide the probability of each outcome larger than 0 by 0.864. The sum of all scaled probabilities will then add up to 1 again. We

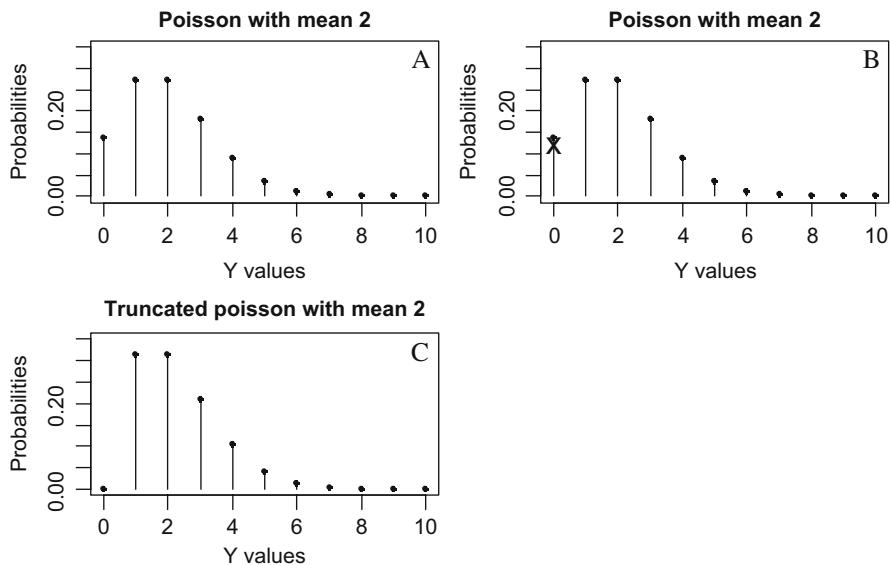


Fig. 8.7 A: Poisson distribution with $\mu = 2$. The sum of all probabilities is 1. B: The zero outcome is dropped from the possible range of outcomes, as indicated by a cross. The sum of all probabilities is equal to 0.864. C: Adjusted probabilities according to Equation (8.15). The vertical lines are slightly higher (because each probability was divided by 0.864), and the probability that $y_i = 0$ is zero. The sum of all scaled probabilities equals 1

therefore need to divide the Poisson probability function by the probability that we have a count larger than 0, and the new probability function is

$$f(y_i; \mu | y_i > 0) = \frac{\mu^{y_i} \times e^{-\mu}}{(1 - e^{-\mu}) \times y_i!} \quad (8.16)$$

The notation ‘ $| y_i > 0$ ’ is used to indicate that y_i is larger than 0. This is called the zero-truncated Poisson distribution. The same can be done for the negative binomial distribution.

The distribution function in Equation (8.15) will be used in GLMs and GAMs to model zero-truncated data. The underlying principle will also be applied in models that have too many zeros (zero inflated Poisson). For further details, see Chapter 11 or Hilbe (2007).

Chapter 9

GLM and GAM for Count Data

9.1 Introduction

A generalised linear model (GLM) or a generalised additive model (GAM) consists of three steps: (i) the distribution of the response variable, (ii) the specification of the systematic component in terms of explanatory variables, and (iii) the link between the mean of the response variable and the systematic part. In Chapter 8, we discussed several different distributions for the response variable: Normal, Poisson, negative binomial, geometric, gamma, Bernoulli, and binomial distributions. One of these distributions can be used for the first step mentioned above. In fact, later in Chapter 11, we see how you can also use a mixture of two distributions for the response variable; but in this chapter, we only work with one distribution at a time.

We spent a lot of time looking at distributions in Chapter 8 because our experience teaching environmental scientists show that in general they are less familiar with some of these distributions, especially the negative binomial. Before reading this chapter, you should ensure that you are familiar with the material described in Chapter 8.

In this chapter, we focus on count data and use the Poisson and negative binomial distributions. In the next chapter we concentrate on logistic regression using the binomial distribution. We also revisit count data in Chapter 11, where we look at data sets with lots of zeros or no zeros. Models for these types of data use a mixture of techniques discussed in this and the next chapter.

Good references on GLM include McCullagh and Nelder (1998), Dobson (2002), and Agresti (2002). It is possible to dedicate an entire book to Poisson or logistic regression (see for examples: Hosmer and Lemeshow, 2000; Collet, 2003). Fox (2002), Ruppert et al. (2003), Wood (2006), and Keele (2008) are excellent GAM references.

We start this chapter showing that the linear regression model is also a GLM. This is merely a pedagogical choice as it allows us to start with something familiar, and after all, the Gaussian linear regression can also be used for count data, even though it is not the best option. In Section 9.3, Poisson GLM is introduced using an artificial data set that we know the regression parameters for. It allows us to demonstrate what

the model is actually doing. In Section 9.4, we give the likelihood criterion and show how parameters can be estimated. In Sections 9.5, 9.6, 9.7, 9.8, and 9.9, we discuss Poisson GLM using a real data set and focus on overdispersion, model selection, and model validation. In Section 9.10, we present the negative binomial distribution and show how it can be used if there is overdispersion. Finally we look at GAM.

9.2 Gaussian Linear Regression as a GLM

A GLM consists of three steps:

1. An assumption on the distribution of the response variable Y_i . This also defines the mean and variance of Y_i .
2. Specification of the systematic part. This is a function of the explanatory variables.
3. The relationship between the mean value of Y_i and the systematic part. This is also called the link between the mean and the systematic part.

We discuss these three steps for the Gaussian linear regression model.

Step 1: In a Gaussian linear regression, we assume that the response variable Y_i is normally distributed with mean μ_i and variance σ^2 . The index i refers to a case or observation.

Step 2: In the second step, we specify the systematic part of the model. This means that we need to select the explanatory variables. Define the predictor function $\eta(X_{i1}, \dots, X_{iq})$ by:

$$\eta(X_{i1}, \dots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \dots + \beta_q \times X_{iq} \quad (9.1)$$

The systematic part is given by the predictor function $\eta(X_{i1}, \dots, X_{iq})$.

Step 3: In the third step, we need to specify the link between the expected value of Y_i (which is μ_i) and the predictor function $\eta(X_{i1}, \dots, X_{iq})$. We use the identity link, which means that $\mu_i = \eta(X_{i1}, \dots, X_{iq})$.

These three steps give the following GLM:

$$\begin{aligned} Y_i &\sim N(\mu_i, \sigma^2) \\ E(Y_i) &= \mu_i \quad \text{and} \quad \text{var}(Y_i) = \sigma^2 \\ \mu_i &= \eta(X_{i1}, \dots, X_{iq}) \end{aligned} \quad (9.2)$$

This model is also called a GLM with Gaussian distribution and identity link. Combining some of the elements in Equation (9.2) gives

$$E(Y_i) = \eta(X_{i1}, \dots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \dots + \beta_q \times X_{iq}$$

which is our familiar linear regression model from Chapter 2 and Appendix A. We can also write it as:

$$Y_i = \alpha + \beta_1 \times X_{i1} + \cdots + \beta_q \times X_{iq} + \varepsilon_i$$

where ε_i is normally and independently distributed with mean 0 and variance σ^2 . Examples and further details of the Gaussian GLM with identity link function are given in Appendix A. In principle, you can use the Gaussian distribution to analyse count data, but the residuals often show heterogeneity. Options to solve this are a data transformation or using generalised least squares as discussed in Chapter 4.

The formulation of a generalised additive model with a Gaussian distribution is similar to the linear regression model, except that in step 2 we use smoothers in the predictor function:

$$\eta(X_{i1}, \dots, X_{iq}) = \alpha + f_1(X_{i1}) + \cdots + f_q(X_{iq})$$

Obviously, we can also have a predictor function with smoothers and parametric or nominal variables.

9.3 Introducing Poisson GLM with an Artificial Example

In this section, we show the model formulation for a Poisson GLM, and we use an artificial example to demonstrate what the model is doing. We need the following three steps for a Poisson GLM:

1. Y_i is Poisson distributed with mean μ_i . By definition of this distribution, the variance of Y_i is also equal to μ_i .
2. The systematic part is given by $\eta(X_{i1}, \dots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \cdots + \beta_q \times X_{iq}$.
3. There is a logarithmic link between the mean of Y_i and the predictor function $\eta(X_{i1}, \dots, X_{iq})$. The logarithmic link (also called a log link) ensures that the fitted values are always non-negative.

As a result of these three steps, we get

$$\begin{aligned} Y_i &\sim P(\mu_i) \\ E(Y_i) &= \mu_i \quad \text{and} \quad \text{var}(Y_i) = \mu_i \\ \log(\mu_i) &= \eta(X_{i1}, \dots, X_{iq}) \quad \text{or} \quad \mu_i = e^{\eta(X_{i1}, \dots, X_{iq})} \end{aligned} \tag{9.3}$$

The Poisson GLM is particularly useful for count data as these tend to be heterogeneous and are always non-negative; both aspects are dealt with by the Poisson GLM.

In the remaining part of this section, we use an artificial data set to explain what a Poisson GLM model is doing. Creating artificial data is simple; choose some

arbitrary values for an intercept and slope, then choose arbitrary values for a covariate, and calculate some fitted values. We will start with the covariate X_i , which takes the values 0, 1, 2, 3, 4, 5, ..., 100. We arbitrarily choose an intercept of 0.01 with a slope of 0.03 and calculate the fitted values μ_i using the equation:

$$\mu_i = \exp(0.01 + 0.03 \times X_i)$$

The problem is that in reality, we never measure a count of $\exp(0.01)$ or $\exp(0.03 + 0.01 \times 1)$, because a count is an integer. We therefore sampled one value from a Poisson distribution with mean μ_i and the resulting value is Y_i . This process is repeated for each $i = 1, \dots, 101$. A scatterplot of X_i and Y_i is given in Fig. 9.1. We fitted a Poisson GLM on these data (on the X_i and Y_i), which gave an estimated intercept and slope, and these allowed us to draw the fitted line in Fig. 9.1. Note the line shows an exponential relationship. The scatter of points around the line in Fig. 9.1 gives an idea of how much variation to expect from a Poisson distribution with values between 0 and 30 (the range of the vertical axis).

The same exponential line is shown in Fig. 9.2, except that the third axis now shows the probability of other realisation. At several values along the covariate, where $X = 2, 15, 30, 50$, and 75, we calculated the fitted values (the Y values in Fig. 9.2), which are the means μ_i of the Poisson distributions in Fig. 9.2. Note how the shape of the Poisson density curves change from small skewed curves to wide symmetric curves.

In this section, we pretended that we knew the intercept α and slope β , which allowed us to calculate the fitted values μ_i used to generate the count data Y_i . Obviously, in real life, the situation is the opposite way around. In real life, we measure Y_i and X_i , and do not know α and β (and therefore also μ_i). Hence, we need a mechanism that estimates the values of α and β , and this is discussed in the next section.

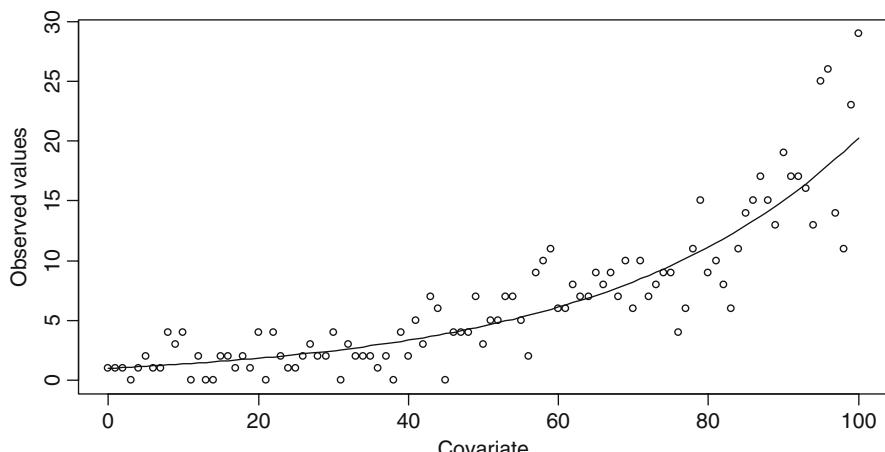


Fig. 9.1 Artificial data with a GLM Poisson model fitted. The fitted line is obtained from the GLM model, and X is the covariate with values from 0 to 100

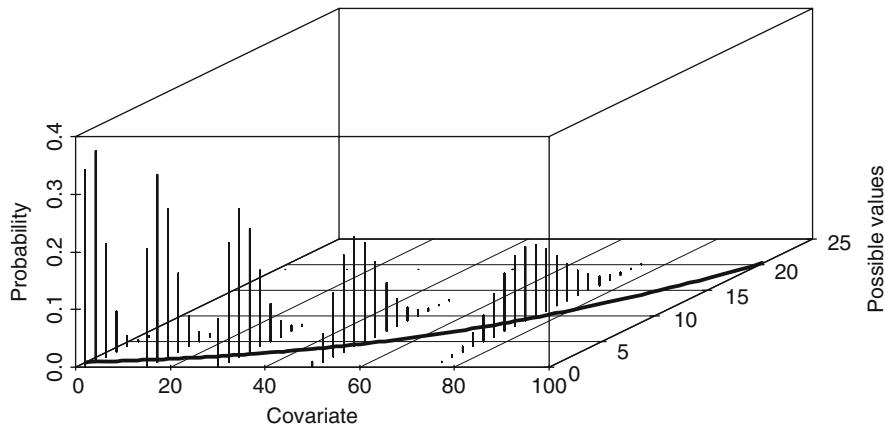


Fig. 9.2 Example of a Poisson GLM. The plane in the x - y axes shows the same exponential curve as in Fig. 9.1. The vertical lines along the third axis show Poisson probability curves at different values of the covariate: $X = 2, 15, 30, 50$, and 75 . The widths of the probability curves show the spread of the data. This is the same graph as Fig. 2.5, except that we use a Poisson GLM here

9.4 Likelihood Criterion

The Poisson distribution was discussed in Chapter 8. Recall that it is given by

$$f(y_i; \mu_i) = \frac{\mu_i^{y_i} \times e^{-\mu_i}}{y_i!} \quad y_i \geq 0, \quad y_i \text{ integer}$$

It gives the probability that a particular y_i value is observed for a given mean μ_i . Within the context of a GLM, we add an index i to μ , and μ_i is a function of the covariates:

$$\mu_i = e^{\alpha + \beta_1 X_{1i} + \dots + \beta_q X_{qi}}$$

The unknown parameters that we need to estimate are the intercept and slopes. In linear regression, we used ordinary least squares to minimise the residual sum of squares. Here, we use maximum likelihood estimation.

The principle of maximum likelihood estimation is that we specify a joint likelihood criterion L for all observed data y_1 to y_n , and we maximise this likelihood criterion as a function of the unknown regression parameters. Formulated differently, what are the values of the regression parameters such that the probability L of the observed data is the highest? The starting point is

$$L = \text{Probability}(Y_1 = y_1 \text{ and } Y_2 = y_2 \text{ and } \dots \text{ and } Y_n = y_n)$$

Because we assume independence of the observations, we can use the basic probability rule $P(A \text{ and } B) = P(A) \times P(B)$. As a result the likelihood function, L can be written as

$$L = \prod_i \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!}$$

The roman pillar symbol stands for multiplication, and the Poisson distribution function was used for the probability that Y_i is y_i . From this point onwards, it is merely a matter of mathematics; how can we maximise L as a function of the regression parameters? To simplify the maximisation process, we make the likelihood criterion L additive by working with the logarithm of the likelihood:

$$\begin{aligned}\log(L) &= \sum_i (\log(\mu^{y_i} \times e^{-\mu_i}) - \log(y_i!)) \\ &= \sum_i (\log(\mu^{y_i}) + \log(e^{-\mu_i}) - \log(y_i!)) \\ &= \sum_i (y_i \times \log(\mu_i) - \mu_i - \log(y_i!)) \\ &= \sum_i (y_i \times \mathbf{X}_i \times \boldsymbol{\beta} - e^{\mathbf{X}_i \times \boldsymbol{\beta}} - \log(y_i!))\end{aligned}\tag{9.4}$$

To speed up the numerical optimisation routines, we could drop the $\log(y_i!)$ term as it does not contain any regression parameters. You may remember from high school mathematics that to optimise a function, we need to obtain first-order derivatives, set them to 0 and solve the equations. The first-order derivatives are given by

$$\frac{\partial \log(L)}{\partial \boldsymbol{\beta}} = \sum_i (y_i \times \mathbf{X}_i - \mathbf{X}_i \times e^{\mathbf{X}_i \times \boldsymbol{\beta}}) = \sum_i \mathbf{X}_i \times (y_i - \mu_i)$$

Setting these to 0 gives

$$\sum_i \mathbf{X}_i \times (y_i - \mu_i) = \mathbf{0}\tag{9.5}$$

For the Gaussian linear regression model with an identity link, this gives a closed form solution. This means we get nice expressions for the unknown parameters that can easily be calculated. However, for most of the other distributions and link functions, this is not the case. Instead, we get a set of equations that have to be solved iteratively. A so-called iteratively reweighted least squares (IRWLS) algorithm is applied, and the numerical output of the GLM function in R has a sentence telling you how many iterations were carried out. To obtain standard errors for the parameters, we also need second-order derivatives of the log likelihood function, but we do not present them here.

If you open a book on GLM, it will be hard to find the likelihood equations for a Poisson GLM, as most books present these equations in terms of the general notation we used in Chapter 8. The advantage of this general notation is that, provided we use a canonical link (e.g. the log for a Poisson, or identity link for the Gaussian distribution), the internal mathematics of all GLMs can be written in the same way and with the same variable names that we used in Chapter 8. This makes it easy to program. However,

from a pedagogical point of view, we decided to focus first on the Poisson GLM, and then to mention the possibility of rewriting it in abstract, and general, mathematical notation. We refer the interested reader to McCullagh and Nelder (1989).

9.5 Introducing the Poisson GLM with a Real Example

9.5.1 Introduction

In Section 9.3, we arbitrarily chose a set of regression parameters and created artificial count data. It allowed us to explain the underlying concept of Poisson GLM and give an impression of how much variation can be expected in the data if they are from a Poisson distribution. In Section 9.4, we formulated the maximum likelihood criterion and presented the first-order derivatives. Luckily, other people have written software code that uses the log likelihood criterion and the equations for first-order derivatives to obtain parameter estimates. In this section, we show how to use the software and present a detailed example. Because we are now going to use a real example, all the misery will come at the same time.

The data used here (and in various other sections in this chapter) are fully analysed in Chapter 16 as a case study. It should be noted that a Poisson GLM is not the best tool to analyse these data, but it serves as a convenient example of how to progress through all steps of a GLM for count data.

The data set consists of roadkills of amphibian species at 52 sites along a road in Portugal. A scatterplot of the response variable roadkills against a possible explanatory variable ‘distance to the natural park’, denoted by D.PARK, is given in Fig. 9.3. The biological interpretation of ‘distance to the park’ is given in Chapter 16.

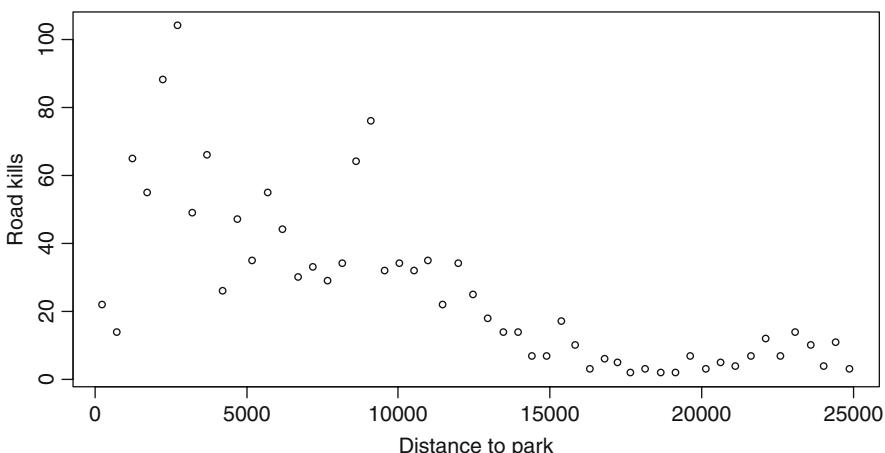


Fig. 9.3 Scatterplot of amphibian road kills versus distance (in metres) to a nearby Natural Park

The data are counts, and there seems to be a non-linear, perhaps exponential, relationship between roadkills and D.PARK. Also note that the variation is larger for larger values of roadkills. Taken together, this gives us all the ingredients for a Poisson GLM. Starting with only D.PARK as an explanatory variable, and ignoring the other 10 explanatory variables, is a pedagogical choice for presenting Poisson GLM in a textbook and is not a general recommendation for analysing these data. The following Poisson GLM was applied.

1. Y_i , the number of killed animals at site i , is Poisson distributed with mean μ_i .
2. The systematic part is given by $\eta(D.PARK_i) = \alpha + \beta \times D.PARK_i$.
3. There is a logarithm link between the mean of Y_i and the predictor function $\eta(D.PARK_i)$.

As a result of these three steps, we have

$$\begin{aligned} Y_i &\sim p(\mu_i) \\ E(Y_i) &= \mu_i \quad \text{and} \quad \text{var}(Y_i) = \mu_i \\ \log(\mu_i) &= \alpha + \beta \times D.PARK_i \quad \text{or} \quad \mu_i = e^{\alpha + \beta \times D.PARK_i} \end{aligned} \tag{9.6}$$

We now discuss how to fit this model in R.

9.5.2 R Code and Results

The following R code accesses the data, produces Fig. 9.3, applies the GLM, and presents the results.

```
> library(AED); data(RoadKills)
> RK <- RoadKills #Saves some space in the code
> plot(RK$D.PARK, RK$TOT.N, xlab = "Distance to park",
       ylab = "Road kills")
> M1 <- glm(TOT.N ~ D.PARK, family = poisson, data = RK)
> summary(M1)
```

The only new code here compared to linear regression (see Chapter 2 and Appendix A) is using the `glm` command instead of the `lm` command and the option `family = poisson`. Using `family = gaussian` applies linear regression, but we will not do that here (in fact, it is easier just to use the function `lm` for linear regression). The output of the `summary` command is slightly different from the `summary` output of an `lm` command and is given by:

```

Call:
glm(formula = TOT.N ~ D.PARK, family = poisson)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-8.1100 -1.6950 -0.4708  1.4206  7.3337 

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) 4.316e+00 4.322e-02 99.87 <2e-16 ***
D.PARK      -1.059e-04 4.387e-06 -24.13 <2e-16 ***
(Dispersion parameter for poisson family taken to be 1)

Null deviance: 1071.4 on 51 degrees of freedom
Residual deviance: 390.9 on 50 degrees of freedom
AIC: 634.29

Number of Fisher Scoring iterations: 4

```

The first two lines tell us which model has been fitted, which is handy if you save the output into a word processor document. Basic numerical information on the residuals is also provided, although in Section 9.8 we present more useful graphical tools that can be used for the model validation process. The estimated intercept and slope are 4.31 and -0.000106 , respectively. Keep in mind that distance to the park is expressed in metres. To avoid parameter estimates with lots of zeros, you could (and perhaps should) express it in kilometres, as it will save some ink when presenting the estimated slope on paper. We also get a z -statistic and corresponding p -value for testing the null hypothesis that the slope (and intercept) is equal to 0 and an AIC, which can be used for model selection. The z -statistic is used because we know the variance. In a Gaussian model, the variance is estimated as well, and therefore, a t -statistic is used.

9.5.3 Deviance

The null and residual deviances are new phrases, and these are sort of maximum likelihood equivalents of the total sum of squares and the residual sum of squares, respectively. For the Poisson GLM, the residual deviance is defined as twice the difference between the log likelihood of a model that provides a perfect fit (also called the saturated model) for the model under study:

$$\text{Residual deviance} = 2 \log(L(\mathbf{y}; \mathbf{y})) - 2 \log(L(\mathbf{y}; \boldsymbol{\mu})) = 2 \sum_i (y_i \log \frac{y_i}{\mu_i} - (y_i - \mu_i))$$

The notation \mathbf{y} refers to a vector of all observations y_1 to y_n , and the same holds for the mean μ . The null deviance is the residual deviance in the model that only contains an intercept. Hence, the null deviance corresponds to the worst possible model (only an intercept), the residual deviance of the model under study, and the deviance of the saturated model from the best possible fit.

We do not have an R^2 in GLM models, but the closest we can get is the explained deviance, which is calculated as

$$100 \times \frac{\text{null deviance} - \text{residual deviance}}{\text{null deviance}} = 100 \times \frac{1071.4 - 390.9}{1071.4} = 63.51\%$$

So the explanatory variable distance to the park explains 63.51% of the variation in road kills. Dobson (2002) called this proportional increase in explained deviance the pseudo R^2 .

The smaller the residual deviance, the better is the model. Some statistics programs also quote a p -value as it is supposedly Chi-square distributed with $n - p$ degrees of freedom, where p is the number of regression parameters in the model and n the number of observations. However, using the residual deviance as a goodness-of-fit measure is not without controversy; see McCullagh and Nelder (pg. 118–119, 1989). They argue that (at least for the binomial GLM) a large value of the residual deviance cannot always be seen as evidence of a poor fit.

The residual deviance is also sometimes called the deviance.

9.5.4 Sketching the Fitted Values

Before discussing how to assess the numerical output presented in Section 9.5.2, we will outline what the model is doing. But first we need to calculate the predicted values from the model and add these as a line in Fig. 9.3.

The function `predict` produces either predicted values on the scale of the predictor function or on the scale of the response variable. In the first case, we use the values $\eta(D.PARK_i) = 4.13 - 0.0000106 \times D.PARK_i$, and in the second case, $\exp(\eta(D.PARK_i)) = \exp(4.13 - 0.0000106 \times D.PARK_i)$. If we want to show how good (or bad) the model fits the observed data, we should use the predicted values on the scale of the response variable (after taking the exponential).

Drawing the line is now simply a matter of sticking in a couple of values for `D.PARK` and calculating the fitted values. Instead of doing this manually, we can do it with a few commands in R. The code uses the `plot` command for Fig. 9.3 and the `glm` command we have already run.

```
> G <- predict(M1, newdata = MyData, type = "link",
+               se = TRUE)
> F <- exp(G$fit)
> FSEUP <- exp(G$fit + 1.96 * G$se.fit)
> FSELLOW <- exp(G$fit - 1.96 * G$se.fit)
> lines(MyData$D.PARK, F, lty = 1)
> lines(MyData$D.PARK, FSEUP, lty = 2)
> lines(MyData$D.PARK, FSELLOW, lty = 2)
```

You will find similar (and more extensive code) in the so-called white book on the S language (on which R is based), written by Chambers and Hastie (1992). We first create a new data frame `MyData`. The variables inside this data frame must have exactly the same names as the explanatory variables in the `glm` command; in this case there is only `D.PARK`. In the data frame, you can specify new values for the explanatory variables. The `predict` command takes as arguments the object from the `glm` function (`M1`), the data frame with the new values of the explanatory variables, an argument `type` that tells the `predict` function at which level to predict (either the scale of the predictor function, or the response variables, and whether you want to have confidence intervals around the predicted line. We predicted at the level of the predictor function; so we get confidence bands that do not contain 0 and are asymmetric. Obviously, we have to do some basic maths ourselves, and the results are given in Fig. 9.4. Note the exponential shape of the curve and the increase in the width of the confidence bands for larger fitted values.

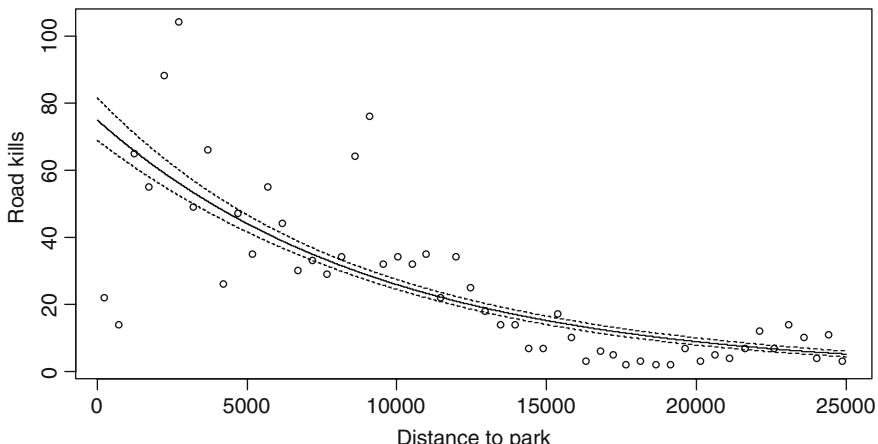


Fig. 9.4 Observed roadkills with a fitted Poisson GLM curve (*solid line*) and 95% confidence bands (*dotted lines*). Note the clear exponential shape of the curve. For smaller fitted values, there are groups of residuals above and below the fitted line. This is not good, and we need to deal with this in the model validation!

9.6 Model Selection in a GLM

9.6.1 Introduction

So far, we have only discussed the interpretation of the model in terms of an exponential curve fitted through a set of points; we now concentrate on things like model selection, hypothesis testing, and model validation. However, applying a model selection with only one explanatory variable is a bit unrealistic, so we now add a few more explanatory variables. The amphibian roadkills data set contains 17 explanatory variables. A list of these variables and abbreviations is given in Table 16.1. Some of the explanatory variables were square root transformed because of large values. Using variance inflation factors (Appendix A), a sub-selection of nine variables is made in Chapter 16 and we use the same sub-selection here. Note, this is still a relatively high number of explanatory variables for a data set with only 52 observations! A Poisson GLM for the roadkills data with nine variables is specified in a very similar way as in Equation (9.4), except that the systematic part now contains all nine explanatory variables (we have no biological reasons to believe there are interactions).

9.6.2 R Code and Output

The following R code implements the Poisson GLM with nine explanatory variables.

```
> RK$SQ.POLIC    <- sqrt(RK$POLIC)
> RK$SQ.WATRES  <- sqrt(RK$WAT.RES)
> RK$SQ.URBAN   <- sqrt(RK$URBAN)
> RK$SQ.OLIVE    <- sqrt(RK$OLIVE)
> RK$SQ.LROAD    <- sqrt(RK$L.P.ROAD)
> RK$SQ.SHRUB    <- sqrt(RK$SHRUB)
> RK$SQ.DWATCOUR <- sqrt(RK$D.WAT.COUR)
> M2 <- glm(TOT.N ~ OPEN.L + MONT.S + SQ.POLIC +
  D.PARK + SQ.SHRUB + SQ.WATRES + L.WAT.C +
  SQ.LROAD + SQ.DWATCOUR, family = poisson,
  data = RK)
> summary(M2)
```

The code is self-explanatory, and the relevant output of the `summary` command is given by

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.749e+00	1.567e-01	23.935	< 2e-16
OPEN.L	-3.025e-03	1.580e-03	-1.915	0.055531

```
MONT.S      8.697e-02  1.359e-02   6.398 1.57e-10
SQ.POLIC   -1.787e-01 4.676e-02  -3.822 0.000133
SQ.SHRUB   -6.112e-01 1.176e-01  -5.197 2.02e-07
SQ.WATRES  2.243e-01 7.050e-02   3.181 0.001468
L.WAT.C    3.355e-01 4.127e-02   8.128 4.36e-16
SQ.LPROAD  4.517e-01 1.348e-01   3.351 0.000804
SQ.DWATCOUR 7.355e-03 4.879e-03   1.508 0.131629
D.PARK     -1.301e-04 5.936e-06  -21.923 < 2e-16
```

Dispersion parameter for poisson family taken to be 1

Null deviance: 1071.44 on 51 degrees of freedom

Residual deviance: 270.23 on 42 degrees of freedom

AIC: 529.62

9.6.3 Options for Finding the Optimal Model

We want to know which explanatory variables are important, and because some terms are not significant, it is time for a model selection. The process is similar to the one used for linear regression (Appendix A). We can use either a selection criterion like the AIC or use a hypothesis testing approach.

Automatic forward, backward, and forward and backward selection can be applied with the command `step(M2)`. Results are not presented here, but a backward selection indicates that no term should be dropped.

For the hypothesis testing approach, we have three options:

1. Test the null hypothesis $H_0: \beta_i = 0$ using the z -statistic. This is the equivalent of the t -statistic in linear regression. This approach suggests to drop first `SQ.DWATCOUR` as it is the least significant term and then to refit the model and see whether there are still non-significant terms in the model.
2. Use the `drop1(M2, test = "Chi")` command, which drops one explanatory variable, in turn, and each time applies an analysis of deviance test. We explain this process below.
3. Use the `anova(M2)` command, which applies a series of analysis of deviance tests by removing each term sequential. We explain at the end of Subsection 9.6.5 how this process works.

Steps 2 and 3 are similar to the `anova` and `drop1` functions in linear regression, except that in linear regression we used an F test based on residual sum of squares of a full and a nested model. A nested model is defined as a model that is obtained from the full model by setting certain parameters equal to 0. We do not have residual sum of squares in Poisson GLM. Well, actually we do, but they are not used in these tests (residuals are discussed in Section 9.8). Instead, we use the residual deviance of two nested models.

9.6.4 The Drop1 Command

Suppose we have two models: model M_1 contains all nine explanatory variables, and in model M_2 we dropped the explanatory variable OPEN.L. So now the number of parameters for M_1 is $p_1 = 9$ and for M_2 is $p_2 = 8$. Obviously, the deviance of M_1 will always be equal or lower than the deviance of M_2 , simply because it has one extra parameter. The null hypothesis is that the regression parameter β for OPEN.L equals 0. Under the null hypothesis, both deviances are equal, and therefore, a large difference between the deviances is evidence against the null hypothesis.

Let D_1 and D_2 be the deviances of models M_1 and M_2 , respectively. The difference between D_2 and D_1 is asymptotically Chi-square distributed with $p_1 - p_2$ degrees of freedom. In formula

$$D_2 - D_1 \sim X^2_{p_1 - p_2} \quad (9.7)$$

The drop1(M2, test = "Chi") command drops each explanatory variable in turn, and each time it calculates the difference in Equation (9.7) and compares the difference to a Chi-square distribution; see the following output.

Single term deletions

```
Model: TOT.N ~ OPEN.L + MONT.S + SQ.POLIC + SQ.SHRUB +
       SQ.WATRES + L.WAT.C + SQ.LPROAD + SQ.DWATCOUR +
       D.PARK
```

	DF	Deviance	AIC	LRT	Pr(Chi)
<none>		270.23	529.62		
OPEN.L	1	273.93	531.32	3.69	0.0546474
MONT.S	1	306.89	564.28	36.66	1.410e-09
SQ.POLIC	1	285.53	542.92	15.30	9.181e-05
SQ.SHRUB	1	298.31	555.70	28.08	1.167e-07
SQ.WATRES	1	280.02	537.41	9.79	0.0017539
L.WAT.C	1	335.47	592.86	65.23	6.648e-16
SQ.LPROAD	1	281.25	538.64	11.02	0.0009009
SQ.DWATCOUR	1	272.50	529.89	2.27	0.1319862
D.PARK	1	838.09	1095.48	567.85	< 2.2e-16

The model containing all explanatory variables has a deviance of 270.3. If we drop OPEN.L, the deviance is 273.93: a difference of 3.69. The statistic $X^2 = 3.69$ follows (approximately) a Chi-square distribution with 1 degree of freedom, which gives a p -value of 0.054. This can be double checked with the R command: `1 - pchisq(3.69, 1)`.

Note that the analysis of deviance does not give exactly the same p -value as the z -statistic. This is because both tests are approximate. If in doubt, use the analysis of deviance test. The advantage of using the analysis of deviance test is that it also gives a p -value for a nominal variable.

9.6.5 Two Ways of Using the Anova Command

The same *p*-value for OPEN.L can be obtained by fitting a model with all explanatory variables (which is M2), a model without OPEN.L, and then use the *anova* command to compare the two models with an analysis of deviance. This is done with the following R code:

```
> M3 <- glm(TOT.N ~ MONT.S + SQ.POLIC + D.PARK +
           SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LROAD +
           SQ.DWATCOUR, family = poisson, data = RK)
> anova(M2, M3, test = "Chi")
```

The output is given by

Analysis of Deviance Table					
	Resid.	Df	Resid.	Dev Df	Deviance P(> Chi)
1		42		270.232	
2		43		273.925 -1	-3.693 0.055

If you use this output in a paper or report, then you should write that the difference in deviance is 3.69 and approximately follows a Chi-square distribution with 1 degree of freedom. We have seen papers where a Chi-square distribution with 43 degrees of freedom was quoted from the output above, which is clearly wrong!

Be careful when using the command *anova*(M2); it applies an analysis of deviance test, but now the terms are removed sequentially and the order depends on the order they were typed. This is useful if all explanatory variables are independent or if the last term is an interaction.

9.6.6 Results

Using the *drop1* function, we decided to remove the variable SQ.DWATCOUR. Refitting the model resulted in all explanatory variables being significant at the 5% level. This suggests that we are finished with the model selection process, and can proceed to the model validation process. However, things are never that easy. The results of the *summary* command presented above had a small sentence that said: ‘overdispersion parameter for Poisson family taken to be 1’. This does not mean that the overdispersion really is 1; it just says it was taken as 1. We promised more misery, and overdispersion is the next stage.

In the next section, we show that all the results presented in this section can be put in the bin, because of overdispersion. If you analyse your own data, you should always first check for overdispersion, before doing any model selection or interpretation of the results. The reason why we did not start by looking at overdispersion was because we wanted to make sure you could read the output and judge whether there is overdispersion. For your own data, you should always start by checking for overdispersion and act accordingly. This is discussed in the next section.

9.7 Overdispersion

9.7.1 Introduction

Overdispersion means the variance is larger than the mean. How do you know your model is overdispersed? There are two options. The first is based on the χ^2 approximation of the residual deviance. If there is overdispersion, then D/ϕ is Chi-square distributed with $n - p$ degrees of freedom, and this leads to the following estimator for ϕ :

$$\hat{\phi} = \frac{D}{n - p} \quad (9.8)$$

In this case, it is $270.23/42 = 6.43$. If this ratio is about 1, then you can safely assume there is no overdispersion and proceed to the model validation process. In this case the ratio is larger than 1 and provides evidence for overdispersion. Note this only identifies overdispersion. The model (and software) does not take into account of the overdispersion and we therefore cannot present the results as they are. Also note that the use of the estimator in Equation (9.8) is not without criticism.

The second option is to use a different estimator based on the so-called Pearson residuals and let the software make the corrections required for overdispersion (i.e. correct the standard errors and tell us the magnitude of the overdispersion based on the estimator using the Pearson residuals). But we have not yet discussed residuals for Poisson GLMs yet. This will be done in Section 9.8.

9.7.2 Causes and Solutions for Overdispersion

Hilbe (2007) discriminates between apparent and real overdispersion. Apparent overdispersion is due to missing covariates or interactions, outliers in the response variable, non-linear effects of covariates entered as linear terms in the systematic part of the model, and choice of the wrong link function. These are mainly model misspecifications. There are a couple of interesting examples in Hilbe (pg. 52–61, 2007). For example, he simulates a Poisson variable using five explanatory variables X_1 to X_5 , applies a Poisson model using only explanatory variables X_2 to X_4 , and shows how this causes overdispersion. Similar examples are given for the effects of outliers and using the wrong link function.

Real overdispersion exists when we cannot identify any of the previous mentioned causes. This can be because the variation in the data really is larger than the mean. Or there may be many zeros (which may, or may not, cause overdispersion), clustering of observations, or correlation between observations.

If adding covariates and interactions does not help, there is a quick-fix that can be tried before considering more complicated methods like the negative binomial GLM.

9.7.3 Quick Fix: Dealing with Overdispersion in a Poisson GLM

We can deal with overdispersion in the GLM by using a quasi-Poisson GLM, which consists of the following steps:

1. The mean and variance of Y_i are given by $E(Y_i) = \mu_i$ and $\text{var}(Y_i) = \phi \times \mu_i$.
2. The systematic part is given by $\eta(X_{i1}, \dots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \dots + \beta_q \times X_{iq}$.
3. There is a logarithmic link between the mean of Y_i and the predictor function $\eta(X_{i1}, \dots, X_{iq})$.

The difference between the Poisson GLM and the Poisson GLM with overdispersion is that we no longer explicitly specify a Poisson distribution, but only a relationship between the mean and variance of Y_i .

Although we do not specify a Poisson distribution, we still use the same type of model structure in terms of the link function and predictor function. If the dispersion parameter $\phi = 1$, we get the same results (in terms of estimated parameters and standard errors) as the Poisson GLM.

If $\phi > 1$, we talk about overdispersion, and if $\phi < 1$, we have underdispersion. The latter means that the variance of the response variable is smaller than you would expect from a Poisson distribution. Reasons for underdispersion are the model is fitting a couple of outliers rather too well or there are too many explanatory variables or interactions in the model (overfitting). If this is not the case, then the consensus is not to correct for underdispersion. Models that take underdispersion into account are discussed in Chapter 7 of Hilbe (2007).

If $\phi > 1$, we need to correct for the overdispersion, which basically means refitting the model, estimating the parameter ϕ , and ‘making some corrections’. Before addressing these corrections, we look at the following questions first:

1. How do we estimate the dispersion parameter ϕ ?
2. How much larger than 1 should it be before we need to make a correction?
3. What is the effect of introducing a dispersion parameter ϕ ?
4. At which point do we decide to do take an alternative approach?

The first question can only be answered in detail towards the end of Section 9.8 because the estimation of ϕ is based on residuals and we have not yet defined residuals for a GLM. The second question can only be answered in light of the third question. The price we pay for introducing a dispersion parameter ϕ , is that the standard errors of the parameters are multiplied with the square root of ϕ . For example, if ϕ is equal to 9, then all standard errors are multiplied by 3, and the parameters become less significant. If the parameters of a Poisson GLM are highly significant, then a small correction of the standard errors due to overdispersion, say $\phi = 1.5$, is not going to make any differences in the biological conclusions. But if you have a parameter with a p -value of 0.03, then multiplying the standard error with the square root of 1.5 may change the p -value in something that is no longer significant at the 5% level. So, it all depends: In general a ϕ larger than 1.5 means

that some action needs to be taken to correct it. Various tests for overdispersion are discussed in Hilbe (2007). For the fourth question, if ϕ is larger than 15 or 20, then you also need to consider other methods (e.g. the negative binomial GLM or zero-inflated models), see the negative binomial model in Section 9.10 and the models for zero-inflated data in Chapter 11.

9.7.4 R Code and Numerical Output

In R, the following command is required for this quick fix approach to correct for overdispersion.

```
> M4 <- glm(TOT.N ~ OPEN.L + MONT.S + SQ.POLIC +
  SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD +
  SQ.DWATCOUR + D.PARK,
  family = quasipoisson, data = RK)
```

You can see the only difference is specifying the `family` option as `quasipoisson` instead of `poisson`. This gives the impression that there is a quasi-Poisson distribution, but there is no such thing! All we do here is specify the mean and variance relationship and an exponential link between the expected values and explanatory variables. It is a software issue to call this ‘quasipoisson’. Do not write in your report or paper that you used a quasi-Poisson distribution. Just say that you did a Poisson GLM, detected overdispersion, and corrected the standard errors using a quasi-GLM model where the variance is given by $\phi \times \mu$, where μ is the mean and ϕ the dispersion parameter. To get the numerical output for this model, use `summary(M4)`, which gives

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.749e+00	3.814e-01	9.830	1.86e-12
OPEN.L	-3.025e-03	3.847e-03	-0.786	0.43604
MONT.S	8.697e-02	3.309e-02	2.628	0.01194
SQ.POLIC	-1.787e-01	1.139e-01	-1.570	0.12400
SQ.SHRUB	-6.112e-01	2.863e-01	-2.135	0.03867
SQ.WATRES	2.243e-01	1.717e-01	1.306	0.19851
L.WAT.C	3.355e-01	1.005e-01	3.338	0.00177
SQ.LPROAD	4.517e-01	3.282e-01	1.376	0.17597
SQ.DWATCOUR	7.355e-03	1.188e-02	0.619	0.53910
D.PARK	-1.301e-04	1.445e-05	-9.004	2.33e-11

Dispersion parameter for quasipoisson family taken to
be 5.928003

Null deviance: 1071.44 on 51 degrees of freedom

Residual deviance: 270.23 on 42 degrees of freedom

AIC: NA

Note that the ratio of the residual deviance and the degrees of freedom is still larger than 1, but that is no longer a problem as we now allow for overdispersion. The dispersion parameter ϕ is estimated as 5.93. This means that all standard errors have been multiplied by 2.43 (the square root of 5.93), and as a result, most parameters are no longer significant! We can move onto model selection.

9.7.5 Model Selection in Quasi-Poisson

The model selection process in quasi-Poisson GLMs is similar to Poisson GLMs; however, there are small, but important differences. First of all, in quasi-Poisson models the AIC is not defined. Hence, there is no automatic backward or forward selection with the `step` function! The hypothesis testing approach is also slightly different. The analysis of deviance approach to compare two nested models M_1 (full model) and M_2 (nested model) uses a different test statistic:

$$\frac{D_2 - D_1}{\phi(p_1 - p_2)} \sim F_{p_1 - p_2, n - p_1} \quad (9.9)$$

where ϕ is the overdispersion parameter, and $p_1 + 1$ and $p_2 + 1$ are the number of regression parameters in models M_1 and M_2 , respectively. The ‘+1’ is for the intercept. Under the null-hypothesis, the regression parameters of the omitted explanatory variables are equal to zero, and the F -ratio follows an F -distribution with $p_1 - p_2$ and $n - p_1$ degrees of freedom (n is the number of observations).

Using the command `drop1(m4, test = "F")` gives us the equivalent of the `drop1` function for the Poisson GLM; one term is dropped in turn. The output is as follows.

Single term deletions					
	Df	Deviance	F value	Pr(F)	
<none>		270.23			
OPEN.L	1	273.93	0.5739	0.452926	
MONT.S	1	306.89	5.6970	0.021574	
SQ.POLIC	1	285.53	2.3776	0.130585	
SQ.SHRUB	1	298.31	4.3635	0.042814	
SQ.WATRES	1	280.02	1.5217	0.224221	
L.WAT.C	1	335.47	10.1389	0.002735	
SQ.LPROAD	1	281.25	1.7129	0.197727	
SQ.DWATCOUR	1	272.50	0.3526	0.555802	
D.PARK	1	838.09	88.2569	7e-12	

These results suggest dropping `SQ.DWATCOUR` from the model and then refitting the model with the remaining terms to see if there are still any non-significant

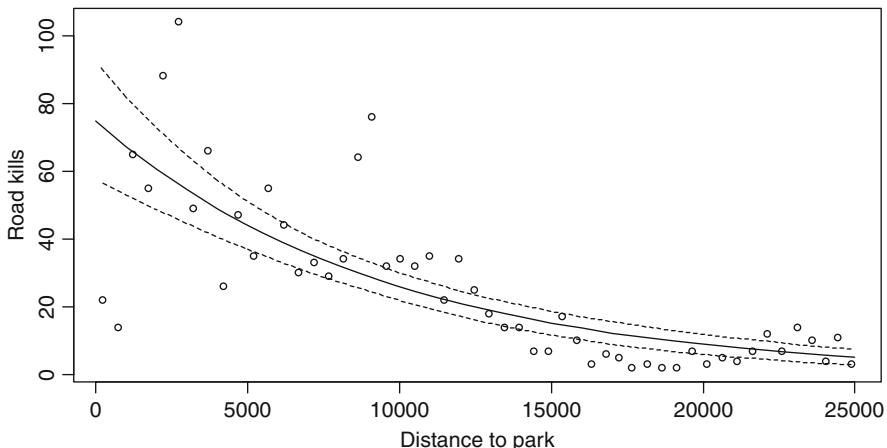


Fig. 9.5 Fitted line of the optimal quasi-Poisson model using only D.PARK as the explanatory variables. R code to make this graph is given on the book's website

terms. After doing this, some terms are still non-significant so the process has to be repeated. The variables were dropped in the following order: OPEN.L, SQ.WATRES, SQ.LPROAD, SQ.SHRUB, SQ.POLIC, MONT.S, and L.WAT.C. Finally, we ended up with a model that only contained D.PARK. So, ignoring overdispersion can result in a completely different biological conclusion!

We finally present the numerical output of the quasi-Poisson model that uses only D.PARK. Its estimated parameters, standard errors, etc. are given below and the fitted line is presented in Fig. 9.5. Note that the confidence intervals around the line are now larger than before due to the overdispersion correction.

```

Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.316e+00 1.194e-01 36.156 < 2e-16
D.PARK      -1.058e-04 1.212e-05 -8.735 1.24e-11

Dispersion parameter for quasipoisson family taken to
be 7.630148

Null deviance: 1071.4 on 51 degrees of freedom
Residual deviance: 390.9 on 50 degrees of freedom

```

9.8 Model Validation in a Poisson GLM

Just as in linear regression, we have to apply a model validation after we have decided on the optimal GLM, and the residuals are an important tool for this. Earlier in linear regression and additive modelling, these were defined as

Linear regression : $\hat{\varepsilon}_i = y_i - \hat{\mu}_i = y_i - \hat{\alpha} - \hat{\beta}_1 \times X_{i1} - \cdots - \hat{\beta}_q \times X_{iq}$

Additive modelling : $\hat{\varepsilon}_i = y_i - \hat{\mu}_i = y_i - \hat{\alpha} - \hat{f}_1(X_{i1}) - \cdots - \hat{f}_q(X_{iq})$

We used the notation $\hat{\cdot}$ to indicate that we are working with estimated values, parameters, or smoothing functions. To save space, we focus on the GLM, but the approach is identical for the GAM.

The question is as follows: What are residuals in a GLM? An obvious starting point would be to define residuals in exactly the same way as we do for linear regression using $y_i - \mu_i$, which is the vertical distance between an observation and the solid line in Fig. 9.5. The next question is whether a large residual at $D.PARK = 1000$ m is any worse than a large residual at $D.PARK = 20000$ m? The answer is not as easy as it may look, and we discuss this next!

9.8.1 Pearson Residuals

As for larger fitted values (left part of the fitted line) with Poisson distributions, we can allow for more variation around the line than with other distributions. Therefore, while we still want to see small residuals $y_i - \mu_i$ for small values of μ_i , residuals are allowed to be larger for larger μ_i . That makes a plot of the residuals $y_i - \mu_i$ versus fitted values μ_i , one of our prime graphs in Chapters 2 and 4, not particularly useful here.

In Chapter 4, we had a similar problem and our solution was to divide the residuals $y_i - \mu_i$ by the square root of the variance of Y_i , also called the normalised residuals. Here, we can do the same and call them the Pearson residuals.

$$\hat{\varepsilon}_i^P = \frac{y_i - \hat{\mu}_i}{\sqrt{\text{var}(Y_i)}} = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i}} \quad (9.10)$$

For this, each residual is divided by the square root of the variance. The name ‘Pearson’ (for a Poisson GLM) is because squaring and summing all the Pearson residuals gives you the familiar Pearson Chi-square goodness of fit criteria.

When we use an overdispersion parameter ϕ , the variance is adjusted with this parameter, and we divide the residuals $y_i - \mu_i$ by the square root of $\phi\mu_i$.

It is also possible to define standardised Pearson residuals by dividing the Pearson residuals by the square root of $1 - h_i$, where h_i is the leverage of observation i ; see also Appendix A.

9.8.2 Deviance Residuals

Recall that the residual deviance is the GLM equivalent of the residual sum of squares; the smaller the better. It would be nice to know the contribution of each observation (case) to the residual deviance. Perhaps some observations are not fitted well by the model, and this can be detected by looking at the deviance residuals. They are defined by

$$\hat{\varepsilon}_i^D = \text{sign}(y_i - \mu_i)\sqrt{d_i} \quad (9.11)$$

The notation ‘sign’ stands for sign and has the value 1 if y_i is larger than μ_i , and -1 if y_i is smaller than μ_i . The quantity d_i is the contribution of the i th observation to the deviance. The d_i was formulated in Section 9.5.3. The sum of squares of the deviance residuals d_i equals the residual deviance D .

9.8.3 Which One to Use?

So, we have three types of residuals in a GLM: (i) the ordinary residuals $y_i - \mu_i$, also called the response residuals, (ii) the Pearson residuals, and (iii) deviance residuals. In fact, there are more types of residuals (e.g. working residuals and Anscombe residuals, see McCullagh and Nelder (1989)), but these are the most popular ones for the purpose of model validation. Which one should we use?

By default, R uses the deviance residuals, and for most data sets used in this book, there is not much difference between using Pearson or deviance residuals for a Poisson GLM. This may not, however, be the case for data sets with lots of zeros (small variance) or for Binomial GLMs. McCullagh and Nelder (p. 398, 1989) recommend using the deviance residuals for model checking as these have distributional properties that are closer to the residuals from a Gaussian linear regression model than the alternatives; use Pierce and Schafer (1986) for a justification.

However, it should be noted that we are not looking for normality from the Pearson or deviance residuals. It is all about lack of fit and looking for patterns in the deviance or Pearson residuals.

9.8.4 What to Plot?

We need to take the residuals of choice (e.g. deviance) and plot them against (i) the fitted values, (ii) each explanatory variable in the model, (iii) each explanatory variable not in the model (the ones not used in the model, or the ones dropped during the model selection procedure), (iv) against time, and (v) against spatial coordinates, if relevant. We do not want to see any patterns in these graphs. If we do, then there is something wrong, and we need to work out what it is.

If there are patterns in the graph with residuals against omitted explanatory variables, then the solution is simple; include them in the model. If there are patterns in the graph showing residuals against each explanatory variable used in the model, then either include quadratic terms, use GAM, or conclude that there is violation of independence. If you plot the residuals against time or spatial coordinates, and there are patterns, conclude you are violating the assumption of independence. Patterns in spread (detected by plotting residuals against fitted values) may indicate overdispersion or use of the wrong mean-variance relationship (e.g. wrong choice of distribution).

Violation of independence nearly always means that an important covariate was excluded from the model. If you did not measure it, then if possible, go back into the field and measure it now. That is assuming you have any idea of what the

missing covariate might be! If this is not an available solution, then curse yourself for a poor experimental design and hope that applying a generalised linear mixed model or generalised estimation equation (GEE) will bale you out. See Chapters 12 and 13.

9.9 Illustration of Model Validation in Quasi-Poisson GLM

To explain model validation, we use the optimal quasi-Poisson GLM for the amphibian roadkills data. Recall from Section 9.7.5 that there was an overdispersion of 7.63 and that the only significant explanatory variable was D.PARK. Figure 9.6 shows the standard output from a `plot` command, and Fig. 9.7 contains the response residuals, Pearson residuals, scaled Pearson residuals (we divided the Pearson residuals by the square root of the overdispersion parameter), and the deviance residuals. Both figures indicate that there is a clear pattern in the residuals. Note that it is hard to detect any differences between Pearson and deviance residuals. Some additional exploration into the residuals against other explanatory variables and spatial locations is done in Chapter 16.

As in linear regression, we can also use leverage and the Cook distance statistic. There are no influential observations.

The following R code was used to produce Figs. 9.6 and 9.7.

```
> M5 <- glm(TOT.N ~ D.PARK, family = quasipoisson, data = RK)
> EP <- resid(M5, type = "pearson")
> ED <- resid(M5, type = "deviance")
> mu <- predict(M5, type = "response")
> E <- RK$TOT.N - mu
> EP2 <- E / sqrt(7.630148 * mu)
> op <- par(mfrow = c(2, 2))
> plot(x = mu, y = E, main = "Response residuals")
> plot(x = mu, y = EP, main = "Pearson residuals")
> plot(x = mu, y = EP2,
       main = "Pearson residuals scaled")
> plot(x = mu, y = ED, main = "Deviance residuals")
> par(op)
```

The first line re-applies the quasi-Poisson model, even though we could have omitted it as we had already applied it in the previous subsection. EP and ED are the Pearson and deviance residuals, respectively. Unfortunately, the function `resid` ignores the overdispersion; so we need to manually divide the Pearson residuals by the square root of 7.63 or calculate these residuals from scratch (as we did here). The rest of the code plots the residuals and should be self explanatory.

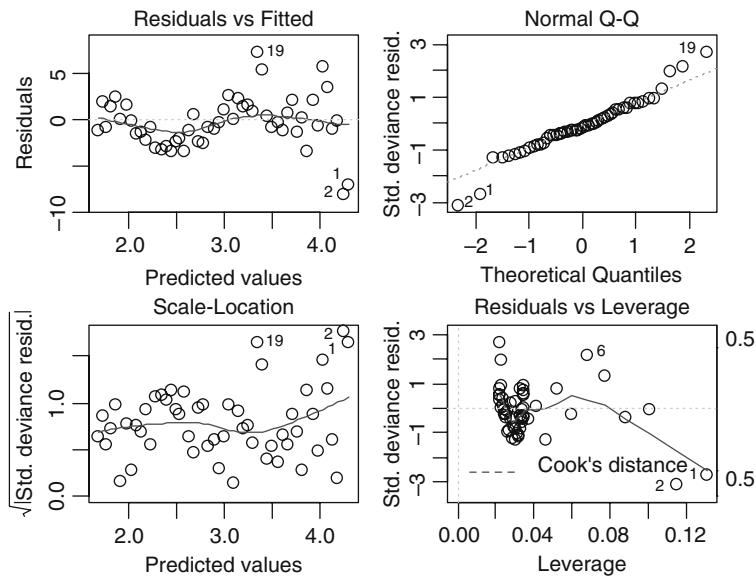


Fig. 9.6 Standard output from a GLM function applied on the amphibian roadkills data obtained by the `plot` command

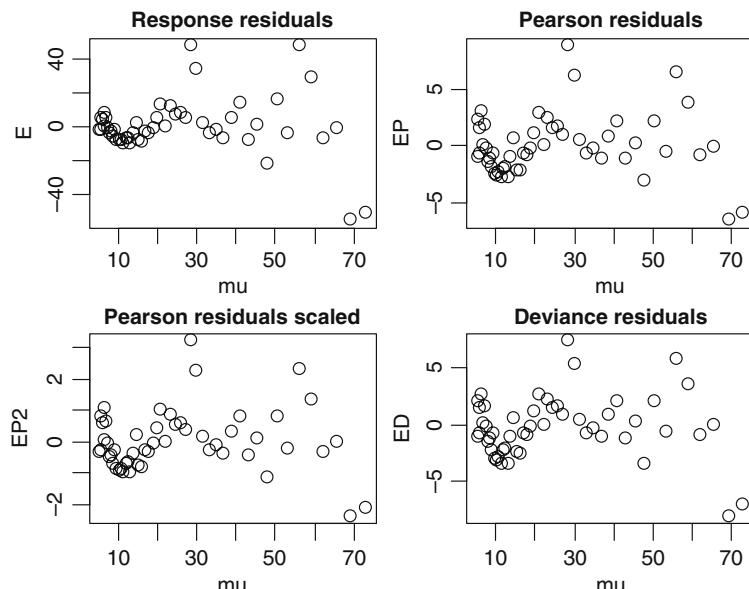


Fig. 9.7 Response residuals (observed minus fitted values, also called ordinary residuals), Pearson residuals, scaled Pearson residuals (the overdispersion is taken into account) and the deviance residuals for the optimal quasi-Poisson model applied on the amphibian roadkills data

The last thing we explain is how the overdispersion parameter ϕ in a Poisson GLM is estimated by R. It takes the Pearson residuals, squares them, adds them all up, and then divides the sum by $n - p$, where n is the number of observations and p the number of regression parameters (slopes) in the model. Check it with the R command `sum(EP ^2) / (52 - 1)`.

9.10 Negative Binomial GLM

9.10.1 Introduction

In the previous sections of this chapter, we applied Poisson GLM on the amphibian roadkills data set and found that there is an overdispersion of 7.63. Consequently, all standard errors were corrected by multiplying them with the square root of 7.63 when we applied the quasi-Poisson model. An alternative approach is to apply the negative binomial model. In Chapter 16, a negative binomial GAM is applied on the amphibian roadkills data, but for illustration purposes we apply the negative binomial GLM here.

Books that contain a chapter on the negative binomial GLM are for example Venables and Ripley (2002), Agresti (2002), or Gelman and Hill (2007). A book dedicated to negative binomial regression is Hilbe (2007). If you are going to apply the negative binomial GLM, then this book is a ‘must read’. It even discusses negative binomial GLMM models. Stata, rather than R, is used for this book, but this does not dominate the text.

Just as for Gaussian and Poisson GLMs, we specify the model with three steps. The NB GLM is given by

1. Y_i is negative binomial distributed with mean μ_i and parameter k (see also Chapter 8). By definition, the variance of Y_i is also equal to μ_i and its variance is $\mu_i + \mu_i^2 / k$.
2. The systematic part is given by $\eta(X_{i1}, \dots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \dots + \beta_q \times X_{iq}$.
3. There is a logarithm link between the mean of Y_i and the predictor function $\eta(X_{i1}, \dots, X_{iq})$. The logarithmic link (also called log link) ensures that the fitted values are always non-negative.

As a result of these three steps, we have

$$\begin{aligned} Y_i &\sim NB(\mu_i, k) \\ E(Y_i) &= \mu_i \quad \text{and} \quad \text{var}(Y_i) = \mu_i + \frac{\mu_i^2}{k} \\ \log(\mu_i) &= \eta(X_{i1}, \dots, X_{iq}) \quad \text{or} \quad \mu_i = e^{\eta(X_{i1}, \dots, X_{iq})} \end{aligned} \tag{9.12}$$

To estimate the regression parameters, we need to specify the likelihood criterion, and obtain the first-order and second-order derivatives. The process is the same as

for the Poisson GLM in Section 9.4. To avoid repetition, we only show how the log likelihood criterion is derived.

Recall from Chapter 8 that the negative binomial probability function is given by

$$f(y_i; k, \mu_i) = \frac{\Gamma(y_i + k)}{\Gamma(k) \times \Gamma(y_i + 1)} \times \left(\frac{k}{\mu_i + k} \right)^k \times \left(1 - \frac{k}{\mu_i + k} \right)^{y_i} \quad (9.13)$$

These probability functions are then used in the log likelihood criterion:

$$\log(L) = \sum_i \log(f(y_i; k, \mu_i)) \quad (9.14)$$

It is now a matter of substituting Equation (9.13) into the log likelihood function in (9.14), and using high school mathematics to simplify things. There is some contradiction in the literature regarding how much you should simplify this equation. For example, Equation (5.30) in Hilbe (2007) looks very different from the one we have here, but it is exactly the same thing, just written down differently. If you start inspecting these equations, do not panic if you find differences; some textbooks have small mistakes! Keeping it simple gives us

$$\begin{aligned} \log(L) &= \sum_i \log(f(y_i; k, \mu_i)) \\ &= \sum_i (k \times \log \left(\frac{k}{\mu_i + k} \right) + y_i \times \log \left(\frac{\mu_i}{\mu_i + k} \right) + \log(\Gamma(y_i + k)) \\ &\quad - \log(\Gamma(k)) - \log(\Gamma(y_i + 1))) \end{aligned} \quad (9.15)$$

This can be further simplified. It is also possible to express the NB probability function in Equation (9.13) as an exponential function. The advantage of this is that the whole model can be written in the same notation as the other GLMs; see also Section 13.2.2 in Hardin and Hilbe (2007).

The function `glm.nb` from the MASS package can be used to apply the negative binomial GLM in R. We start with all 11 explanatory variables again.

```
> library(MASS)
> M6 <- glm.nb(TOT.N ~ OPEN.L + MONT.S + SQ.POLIC +
+ SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD +
+ SQ.DWATCOUR + D.PARK, link = "log", data = RK)
```

You can choose from the logarithmic, identity, and square root link function, and an example with the identity link can be found in Agresti (2002). Here, we use the logarithmic link (which is also the default link in the function `glm.nb`, but not the canonical link function); so we can compare the results with those from the Poisson GLM. The command `summary(M6, cor = FALSE)` gives the relevant numerical output.

```

              Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.951e+00 4.145e-01 9.532 <2e-16
OPEN.L      -9.419e-03 3.245e-03 -2.903 0.0037
MONT.S      5.846e-02 3.481e-02 1.679 0.0931
SQ.POLIC   -4.618e-02 1.298e-01 -0.356 0.7221
SQ.SHRUB   -3.881e-01 2.883e-01 -1.346 0.1784
SQ.WATRES   1.631e-01 1.675e-01 0.974 0.3301
L.WAT.C     2.076e-01 9.636e-02 2.154 0.0312
SQ.LPROAD   5.944e-01 3.214e-01 1.850 0.0644
SQ.DWATCOUR -1.489e-05 1.139e-02 -0.001 0.9990
D.PARK      -1.235e-04 1.292e-05 -9.557 <2e-16

Dispersion parameter for Negative Binomial(5.5178)
family taken to be 1

Null deviance: 213.674 on 51 degrees of freedom
Residual deviance: 51.803 on 42 degrees of freedom
AIC: 390.11
Theta: 5.52
Std. Err.: 1.41
2 x log-likelihood: -368.107

```

The output is similar to the Poisson GLM output, except we also get a parameter theta, which is the k in the negative binomial variance function. We also get its standard error, but care is needed with its use as the interval is not symmetric and we are testing on the boundary. Note that as half of the regression parameters are not significant at the 5% level, a model selection is required.

The available tools for a model selection are similar to those we have seen in the previous section: hypothesis testing and using a model selection tool like the AIC. For hypothesis testing, we can use

1. The z -statistic (table above).
2. Analysis of deviance tables obtained by the `anova(M6, test = "Chi")` command (this is doing sequential testing).
3. Drop each term in turn and compare the full model with a nested model using the `drop1(M6, test = "Chi")` command.
4. Manually specifying a nested model, call it for example `M7`, and use the command `anova(M6, M7, test = "Chi")`.

An automatic backward (or forward) selection procedure based on the AIC can be applied by the command `step(M6)` or `stepAIC(M6)`. The latter option is the main advantage over quasi-Poisson, where we do not have a likelihood function and therefore cannot use AIC and automatic selection procedures.

A negative binomial model can also be overdispersed, and the approach described earlier of using the ratio of the residual deviance and the degrees of freedom can be used. In this case, there is a small amount of overdispersion. A quasi-negative binomial option does not exist.

Hilbe (2007) discusses a large range of extensions that can be applied (see his Table 5.1). It is even possible to model the parameter k as a function of covariates, but you may have to program your own model in R. Another exotic cousin of the negative binomial model is the NB-P model, which has as variance $\mu_i + \mu_i^p/k$. If $p = 2$, we end up with the ordinary NB GLM again. These are all useful options if there is overdispersion in the NB GLM, but appropriate R software is scarce.

9.10.2 Results

The intermediate results of the model selection (using first the AIC and then some fine tuning using hypothesis testing) is not given here, but the final model contains the explanatory variables OPEN.L and D.PARK. You could also decide to use L.WAT.C as well because its p -value in a model with OPEN.L and D.PARK is 0.02. We decided to drop it, because these p -values are approximate, and it is so close to the magic 5% level.

Our optimal model and its numerical and graphical output are obtained by the following R code.

```
> M8 <- glm.nb(TOT.N ~ OPEN.L + D.PARK, link = "log",
+                 data = RK)
> summary(M8)
> drop1(M8, test = "Chi")
> op <- par(mfrow = c(2, 2))
> plot(M8)
> par(op)
```

The output from the `drop1` function is given below. Both explanatory variables are significant at the 5% level.

```
Single term deletions
Model: TOT.N ~ OPEN.L + D.PARK

          Df Deviance    AIC      LRT   Pr(Chi)
<none>        51.84 385.43
OPEN.L     1    59.73 391.32    7.89  0.004967
D.PARK     1   154.60 486.19 102.76 < 2.2e-16
```

The `summary` command gives

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	4.6717034	0.1641768	28.455	<2e-16
OPEN.L	-0.0093591	0.0031952	-2.929	0.0034
D.PARK	-0.0001119	0.0000113	-9.901	<2e-16

Dispersion parameter for Negative Binomial(4.1328)
 family taken to be 1

Null deviance: 170.661 on 51 degrees of freedom
 Residual deviance: 51.839 on 49 degrees of freedom
 AIC: 387.43
 Theta: 4.133
 Std. Err.: 0.980
 $2 \times \log\text{-likelihood}$: -379.432

Theta is the parameter k from the variance function. Note that the analysis of deviance results gives slightly different p -values compared to the z -statistics, but the biological conclusions will be similar. The graphical validation plots are presented in Fig. 9.8 and do not show any problems.

The model seems to suggest that the further away you are from the park, the fewer roadkills. Open land cover also has a negative effect of roadkill numbers.

So, which model is better, the quasi-Poisson or the negative binomial GLM? The answer is simple: the quasi-Poisson model has patterns in the residuals and the

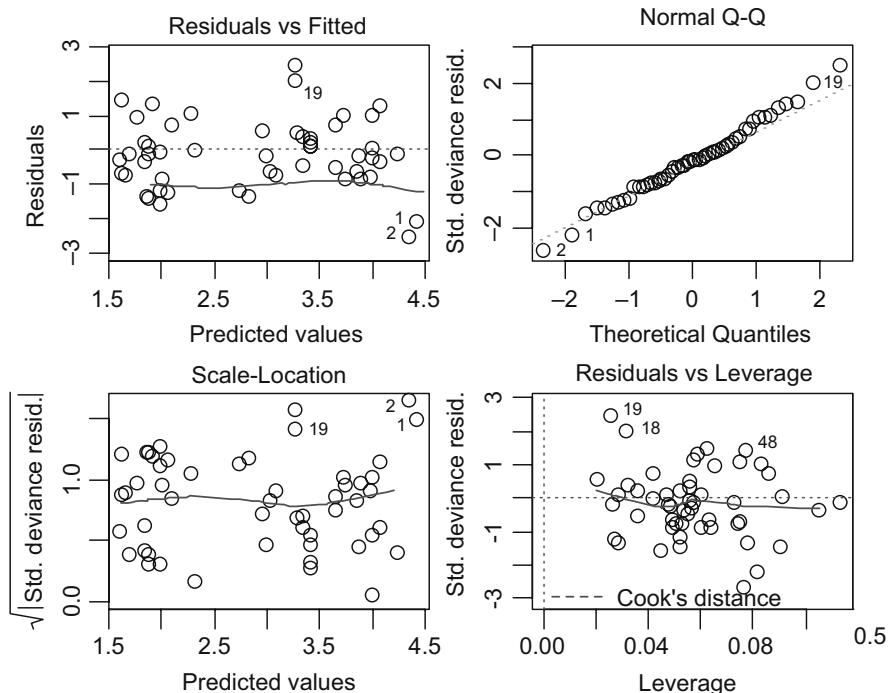


Fig. 9.8 Graphical validation tools for the negative binomial GLM. The graphs do not indicate any problems. We also plotted Pearson residuals versus the fitted values (not shown here), and this graph did not show any problems neither

negative binomial has no patterns, so this is the preferred model. Adding OPEN.L as an explanatory variable to the quasi-Poisson model does not remove the pattern. A bonus of the negative binomial GLM is that the AIC is defined, which allows us to do automatic selection procedures.

If the residual graphs do not show a clear winner, then you can also apply a test to compare the NB and Poisson GLMs; they are nested. The variance of the Poisson is: $\text{var}(Y_i) = \mu_i$, and for the NB we have $\text{var}(Y_i) = \mu_i + \mu_i^2/k$. We can also write the variance of the NB model as $\text{var}(Y_i) = \mu_i + \alpha \times \mu_i^2$. The models will give the same variance if $\alpha = 0$; so we can use a likelihood ratio test and the null hypothesis is $H_0: \alpha = 0$. However, we are testing on the boundary again (the alternative is $H_1: \alpha > 0$). We saw a similar problem when we tested the significance of a random effect in Chapter 5, and the same solution of dividing the *p*-value by 2 can be applied. The Poisson model with OPEN.L and D.PARK is fitted with

```
> M9 <- glm(TOT.N ~ OPEN.L + D.PARK, family = poisson,
  data = RK)
```

The log likelihood test is obtained by

```
> llhNB = logLik(M8)
> llhPoisson = logLik(M9)
> d <- 2 * (llhNB - llhPoisson)
> pval <- 0.5 * pchisq(as.numeric(d), df = 1,
  lower.tail = FALSE)
```

The statistic is equal to 244.66, and the *p*-value is $p < 0.001$. Note that we divided the *p*-value by 2. Hence, there is strong support for the negative binomial model. The same result can be obtained with the command `odTest(M8)` from the `pscl` package, which is not part of the base installation.

The amphibian roadkills data set is further analysed in Chapter 16. A comparison of the Poisson, quasi-Poisson, negative binomial, and three alternative models in case there are lots of zeros (the hurdle model, zero-inflated Poisson, and zero-inflated negative binomial models) is presented in Chapter 11.

9.11 GAM

Having explained Gaussian additive models in detail in Chapter 3 and the Poisson and negative binomial GLM in detail in earlier sections in this chapter, it is rather simple to explain Poisson or negative binomial GAM. A Poisson GAM has these assumptions:

1. Y_i is Poisson distributed with mean μ_i . By definition the variance of Y_i is also equal to μ_i .

2. The systematic part is given by $\eta(X_{i1}, \dots, X_{iq}) = \alpha + f_1(X_{i1}) + \dots + f_q(X_{iq})$, where the f_j s are smoothing functions.
3. There is a logarithm link between the mean of Y_i and the predictor function $\eta(X_{i1}, \dots, X_{iq})$. The logarithmic link ensures that the fitted values are always non-negative.

As a result of these three assumptions, we have

$$\begin{aligned} Y_i &\sim P(\mu_i) \\ E(Y_i) &= \mu_i \quad \text{and} \quad \text{var}(Y_i) = \mu_i \\ \log(\mu_i) &= \eta(X_{i1}, \dots, X_{iq}) \quad \text{or} \quad \mu_i = e^{\eta(X_{i1}, \dots, X_{iq})} \end{aligned} \tag{9.16}$$

For a negative binomial GAM, we only have to change step 1 from the Poisson distribution to a negative binomial distribution and the variance is then given by $\mu_i + \mu_i^2/k$. A detailed example of the negative binomial GAM is given in Chapter 16. Below, we present a short example of a GAM that also illustrates the use of the offset variable in Poisson and NB GLMs and GAMs.

9.11.1 Distribution of larval Sea Lice Around Scottish Fish Farms

The data used in this example are taken from Penston et al. (2008). Plankton tows were taken approximately weekly at two depths (0 and 5 m) at five stations for two years. In the original paper, numbers of *nauplii* and *copepodids* were analysed in two separate univariate analysis where production week (time expressed in weeks since March 2002, when the local farms stocked their cages with lice-free, juvenile fish), station, and depth were the covariates. There are five stations labelled as A, C, E, F, and G. Stations C and G are beside salmon farms, stations A and F are landward of these farms, and station E is seaward of the farms. Here, we only use copepodoids. Further biological details can be found in Penston et al. (2008).

There are three potential problems with the analysis of these data: we have longitudinal (over time) data at each station, there may be correlation between adjacent stations, and there is a large variation in the sampled water volume. As to the first two problems, we follow the same strategy as the paper by showing there is no temporal correlation *within* each of the residual time series, and that there are no strong Pearson correlations *between* the 5 residual time series. The third problem of different volumes per observation was discussed in Chapter 8. Define Y_i as the number of copepodoids measured for observation i . We could have used a notation Y_{sk} , referring to observation s at station k , but we will keep the notation simple and stick to Y_i . As Y_i is a count, a Poisson, negative binomial or geometric distribution is appropriate. We start with the Poisson distribution. So far in this chapter, we assumed that Y_i is Poisson distributed with mean μ_i , which we wrote as $P(\mu_i)$ with its probability function as

$$f(y_i; \mu_i) = \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!} \quad y_i \geq 0, \text{ } y_i \text{ integer} \quad (9.17)$$

The problem with these data is that the water volumes differ per observation, see Fig. 9.9. We may measure a large number of copepodids simply because the water volume was large. The easiest solution is to work with densities, and analyse these with a Gaussian distribution. The disadvantage of this is that the fitted values may become negative, there may be heterogeneity, etc. It is also an option to use Volume as an explanatory variable, but then you would be modelling a functional relationship between Volume and numbers of copepodids. A neater approach is to use Volume as an offset; this process works as follows.

Assume that Y_i is Poisson distributed with mean $\mu_i \times V_i$. V_i is also called the exposure or intensity parameter of the Poisson process, and μ_i is the expected number of copepodids for a one unit volume. The expected value and variance are: $E(Y_i) = \mu_i \times V_i$ and $\text{var}(Y_i) = \mu_i \times V_i$. The following simple algebra leads to a GLM (or GAM) with an offset variable.

$$E(Y_i) = \mu_i \times V_i \Rightarrow \log(E(Y_i)) = \log(\mu_i) + \log(V_i) = \alpha + \beta \times X_{i1} + f(X_{i2}) + \log(V_i)$$

The term $\log(V_i)$, where \log is the natural log, is the offset. Using basic mathematics, we have placed the V_i inside the predictor function, but note there is no regression parameter in front of this term. The other terms α and β are the regression parameters and $f()$ is a smoothing function. R will estimate the regression parameters and smoothers, and you can express the fitted values of the model either as μ_i or as $\mu_i \times V_i$.

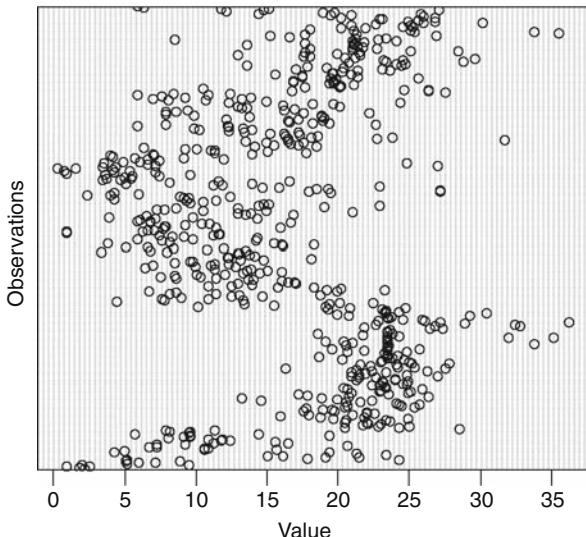


Fig. 9.9 Cleveland dotplot of the sampled volumes. Note that there are considerable differences in volumes! The graph was produced with the R command `dotchart` (`Volume`, `xlab = "Value"`, `ylab = "Observations"`)

The offset can be used for a Poisson, negative binomial, and geometric distribution. The advantages of the offset approach compared to analysing densities are that the fitted values are always positive, the confidence intervals around the fitted values do not contain negative values, and we allow for heterogeneity within the context of a Poisson or NB distribution.

To use an offset variable in a GLM or GAM in R, use the following code.

```
> library(AED); data(Lice)
> Lice$LVol <- log(Lice$Volume)
> Lice$fStation <- factor(Lice$Station)
> L0 <- glm(Copepod ~ offset(LVol) + fStation,
  family = poisson, data = Lice)
```

The first two commands import the data. The variable `LVol` contains the natural log transformed volumes, and `offset(LVol)` ensures that the `glm` function is not putting a parameter in front of it. The only problem is that unfortunately, the model itself is rubbish; we have only shown it to illustrate how to use an offset in a GLM or GAM. So we will now move on and do it for real. There are three explanatory variables, `Station`, `Depth` (both are factors), and `Production_week`. Simple scatterplots indicate no clear relationships, and we therefore used a GAM. We start with a Poisson distribution. The most complicated model that we can apply contains a smoother for production week for each station and depth combination, the main terms station and depth, and the interaction between station and depth. This is the GAM equivalent of 3-way interaction. The problem is that such a model ended in an error message (numerical convergence problems), and we therefore switched to a negative binomial distribution. The following code was used.¹

```
> library(mgcv)
> Lice$PW <- Lice$Production.week #saves some space
> Lice$fDepth <- factor(Lice$Depth)
> L1 <- gam(Copepod ~ offset(LVol) +
  s(PW, by=as.numeric(Station=="A")) +
  s(PW, by=as.numeric(Station=="C")) +
  s(PW, by=as.numeric(Station=="E")) +
  s(PW, by=as.numeric(Station=="F")) +
  s(PW, by=as.numeric(Station=="G")) +
  s(PW, by=as.numeric(Station=="A")) +
  s(PW, by=as.numeric(Station=="C")) +
  s(PW, by=as.numeric(Station=="E")) +
```

¹We used R version 2.6.0. More recent R versions require slightly different code; see the book website for updated code.

```
s(PW, by=as.numeric(Depth=="5m" & Station=="F")) +
s(PW, by = as.numeric(Depth=="5m" & Station=="G")) +
fDepth * fStation,
family = negative.binomial(1), data = Lice)
```

This model also gave a warning message, but including the option `gamma = 1.4` allows the code to run. This option helps against overfitting by the smoothers (Wood, 2006); it puts a heavier penalty on each degrees of freedom in the GCV score (Chapter 3).

A backward selection resulted in various numerical problems, and therefore in the original paper, Penston et al. (2008) adopted a slightly different approach for the model selection process. They estimated the parameter k (used in the NB variance function) from one of the larger models, e.g. from `L3`, and kept it fixed during the backwards selection. This gave an optimal model, and the whole backward selection process was then repeated using the k from the first optimal model. Both selection rounds ended up in the same model, namely,

```
> L3 <- gam(Copepod ~ offset(LVol) +
  s(PW, by = as.numeric(Depth=="0m")) +
  s(PW, by = as.numeric(Depth=="5m")) +
  fDepth + fStation, data = Lice,
  family = negative.binomial(1), gamma = 1.4)
```

This model contains a smoother for production week for each depth together with depth and station as factors. We can compare this model with its Poisson equivalent using the likelihood ratio test:

```
> L4 <- gam(Copepod ~ offset(LVol) +
  s(PW,by = as.numeric(Depth=="0m")) +
  s(PW,by = as.numeric(Depth=="5m")) +
  fDepth + fStation, data = Lice,
  family = poisson, gamma = 1.4)
> llhNB <- logLik(L3); llhPoisson <- logLik(L4)
> d <- 2 * (llhNB - llhPoisson)
> pval <- 0.5 * pchisq(as.numeric(d), df = 1,
  lower.tail = FALSE)
```

The likelihood ratio statistic is 2137.20, which is strong evidence to choose the NB GAM over the Poisson GAM. The numerical output of the NB GAM is obtained by the `summary(L3)` command:

```
Family: Negative Binomial(0.3569). Link function: log
Parametric coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     -1.7030    0.1956  -8.708  < 2e-16
factor(Depth)5m -1.3921    0.2203  -6.319  5.2e-10
```

```

factor(Station)C -0.3496      0.2513   -1.391   0.16470
factor(Station)E -0.4661      0.2546   -1.830   0.06769
factor(Station)F -0.8455      0.2656   -3.183   0.00153
factor(Station)G  0.1102      0.2524   0.437    0.66253

```

Approximate significance of smooth terms:

	edf	F	p-value
s(PW) :as.numeric(Depth=="0m")	8.36	15.62	< 0.001
s(PW) :as.numeric(Depth=="5m")	6.14	5.45	< 0.001
R-sq. (adj) =	0.212	Deviance explained =	72.6%
GCV score =	1.0644	Scale est.	= 1. n = 608

The model explains 72.6% of the null deviance. The *p*-values for the levels of station only indicate which stations are significantly different from the baseline station A (Dalgaard, 2002). A post-hoc test can be applied to investigate which sites are different from each other. The fitted values are given in Fig. 9.10.

Further discussions on the results, model validation (there was no significant temporal auto-correlation within the four residual time series), biological interpretation, and analyses can be found in Penston et al. (2008). Besides the NB GAM, it may also be an option to apply a zero-inflated GAM. These models are discussed in Chapter 11.

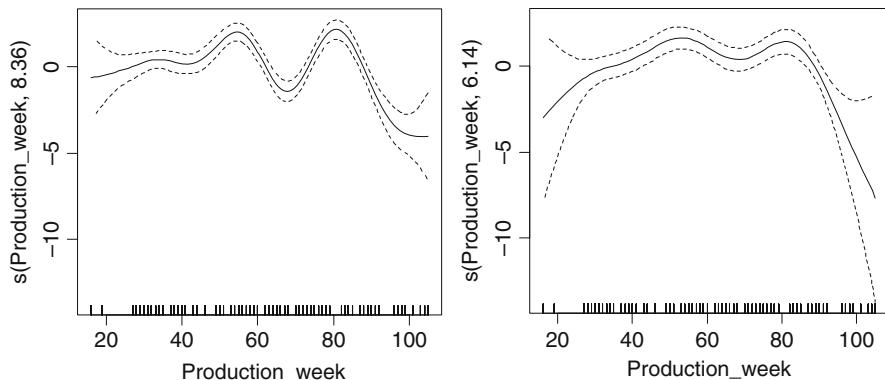


Fig. 9.10 Estimated smoothing curves for depth at the surface (*left*) and depth and 5 m (*right*). The *solid line* is the smoother and the dotted lines are 95% point-wise confidence bands

Chapter 10

GLM and GAM for Absence–Presence and Proportional Data

10.1 Introduction

In the previous chapter, count data with no upper limit were analysed using Poisson generalised linear modelling (GLM) and negative binomial GLM. In Section 10.2 of this chapter, we discuss GLMs for 0–1 data, also called absence–presence or binary data, and in Section 10.3 GLM for proportional data are presented. In the final section, generalised additive modelling (GAM) for these types of data is introduced. A GLM for 0–1 data, or proportional data, is also called logistic regression.

When we wrote Chapters 8, 9, 10, and 11, we had a dilemma how to structure the material. The options were as follows:

1. First present the GLM as abstract formulae, and then show the Poisson, negative binomial, and logistic GLMs as special cases. The disadvantage of this approach is that the reader has to go through a grilling mathematical section. This approach may work for the more mathematically skilled reader, but it did not seem appropriate for our target audience.
2. Present every GLM family in detail, and explain all the procedures every time. This approach may be better for a ‘GLM-only’ book, but it duplicates a lot of text.
3. First present the Poisson GLM in detail, and then present logistic regression (and other GLMs) with help of a couple of examples. The disadvantage of this approach is that the reader has to read the Poisson GLM chapter, even if he or she has absence–presence data.

We decided to go for the third approach because we want to discuss not only the Poisson GLM, but also the logistic GLM, negative binomial GLM, and in Chapter 11 zero-truncated, and zero-inflated GLMs. In Chapter 9, we used a considerable number of pages explaining the Poisson GLM, and the good news is that the mathematical background for this chapter is much the same. However, this does mean you need to have read Chapters 8 and 9 before starting this chapter.

Many statistical textbooks describe logistic regression and we could fill an entire page with references. Some books are dedicated entirely to logistic regression and

some only contain a chapter. Some are for medical science, some for econometrics, and some for ecology. Our favourites are McCullagh and Nelder (1989), Agresti (2002), and Fitzmaurice et al. (2004). The second reference is probably a ‘must read’, and the first one is a ‘must cite’.

10.2 GLM for Absence–Presence Data

We illustrate the binomial GLM for absence–presence data with help of two examples. In Section 10.2.1, we model the probability that a wild boar has tuberculosis (Tb) as a function of the length of the animal (length from the nose to the tail joint along the back of the animal, expressed in centimetres). Another potential explanatory variable is age, but it is collinear with length and unbalanced. This example serves as a simple explanation of binomial GLM. A more detailed binomial GLM is presented in Section 10.2.2, which deals with the presence and absence of parasites in cod.

10.2.1 Tuberculosis in Wild Boar

Vicente et al. (2006) analysed the distribution of tuberculosis-like lesions in wild boar *Sus scrofa* to explore the potential importance of wild boar in the maintenance of tuberculosis in south central Spain. Here, we model Tb as a function of the continuous explanatory variable length (as defined above); it is denoted by LengthCT (CT is an abbreviation of *cabeza-tronco*, which is Spanish for head-body). Tb is a vector of zeros and ones, representing absence and presence of Tb, respectively.

The first thing we do in any data analysis is a data exploration. Useful tools for most types of data are a boxplot and a scatterplot; see Fig. 10.1.

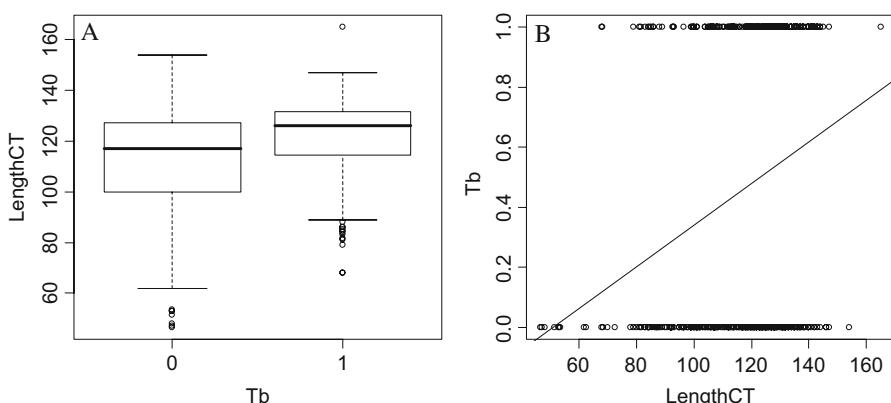


Fig. 10.1 **A:** Boxplot of LengthCT conditional on the variable TB. **B:** Scatterplot of LengthCT versus TB. A regression line was added to aid visual interpretation

The boxplot of LengthCT conditional on Tb (Fig. 10.1A) shows that animals with Tb have larger LengthCT values. The pairplot is less useful due to the 0–1 nature of Tb. When we make a pairplot, we tend to add the fit of a linear regression model. In this case:

$$Tb_i = \alpha + \beta \times CTLength_i + \varepsilon_i.$$

The question is now as follows: How sensible is it to apply linear regression on these data, and what is the interpretation of the fitted line? Let us start with the latter question. The fitted line in Fig. 10.1B suggests that an animal of LengthCT = 100 cm has approximately 0.35 Tb. However, this is a rather strange statement; an animal has Tb or it does not have Tb. It cannot have 0.35 Tb! It seems that our linear regression model is impractical.

To produce a model with fitted values that make more sense, define π_i as the probability that animal i is infected with Tb, and $1 - \pi_i$ is the probability that it is not infected. If we now imagine the vertical axis in Fig. 10.1B showing π_i , we can say that an animal of LengthCT = 100 cm has a probability of 0.35 of being infected with Tb. At least, the fitted values of the linear regression model now make a little bit more sense.

So, the vertical axis in Fig. 10.1B represents the probability that an animal is infected with Tb. Based on the line in Fig. 10.1B, this means that an animal with LengthCT = 47 cm, has a probability of –0.03 of being infected. But probabilities are supposed to be between 0 and 1! And the underlying theory of linear regression tells us that there are realisations (possible outcomes) with probabilities larger than 1 or smaller than 0. It seems we have a serious problem with the linear regression model applied on presence and absence data! The binomial GLM provides a framework to solve all these problems.

To formulate the binomial GLM in a general notation, let Y_i be 1 if animal i is infected with TB and 0 if not infected. A binomial GLM is specified with the same three steps as the Poisson and negative binomial GLMs:

1. An assumption on the distribution of the response variable Y_i . This also defines the mean and variance of Y_i .
2. Specification of the systematic part. This is a function of the explanatory variables.
3. The relationship between the mean value of Y_i and the systematic part. This is also called the link between the mean and the systematic part.

We discuss each of these points in more details next.

Step 1: We assume that Y_i is binomial distributed with probability π_i and $n_i = 1$ independent trials; see also Section 8.6. Recall that this is actually a Bernoulli distribution. As a result, the expected mean and variance of Y_i are given by: $E(Y_i) = \pi_i$ and $\text{var}(Y_i) = \pi_i \times (1 - \pi_i)$. The π_i plays the same role as the μ_i in Poisson regression and negative binomial regression.

Step 2: The systematic part of the model is specified by the predictor function:

$$\eta(LengthCT_i) = \alpha + \beta \times LengthCT_i$$

Step 3: In this step, we need to define the relationship between the expected value of Y_i , π_i , and the predictor function η . We already argued that the identity link (as imposed by the linear regression model) gives non-sensible results; fitted probabilities and possible realisations are smaller than 0 or larger than 1. So, we need a function that maps the values of η between 0 and 1. There are various options, e.g. the logit link, probit link, clog–log link, and log–log link, but the logit link is the default (canonical) link and is probably the most used one. We will explain it first and then quickly discuss the differences with some of the other ones.

The logit link works as follows. Recall that the problem is that π_i is bounded by a lowest value of 0 and a highest value of 1, and the fitted values obtained by the predictor function η and identity link function ignore this on both sides. Define the odds as follows:

$$O_i = \frac{\pi_i}{1 - \pi_i}$$

The odds are an unusual concept for most scientists. We are familiar with probabilities; it tells us how likely things are with a value between 0 and 1. The odds are doing the same thing, but on a different numerical scale. They are used in for example gambling offices; the odds that a race horse will win can be 9 to 1. This means that if you organise 10 races, it is likely that the horse will win 9 times and lose once. In terms of probabilities: the probability that a particular horse will win is 0.9. This is the same statement as saying that the odds are 9. The nice thing about odds is that they do not have an upper bound. Take a series of values for π_i , say 0.1, 0.2, 0.3, . . .), and 0.9, and calculate the odds; they go from something close to zero to something very large. See also Table 10.1 where it shows how probabilities between 0 and 1 are transformed into odds between 0 and infinity.

So, by going from probabilities to odds, we managed to get rid of the upper boundary, but we still have the lower boundary; odds still cannot be negative. The solution is simple; take the natural logarithm of the odds, also called the log odds. The last row in Table 10.1 gives examples of log odds, which are no longer bounded

Table 10.1 Various probabilities, odds, and log odds. The table shows how odds and log odds are calculated from probabilities. The table was taken from Zuur et al. (2007)

P _i	0.001	0.1	0.3	0.4	0.5	0.6	0.7	0.9	0.999
1 – P _i	0.999	0.9	0.7	0.6	0.5	0.4	0.3	0.1	0.001
O _i	0.001	0.11	0.43	0.67	1	1.5	2.33	9	999
Ln(O _i)	-6.91	-2.20	-0.85	-0.41	0	0.41	0.85	2.20	6.91

by a lower or upper limit. In a logistic regression, we model the log odds as a linear function of the explanatory variables. This gives the following:

$$\log(O_i) = \eta(LengthCT)$$

Instead of $\log(O_i)$ we can also write $\text{logit}(\pi_i)$. The entire binomial GLM for the Tb data is now given by

$$\begin{aligned} Y_i &\sim B(1, \pi_i) \\ E(Y_i) &= \pi_i \quad \text{and} \quad \text{var}(Y_i) = \pi_i \times (1 - \pi_i) \\ \text{logit}(\pi_i) &= \alpha + \beta \times LengthCT_i \end{aligned}$$

The last line can also be written with some simple mathematics as

$$\pi_i = \frac{e^{\alpha+\beta \times LengthCT_i}}{1 + e^{\alpha+\beta \times LengthCT_i}}$$

Whatever the values of α , β and $LengthCT_i$, the fitted values for π_i are always between 0 and 1. As this model cannot produce fitted values outside the 0 – 1 range, the binomial distribution ensures we only get sensible realisations.

10.2.1.1 R Code, Results and Fitted Values

The following R code accesses the data, applies the GLM, and presents the numerical output.

```
> library(AED); data(Boar)
> B1 <- glm(Tb ~ LengthCT, family = binomial,
  data = Boar)
> summary(B1)
```

The output is:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-3.892109	0.671152	-5.799	6.67e-09
LengthCT	0.031606	0.005588	5.656	1.55e-08

```
Dispersion parameter for binomial family taken to be 1
Null deviance: 700.76 on 507 degrees of freedom
Residual deviance: 663.56 on 506 degrees of freedom
149 observations deleted due to missingness
AIC: 667.56
```

For the moment, we are focussing on the estimated parameters, and the consequences for the graphical interpretation of the model. Deviances, overdispersion, and AIC are discussed in the next example.

The estimated intercept and slope are -3.89 and 0.03 respectively, and are significant at the 5% level. This means that the probability that an animal is of LengthCT_i is infected, is given by:

$$\pi_i = \frac{e^{-3.89+0.03 \times \text{LengthCT}_i}}{1 + e^{-3.89+0.03 \times \text{LengthCT}_i}}$$

If we fill in a couple of values for LengthCT_i , we can calculate the corresponding π_i , and make a sketch of the relationship. Instead of doing this manually, we use the `predict` command in R:

```
> MyData <- data.frame(LengthCT = seq(from = 46.5,
                                         to = 165, by = 1))
> Pred <- predict(B1, newdata = MyData, type = "response")
> plot(x = Boar$LengthCT, y = Boar$Tb)
> lines(MyData$LengthCT, Pred)
```

We first created a data frame `MyData` with new values for the covariate between 46.5 and 165 , with steps of 1 cm. The values 46.5 and 165 are the smallest and largest values of the observed animals, and using this range prevents predictions outside the range of observed values. The resulting graph is given in Fig. 10.2. The fitted line shows the pattern of a typical sigmoid curve. Note that the fitted values are always between 0 and 1 ! At small lengths, the probability of sampling Tb infected animals is small, whereas the probability increases rapidly from about 70 – 80 cm up to about 140 cm, and then the rate of change becomes smaller again (but the probability of infection stays high).

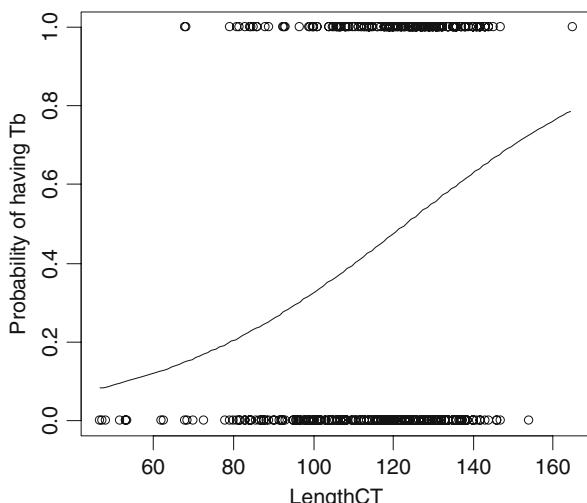


Fig. 10.2 Graph showing the fitted values (*solid line*) obtained by the binomial GLM applied on the boar data. The dots are the observed values

10.2.1.2 General Comments

For a binomial GLM applied on binary data, the mean is given by π_i and the variance by $\pi_i \times (1 - \pi_i)$. To visualise the mean–variance relationship, we plotted the estimated values of π_i versus $\pi_i \times (1 - \pi_i)$, see Fig. 10.3. Note that the variance is the largest for intermediate values of π_i .

Besides the logit link, other link functions are available and a comparison of GLMs with different link functions can be found in, for example, Hardin and Hilbe (2007). Most binary GLMs in the literature use the logit link, but the probit link is a good second choice. We will not discuss the probit link or any of the other link functions here, but the main difference is the shape of the fitted line in Fig. 10.2. In fact, we suggest that you plot the fitted curves obtained from a probit link and a clog–log link yourself. All that is needed is to modify the code in the `family` option inside the `glm` command to `family = binomial(link = "probit")` or `family = binomial(link = "cloglog")`. You will see that the fitted curve is slightly different in the lower and upper parts.

The logit and probit link functions assume that you have approximately an equal number of zeros and ones. The clog–log may be an option if you have considerably more zeros than ones, or vice versa; the sigmoidal curve is asymmetrical. If you do decide to use any of the non-standard link functions (that is other than the logit) for a binary GLM, Hardin and Hilbe (2007) give examples how you can compare different link functions and some tools are based on the AIC, BIC, and deviance.

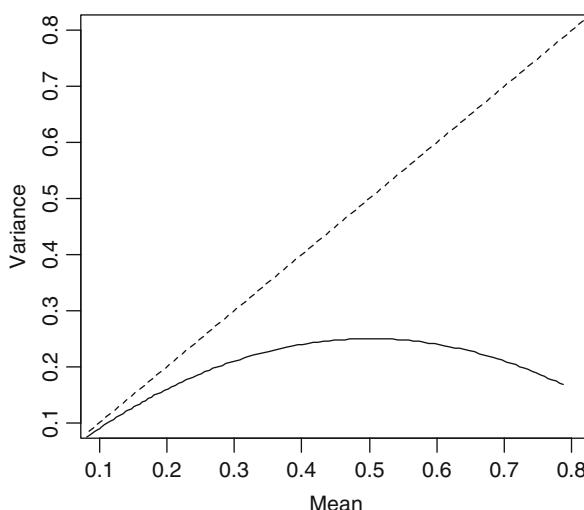


Fig. 10.3 Relationship between the mean π_i and variance $\pi_i \times (1 - \pi_i)$ is given by the *solid line*. The dotted line is the line for which the mean equals the variance. Note that the variance is the largest for intermediate values of π_i

10.2.2 Parasites in Cod

The red king crab *Paralithodes camtschaticus* was introduced to the Barents Sea in the 1960s and 1970s from its native area in the North Pacific. The leech *Johannsonia arctica* uses the carapace of this crab to deposit eggs. The leech is a vector for a trypanosome blood parasite of marine fish, including cod. Hemmingsen et al. (2005) examined a large number of cod for trypanosome infections during annual cruises along the coast of Finnmark in North Norway. These cruises covered three years and were divided in four ‘stations’ or areas. Full details of the research and results can be found in their paper. Their statistical analyses were carried out using Chi-square statistics and analysis of variance and are in principle all correct. Here, we use a subset of the data and repeat their analyses with GLM.

The response variable is Prevalence, which is coded as 1 if the parasite is present and 0 else. Possible explanatory variables are year, area, and the depth that fish were caught. The problem is that not all areas have the same depth; hence, purely because of the study design, depth and area are collinear (just make a boxplot of depth conditional on area, and you will see that this is indeed the case). Other explanatory variables are sex, length, weight, stage, and age of the fish. Except for sex, all these variables are highly collinear and an arbitrary choice on which one to use has to be made. However, the aim of this text is not to present a full blown analysis, but merely to explain binomial GLM. Hemmingsen et al. (2005) used a model with the main terms year, area, and length, and an interaction term year \times area, and we will also use this set of covariates. We have 1254 observations, but with a few missing values in the spreadsheet. We could remove them, but we prefer using the data as they are and guide you through the problems.

This is clearly a binomial GLM as the response variable is coded as 0 – 1. The following model is applied.

$$\begin{aligned} Y_i &\sim B(1, \pi_i) \\ E(Y_i) &= \pi_i \quad \text{and} \quad \text{var}(Y_i) = \pi_i \times (1 - \pi_i) \\ \text{logit}(\pi_i) &= \text{Year}_i + \text{Area}_i + \text{Year}_i \times \text{Area}_i + \text{Length}_i \end{aligned}$$

We have written down the systematic part of the model in a semi-mathematical notation because Year_i and Area_i are fitted as factors (each have three levels) and Length_i is a continuous variable. The R code to fit this model is given by

```
> library(AED); data(ParasiteCod)
> ParasiteCod$fArea <- factor(ParasiteCod$Area)
> ParasiteCod$fYear <- factor(ParasiteCod$Year)
> Par1 <- glm(Prevalence ~ fArea * fYear + Length,
               family = binomial, data = ParasiteCod)
```

The `family = binomial` option and a response variable with zeros and ones is the only difference compared the GLMs used in the previous chapters.

The `summary(Par1)` command gives a lot of output due to the three levels for Area and Year. If we omit for the moment, the estimated values, standard errors, z -values, and p -values, we have

```
...
Dispersion parameter for binomial family taken to be 1
Null deviance: 1727.8 on 1247 degrees of freedom
Residual deviance: 1495.2 on 1235 degrees of freedom
6 observations deleted due to missingness
AIC: 1521.2
```

The good news is that in a Bernoulli GLM (the response variable is a vector with zeros and ones), overdispersion cannot occur. A justification for this can be found in McCullagh and Nelder (p. 125, 1989). The rest of the information is similar as for the Poisson GLM (Chapter 9). For example, the AIC can be used for model selection. And we also have the `step` function, which can be used for automatic model selection. The hypothesis testing procedures are also identical to Poisson GLM. Because we have factors with more than two levels in the model, we use `drop1(Par1, test = "Chi")` as it gives one p -value for the interaction. The output below shows that the Area \times Year interaction is highly significant at the 5% level, but not the variable Length.

```
Single term deletions. Model:
Prevalence ~ fArea * fYear + Length
      Df Deviance    AIC      LRT   Pr(Chi)
<none>          1495.16  1521.16
Length           1     1498.64  1522.64    3.47    0.06
fArea:fYear      6     1537.60  1551.60   42.44 <0.001
```

The full output from the `summary` command is not shown here, but just as in linear regression and Poisson GLM, it tells you which levels of a factor are different from the baseline level, and in the case of an interaction, which combinations are different from the baseline (Area 1 and Year 1999). To see which area–year combinations are different from *each other*, you can change these baselines to other values, and do some post-hoc testing (see Chapter 10 in Dalgaard (2002)).

The graphical model validation in a Binomial GLM with a 0 – 1 response variable is some sort of an art, and Fig. 10.4 shows why. So far, we said: You should not see any patterns in the residuals. Because the observed data are zeros and ones, we now see two clear bands in these graphs. This makes it rather difficult to say anything sensible about these graphs, and one can wonder whether there is any point in using them. In cases where you have a large data set, like we have in this example (1254 observations), it may be an option to extract the residuals, put them in groups of, say 10, calculate an average of the residuals per group, and use these in graphical validation plots. The groups can be based on the order of the fitted values, or on the order of a covariate.

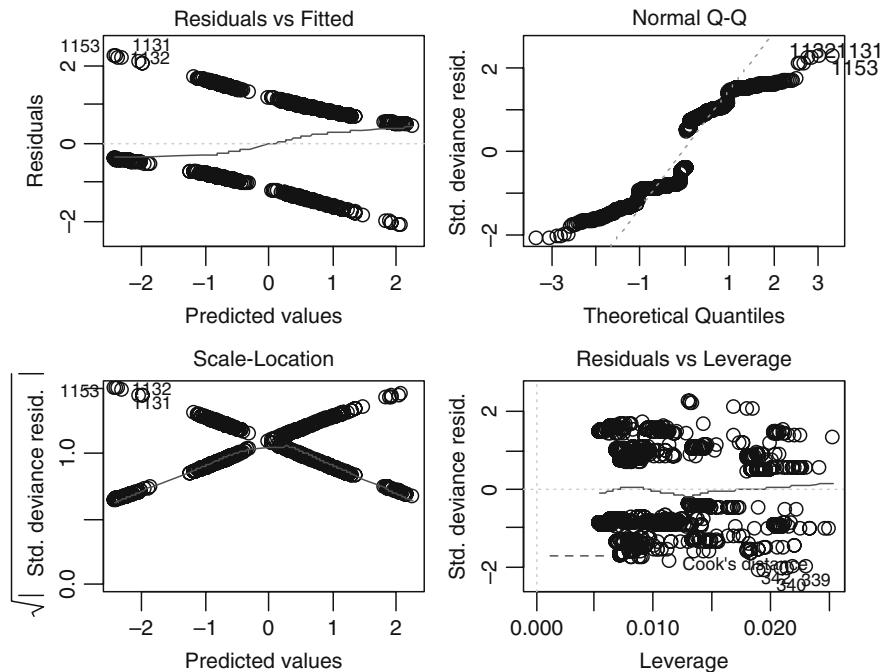


Fig. 10.4 Standard graphical validation tools for a GLM. Because we are working with a response variable that has only zeros and ones, we can see two bands of points in all but the leverage plot

The easy mistake to make with the model selection process for this data set is ignoring the missing values. Once an explanatory variable with missing values is dropped from the selection process, the new data set may have more observations, and therefore, AICs are not comparable! This also holds for analysis of deviance tables. But luckily, the `drop1` command does it properly by removing the observations with NAs, but that only works for one round. It is therefore better to remove missing values before doing the model selection process.

10.3 GLM for Proportional Data

In the previous section, the response variable Y_i was binary and a Bernoulli distribution was used. The notation for this was $B(1, \pi_i)$, where π_i is the probability on ‘success’.

Vicente et al. (2006) analysed data from a number of estates with wild boar and red deer in Spain. At each estate i , a group of n_i animals was sampled. The data set contains information on the tuberculosis (Tb) disease in both species, and on the parasite *Elaphostrongylus cervi*, which only infects red deer. Both variables are recorded as the number of animals that are positive for Tb or have the parasite

E. cervi. So the response variable Y_i is the number of animals that test positive out of n_i animals. There is also information on the main characteristics of the habitat and management (fencing) at each estate: The percentage of open land, scrubs and pine plantation, number of *quercus* plants per area, number of *quercus* trees per area, a wild boar abundance index, a reed deer abundance index, estate size (ha), and whether the estate is fenced (1 = yes, 0 = no).

Data like these are typically analysed using a GLM with a binomial distribution (Chapter 8). Let us focus first on *E. cervi* in deer. Define Y_i as the number of deer at estate i that have *E. cervi*, and n_i is the number of sampled deer. The binomial GLM is as follows:

$$Y_i \sim B(n_i, \pi_i)$$

$$E(Y_i) = \pi_i \times n_i \quad \text{and} \quad \text{var}(Y_i) = n_i \times \pi_i \times (1 - \pi_i)$$

$$\text{logit}(\pi_i) = \alpha + \beta_1 \times \text{OpenLand}_i + \beta_2 + \text{ScrubLand}_i + \cdots + \beta_8 \times \text{Fenced}_i$$

We also assume that the n_i deer are independent and that each animal at estate i has the same probability π_i of having the parasite. If this is not the case, then you should work with the individual binary data per animal and use generalised linear mixed modelling techniques (Chapter 13). The logistic regression model can be fitted in R with the following code. The first two commands import the data. The function `corvif` is our own function that calculates variance inflation factors to detect collinearity. It is available in the `AED` package, but a similar function can be obtained from the `car` package. The variable `PinePlantation` was dropped due to collinearity. The remaining code applies the binomial GLM.

```
> library(AED); data(Tbdeer)
> Z <- cbind(Tbdeer$OpenLand, Tbdeer$ScrubLand,
   Tbdeer$QuercusPlants, Tbdeer$QuercusTrees,
   Tbdeer$ReedDeerIndex, Tbdeer$EstateSize,
   Tbdeer$Fenced)
> corvif(Z)
> DeerNegCervi <- Tbdeer$DeerSampledCervi -
   Tbdeer$DeerPosCervi
> Tbdeer$ffFenced <- factor(Tbdeer$Fenced)
> Deer1 <- glm(cbind(DeerPosCervi, DeerNegCervi) ~
   OpenLand + ScrubLand + QuercusPlants +
   QuercusTrees + ReedDeerIndex + EstateSize + ffFenced,
   family = binomial, data = Tbdeer)
> summary(Deer1)
```

Note that the response variable is a data frame consisting of two columns; the number of positives and the number of negatives (which is `DeerNegCervi`). It is also possible to fit the model with the following code:

```
> Tbdeer$DeerPosProp <- Tbdeer$DeerPosCervi /
  Tbdeer$DeerSampledCervi
> Deer2 <- glm(DeerPosProp ~ OpenLand + ScrubLand +
  QuercusPlants + QuercusTrees +
  ReedDeerIndex + EstateSize + fFenced,
  family = binomial, data = Tbdeer,
  weights = DeerSampledCervi)
```

The variable `DeerPosProp` contains the proportion (as a value between 0 and 1) of animals that are positive (presence of the parasite). Both approaches give exactly the same results. The summary output from model `Deer2` is as follows.

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.843e+00	7.772e-01	4.945	7.61e-07
OpenLand	-3.950e+00	6.383e-01	-6.187	6.12e-10
ScrubLand	-7.696e-01	6.140e-01	-1.253	0.210042
QuercusPlants	-3.633e-04	2.308e-02	-0.016	0.987439
QuercusTrees	2.290e-03	5.326e-02	0.043	0.965707
ReedDeerIndex	6.689e-02	2.097e-02	3.191	0.001419
EstateSize	-8.218e-05	2.478e-05	-3.316	0.000913
fFenced1	-2.273e+00	5.954e-01	-3.819	0.000134

Dispersion parameter for binomial family taken to be 1
 Null deviance: 234.85 on 22 degrees of freedom
 Residual deviance: 152.79 on 15 degrees of freedom
 (9 observations deleted due to missingness)
 AIC: 227.87

This output is similar to the output of the Poisson GLM in Chapter 9. In a binomial GLM with $n_i > 1$, we can have overdispersion. This seems to be the case here, and we have to fit a quasi-binomial model. This is doing the same thing as in a quasi-Poisson GLM by adding an overdispersion parameter ϕ to the variance of Y_i ; $\text{Var}(Y_i) = \phi \times n_i \times \pi_i \times (1 - \pi_i)$. The R programming process is similar to a quasi-Poisson process; first we need to fit a model with the `family = quasibinomial` option (call the resulting object `Deer3`) and the `drop1(Deer3, test = "F")` command can be used to assess which term to drop. The final model contains only `OpenLand`:

```
> Deer4 <- glm(cbind(DeerPosCervi, DeerNegCervi) ~
  OpenLand, data = Tbdeer,
  family = quasibinomial)
> drop1(Deer4, test = "F")
```

The analysis of deviance test with the `drop1` command is not presented here, but it gives a p -value of 0.02 for `OpenLand`, and the `summary` command gives

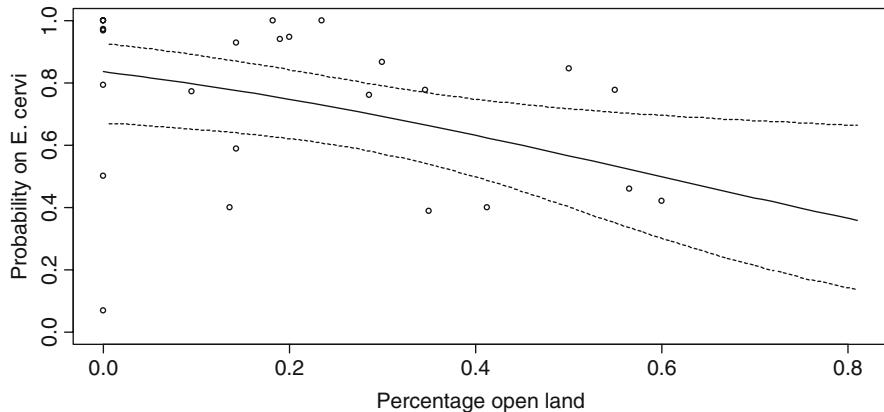


Fig. 10.5 Fitted values (*solid line*) and 95% confidence bands for the optimal binomial GLM model applied on the red deer data

a negative slope. These results suggest that the larger the percentage of open land, the smaller the probability of sampling deer with *E. cervi*. The results can also be visualised (Fig. 10.5) using the R code:

```
> MyData <- data.frame(OpenLand =
  seq(from = min(Tbdeer$OpenLand),
      to = max(Tbdeer$OpenLand), by = 0.01))
> P1 <- predict(Deer4, newdata = MyData, type = "link",
  se = TRUE)
> plot(MyData$OpenLand, exp(P1$fit) / (1+exp(P1$fit)),
  type = "l", ylim = c(0, 1),
  xlab = "Percentage open land",
  ylab = "Probability on E. cervi")
> lines(MyData$OpenLand, exp(P1$fit+1.96*P1$se.fit) /
  (1 + exp(P1$fit + 1.96 * P1$se.fit)), lty = 2)
> lines(MyData$OpenLand, exp(P1$fit-1.96*P1$se.fit) /
  (1 + exp(P1$fit - 1.96 * P1$se.fit)), lty = 2)
> points(Tbdeer$OpenLand, Tbdeer$DeerPosProp)
```

The data frame `MyData` contains new values for the explanatory variable `OpenLand`, and we use these for the predictions. The `predict` function predicts at the level of the predictor function, and therefore, we need to transform the fitted values (and the confidence bands) with the logistic link function. This ensures that the confidence bands are between 0 and 1.

The model validation process in a binomial GLM is identical to the one in a Poisson GLM; plot the Pearson or deviance residuals against the fitted values and plot the residuals versus each explanatory variable in the model (and also against the variables that were dropped).

10.4 GAM for Absence–Presence Data

Having explained additive modelling in detail in Chapter 2 and binomial GLM in detail in all the earlier sections in this chapter, binomial GAM is just a combination of the two, and we now give a short example to illustrate the method. In Section 10.2, we analysed the presence of parasites in cod, and assumed that $Y_i \sim B(1, \pi_i)$ and

$$\text{logit}(\pi_i) = \alpha + \text{Year}_i + \text{Area}_i + \text{Year}_i \times \text{Area}_i + \text{Length}_i$$

Because all explanatory variables are nominal or continuous, the model is called a generalised *linear* model. If you are unsure that length has a linear effect, or if a plot of residuals (obtained by a GLM) against Length shows a clear pattern, we can use:

$$\text{logit}(\pi_i) = \alpha + \text{Year}_i + \text{Area}_i + \text{Year}_i \times \text{Area}_i + f(\text{Length}_i)$$

where $f(\text{Length}_i)$ is a smoothing function of Length . Such a model is called a generalised *additive* model. The only difference between a GLM and a GAM is that the latter contains at least one smoothing function in the predictor function. The following R code applies a GAM on the cod parasite data. Length is fitted as a smoother.

```
> library(AED); data(ParasiteCod)
> library(mgcv)
> ParasiteCod$fArea <- factor(ParasiteCod$Area)
> ParasiteCod$fYear <- factor(ParasiteCod$Year)
> P2 <- gam(Prevalence ~ fArea * fYear + s(Length),
family = binomial, data = ParasiteCod)
```

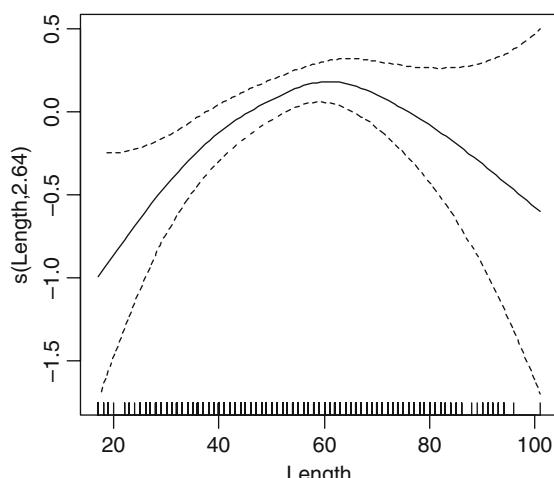


Fig. 10.6 Estimated smoother for Length obtained by the GAM applied on the cod parasite data. The solid line is the smoother, and the dotted lines are 95% confidence bands

The only difference compared to the `gam` commands in Chapter 3 is the `family = binomial` option. The same model selection and model validation steps should be applied as we did with logistic regression and discussed in previous sections. The `anova(P2)`, `summary(P2)`, and `plot(P2)` commands can be used. No numerical output is presented here, but the smoother of `Length` is significant at the 5% level (2.63 degrees of freedom, $X^2 = 17.08$, $p = 0.009$). The estimated smoother is presented in Fig. 10.5. Although 2.63 degrees is not strong evidence against a GLM (1 degree of freedom is identical to a GLM), the curve clearly shows a non-linear `Length` effect. Fish around 60 have a higher probability of having parasites than smaller and larger fishes (Fig. 10.6).

10.5 Where to Go from Here?

In Chapters 12 and 13, we extend GLM and GAM to allow for nested data, and temporal and spatial correlations, leading to the methods of generalised estimation equations, generalised linear mixed modelling, and generalised additive mixed modelling.

Chapter 11

Zero-Truncated and Zero-Inflated Models for Count Data

11.1 Introduction

In this chapter, we discuss models for zero-truncated and zero-inflated count data. Zero truncated means the response variable cannot have a value of 0. A typical example from the medical literature is the duration patients are in hospital. For ecological data, think of response variables like the time a whale is at the surface before re-submerging, counts of fin rays on fish (e.g. used for stock identification), dolphin group size, age of an animal in years or months, or the number of days that carcasses of road-killed animals (amphibians, owls, birds, snakes, carnivores, small mammals, etc.) remain on the road. These are all examples for which the response variable cannot take a value of 0.

On their own, zero-truncated data are not necessarily a problem. It is the underlying assumption of Poisson and negative binomial distributions that may cause a problem as these distributions allow zeros within their range of possible values. If the mean is small, and the response variable does not contain zeros, then the estimated parameters and standard errors obtained by GLM may be biased. In Section 11.2, we introduce zero-truncated Poisson and zero-truncated negative binomial models as a solution for this problem. If the mean of the response variable is relatively large, ignoring the truncation problem, then applying a Poisson or negative binomial (NB) generalised linear model (GLM), is unlikely to cause a problem. In such cases, the estimated parameters and standard errors obtained by Poisson GLM and truncated Poisson GLM tend to be similar (the same holds for the negative binomial models).

In ecological research, you need to search very hard to find zero-truncated data. Most count data are zero *inflated*. This means that the response variable contains more zeros than expected, based on the Poisson or negative binomial distribution. A simple histogram or frequency plot with a large spike at zero gives an early warning of possible zero inflation. This is illustrated by the graph in Fig. 11.1, which shows the numbers of parasites for the cod dataset that was used in Chapter 10 to illustrate logistic regression. In addition to presence and absence of parasites in cod, Hemmingsen et al. (2005) also counted the number of parasites, expressed as intensity.

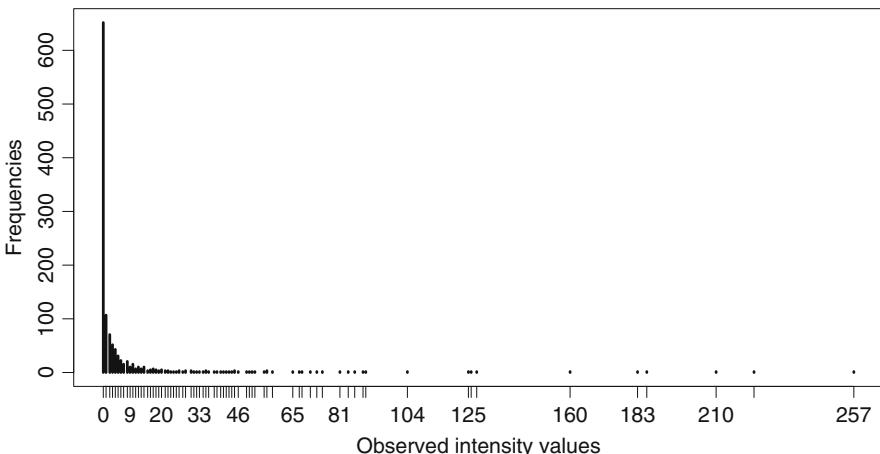


Fig. 11.1 Plot of the frequencies for the response variable `Intensity` from cod parasite data. There are 654 zeros, 108 ones, 71 twos, 52 threes, 44 fours, 31 fives, etc. Note the large numbers of zeros indicating zero inflation. R code to make this graph is presented in Section 11.4

In this chapter, four models are discussed that can deal with the excessive number of zeros; zero-inflated Poisson (ZIP), zero-inflated negative binomial (ZINB) models, zero-altered Poisson (ZAP), and zero-altered negative binomial (ZANB) models. There are two main distinctions in these abbreviations; ZI versus ZA, and P versus NB. The latter pair of Poisson versus negative binomial should be familiar territory with the negative binomial models (ZINB and ZANB) coping with a certain degree of overdispersion. Furthermore, because a Poisson GLM is nested in a NB GLM, the ZIP is nested in a ZINB, and a ZAP is nested in a ZANB. The difference between ZI and ZA is slightly more complicated and is related to the nature of the zeros. We discuss this further in Sections 11.3 and 11.4. What we call ZI is also called mixture models in the literature, and our ZA is normally known as two-part models.

In the past, software for mixture and two-part models used to be in obscure functions, and different software packages gave different results. It is only recently that these methods have become more popular and a growing number of people are using the software. This means that most of the bugs have now been filtered out, and publications with mixture and two-part models applied on ecological data are appearing more frequently (Welsh et al., 1996; Agarwal et al., 2002; Barry and Welsh, 2002; Kuhnert et al., 2005; Minamia et al., 2007; and Potts and Elith, 2006 among several others). There are also many applications outside ecology; see, for example, Lambert (1992), Ridout et al. (1998), Xie et al. (2001), and Carrivick et al. (2003) among many others in the fields of social science, traffic accident research, econometrics, psychology, etc. A nice overview and comparison of Poisson, NB, and zero-inflated models in R is given in Zeileis et al. (2008). This paper also gives a couple of useful references to publications using mixture and two-part models.

If you start digging into zero-inflated models, you have to rely mainly on papers as few statistical textbooks cover this topic in any detail. A few exceptions are

Cameron and Trivedi (1998), Hardin and Hilbe (2007), or Hilbe (2007), but only a small number of pages are dedicated to mixture and two-part models. As papers tend to present things in a compact and condensed format, we decided to use this chapter to explain these methods in more detail. We assume that you are fully familiar with the methods discussed in Chapters 8, 9, and 10.

A detailed explanation of the underlying principle of mixture and two-part models is given in Sections 11.2–11.5, and in Section 11.6, we compare the different models and discuss how to choose between them.

11.2 Zero-Truncated Data

In this section, we discuss models that can be used when the response variable is a count and cannot obtain the value of zero. In this case, we refer to the variable as being zero truncated. In Section 11.2.1, we discuss the underlying mathematics for zero-truncated Poisson models and the negative binomial equivalent. In Section 11.2.2, we give an example and discuss software. If you are not interested in the underlying mathematics, you can skip Section 11.2.1 (but you should still try and read the summary at the end of that section) and go straight to the example.

Knowledge of the material discussed in this section is required for ZAP and ZANB models discussed in Section 11.5.

11.2.1 The Underlying Mathematics for Truncated Models

11.2.1.1 Mathematics for the Zero-Truncated Poisson Model

Let Y_i be the response variable for observation i . We assume it is Poisson distributed with mean μ_i . We have already discussed in Chapter 8, how the Poisson probability function can be adjusted to exclude zeros, and we briefly revisit it here. The starting point was the Poisson probability function:

$$f(y_i; \mu_i | y_i \geq 0) = \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!} \quad (11.1)$$

Recall that y_i is a possible outcome of Y_i . The function gives the probability for each integer value of y_i that is equal or larger than 0 for a given mean μ_i . For example, the probability that $y_i = 0$ is

$$f(0; \mu_i) = \frac{\mu^0 \times e^{-\mu_i}}{0!} = e^{-\mu_i}$$

Recall from Chapter 8 that we can exclude the probability that $y_i = 0$ from the Poisson distribution by dividing its probability function in Equation (11.1) by 1 minus the probability that $y_i = 0$, resulting in

$$f(y_i; \mu_i | y_i > 0) = \frac{\mu_i^{y_i} \times e^{-\mu_i}}{(1 - e^{-\mu_i}) \times y_i!} \quad (11.2)$$

From this point onwards, truncated Poisson GLM follows ordinary Poisson GLM. We use the same mean and variance relationships, the same systematic component, and the same link function. Hence, the mean value μ_i is modelled as an exponential function of the predictor function:

$$\mu_i = e^{\alpha + \beta_1 \times X_{1i} + \dots + \beta_q \times X_{qi}}$$

To find the regression parameters, we need to specify a likelihood criterion. The only difference with Poisson GLM is that we use the probability function in Equation (11.2) instead of the one in Equation (11.1), and this gives

$$L = \prod_i f(y_i; \mu_i | y_i > 0) = \prod_i \frac{\mu_i^{y_i} \times e^{-\mu_i}}{(1 - e^{-\mu_i}) \times y_i!} \quad (11.3)$$

In Chapter 9, we explained that this expression is based on the probability rule that $\Pr(A \text{ and } B) = \Pr(A) \times \Pr(B)$ if A and B are independent. The f s in Equation (11.3) are the probabilities. The principle of maximum likelihood states that for the given data, we need to maximise L as a function of the regression parameters. To aid the numerical optimisation routines, we use the log-likelihood so that we can work with a sum instead of a product:

$$\log(L) = \sum_i \log(f(y_i; \mu_i | y_i > 0)) = \sum_i \log \left(\frac{\mu_i^{y_i} \times e^{-\mu_i}}{(1 - e^{-\mu_i}) \times y_i!} \right) \quad (11.4)$$

Using matrix notation, we replace the $\beta_1 \times X_{1i} + \dots + \beta_q \times X_{qi}$ by $\mathbf{X}_i \times \boldsymbol{\beta}$, where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_q)$, and \mathbf{X}_i contains all explanatory variables for observation i . A bit of high school mathematics gives

$$\begin{aligned} \log(L) = & - \sum_i e^{\mathbf{X}_i \times \boldsymbol{\beta}} + \sum_i y_i \times \mathbf{X}_i \times \boldsymbol{\beta} - \sum_i \log(1 - e^{\mathbf{X}_i \times \boldsymbol{\beta}}) \\ & - \sum_i \log(\Gamma(y_i + 1)) \end{aligned} \quad (11.5)$$

Just as for the Poisson GLM, we end up with a maximum likelihood criterion that needs to be maximised as a function of the regression parameters. The algorithm needs first-order and second-order derivatives (which can easily be determined and we leave this as an exercise for the reader), and then it is purely a matter of numerical optimisation, though we end up with a slightly different algorithm compared to Poisson GLM. Details can be found in Barry and Welsh (2002) or Hilbe (2007).

11.2.1.2 Mathematics for the Negative Binomial Truncated Model

The NB truncated model follows the same steps. The starting point is the probability function for y larger or equal to 0 (Chapter 9):

$$f(y_i; k, \mu_i | y_i \geq 0) = \frac{\Gamma(y_i + k)}{\Gamma(k) \times \Gamma(y_i + 1)} \times \left(\frac{k}{\mu_i + k} \right)^k \times \left(1 - \frac{k}{\mu_i + k} \right)^{y_i} \quad (11.6)$$

The probability that $y_i = 0$ is given by

$$f(0; k, \mu_i) = \frac{\Gamma(0 + k)}{\Gamma(k) \times \Gamma(0 + 1)} \times \left(\frac{k}{\mu_i + k} \right)^k \times \left(1 - \frac{k}{\mu_i + k} \right)^0 = \left(\frac{k}{\mu_i + k} \right)^k$$

To exclude the probability that $y_i = 0$, we divide the probability function in Equation (11.6) by 1 minus the probability that $y_i = 0$, resulting in

$$f(y_i; \mu_i | y_i > 0) = \frac{\Gamma(y_i + k)}{\Gamma(k) \times \Gamma(y_i + 1)} \times \left(\frac{k}{\mu_i + k} \right)^k \times \left(1 - \frac{k}{\mu_i + k} \right)^{y_i} \Bigg/ \left(1 - \left(\frac{k}{\mu_i + k} \right)^k \right) \quad (11.7)$$

We can follow the same steps as in Equations (11.3) and (11.4) and also use the logarithmic link function. The end result is as follows:

$$\log(L) = \log(L_{NB}) - \log \left(1 - \left(\frac{k}{\mu_i + k} \right)^k \right) \quad (11.8)$$

where $\log(L_{NB})$ is the log likelihood from the NB GLM (see Chapter 9). Note that the notation in Hardin and Hilbe (2007) and Hilbe (2007) uses a slightly different parameterisation of $k = 1/\alpha$.

11.2.1.3 Summary

For those of you who skipped all the mathematical text in this subsection, here is a short summary. We adjusted the probability functions for the Poisson and negative binomial (NB) distributions to exclude the probability of a zero observation. We then specified the log likelihood criterion for the zero-truncated Poisson and NB models. First-order and second-order derivatives can easily be derived. It is now only a matter of numerical optimisation to find the regression parameters. Software code exists to fit these models in R, and an example is given in the next section.

11.2.2 Illustration of Poisson and NB Truncated Models

In this section, we illustrate zero-truncated models. The data are unpublished (at the time of writing) and were donated by António Mira (University of Évora, Portugal). The response variable is the number of days that carcasses of road-killed

animals remain on the road. For illustrative purposes, we only use snakes (*Coronella girondica*, *Coluber hippocrepis*, *Elaphe scalaris*, and *Macroprotodon cucullatus*). We removed some observations because of the unbalanced design (different sample sizes), and the remaining data set contains 130 observations. There are also potential issues with spatial and temporal correlation, but in this subsection, we only focus on the zero truncation.

Figure 11.2 shows a frequency plot of the number of days that snake carcasses remain on a road. The value of 1 does not represent 24 hours exactly, rather it is just that we start counting with 1 because each carcass is on the road for at least a couple of hours. The number of days will never be zero. Except for the lucky snakes that made it to the other side of the road. They will have a value of zero, but of course, are not (yet) part of this dataset.

The following R code accesses the data and produces the frequency plot in Fig. 11.2. The code is self explanatory.

```
> library(AED); data(Snakes)
> plot(table(Snakes$N_days))
```

Ignoring the zero truncation problem and analysing these data with a Poisson GLM is already a major challenge! The explanatory variables are *Size_cm* (mean size of adults of each species), *PDayRain* (proportion of days with rain), *Tot_Rain* (total rainfall in mm), *Temp_avg* (average daily mean temperature), *Road* (identity of the road representing traffic intensity; EN114 has high traffic, EN4 has medium traffic, and EN370.EN114_4 has low traffic), *Road_Loc* (location on the road; L = paved lane and V = paved verge), *Season*, and *Species*.

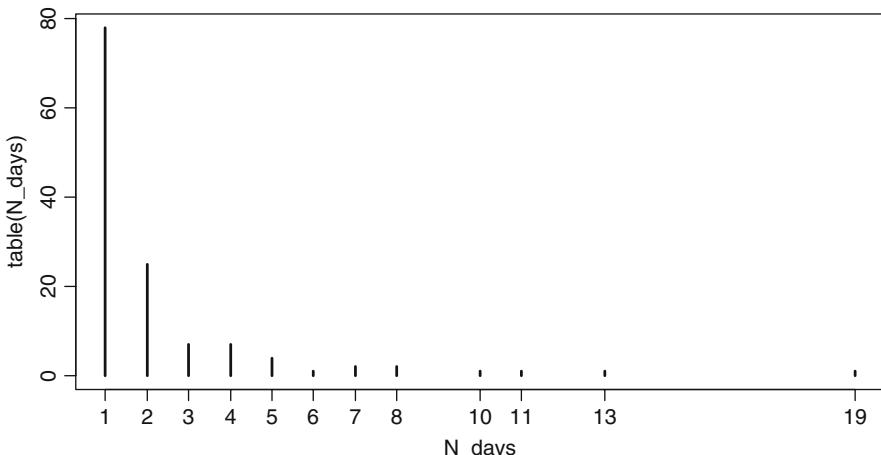


Fig. 11.2 Frequency plot of the response variable *N_days*, the number of days snake carcasses remain on the road. Note that a value of 0 cannot occur

The variables `Size_cm`, `PDayRain`, `Tot_Rain`, and `Temp_avg` are continuous; all others are nominal.

Exploring these data using pairplots and correlation coefficients for the continuous variables, and boxplots of each continuous explanatory variable conditional on each nominal explanatory variable, showed that `Season` is collinear with both `Temp_avg` and `Tot_Rain`, and there is also collinearity between `PDayRain` and `Temp_avg`. We therefore omitted `Season` and `Temp_avg`. All observations from the same species had the same size, and therefore, the covariate `Species` was also dropped. From a biological point of view, it may be argued that `Species` is a more useful covariate than size; however, the degrees of freedom rapidly increase if various two-way interactions with species are included in the model.

Using common sense, it can be argued that there may be interactions; perhaps, carcasses of bigger animals at sites with less rain stay longer on the road? Not all 2-way interactions can be fitted due to the experimental design. We started our data analysis with a Poisson GLM and quickly noticed overdispersion. Therefore, a quasi-Poisson model was applied. The results of this model are not presented here, but there is an overdispersion of 1.5 and various terms are not significant. The aim of this section is to show the difference between a GLM and a zero-truncated GLM, and because there is no such thing as a zero-truncated quasi-Poisson model, we switch to a negative binomial model as NB models allow for a more flexible approach to overdispersion. R code for the NB GLM, ignoring the zero truncation, is given by

```
> library(MASS)
> M1 <- glm.nb(N_days ~ Size_cm + PDayRain + Tot_Rain +
+ Road + Road_Loc + Size_cm:PDayRain +
+ Size_cm:Tot_Rain + Size_cm:Road +
+ Size_cm:Road_Loc + PDayRain:Tot_Rain +
+ PDayRain:Road + PDayRain:Road_Loc +
+ Tot_Rain:Road, data = Snakes)
```

Similar code was used in Chapter 9. The results of the `summary(M1)` command are not presented here, but show that various terms are not significant at the 5% level. The optimal model was found using `step(M1)`, and further fine tuning was done with the `drop1(M1, test = "Chi")` command. The optimal model is given by

```
> M2A <- glm.nb(N_days ~ PDayRain + Tot_Rain +
+ Road_Loc + PDayRain:Tot_Rain, data = Snakes)
```

The two-way interaction `PDayRain:Tot_Rain` and the main term `Road_Loc` were significant at the 5% level. The explained deviance of this model is 40%. The parameter k (theta in the R output) in the variance function $\mu_i + \mu_i^2/k$ is equal to 6.72. Interestingly, the model selection process for the quasi-Poisson GLM gave the same results.

So far, we have used the `glm.nb` function from the MASS package for negative binomial GLM; but it can also be done in other packages, for example, in the VGAM (Vector Generalized Additive Models) package with the code:

```
> library(VGAM)
> M2B <- vglm(N_days ~ PDayRain + Tot_Rain + Road_Loc +
+ PDayRain:Tot_Rain, family = negbinomial,
+ data = Snakes)
> summary(M2B)
```

The VGAM package does not come with the base installation of R; so you will need to download and install it. Actually, this package is rather interesting as it contains many statistical techniques closely related to those we use in this book. For example, it has tools for multivariate (multiple response variables) GLMs and GAMs (Yee and Wild, 1996), and it is one of the few packages that can do zero-truncated models! It is certainly worthwhile having a look at the package description at www.stat.auckland.ac.nz/~yee/VGAM. The zero-truncated NB model is run with the following R code.

```
> M3A <- vglm(N_days ~ PDayRain + Tot_Rain + Road_Loc +
+ PDayRain:Tot_Rain, family = posnegbinomial,
+ control = vglm.control(maxit = 100),
+ data = Snakes)
```

The `family = posnegbinomial` argument ensures that a zero-truncated NB model is applied. The `summary` command can be used to obtain estimated parameters and standard errors, but the `anova` and `drop1` functions have not yet been implemented in the VGAM package.

The option `family = pospoisson` runs a zero-truncated Poisson GLM, and if `vglm` is replaced by `vgam`, we obtain a zero-truncated GAM. To run an ordinary Poisson GLM, use `family = poissonff`; the extra `ff` is due to VGAM's incompatibility with the ordinary `family` option in R and is specific to this package. Another 'problem' with VGAM is that it overwrites existing functions. You can overcome this by using, for example, `stats:::resid` after you have typed the `library(VGAM)` command. The `stats:::` ensures that you use the `resid` function from the `stats` package (which is the one used in all chapters so far) and not VGAM's `resid` function, which is not compatible with `glm` and `lm` objects.

It is interesting to compare the parameters and standard errors estimated using NB GLM and truncated NB GLM. The following code looks intimidating, but only collates the corresponding estimated regression parameters in a table:

```
> Z <- cbind(coef(M2A), coef(M3A)[-2])
> ZSE <- cbind(sqrt(diag(vcov(M2A))),
+ sqrt(diag(vcov(M3A)))[-1]))
```

```
> Comp <- cbind(z[,1], z[,2], zSE[,1], zSE[,2])
> Comb <- round(Comp, digits = 3)
> colnames(Comb) <-
      c("NB", "Trunc.NB", "SE NB", "SE Trunc.NB")
> Comb
```

The `coef` command extracts the estimated parameters and the `vcov` the covariance matrix of the estimated parameters. The diagonal elements of this matrix are the estimated variances; hence, the square root of these gives the standard errors. `[-2]` ensures that only regression parameters are extracted and not the parameter k . The `cbind` command prints the columns next to each other, and the `colnames` command adds labels. The output is as follows:

	NB	Trunc.NB	SE NB	SE Trunc.NB
(Intercept)	0.365	-2.035	0.112	0.267
PDayRain	-0.001	0.114	0.193	0.449
Tot_Rain	0.120	0.254	0.020	0.065
Road_LocV	0.449	1.077	0.148	0.368
PDayRain:Tot_Rain	-0.109	-0.234	0.022	0.070

The first two columns are the estimated parameters obtained by NB GLM and truncated NB GLM. As you can see, the estimated parameters obtained using these two methods are rather different! The same holds for the standard errors in the third and fourth columns. Also note that the standard errors of the truncated NB are all larger.

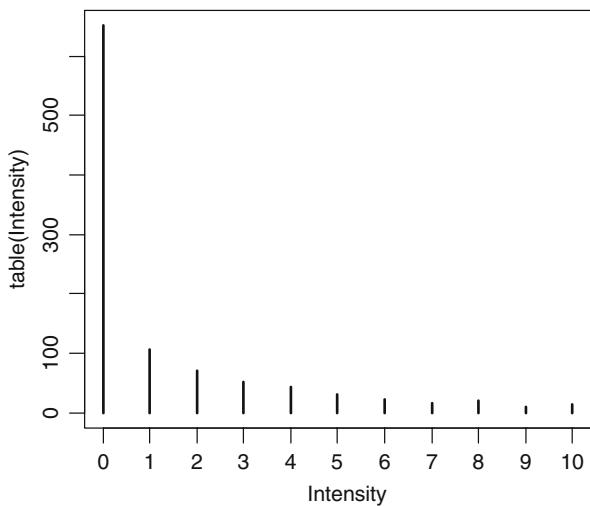
Differences between NB GLM and truncated NB GLM will become smaller if the observed values are further away from zero. But in this case, with 93% of the observations smaller than 5, it makes a substantial difference!

11.3 Too Many Zeros

Zero inflation means that we have far more zeros than what would be expected for a Poisson or NB distribution. Let us have another look at Fig. 11.1, but only at the frequencies between 0 and 10 (see Fig. 11.3). If the data followed a Poisson distribution, you would not expect 651 zeros! It depends a bit on the value of the mean of the Poisson distribution, but 100 zeros would be more likely (see also the shapes of the Poisson probability functions in Chapter 8).

Ignoring zero inflation can have two consequences; firstly, the estimated parameters and standard errors may be biased, and secondly, the excessive number of zeros can cause overdispersion. Before discussing two techniques that can cope with all these zeros, we need to ask the question: Why do we have all these zeros?

Fig. 11.3 Intensity of parasites in cod. This is the same graph as Fig. 11.1, except that only frequencies between 0 and 10 are shown



11.3.1 Sources of Zeros

If we assume a Poisson distribution for the data presented in Fig. 11.3, then we would expect approximately 100–150 zeros. These are at the lower part of the vertical line at intensity = 0. All the other zeros are excess zeros and more than we expect. Some authors try to make a distinction between these two groups of zeros. For example, Kuhnert et al. (2005) and Martin et al. (2005) discriminate between various types of errors that may be causing the zeros in the context of bird abundances in forest patches.

1. First of all, there are structural errors. This means that a bird is not present because the habitat is not suitable.
2. The second is design error, where poor experimental design or sampling practises are thought to be the reason. As an example, imagine counting the number of puffins on the cliffs in the winter. It is highly likely that all samples will be 0 as it is the wrong season and they are all at sea. Another design error is sampling for too short a time period or sampling too small an area.
3. The third cause for zeros is observer error. Some bird species look similar, or are difficult to detect. The less experienced the observer, the more likely he/she will end up with zero counts for bird species that are difficult to identify. Alternatively, the observer may be highly experienced, but it is extremely difficult to detect a tiny dark bird in a dark field on a dark day.
4. The ‘bird’ error. This means that the habitat is suitable, but the site is not used.

There is even a fifth type of zero, the so-called naughty naughts (Austin and Meyers, 1996). For non-native English readers, this can be translated as the bad

zeros. These are zeros due to sampling outside the habitat range that an animal lives in; for example, sampling for elephants in the sea. Any such zeros should be removed.

The zeros due to design, survey, and observer errors are also called false zeros or false negatives. In a perfect world, we should not have them. The structural zeros are called positive zeros, true zeros, or true negatives. It should be noted that these definitions of true and false zeros are open to discussion. In some studies, a false zero may actually be a true zero; see also Martin et al. (2005) for a discussion.

11.3.2 Sources of Zeros for the Cod Parasite Data

Hemmingset al. (2005) looked at the effect of introducing the red king crab *Paralithodes camtschaticus* in the Barents Sea. This species is a host for the leech *Johannsonia arctica*, which in turn is a vector for a trypanosome blood parasite of marine fish, including cod. The data set contains a large number of zeros. Let us discuss what type of zeros we have.

First of all, there are fish that have not been exposed to the parasite, either because they were caught at a place where there are no red king crabs or they had migrated long distances and arrived when Hemmingset al. turned up to catch them. These zeros can probably be labelled as zeros due to ‘poor’ experimental design; however, we put quotation marks around poor as there is not much the biologists can do about it. None the less they are still false zeros that we need to deal with. We also have zeros because of observer errors. Apparently, it is not always easy to detect trypanosomes in fish with light infections, even for experienced parasitologists (Ken MacKenzie, personal communication). So these are also false zeros. The other type of zeros, the true zeros or the true negatives, come from fish that may have been in contact with red king crabs; but for some reason, they have zero parasites. There may be many reasons for this, including habitat, immunity, and environmental conditions.

11.3.3 Two-Part Models Versus Mixture Models, and Hippos

In the next section, four models are used to analyse the zero-inflated data: ZIP, ZINB, ZAP, and ZANB (see also Table 11.1). We have already discussed the difference between the P and the NB. That is Poisson versus negative binomial, where the negative binomial allows for extra overdispersion in the positive (non-zero) part of the data. The difference between the mixture and two-part models is how they deal with the different types of zeros. The two-part models (ZAP and ZANB) are probably easier to explain; they consist of two parts:

Table 11.1 Overview of ZIP, ZAP, ZINB and ZANB models. All models can cope with overdispersion due to excessive numbers of zeros. The negative binomial models can also cope with overdispersion due to extra variation in the count data. The ZIP and ZINB are mixture models in the sense that they consist of two distributions. The ZAP and ZANB are also called hurdle models, conditional models, or compatible models

Model	Full name	Type of model	Overdispersion
ZIP	Zero-inflated Poisson	Mixture	Zeros
ZINB	Zero-inflated negative binomial	Mixture	Zeros and counts
ZAP	Zero-altered Poisson	Two-part	Zeros
ZANB	Zero-altered negative binomial	Two-part	Zeros and counts

1. In first instance, the data are considered as zeros versus non-zeros and a binomial model is used to model the probability that a zero value is observed. It is possible to use covariates in this model, but an intercept-only model is also an option.
2. In the second step, the non-zero observations are modelled with a truncated Poisson (ZAP) or truncated negative binomial (ZANB) model, and a (potentially different) set of covariates can be used. Because the distributions are zero truncated, they cannot produce zeros.

You can use specific software for ZAPs and ZANBs, but it is also possible to carry out these two steps manually with a binomial GLM and a Poisson/NB GLM; both give the same results in terms of estimated parameters and standard errors. The advantage of using specialised ZAP or ZANB software is that it gives one AIC for both models (this can also be calculated manually from the two separate models), and it is more flexible for hypothesis testing for the combined model. Figure 11.4 shows a graphical presentation of the two-part, or hurdle, models for the hippo example. The name hurdle comes from the idea that whatever mechanism is causing the presence of hippos, it has to cross a hurdle before values become non-zero. The important point is that the model does not discriminate between the four different types of zeros.

The ZIP and ZINB models work rather differently. They are also called mixture models because the zeros are modelled as coming from two different processes: the binomial process and the count process. As with the hurdle models, a binomial GLM is used to model the probability of measuring a zero and covariates can be used in this model. The count process is modelled by a Poisson (ZIP) or negative binomial (ZINB) GLM. The fundamental difference with the hurdle models is that the count process can produce zeros (the distribution is not zero truncated).

The underlying process of the mixture model is sketched in Fig. 11.5. In the count process, the data are modelled with, for example, a Poisson GLM, and under certain covariate conditions, we count zero hippos. These are true zeros. But there is also a process that generates only false zeros, and these are modelled with a binomial model. Hence, the binomial GLM models the probability of measuring a false positive versus all other types of data (counts and true zeros).

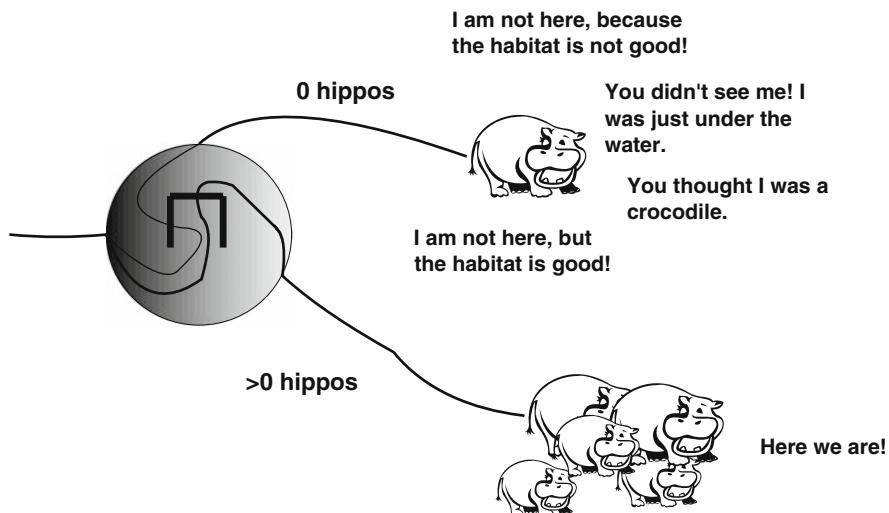


Fig. 11.4 Sketch of a two-part, or hurdle model. There are two processes; one is causing zeros versus non-zeros, the other process is explaining the non-zero counts. This is expressed with the hurdle in the *circle*; you have to cross it to get non-zero counts. The model does not make a distinction between the different types of zeros

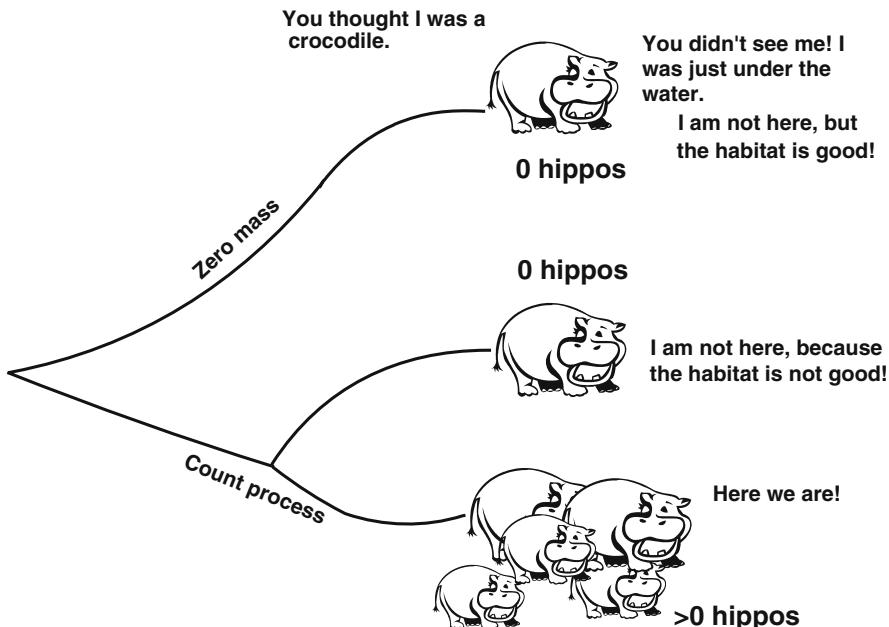


Fig. 11.5 Sketch of the underlying principle of mixture models (ZIP and ZINB). In counting hippos at sites, one can measure a zero because the habitat is not good (the hippos don't like the covariates), or due to poor experimental design and inexperienced observers (or experienced observers but difficult to observe species)

Summarising, the fundamental difference between mixture and two-part models is how the zeros are modelled. Or formulated differently, how do you want to label the zeros in the data? There are many papers where selection criteria (for example, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and estimated parameters) are obtained from Poisson, quasi-Poisson, NB, ZIP, ZINB, ZAP, and ZANB GLMs, and the model with the lowest value is deemed as ‘the best’ model. We do this later in this chapter, but it is perhaps better to choose between the latter four models based on biological knowledge.

It should be noted that labelling the different types of zeros and classifying them into two groups, false and true zeros, is useful for the ecological interpretation, but the bottom line is that in a mixture model, some of the zeros are modelled with the covariates that are also used for the positive count data, and all extra zeros are part of the zeros in the binomial model. For this process to work, it is unnecessary to split the data into true zeros and false zeros.

11.4 ZIP and ZINB Models

We follow the same approach as in Section 11.2; first we discuss the maximum likelihood for the ZIP and ZINB models in Section 11.4.1 and provide an example and R code in Section 11.4.2. If you are not interested in the underlying mathematics, just read the summary at the end of Section 11.4.1, and continue with the example.

11.4.1 Mathematics of the ZIP and ZINB

Let us return to the hippo example in Fig. 11.5 and focus on the question: What is the probability that you have zero counts? Let $\Pr(Y_i)$ be the probability that at site i , we measure a hippo. The answer to the question is

$$\begin{aligned}\Pr(Y_i = 0) &= \Pr(\text{False zeros}) + (1 - \Pr(\text{False zeros})) \\ &\quad \times \Pr(\text{Count process gives a zero})\end{aligned}\tag{11.9}$$

The component $\Pr(\text{False zeros})$ is the upper part of the graph in Fig. 11.5. The second component comes from the probability that it is not a false zero multiplied by the probability that it is a true zero. Basically, we divide the data in two imaginary groups; the first group contains only zeros (the false zeros). This group is also called the observations with zero mass. The second group is the count data, which may produce zeros (true zeros) and as well as values larger than zero. Note that we are not actively splitting the data in two groups; it is just an *assumption* that we have these two groups. We do not know which of the observations with zeros belong to a specific group. All that we know is that the non-zeros (the counts) are in group 2.

Things like ‘probability of false zero’, and 1 minus this probability indicates a binomial distribution, and indeed, this is what we will do. We assume that the

probability that Y_i is a false zero is binomially distributed with probability π_i , and therefore, we automatically have the probability that Y_i is not a false zero is equal to $1 - \pi_i$. Using this assumption, we can rewrite Equation (11.9):

$$\Pr(Y_i = 0) = \pi_i + (1 - \pi_i) \times \Pr(\text{Count process at site } i \text{ gives a zero}) \quad (11.10)$$

So, what do we do with the term $\Pr(\text{Count process gives a zero})$? We assume that the counts follow a Poisson, negative binomial, or geometric distribution. And this is the difference between zero-inflated *Poisson* and zero-inflated *negative binomial* models. Because the geometric distribution is a special case of the NB, it does not have a special name like ZIP or ZINB.

Let us assume for simplicity that the count Y_i follows a Poisson distribution with expectation μ_i . We have already seen its probability function a couple of times, but just to remind you

$$f(y_i; \mu_i | y_i \geq 0) = \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!} \quad (11.11)$$

In Section 11.2, we showed that for a Poisson distribution, the term $\Pr(\text{Count process gives a zero})$ is given by

$$f(y_i = 0; \mu_i | y_i \geq 0) = \frac{\mu^0 \times e^{-\mu_i}}{0!} = e^{-\mu_i} \quad (11.12)$$

Hence, Equation (11.10) can now be written as

$$\Pr(y_i = 0) = \pi_i + (1 - \pi_i) \times e^{-\mu_i} \quad (11.13)$$

The probability that we measure a 0 is equal to the probability of a false zero, plus the probability that it is not a false zero multiplied with the probability that we measure a true zero.

This was the probability that $Y_i = 0$. Let us now discuss the probability that Y_i is a non-zero count. This is given by

$$\Pr(Y_i = y_i) = (1 - \Pr(\text{False zero})) \times \Pr(\text{Count process}) \quad (11.14)$$

Because we assumed a binomial distribution for the binary part of the data (false zeros versus all other types of data) and a Poisson distribution for the count data, we can write Equation (11.14) as follows:

$$\Pr(Y_i = y_i | y_i > 0) = (1 - \pi_i) \times \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!} \quad (11.15)$$

Hence, we have the following probability distribution for a ZIP model.

$$\begin{aligned}\Pr(y_i = 0) &= \pi_i + (1 - \pi_i) \times e^{-\mu_i} \\ \Pr(Y_i = y_i | y_i > 0) &= (1 - \pi_i) \times \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!}\end{aligned}\quad (11.16)$$

The notation $\Pr()$ stands for probability; it is probably better to use the notation in terms of probability functions f :

$$\begin{aligned}f(y_i = 0) &= \pi_i + (1 - \pi_i) \times e^{-\mu_i} \\ f(y_i | y_i > 0) &= (1 - \pi_i) \times \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!}\end{aligned}\quad (11.17)$$

The last step we need is to introduce covariates. Just as in Poisson GLM, we model the mean μ_i of the positive count data as

$$\mu_i = e^{\alpha + \beta_1 \times X_{i1} + \dots + \beta_q \times X_{iq}} \quad (11.18)$$

Hence, covariates are used to model the positive counts. What about the probability of having a false zero, π_i ? The easiest approach is to use a logistic regression with an intercept:

$$\pi_i = \frac{e^\nu}{1 + e^\nu} \quad (11.19)$$

where ν is an intercept. But, what if the process generating false zeros depends on covariates? Nothing stops us from including covariates that model the probability of false zeros:

$$\pi_i = \frac{e^{\nu + \gamma_1 \times Z_{i1} + \dots + \gamma_q \times Z_{iq}}}{1 + e^{\nu + \gamma_1 \times Z_{i1} + \dots + \gamma_q \times Z_{iq}}} \quad (11.20)$$

We used the symbol Z for the covariates as these may be different to the covariates that influence the positive counts. γ s are regression coefficients.

We are now back on familiar territory; we have a probability function in Equation (11.17), and we have unknown regression parameters $\alpha, \beta_1, \dots, \beta_q, \nu, \gamma_1, \dots, \gamma_q$. It is now a matter of formulating the likelihood equation based on the probability functions in Equation (11.17); take the logarithm, get derivatives, set them to zero, and use a very good optimisation routine to get parameter estimates and standard errors. We do not present all the mathematics here, instead see p. 126 in Cameron and Trivedi (1998) or p. 174 in Hilbe (2007).

The only difference between a ZIP and ZINB is that the Poisson distribution for the count data is replaced by the negative binomial distribution. This allows for overdispersion from the non-zero counts. The probability functions of a ZINB are simple modifications of the ones from the ZIP:

$$\begin{aligned}f(y_i = 0) &= \pi_i + (1 - \pi_i) \times \left(\frac{k}{\mu_i + k} \right)^k \\ f(y_i | y_i > 0) &= (1 - \pi_i) \times f_{NB}(y)\end{aligned}\quad (11.21)$$

The function $f_{NB}(y)$ is given in Equation (11.6).

11.4.1.1 The Mean and the Variance in ZIP and ZINB Models

Before giving an example, we need to discuss what the expected mean and variance of a ZIP and ZINB model are. In a Poisson GLM, we have $E(Y_i) = \mu_i$ and $\text{var}(Y_i) = \mu_i$, whereas in an NB GLM we have $E(Y_i) = \mu_i$ and $\text{var}(Y_i) = \mu_i + \mu_i^2/k$. In ZIP and ZINB, this is slightly different due to the definition of the probability functions in Equations (11.17) and (11.21). To derive these means and variances, we need a couple of basic rules:

1. $E(Y) = \sum y \times f(y)$. The summation is over $y = 0, 1, 2, 3, \dots$, etc. The function f is either the Poisson probability function in Equation (11.11) or the NB from Equation (11.6).
2. $\text{var}(Y) = E(Y^2) - E(Y)^2$.
3. $\Gamma(y+1) = y \Gamma(y)$.

Using these rules and a bit of basic mathematics (and a lot of paper), we obtain the following expressions for the mean and variance of a ZIP.

$$\begin{aligned} E(Y_i) &= \mu_i \times (1 - \pi_i) \\ \text{var}(Y_i) &= (1 - \pi_i) \times (\mu_i + \pi_i \times \mu_i^2) \end{aligned} \tag{11.22}$$

You can find these also on p. 126 in Cameron and Trivedi (1998). If the probability of false zeros is zero, that is $\pi_i = 0$, we obtain the mean and variance equations from the Poisson GLM. If $\pi_i > 0$, then the variance is larger than the mean; hence, excessive number of (false) zeros causes overdispersion!

The equations for the ZINB follow the same steps (and are a bit more tedious to obtain) and are as follows.

$$\begin{aligned} E(Y_i) &= \mu_i \times (1 - \pi_i) \\ \text{var}(Y_i) &= (1 - \pi_i) \times (\mu_i + \frac{\mu_i^2}{k}) + \mu_i^2 \times (\pi_i^2 + \pi_i) \end{aligned} \tag{11.23}$$

If the probability of false zeros is 0, we obtain the mean and variance of the NB GLM. Now that we have expressions for the mean and variances of ZIP and ZINB models, we can calculate Pearson residuals:

$$\text{Pearson residual}_i = \frac{Y_i - (1 - \pi_i) \times \mu_i}{\sqrt{\text{var}(Y_i)}}$$

Depending whether a ZIP or ZINB is used, substitute the appropriate variance. μ_i and π_i are given by Equations (11.18) and (11.20), respectively.

11.4.1.2 Summary

If you skipped the mathematics above, here is a short summary. We started asking ourselves how you can measure zero hippos. This is because we can measure either false zeros or true zeros. We then defined π_i as the probability that we measure a

false zero at site i , and for the count data we assumed a Poisson distribution with mean μ_i . This leads to a statement of the form: The probability that we measure 0 hippos is given by the probability that we measure a false zero plus the probability that we do not measure a false zero multiplied with the probability that we measure a true zero. In the same way we can specify the probability that we measure a non-zero count: The probability that we do not measure a false zero multiplied with the probability of the count. Now fill in the distributions, and we get Equation (11.17). The mean values μ_i and π_i can be modelled in terms of covariates. For example, the average number of hippos at site i may depend on the availability of food, and the probability of counting a false zero (false zero) may be because the observer needs better glasses (use observer experience as covariate to model π_i). The rest is a matter of formulating and optimising a maximum likelihood equation, which follows the type of equations we saw in earlier sections and chapters.

It is important to realise that our count process, as modelled by a Poisson process can produce zeros.

11.4.2 Example of ZIP and ZINB Models

We now show an application of ZIP and ZINB models using the cod parasite data. Recall that the choice between a ZIP and ZINB depends whether there is overdispersion in the count data. So, if you apply a ZIP, and there is still overdispersion, just apply the ZINB. We use the `pscl` package (Zeileis et al., 2008) for inflated models.

In Chapter 10, we applied a binomial model for the cod parasite data. However, the numbers of parasites were also measured, and this is a count. The following code loads the data, defines the nominal variables, and removes the missing values (which are present in the response variable). Removing missing values is not really necessary, but it makes the R code for model validation easier, especially when plotting residuals versus the original explanatory variables.

```
> library(AED); data(ParasiteCod)
> ParasiteCod$fArea <- factor(ParasiteCod$Area)
> ParasiteCod$fYear <- factor(ParasiteCod$Year)
> I1 <- is.na(ParasiteCod$Intensity) |
  is.na(ParasiteCod$fArea) |
  is.na(ParasiteCod$fYear) |
  is.na(ParasiteCod$Length)
> ParasiteCod2 <- ParasiteCod[!I1, ]
> plot(table(ParasiteCod2$Intensity),
  ylab = "Frequencies",
  xlab = "Observed intensity values") #Fig. 11.1
```

The `pscl` package is reasonably new, and we are using version 0.92. The function `zeroinfl` applies a zero-inflated model, and the required R code is as follows.

```
> library(pscl)
> f1 <- formula(Intensity ~ fArea*fYear +
+                  Length | fArea * fYear + Length)
> Zip1 <- zeroinfl(f1, dist = "poisson",
+                     link = "logit", data = ParasiteCod2)
```

We could also have typed `zeroinfl(f1)` as we used default settings for the `dist` and `link` options. The `dist` option specifies the distribution for the count data, and the available choices are Poisson, negative binomial, and geometric. The `link = logit` option specifies the logistic link for the false zeros versus the non-false zeros (the true zeros plus the positive counts). But the distribution will always be a binomial. Now let us focus on the more difficult bit, the formula `f1`. The function `zeroinfl` allows the following formulae specifications.

1. $Y \sim X_1 + X_2$. This is equivalent to: $Y \sim X_1 + X_2 | 1$.
2. $Y \sim X_1 + X_2 | X_1 + X_2$
3. $Y \sim X_1 + X_2 | Z_1 + Z_2$

The first option specifies the following link functions for the count data and the binomial data:

$$\mu_i = e^{\alpha + \beta_1 \times X_{i1} + \beta_2 \times X_{i2}} \quad \text{and} \quad \pi_i = \frac{e^\nu}{1 + e^\nu}$$

The mean μ_i for the Poisson count data is modelled in terms of the covariates X_1 and X_2 and the probability π_i for the binomial distribution with a constant. If you think, purely based on biology, that the probability of false zeros is also a function of X_1 and X_2 , then use the second option:

$$\mu_i = e^{\alpha + \beta_1 \times X_{i1} + \beta_2 \times X_{i2}} \quad \text{and} \quad \pi_i = \frac{e^{\nu + \gamma_1 \times X_{i1} + \gamma_2 \times X_{i2}}}{1 + e^{\nu + \gamma_1 \times X_{i1} + \gamma_2 \times X_{i2}}}$$

If you want to model the probability of false zeros with a different set of covariates, say Z_1 and Z_2 , then go for option 3, and use

$$\mu_i = e^{\alpha + \beta_1 \times X_{i1} + \beta_2 \times X_{i2}} \quad \text{and} \quad \pi_i = \frac{e^{\nu + \gamma_1 \times Z_{i1} + \gamma_q \times Z_{i2}}}{1 + e^{\nu + \gamma_1 \times Z_{i1} + \gamma_q \times Z_{i2}}}$$

In this model, the count process is modelled with a different set of covariates compared to the process generating the false zeros. In the theory section, we explained this in terms of measuring no hippos because you forgot your glasses (Z describes the quality of the observer) and X for the count process can be habitat variables.

We went for option 2, but we show in a moment that the model in option 1 is nested in the model in option 2, which means that we can compare them with a likelihood ratio test. Let us return to our R code for the formula.

```
f1 <- formula(Intensity ~ fArea * fYear +
               Length | fArea*fYear + Length)
```

This means that the following link functions (in words) are applied.

$$\mu_i = e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}} \quad \text{and} \quad \pi_i = \frac{e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}}{1 + e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}}$$

You could also copy the code inside the `formula` command directly into the `zeroinfl` command, but the code becomes rather intimidating. The `summary(Zip1)` command gives the estimated parameters, standard errors, z -values, and p -values, but these values are not presented here. The interaction term for the log-link function is significant, and the same can be said for the logistic link function. Hence, the $\text{Area} \times \text{Year}$ term seems to be important for the counts, but also for the probability of measuring false zeros. `Length` has no effect on the false zeros.

However, the ZIP model uses a Poisson distribution for the counts, and the ordinary Poisson GLM applied on these data already showed overdispersion. Before continuing with the model selection and validation, we need to look whether we have dealt properly with the overdispersion. Remember that the ZIP model only deals with zero inflation, not directly with overdispersion in the non-zero count data. If the overdispersion in a Poisson GLM is caused by the excessive number of zeros, then the ZIP will take care of the overdispersion, and we are finished. But if the overdispersion is not caused by the zeros, then the ZIP is not the appropriate model either! The best way to judge whether the ZIP is acceptable is to compare it with a ZINB as these models are nested.

The following code applies a ZINB, and applies a likelihood ratio test, and the output is given as well. The package `lmtest` is not part of the base installation, and you will need to download and install it.

```
> Nb1 <- zeroinfl(f1, dist = "negbin", link = "logit",
                     data = ParasiteCod2
> library(lmtest)
> lrtest(Zip1,Nb1)

Likelihood ratio test
Model 1: Intensity ~ fArea * fYear + Length | fArea *
           fYear + Length
Model 2: Intensity ~ fArea * fYear + Length | fArea *
           fYear + Length

#Df LogLik Df Chisq Pr(>Chisq)
1 26 -6817.6
2 27 -2450.4  1 8734.2  < 2.2e-16
```

Recall from Chapter 9 that with the likelihood ratio test, we are testing whether the variance structure of the Poisson, $\text{var}(Y_i) = \mu_i$, is the same as the

variance structure of the NB, $\text{var}(Y_i) = \mu_i + \mu_i^2 / k$. For the purpose of this test, it is probably easier to use the notation $\text{var}(Y_i) = \mu_i + \alpha \times \mu_i^2$ for the NB, where $\alpha = 1/k$, because the null hypothesis (the Poisson variance equals the NB variance) can then be written as $H_0: \alpha = 0$ (note that we are testing on the boundary, but the `lrtest` function corrects for this). The results of this test provide overwhelming evidence to go for a ZINB, instead of a ZIP. The numerical output of the ZINB is obtained with the command `summary(Nb1)` and is as follows.

```
> summary(Nb1)
Call:
zeroinfl(formula = f1, data = ParasiteCod2,
          dist = "negbin", link = "logit")

Count model coefficients (negbin with log link):
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.724165   0.344488 10.811 < 2e-16
fArea2         0.197832   0.329187  0.601  0.54786
fArea3        -0.646241   0.277843 -2.326  0.02002
fArea4         0.709638   0.252319  2.812  0.00492
fYear2000      0.063212   0.295670  0.214  0.83071
fYear2001     -0.940197   0.605908 -1.552  0.12073
Length        -0.036246   0.005109 -7.094 1.3e-12
fArea2:fYear2000 -0.653255   0.535476 -1.220  0.22248
fArea3:fYear2000  1.024753   0.429612  2.385  0.01707
fArea4:fYear2000  0.534372   0.415038  1.288  0.19791
fArea2:fYear2001  0.967809   0.718086  1.348  0.17773
fArea3:fYear2001  1.003671   0.677373  1.482  0.13842
fArea4:fYear2001  0.855233   0.654296  1.307  0.19118
Log(theta)     -0.967198   0.096375 -10.036 < 2e-16

Zero-inflation model coefficients (binomial with logit
link):
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.19106   0.78312  0.244  0.807249
fArea2         2.01576   0.57396  3.512  0.000445
fArea3         1.90753   0.55093  3.462  0.000535
fArea4        -0.73641   0.86427 -0.852  0.394182
fYear2000     -1.07479   2.01183 -0.534  0.593180
fYear2001      3.29534   0.71139  4.632 3.62e-06
Length        -0.03889   0.01206 -3.226  0.001254
fArea2:fYear2000  0.46817   2.09007  0.224  0.822759
fArea3:fYear2000 -0.79393   2.16925 -0.366  0.714369
fArea4:fYear2000 -12.93002  988.60803 -0.013  0.989565
fArea2:fYear2001 -3.20920   0.83696 -3.834  0.000126
fArea3:fYear2001 -3.50640   0.83097 -4.220 2.45e-05
fArea4:fYear2001 -2.91175   1.10650 -2.631  0.008501
```

```
Theta = 0.3801
Number of iterations in BFGS optimization: 52
Log-likelihood: -2450 on 27 Df
```

The z - and p -values of the parameters for the count model (upper part of the output) are rather different, compared to the ZIP! You would expect this as there is overdispersion. The sentence with the BFGS phrase refers to the number of iterations in the optimisation routines.

The question that we should now focus on is which of the explanatory variables can be dropped from the model. The candidates are the Area \times Year interaction term for the count model (most levels have high p -values) and the Area \times Year interaction term for the logistic model (some levels are not significant). In fact, why don't we just drop each term in turn and select the optimal model using the likelihood ratio statistic or AIC. The options are

1. Drop length from the count model. Call this Nb1A.
2. Drop the Area \times Year term from the count model. Call this Nb1B.
3. Drop length from the logistic model. Call this Nb1C.
4. Drop the Area \times Year term from the logistic model. Call this Nb1D.

The models Nb1 (without dropping anything), Nb1A, Nb1B, Nb1C, and Nb1D are given below.

$$\begin{aligned}
 \text{nb1:} \quad \mu_i &= e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}} \\
 \pi_i &= \frac{e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}}{1 + e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}} \\
 \text{nb1A:} \quad \mu_i &= e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year}} \\
 \pi_i &= \frac{e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}}{1 + e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}} \\
 \text{nb1B:} \quad \mu_i &= e^{\text{Area} + \text{Year} + \text{Length}} \\
 \pi_i &= \frac{e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}}{1 + e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}} \\
 \text{nb1C:} \quad \mu_i &= e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}} \\
 \pi_i &= \frac{e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}}}{1 + e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year}}} \\
 \text{nb1D:} \quad \mu_i &= e^{\text{Area} + \text{Year} + \text{Area} \times \text{Year} + \text{Length}} \\
 \pi_i &= \frac{e^{\text{Area} + \text{Year} + \text{Length}}}{1 + e^{\text{Area} + \text{Year} + \text{Length}}}
 \end{aligned}$$

You can implement these models with the code

```
> #Drop Length from count model
> f1A <- formula(Intensity ~ fArea * fYear |
+                   fArea * fYear + Length)
> #Drop interaction from count model
> f1B <- formula(Intensity ~ fArea + fYear +
+                   Length | fArea * fYear + Length)
> #Drop Length from binomial model
> f1C<-formula(Intensity ~ fArea * fYear +
+                   Length | fArea * fYear)
> #Drop interaction from binomial model
> f1D<-formula(Intensity ~ fArea * fYear +
+                   Length | fArea + fYear + Length)
```

```
> Nb1A <- zeroinfl(f1A, dist = "negbin",
+                      link = "logit", data = ParasiteCod2)
> Nb1B <- zeroinfl(f1B, dist = "negbin",
+                      link = "logit", data = ParasiteCod2)
> Nb1C <- zeroinfl(f1C, dist = "negbin",
+                      link = "logit", data = ParasiteCod2)
> Nb1D <- zeroinfl(f1D, dist = "negbin",
+                      link = "logit", data = ParasiteCod2)
```

Just as we did in Chapters 4, 5, and 6, we use the likelihood ratio test to compare each nested model Nb1A, Nb1B, Nb1C, and Nb1D with the full model Nb1, and if a term is not significant, drop the least significant one. The required code is

```
> lrtest(Nb1,Nb1A); lrtest(Nb1,Nb1B)
> lrtest(Nb1,Nb1C); lrtest(Nb1,Nb1D)
```

Table 11.2 shows the results. The AIC values were obtained with the command `AIC(Nb1A, Nb1B, Nb1C, Nb1D)`. The model, in which the Area \times Year interaction was dropped from the count data model gave the lowest AIC and an associated *p*-value of 0.026; so we might as well drop it. These tests are approximate, and therefore, *p* = 0.026 is not convincing. The AICs of the model with and without the Area \times Year interaction are also similar.

This means that we continue with the model selection procedure and test whether Length, Area, or Year can be dropped from the count model and length and the Area \times Year interaction from the logistic model. Results are not shown here, but no further terms could be dropped. This means that we can now say: ‘Thank you for producing the numerical output from the first ZINB model, but it is not the information we need’. The parameters of the optimal model are given by

```
> summary(Nb1B)

Call:
zeroinfl(formula = f1B, data = ParasiteCod2,
          dist = "negbin", link = "logit")

Count model coefficients (negbin with log link):
Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.497280 0.326888 10.699 < 2e-16
fArea2      0.254160 0.229988  1.105  0.26912
fArea3     -0.200901 0.205542 -0.977  0.32836
fArea4      0.912450 0.195039  4.678 2.89e-06
fYear2000    0.462204 0.173067  2.671  0.00757
fYear2001   -0.128524 0.166784 -0.771  0.44094
Length     -0.034828 0.004963 -7.017 2.27e-12
Log(theta)  -0.985648 0.095385 -10.333 < 2e-16
```

Table 11.2 Results of the model selection in ZINB

Dropped term	df	AIC	Likelihood ratio test		
None	27	4954.897			
Length from μ_i	26	4994.993	$X^2 = 42.096$	(df = 1, $p < 0.001$)	
Area \times Year from μ_i	21	4957.146	$X^2 = 14.249$	(df = 6, $p = 0.026$)	
Length from π_i	26	4965.019	$X^2 = 12.122$	(df = 1, $p < 0.001$)	
Area \times Year from π_i	21	4961.751	$X^2 = 18.853$	(df = 6, $p = 0.004$)	

Zero-inflation model coefficients (binomial with logit link) :

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.16057	0.85842	-0.187	0.851617
fArea2	2.18198	0.65106	3.351	0.000804
fArea3	2.23765	0.61803	3.621	0.000294
fArea4	-0.50954	0.90067	-0.566	0.571570
fYear2000	-0.60158	1.55344	-0.387	0.698564
fYear2001	3.71075	0.72278	5.134	2.84e-07
Length	-0.03588	0.01150	-3.121	0.001801
fArea2:fYear2000	0.40925	1.61583	0.253	0.800055
fArea3:fYear2000	-1.81000	1.83708	-0.985	0.324495
fArea4:fYear2000	-10.94642	285.39099	-0.038	0.969404
fArea2:fYear2001	-3.71145	0.84033	-4.417	1.00e-05
fArea3:fYear2001	-3.99409	0.81410	-4.906	9.29e-07
fArea4:fYear2001	-3.37317	1.09981	-3.067	0.002162

Theta = 0.3732

Number of iterations in BFGS optimization: 45

Log-likelihood: -2458 on 21 Df

For publication, you should also give one p -value for the Area and Year terms in the count model, and one p -value for the interaction term in the logistic model. Just drop these terms in turn, use the likelihood ratio test, and quote the Chi-square statistic, degrees of freedom and a p -value. If you are not 100% sure, here are our results for the count model: Length ($X^2 = 41.604$, df = 1, $p < 0.001$), Year ($X^2 = 12.553$, df = 2, $p = 0.002$), Area ($X^2 = 47.599$, df = 3, $p < 0.001$), and for the logistic model: length ($X^2 = 10.155$, df = 1, $p = 0.001$) and the Area \times Year interaction ($X^2 = 47.286$, df = 6, $p < 0.001$).

This was the model selection. There are two more things we need to do; model validation and model interpretation of the optimal ZINB model.

11.4.2.1 Model Validation

The keyword is again residuals. You need to plot Pearson residuals against the fitted values and Pearson residuals against each explanatory variable and you should

not see any pattern. It is also useful to plot the original data versus the fitted data; hopefully, they form a straight line.

If you fit a ZIP model with the function `zeroinfl`, Pearson residuals for the count data can be obtained by the R command:

```
> EP <- residuals(Zip1, type = "pearson") .
```

There are multiple packages for zero-inflated data, and it is not always clear how exactly residuals are calculated. Because we believe in ‘know what you are doing’, we show you how to get the Pearson residuals using the equations we derived in the previous subsection.

Let us extract the probabilities π_i , the probability of a false zero. They are obtained by

```
> EstPar <- coef(Nb1B, model = "zero")
> Z <- model.matrix(Nb1B, model = "zero")
> g <- Z %*% EstPar
> p <- exp(g) / (1 + exp(g))
```

The p in the code is π_i . The `coeff` command with the option `model = "zero"` gives the estimated parameters presented above (`Nb1B` is our optimal ZINB model). The μ_i from Equation (11.18) is obtained by

```
> EstPar2 <- coef(Nb1B, model = "count")
> X <- model.matrix(Nb1B, model = "count")
> g <- X %*% EstPar2
> mu1 <- exp(g)
```

Using Equation (11.23), the expected values of a ZINB model are given by

```
> mu <- (1 - p) * mu1
```

If you compare this result with the results of `fitted(Nb1B)` or `predict(Nb1B)`, you should get the same values. Finally, we show how to get the variance and Pearson residuals:

```
> k <- Nb1B$theta
> VarY <- ((mu^2) / k + mu) * (1 - p) +
  (mu^2) * (p^2 + p)
> EP <- (ParasiteCod2$Intensity - mu) / sqrt(VarY)
```

These should give the same results as the `residuals` command; but it is good to know that we can do it ourselves! The rest is a matter of plotting these residuals against everything we have and hope that there are no clear patterns. We do not show these graphs here.

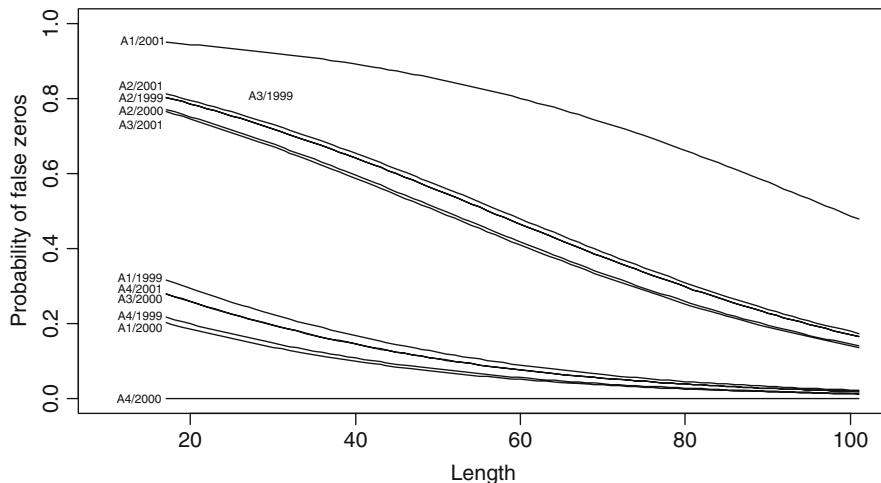


Fig. 11.6 Fitted curves for the logistic regression model. The vertical axis shows the probability of measuring a false zero, and the horizontal axis length of cod. Each line corresponds to an area and year combination

11.4.2.2 Model Interpretation

The question we now focus on is: What does it all mean? To answer this question, we sketch the results of the model. There are two components to plot; the logistic model for the false zeros versus all other data, and the count model versus all other data. We first focus on the logistic regression part. Fitted values can be obtained by the `predict` function, or you can do it manually (which is what we did). We took the estimated intercepts and slopes from the zero-inflated part of the optimal ZINB model (`Nb1B`), created length values from 17 to 100 cm, and calculated the fitted values for each area and year combination. This is a straightforward exercise and was explained in Chapter 10. The results are given in Fig. 11.6. It seems that the highest probabilities of false zeros are obtained for small fish in area 1 in 2001, in area 2 in all years, and in area 3 in 1999 and 2001. Explained differently, in these areas and these years, you are likely to catch small cod with zero parasites, but these zeros are false zeros.

A similar graph was drawn for the count data. In this case, fitted values are obtained from Equation (11.23). Regression coefficients were taken from the upper part of the `summary(Nb1B)` output. Area 4 in 1999 and 2000 has the highest values. This information can also be derived from the estimated regression parameters; so the need for a graph is limited.

11.5 ZAP and ZANB Models, Alias Hurdle Models

In the previous section, we assumed the zeros for the cod data consist of false zeros and true zeros. In this section, we do not discriminate between the four types of zeros; they are just zeros.

We follow the same approach as in Section 11.2; first we present the probability functions for the two-part models and give the maximum likelihood equations in Section 11.5.1, and an example plus the R code is presented in Section 11.5.2. If you are not interested in the underlying mathematics, just read the summary at the end of Section 11.5.1.

11.5.1 Mathematics of the ZAP and ZANB

In the hurdle model (ZAP and ZANB), we consider the data on a presence and absence level and analyse the presence data with a count model. Actually, if you apply two analyses, one binomial GLM and one Poisson (or NB) GLM, you will get the same estimated regression parameters.

A small difference is that with ZIP and ZINB, the binomial GLM models the probability of a false zero versus other types of data, whereas in ZAP and ZANB, the binomial GLM models the probability of presence versus absence. Hence, the estimated regression parameters obtained by ZAP and ZANB should have opposite signs compared to those obtained by ZIP and ZINB due to the definition of π .

The underlying idea for the hurdle model is that there are two ecological processes playing a role. In the context of the hippo example, one process is causing the absence of hippos, and at those sites where hippos are present, there is a second process influencing the number of hippos. The probability function for a hurdle model is build up accordingly. The binomial distribution is used to model the absence and presence of hippos, and a Poisson (or negative binomial or geometric) distribution for the counts. This leads to the following probability function for the Poisson ZAP:

$$f_{\text{ZAP}}(y; \beta, \gamma) = \begin{cases} f_{\text{binomial}}(y = 0; \gamma) & y = 0 \\ (1 - f_{\text{binomial}}(y = 0; \gamma)) \times \frac{f_{\text{Poisson}}(y; \beta)}{1 - f_{\text{Poisson}}(y = 0; \beta)} & y > 0 \end{cases} \quad (11.24)$$

So, the probability of measuring zero hippos is modelled with a binomial distribution, where π_i is the probability that $y_i = 0$. Hence, $1 - \pi_i$ is the probability that we do not measure zero hippos. Just as for the ZIP model, π_i is modelled in terms of covariates Z and regression parameters γ ; see also Equation (11.20). To measure a non-zero count, the ecosystem needs to cross a hurdle to produce a non-zero value and the Poisson count process has to exclude the probability of zero values, which we called a zero-truncated Poisson distribution in Section 11.2. So, the second part in the above equation says that the probability of measuring a non-zero value equals the probability that it is not a zero multiplied with the probability determined by a zero-truncated Poisson. The mean of the Poisson distribution, μ_i , is modelled in terms of covariates X and regression parameters β ; see also Equation (11.18).

The next task is to find the optimal regression parameters γ and β . As with the ZIP, a likelihood criterion is formulated using the probability function in Equation (11.24). Finding the regression parameters that optimise the log-likelihood is a matter of numerical optimisation, and the required formulae can be found in

Section 4.7.1 in Cameron and Trivedi (1998). The function `hurdle` from the `pscl` package in R will do the hard work for you.

The difference between a ZAP and a ZANB is due to the assumption for the distribution of the count data. If we assume a Poisson distribution, we end up with a ZAP and if a negative binomial distribution is used, we get a ZANB. Justification for the ZANB is extra overdispersion in the count data.

In Equations (11.22) and (11.23), we formulated the mean and variance for the ZIP and ZINB. For the ZAP, these are as follows.

$$E_{\text{ZAP}}(Y_i; \pi_i, \mu_i) = \frac{1 - \pi_i}{1 - e^{-\mu_i}} \times \mu_i$$

$$\text{Var}_{\text{ZAP}}(Y_i; \pi_i, \mu_i) = \frac{1 - \pi_i}{1 - e^{-\mu_i}} \times (\mu_i + \mu_i^2) - \left(\frac{1 - \pi_i}{1 - e^{-\mu_i}} \times \mu_i \right)^2$$

And for the ZANB, we have

$$E_{\text{ZANB}}(Y_i; \pi_i, \mu_i, k) = \frac{1 - \pi_i}{1 - P_0} \times \mu_i \quad \text{where } P_0 = \left(\frac{k}{\mu_i + k} \right)^k$$

$$\text{Var}_{\text{ZANB}}(Y_i; \pi_i, \mu_i, k) = \frac{1 - \pi_i}{1 - P_0} \times \left(\mu_i^2 + \mu_i + \frac{\mu_i^2}{k} \right) - \left(\frac{1 - \pi_i}{1 - P_0} \times \mu_i \right)^2$$

The mean and variance can be used to calculate the Pearson residuals.

11.5.2 Example of ZAP and ZANB

The whole modelling process in two-part models is identical compared to mixture models. First you need to decide whether you need a ZAP or ZANB. The best option is to run them both and compare them with a likelihood ratio test. This can be done with the following R code.

```
> H1A <- hurdle(f1, dist = "poisson", link = "logit",
+                 data = ParasiteCod2)
> H1B <- hurdle(f1, dist = "negbin", link = "logit",
+                 data = ParasiteCod2)
```

The command `lrtest(H1A, H1B)` produces a Chi-square statistic of 8752.50 (which is overwhelming evidence in favour of the negative binomial model) and the command `AIC(H1A, H1B)`, gives an AIC of 13687.59 for the ZAP and 4939.08 for the ZANB, confirming the choice for the ZANB. The `summary(H1B)` gives the estimated parameters, but because the model has various nominal variables with multiple levels, it is better to compare the full model `H1B`, with models in which a particular term is dropped, and use the `lrtest` command to get one *p*-value for the interaction term in the count model and in the binomial model. R code for these analyses were provided in Section 11.4 and are not repeated here (the code

can also be found on the book's website). In the first round of model simplification, length was dropped from the binomial model, and in the second (and last) round, the Area \times Year interaction term was dropped from the Poisson model. The code and estimated regression parameters for the optimal ZANB model are as follows.

```
> fFinal <- formula(Intensity ~ fArea + fYear +
+ Length | fArea*fYear )
> HFinal <- hurdle(f1, dist = "negbin", link = "logit",
+ data = ParasiteCod2)
> summary(HFinal)
```

Call:

```
hurdle(formula = f1, data = ParasiteCod2,
       dist = "negbin", link = "logit")
```

Count model coefficients (truncated negbin with log link):

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.366059	0.399420	8.427	< 2e-16
fArea2	0.379211	0.380945	0.995	0.31952
fArea3	-0.504376	0.312256	-1.615	0.10625
fArea4	0.893614	0.291517	3.065	0.00217
fYear2000	-0.040511	0.328434	-0.123	0.90183
fYear2001	-0.757718	0.688097	-1.101	0.27082
Length	-0.037309	0.005867	-6.359	2.03e-10
fArea2:fYear2000	-0.639059	0.616450	-1.037	0.29989
fArea3:fYear2000	1.193440	0.494530	2.413	0.01581
fArea4:fYear2000	0.510433	0.476990	1.070	0.28457
fArea2:fYear2001	0.707730	0.819333	0.864	0.38770
fArea3:fYear2001	0.912374	0.775776	1.176	0.23956
fArea4:fYear2001	0.601263	0.746292	0.806	0.42043
Log(theta)	-1.498146	0.239114	-6.265	3.72e-10

Zero hurdle model coefficients (binomial with logit link):

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.085255	0.295071	0.289	0.7726
fArea2	-1.321373	0.285258	-4.632	3.62e-06
fArea3	-1.449183	0.243885	-5.942	2.81e-09
fArea4	0.300728	0.271105	1.109	0.2673
fYear2000	0.395069	0.343817	1.149	0.2505
fYear2001	-2.652010	0.433404	-6.119	9.42e-10
Length	0.006933	0.004655	1.489	0.1364
fArea2:fYear2000	-0.080344	0.507970	-0.158	0.8743
fArea3:fYear2000	0.870585	0.450277	1.933	0.0532
fArea4:fYear2000	0.864622	0.592387	1.460	0.1444

```
fArea2:fYear2001 2.737488 0.532903 5.137 2.79e-07
fArea3:fYear2001 2.718986 0.499487 5.444 5.22e-08
fArea4:fYear2001 2.541437 0.518245 4.904 9.39e-07
Theta: count = 0.2235
Number of iterations in BFGS optimization: 29
Log-likelihood: -2442 on 28 Df
```

The difference between the optimal ZINB and ZANB is that length is not significant in the binomial part of the ZANB. For the rest, both models are the same in terms of selected explanatory variables.

It is also interesting to compare the estimated parameters of the optimal ZINB and ZANB models. For the count part of the model, the sign and magnitude of the significant parameters are very similar. Plotting the fitted values as in Fig. 11.7 gives a similar graph. Hence, the biological conclusions for the count part are similar. For the binomial part of the model, things look different, at least in the first instance. However, the p -values of corresponding terms in both tables give the same message. The magnitudes of the significant parameters are similar as well. It is only the sign of the regression parameters that are different. But this is due to the opposite definition of the π s in both methods!

In summary, for the cod parasite data, the ZINB and ZANB give similar parameter estimates. The difference is how they treat the zeros. The ZINB labels the excessive number of zeros (which occur at small fish and in certain areas in particular years) as false zeros, whereas the ZANB models the zeros versus the non-zeros (and identifies the area \times year interaction as a driving factor for this), and the non-zeros with a truncated NB GLM jointly.

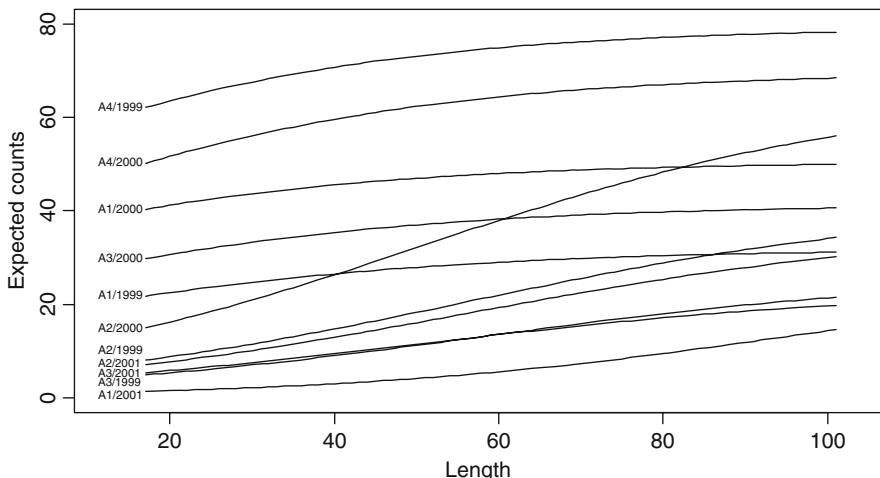


Fig. 11.7 Fitted curves for the count model. The vertical axis shows the expected counts (assuming a ZINB distribution) and the horizontal axis length of cod. Each line corresponds to an area and year combination

11.6 Comparing Poisson, Quasi-Poisson, NB, ZIP, ZINB, ZAP and ZANB GLMs

In the previous sections and chapters, we applied Poisson, quasi-Poisson, NB GLM, ZIP, ZINB, ZAP, and ZANB models on the cod parasite data. The question is now: What is the best model? There are many ways to answer this question.

Option 1: Common Sense

The first option is common sense. First, you should decide whether there is overdispersion. If there is no overdispersion, you are lucky and you can stick to the Poisson GLM. If there is overdispersion, ask yourself why you have overdispersion; outliers, missing covariates, or interactions are the first things you should consider. Small amounts of overdispersion can be modelled with quasi-Poisson. Let us assume that this is not the case. Do you have overdispersion due to excessive number of zeros or due more to variation in the count data? Make a frequency plot of the data and you will know whether it is zero inflation. If there is zero inflation, go the route of ZIP, ZAP, ZINB, and ZANB. If the overdispersion is not caused by excessive number of zeros, but due to more variation than expected by the Poisson distribution in the positive part of the count data, use the negative binomial distribution. In case of zero inflation *and* extra variation in the positive count data, use ZINB or ZANB. The choice between ZINB and ZANB (or ZIP and ZAP) should be based on *a priori* knowledge of the cause of your excessive number of zeros.

Option 2: Model Validation

A second option to help decide on the best model (if there is such a thing) is to plot the residuals of each model and see whether there are any residual patterns. Drop each model that shows patterns in the residuals.

Option 3: Information Criteria

Another option is to apply all methods and print all estimated parameters, standard errors and AICs, BICs, etc. in one big table. Compare them, and based on the AICs, judge which one is the best. You can find examples of this approach in most books discussing these statistical methods.

Option 4: Hypothesis Tests – Poisson Versus NB

Formal hypotheses tests can be used to choose between Poisson and negative binomial models as these are nested. This also holds for ZIP versus ZINB and ZAP versus ZANB.

Option 5: Compare Observed and Fitted Values

Potts and Elith (2006) compared the fitted and observed values of all the models. To assess how good each technique predicts the fitted values, they used various tools. For example, high values of the Pearson correlation coefficient and Spearman's rank correlation between fitted and observed values mean that these are similar.

It is also possible to apply a linear regression model of the form $\text{Observed}_i = \alpha + \beta \times \text{Fitted}_i + \varepsilon_i$, where Observed_i are the observed data and Fitted_i the fitted values of a particular method. An estimated intercept of 0 and slope of 1 indicates a perfect fit. Potts and Elith (2006) discuss the interpretation of non-significant slopes.

Other ways to quantify how similar the observed and fitted values are the root mean square errors and mean absolute error (where error is defined as the difference between the observed and fitted values).

All these statistics are discussed in Potts and Elith (2006) and require bootstrapping. We implemented their algorithm, and the results are presented in Table 11.3. Note that the Pearson correlation coefficients and the Spearman rank correlations of all five methods are nearly identical. The ZANB is the only model that gives an intercept of 0. The AIC of this model is also the lowest, and therefore based on these numerical tools, the ZANB can be selected as the best possible model.

Another approach to compare (and select) models is discussed in Ver Hoef and Boveng (2007), who plotted the variance versus the mean and the weights that are used inside the algorithm versus the mean.

Instead of sticking to one of these five methods, you may need multiple approaches to arrive at the best model. The hypothesis testing approach showed that an NB model is preferred above the Poisson GLM. A frequency plot indicated zero inflation; hence, we should apply a ZINB or ZANB. A discussion with one of the involved researchers revealed that we have both false and true zeros. We can either try to determine the contribution from each of these (with a ZINB) or just consider them as zeros and use the ZANB. So, the choice between the ZINB and ZANB depends on the underlying questions with regards to the zeros. If you close your eyes and compare the ZINB and ZANB, then the latter should be selected as judged by the AIC.

Table 11.3 Model comparison tools for the Poisson GLM, quasi-Poisson GLM, NB GLM, ZINB, and ZANB models. The Pearson correlation coefficient (r), Spearman rank correlation (p), intercept and slope (of a linear regression of observed versus fitted), RMSE, MAE (mean absolute error), AIC, log likelihood and degrees of freedom (df).

Model	r	p	Intercept	Slope	RMSE	MAE	AIC	Log lik	Df
Poisson	0.33	0.36	0.32	0.96	18.60	7.50	20377.86	-10175.93	13
Quasi-Poisson	0.33	0.36	0.32	0.96	18.63	7.50	NA	NA	13
NB GLM	0.34	0.37	-0.20	1.07	18.49	7.42	5030.67	-2501.33	14
ZINB	0.33	0.37	0.30	0.96	18.57	7.49	4954.89	-2450.44	27
ZANB	0.34	0.37	-0.06	1.04	18.47	7.47	4937.08	-2441.54	27

11.7 Flowchart and Where to Go from Here

In this chapter, we have discussed tools to analyse zero-inflated models, resulting in four extra models (ZIP, ZAP, ZINB and ZANB) in our toolbox for the analysis of count data. Mixture models and two-part models should be part of every ecologist's toolbox as zero inflation and overdispersion are commonly encountered in ecological data. If you are now confused with the large number of models to analyse count data, Fig. 11.8 will help you to visualise the difference between some of the models discussed in Chapters 9, 10, and 11.

So, where do we go from here? In Chapters 12 and 13, we concentrated on models that allow for correlation and random effects in Poisson and binomial GLMs and GAMs. These models are called generalised estimation equations (GEE), generalised linear mixed modelling (GLMM), and generalised additive mixed modelling (GAMM). At the time of writing this book, software for GEE, GLMM, and GAMM for zero-inflated data consists mainly of research or publication specific code. By this, we mean that papers using random effects or spatial and temporal correlations structures in combination with zero inflation are now being published (e.g. Ver Hoef and Jansen, 2007), but general software code is not yet available. So, a bit of challenging R programming awaits you, should you want to model zero-inflated GLMMs.

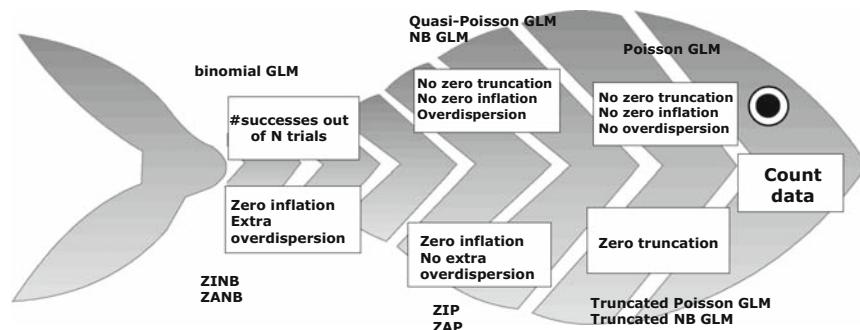


Fig. 11.8 GLMs for count data. Instead of GLM, you can also use GAM. Try sketching in the R functions for each box. If there is no zero truncation, no zero inflation and no overdispersion (*upper right box*), you can apply a Poisson GLM. If there is overdispersion (*upper middle box*), then consider quasi-Poisson or negative binomial GLM. The '#successes out of N trials' box refers to a logistic regression. The trials need to be independent and identical. For zero-truncated data (*lower right box*), you need to apply a zero-truncated Poisson GLM or a zero-truncated negative binomial GLM. If there is zero inflation, you are in the world of ZIP, ZAP, ZINB, and ZINB models. The difference between the P and NB is whether there is overdispersion in the non-zero data. It is a nice exercise to add the names of the corresponding R functions! You can also use the offset in the ZIP, ZAP, ZINB, and ZINB models

Chapter 12

Generalised Estimation Equations

In this chapter, we analyse three data sets; California birds, owls, and deer. In the first data set, the response variable is the number of birds measured repeatedly over time at two-weekly intervals at the same locations. In the owl data set (Chapter 5), the response variable is the number of calls made by all offspring in the absence of the parent. We have multiple observations from the same nest, and 27 nests were sampled. In the deer data, the response variable is the presence or absence of parasites in a deer; the data are from multiple farms.

In the first instance, we apply a generalised linear model (GLM) with a Poisson distribution for the California birds and owl data and a binomial GLM for the deer data. However, such analyses violate the independence assumption; for the California bird data, there is a longitudinal aspect, we have multiple observations per nest for the owls, and multiple deer from the same farm. We therefore introduce generalised estimation equations (GEE) as a tool to include a dependence structure, discuss its underlying mathematics, and apply it on the same data sets.

GEE was introduced by Liang and Zeger (1986), and since their publication, several approaches have been developed to improve the technique. We use the original method as it is the simplest. Useful GEE references are Ziegler et al. (1996), Greene (1997), Fitzmaurice et al. (2004), and a textbook completely dedicated to GEE by Hardin and Hilbe (2002). This chapter heavily depends on the Fitzmaurice et al. (2004) book. Chapter 22 contains a binary GEE case study.

12.1 GLM: Ignoring the Dependence Structure

12.1.1 *The California Bird Data*

Elphick and Oring (1998, 2003) and Elphick et al. (2007) analysed time series of several water bird species recorded in California rice fields. Their main goals were to determine whether flooding fields after harvesting results in greater use by aquatic birds, whether different methods of manipulating the straw in conjunction with flooding influences how many fields are used, and whether the depth that the

fields are flooded to is important. Biological details can be found in the references mentioned above.

Counts were made during winter surveys at several fields. Here, we only use data measured from one winter (1993–1994), and we use species richness to summarise the 49 bird species recorded. The sampling took place at multiple sites, and from each site, multiple fields were repeatedly sampled. Here, we only use one site (called 4mile) for illustrative purposes. There are 11 fields in this site, and each field was repeatedly sampled; see Fig. 12.1. Note that there is a general decline in bird numbers over time. One of the available covariates is water depth per field, but water depth and time are collinear (as can be inferred from making an *xyp*lot of depth versus time for each field), so we avoid using them together as covariates in the models.

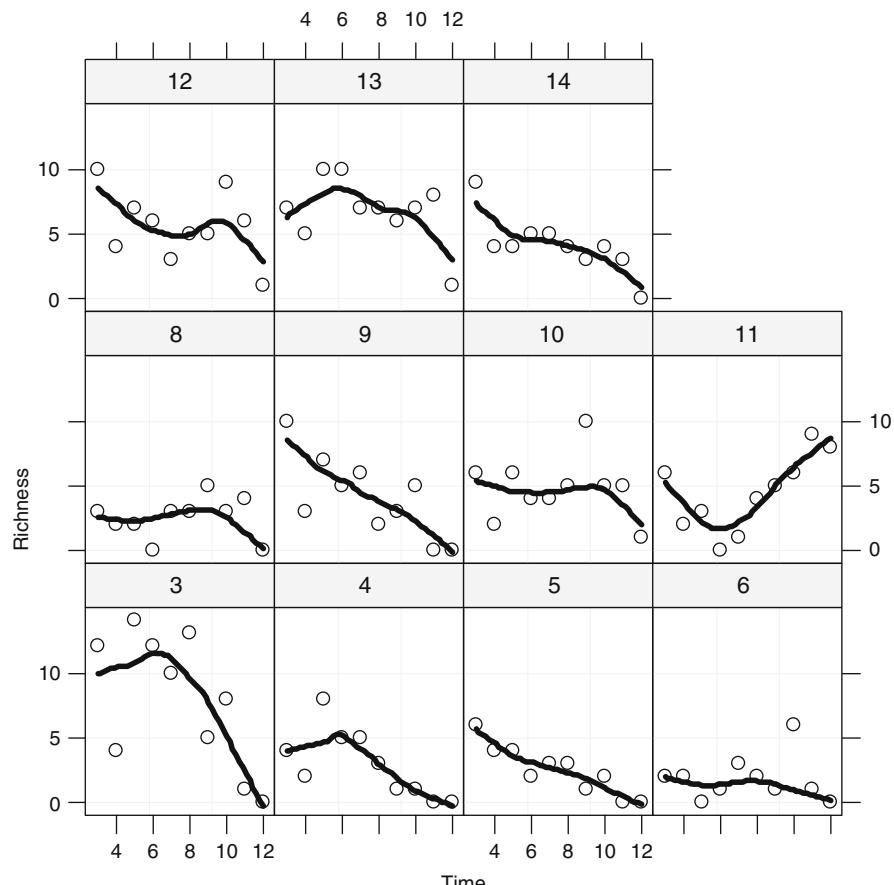


Fig. 12.1 *xyp*lot of species richness plotted against time (expressed in two-weekly periods). Each panel represents a different field. A LOESS smoother was added to aid visual interpretation of the graph

The following R code reads the data, calculates the richness index, and makes the `xyplot` in Fig. 12.1.

```
> library(AED); data(RiceFieldBirds)
> RFBirds <- RiceFieldBirds           #Saves some space
> RFBirds$Richness <- rowSums(RFBirds[, 8:56] > 0)
> RBirds$fField <- factor(RFBirds$FIELD)
> library(lattice)
> xyplot(Richness ~ Time | fField, data= RFBirds,
  panel=function(x, y) {
    panel.grid(h = -1, v = 2)
    panel.points(x, y, col = 1)
    panel.loess(x, y, col = 1, lwd = 2)})
```

The first few lines access the data and the object with the data is renamed into a shorter name. The `rowSums` command is used to calculate species richness (add `na.rm = TRUE` if you have missing values in your data), and the rest is a matter of some simple `xyplot` commands and options to get points and smoothers in the panels (see also Chapter 2). As always in R, things can be done in at least five different ways. Instead of the code in the `panel` function, you can also use:

```
> xyplot(Richness ~ Time | fField, data= RFBirds,
  type = c ("p" , "smooth" , "grid"))
```

It gives the same graph, but the code looks a bit more cryptic. Additional parameters like the span width and line thickness for the smoother can also be specified (just add `span = 0.5, lwd = 2, col = 1` to the command above).

Counts took place approximately every two weeks. As well as species richness, we also have water depth and information on rice debris management. The aim of the analysis presented here is to explain the richness values as a function of depth and management effects. The response variable is a count, and therefore we are in the world of GLMs with a likely candidate model the GLM with a Poisson distribution and log-link function. Actually, it is a bit more complicated as the original data were densities; numbers per field and the sizes of the fields are different. This means that (the log of the) size of the field can be used as an offset variable (Chapter 9). Based on biological knowledge, and an initial analysis using generalised additive modelling (Elphick et al., 2007), the effect of the covariate depth is modelled as a quadratic term. The following three steps define the GLM.

1. Define Y_{is} as the richness measured in field i at time s . We assume that Y_{is} is Poisson distributed with mean μ_{is} . In mathematical notation, we have: $Y_{is} \sim P(\mu_{is})$. Recall that for a Poisson distribution, the mean is the variance.

2. The systematic part of the GLM is given by

$$\eta(\text{Depth}_{is}, \text{SPTREAT}_{is}, \text{AREA}_{is}) = \alpha + \text{offset}(\log(\text{AREA}_{is})) + \\ \beta_1 \times \text{Depth}_{is} + \beta_2 \times \text{Depth}_{is}^2 + \beta_3 \times \text{SPTREAT}_{is}$$

The term SPTREAT is the categorical variable defining management type. It is also possible to include an interaction between depth and the management type and also between the quadratic function of depth and management type. But to keep the models simple, we do not do this.

3. The link between the expected values and systematic component is the log-link:

$$\log(\mu_{is}) = \eta(\text{Depth}_{is}, \text{SPTREAT}_{is}, \text{AREA}_{is})$$

Full details of Poisson GLMs are given in Chapter 9. It is important to realise that the GLM assumes independence of all richness values, including those from the same field (which are separated by only two weeks). For the moment, we will ignore this problem and just carry on with the GLM. Later in this chapter, we will apply GEE to incorporate auto-correlation on the data from the same field and compare results. An initial GLM indicated overdispersion, and we therefore applied a quasi-Poisson GLM with the following R code. Results from the `summary` command are given as well and we will compare them later with the GEE results.

```
> RFBirds$LA <- log(RFBirds$AREA)
> RFBirds$fSptreat <- factor(RFBirds$SPTREAT)
> RFBirds$DEPTH2 <- RFBirds$DEPTH^2
> M0 <- glm(Richness ~ offset(LA) + fSptreat + DEPTH +
             DEPTH2, family = quasipoisson, data = RFBirds)
> summary(M0)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.7911754	0.2136575	-3.703	0.00034
fSptreatrlfld	-0.4931558	0.1666480	-2.959	0.00380
DEPTH	0.0690528	0.0249844	2.764	0.00674
DEPTH2	-0.0016531	0.0006732	-2.455	0.01569

Dispersion parameter for quasipoisson family taken to
be 2.392596

Null deviance: 297.47 on 109 degrees of freedom

Residual deviance: 245.10 on 106 degrees of freedom

AIC: NA

Note that the overdispersion is 2.39. All terms in the model are significant at the 5% level, although the quadratic depth term is only weakly significant with a *p*-value of 0.015.

12.1.2 The Owl Data

In Chapters 5 and 6, we analysed data from a study on vocal begging behaviour when the owl parents bring prey to their nest. In both chapters, we used sibling negotiation as response variable. It was defined as the number of calls made by all offspring in the absence of the parents counted during 30-second time intervals before arrival of a parent divided by the number of nestlings. Just as in the previous section, we can use the (natural) logarithm of the number of nestlings as an offset variable and analyse the number of calls NCalls_{is} at time s in nest i using a Poisson GLM. Hence, we assume that $\text{NCalls}_{is} \sim P(\mu_{is})$, and therefore the mean and variance of NCalls_{is} are equal to μ_{is} . The systematic part is given by

$$\begin{aligned}\eta_{is} = & \alpha + \log(\text{Broodsize}_i) + \beta_1 \times \text{SexParent}_{is} + \beta_2 \times \text{FoodTreatment}_{ij} \\ & + \beta_3 \times \text{ArrivalTime}_{ij} + \beta_4 \times \text{SexParent}_{is} \times \text{FoodTreatment}_{ij} \\ & + \beta_5 \times \text{SexParent}_{is} \times \text{ArrivalTime}_{ij}\end{aligned}$$

Recall from Chapter 5 that the sex of the parent is male or female, food treatment at a nest is deprived or satiated, and arrival time of the parent at the nest was coded with values from 21 (9.00 PM) to 30 (6.00 AM). Note that there is no regression parameter in front of the $\log(\text{Broodsize}_i)$ term; it is modelled as an offset variable. The link between the expected value of Y_{is} , μ_{is} , and the systematic component η_{is} is the log-link:

$$\log(\mu_{is}) = \eta_{is} \Leftrightarrow \mu_{is} = e^{\eta_{is}}$$

The model is fitted with the following R code.

```
> library(AED) ; data(Owls)
> Owls$NCalls <- Owls$SiblingNegotiation
> Owls$LBroodSize <- log(Owls$BroodSize)
> Form <- formula(NCalls ~ offset(LBroodSize) +
+                   SexParent * FoodTreatment +
+                   SexParent * ArrivalTime)
> O1 <- glm(Form, family = poisson, data = Owls)
```

Instead of the name `SiblingNegotiation`, we used the shorter name `NCalls` as it saves some space in the code. The results of the `summary(O1)` command are not shown here, but there is overdispersion. Therefore, we refitted the model with a quasi-Poisson GLM:

```
> O2 <- glm(Form, family = quasipoisson, data = Owls)
> drop1(O2, test = "F")
```

Results of the `drop1` command are not presented here, but indicate that the two two-way interactions are not significant. Using a backwards selection, we ended up with the model containing food treatment and arrival time:

```

> Form <- formula(NCalls ~ offset(LBroodSize) +
+ FoodTreatment + ArrivalTime)
> O3 <- glm(Form, family = quasipoisson, data = Owls)
> summary(O3)

Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.81333   0.53946  7.069 4.39e-12
FoodTreatmentSatiated -0.53230   0.08260 -6.444 2.40e-10
ArrivalTime   -0.12924   0.02205 -5.861 7.60e-09

Dispersion parameter for quasipoisson family taken to be 6.246006
Null deviance: 4128.3 on 598 degrees of freedom
Residual deviance: 3652.6 on 596 degrees of freedom
AIC: NA

```

All regression parameters are highly significant. We will return to these results once the GEE has been discussed.

12.1.3 The Deer Data

Vicente et al. (2006) looked at the distribution and faecal shedding patterns of the first-stage larvae (L1) of *Elaphostrongylus cervi* (*Nematoda: Protostrongylidae*) in red deer across Spain. Effects of environmental variables on *E. cervi* L1 counts were determined using generalised linear mixed modelling (GLMM) techniques. Full details on these data can be found in their paper. In this book, we use only part of their data to illustrate GEE and GLMM (Chapter 13).

In this section, we keep the analysis simple and focus on the relationship between the presence and absence of *E. cervi* L1 in deer and the explanatory variables length and sex of the host. Because the response variable is of the form 0–1, we are immediately in the world of a binomial GLM. The explanatory variables are length and sex of the deer, the first is continuous and sex is nominal. The following three steps define the GLM.

1. Define Y_{is} as 1 if the parasite *E. cervi* L1 is found in animal j at farm i , and 0 otherwise. We assume that Y_{is} is binomially distributed with probability p_{is} . In mathematical notation, we have: $Y_{is} \sim B(1, p_{is})$. Recall that for a binomial distribution, we have $E(Y_{is}) = p_{is}$ and $\text{var}(Y_{is}) = p_{is} \times (1 - p_{is})$.

2. The systematic part of the GLM is given by:

$$\eta(\text{Length}_{is}, \text{Sex}_{is}) = \alpha + \beta_1 \times \text{Length}_{is} + \beta_2 \times \text{Sex}_{is} + \beta_3 \times \text{Length}_{is} \times \text{Sex}_{is}$$

3. The link between the expected values and systematic component is the logistic link:

$$\text{logit}(p_{is}) = \eta(\text{Length}_{is}, \text{Sex}_{is}) \Leftrightarrow$$

$$p_{is} = \frac{e^{\alpha + \beta_1 \times \text{Length}_{is} + \beta_2 \times \text{Sex}_{is} + \beta_3 \times \text{Length}_{is} \times \text{Sex}_{is}}}{1 + e^{\alpha + \beta_1 \times \text{Length}_{is} + \beta_2 \times \text{Sex}_{is} + \beta_3 \times \text{Length}_{is} \times \text{Sex}_{is}}}$$

The notation logit stands for the logistic link (Chapter 10), and p_{ij} is the probability that animal j on farm i has the parasite, Length_{ij} is the length of the deer, and Sex_{ij} tells us whether it is male or female. Instead of the subscripts i and j , we could have used one index k identifying the animal. However, with respect to the methods that are to come, it is more useful to use indices i and j .

The following code accesses the data from our AED package, defines Sex as a nominal variable, and converts the *E. cervi* count data into presence and absence.¹

```
> library(AED); data(DeerEcervi)
> DeerEcervi$Ecervi.01 <- DeerEcervi$Ecervi
> DeerEcervi$Ecervi.01[DeerEcervi$Ecervi > 0] <- 1
> DeerEcervi$fSex <- factor(DeerEcervi$Sex)
> DeerEcervi$CLength <- DeerEcervi$Length -
  mean(DeerEcervi$Length)
```

Note that we centred length. If you do not centre the length, the intercept represents the probability that a deer of length 0 has the parasite. This of course is nonsense as there cannot be any deer of length 0. By centring length, the intercept has the more meaningful interpretation of the probability that an animal of average length has the parasite. The code below applies a GLM on the selected data, drops each allowable term in turn, from the model, and applies a likelihood ratio test that is Chi-square distributed. Note that because the interaction between length and sex is included, we cannot drop the main terms Length and Sex.

```
< DE.glm<-glm(Ecervi.01 ~ Length * fSex,
                 data = DeerEcervi, family = binomial)
> drop1 (DE.glm, test = "Chi")

Single term deletions. Model: Ecervi.01 ~ CLength*fSex
              Df Deviance    AIC    LRT Pr(Chi)
<none>           1003.7 1011.7
CLength:fSex   1    1008.1 1014.1     4.4   0.036

> summary(DE.glm)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.652409  0.109602  5.953 2.64e-09
CLength     0.025112  0.005576  4.504 6.68e-06
```

¹The motivation for this is purely pedagogical; we want to present three GEE examples, one of which is a binomial GEE.

```
fSex2          0.163873   0.174235   0.941    0.3469
CLength:fSex2 0.020109   0.009722   2.068    0.0386

Dispersion parameter for binomial family taken to be 1
Null deviance: 1073.1 on 825 degrees of freedom
Residual deviance: 1003.7 on 822 degrees of freedom
AIC: 1011.7
```

The output from a `drop1` function was discussed in Chapter 10. Recall that it compares the deviance of the specified model with that of nested models. The difference between these two deviances is Chi-square distributed. The Length–Sex interaction term is significant at the 5% level. We will return to the numerical output once the GEE has been discussed.

The problem with this analysis is that the data were obtained from 24 farms. This means that we are sampling deer that may have been in contact with each other, and we can therefore not assume that the presence or absence of parasites on deer from the same farm are independent.

12.2 Specifying the GEE

12.2.1 Introduction

The GLMs presented in the previous section are potentially flawed because the data are longitudinal (California birds) or we have repeated measurements from the same nest (owls) or farm (deer). Hence, the assumption of independence is invalid. We could just ignore the potential existence of dependence and present the analyses of the data obtained by GLM, but this will tend to increase the risk of a Type I error, particularly where within-subject (auto-) correlation is strong.

In Chapters 8, 9, and 10, we have seen how GLM gives us a framework for analysing response data whose inherent stochasticity can be modelled with any one of a number of probability distributions belonging to the exponential family, i.e. can be expressed in the form

$$\exp \left\{ \frac{Y \times \theta - b(\theta)}{a(\phi)} - c(Y, \phi) \right\}$$

where Y is the response variable. Additionally, in Chapters 5, 6, and 7, we looked at ways to model within-subject correlation, and incorporate it into the analysis through, for example, mixed modelling. Liang and Zeger (1986) set out to establish an algorithm that combined these two methodologies.

The modelling of correlation structures is relatively easily managed with normally distributed response data. Although the mathematics appear involved to the non-technical reader, the mechanics of optimisation are computationally trivial, particularly with powerful modern computers. However, the complications

multiply when we are modelling auto-correlation where the data are clearly non-normal and cannot be transformed.

Although this normally means binary (presence/absence), proportional, and count data, the same general arguments can be applied to any response that can be modelled using GLMs, e.g. overdispersed count data using a negative binomial type variance structure. We specifically write ‘negative binomial type’ as we are not going to make any distributional assumptions in the GEE.

Although not yet discussed, we can use generalised linear mixed modelling (Chapter 13) to account for within-subject ‘compound-symmetry’ type correlation. This is the simplest mixed model structure where all we saying is that all observations from a given source (subject) are correlated. No allowance can be made within this procedure for correlation patterns between the observations from the same source, e.g. temporal auto-correlation. One important philosophical and methodological difference from (generalised linear) mixed modelling is that GEEs do not estimate the distributional properties of the subjects themselves. In a mixed model setting, if there are a sufficient number of subjects (or fields, nest, trawls, etc.), we can estimate the variance of the distribution that their effects are drawn from. This is usually taken to be a Normal distribution.

We now specify a GEE, and broadly follow Chapter 11 in Fitzmaurice et al. (2004). GEE models are also called ‘marginal’ models, but this is slightly confusing. In previous chapters, we used the word ‘marginal’ in the context of conditional models (i.e. conditional on a random effect). Here, it means that the model for the mean response only depends on covariates and not on random effects.

12.2.2 Step 1 of the GEE: Systematic Component and Link Function

Suppose we have a response variable Y_{is} and one explanatory variable X_{is} . Y_{is} can be the number of birds in field i at time s , the number of sibling calls in nest i at time s , or the presence or absence of the parasite $E. cervi$ in deer j sampled at farm i . The systematic part in all these models is given by

$$\eta = \alpha + \beta_1 \times X_{is}$$

It is also possible to have more explanatory variables. The relationship between the conditional mean and the systematic component has the same structure as in GLM models. Hence, for count data we use

$$E(Y_{is}) = e^\eta = e^{\alpha + \beta_1 \times X_{is}}$$

and for the 0–1 data

$$E(Y_{is}) = \frac{e^{\alpha + \beta_1 \times X_{is}}}{1 + e^{\alpha + \beta_1 \times X_{is}}}$$

The word conditional is with respect to the explanatory variables (we pretend we know what they are). We should, therefore, write the first part of the equation as $E(Y_{is}|X_{is})$. This reads as follows: The expected value of Y_{is} for given X_{is} . A more general notation for the relationship between the conditional mean and the explanatory variables for the count data is

$$E(Y_{is}|X_{is}) = \mu_{is} \text{ and } g(\mu_{is}) = \alpha + \beta_1 \times X_{is}$$

12.2.3 Step 2 of the GEE: The Variance

For count data, the easiest conditional variance structure of Y_{is} is given by

$$\text{var}(Y_{is}|X_{is}) = \mu_{is}$$

Obviously, we can also opt for a negative binomial type variance structure (Chapter 11), but for the moment we will keep it simple. The notation for a more general model is: $\text{var}(Y_{is} | X_{is}) = \phi \times v(\mu_{is})$, where $v()$ is the variance function, and ϕ is the scale parameter (overdispersion), which we need to estimate or simply set to 1. Choosing $\phi = 1$ and $v(\mu_{is}) = \mu_{is}$ gives a identical variance structure to the one used in Poisson GLM. For the absence–presence deer data, we can choose a binomial variance structure (Chapter 10).

You may wonder why we do not just assume that count data Y_{is} is Poisson distributed with mean μ_{is} , or for 0–1 data a binomial distribution? After all, these give the same mean and variance relationships as specified above. The reason is because we can do the GEE without having to specify a distribution. Furthermore, in the next step, we have to specify a correlation structure between the observations. Assuming that Y_{is} is Poisson or binomial distributed makes this step awkward, we will explain below why.

Basically, all we have done so far is follow the quasi-GLM route by specifying the relationship between the mean and the explanatory variables and the variance structure. The next step specifies the association between the observations. Note we carefully wrote ‘association’ and not correlation.

12.2.4 Step 3 of the GEE: The Association Structure

Now we have to specify an association structure between Y_{is} and Y_{it} , where s and t are two different sampling days on the same field i , two observations from the same nest, or two deer from the same farm. There are many ways of doing this, and the

type of data (e.g. continuous, binary or counts) also affects how the association is defined.

Option 1: The Unstructured Correlation

For continuous data, the obvious tool to define association between the two observations Y_{is} and Y_{it} is the Pearson correlation coefficient. Just as in Chapter 6, we have various options to parameterise the correlation. The most flexible choice is the so-called unstructured correlation, which is given by

$$\text{cor}(Y_{is}, Y_{it}) = \alpha_{st}$$

This correlation structure can be easily understood if we imagine temporally sequential observations coming from the same source, for example, a blood pressure reading taken from the same patient/animal at regular (e.g. hourly) intervals. The correlation matrix can be expressed thus:

$$\left(\begin{array}{cccccccccccccc} 1 & \alpha_{12} & \alpha_{13} & \alpha_{14} & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \alpha_{14} & \alpha_{24} & \alpha_{34} & 1 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & \alpha_{12} & \alpha_{13} & \alpha_{14} & & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} & & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} & & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha_{14} & \alpha_{24} & \alpha_{34} & 1 & & & & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & & \ddots & & & & & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & & & & & 1 & \alpha_{12} & \alpha_{13} & \alpha_{14} & & \\ \vdots & \vdots & \vdots & \vdots & & & & & & \alpha_{12} & 1 & \alpha_{23} & \alpha_{24} & & \\ \vdots & \vdots & \vdots & \vdots & & & & & & \alpha_{13} & \alpha_{23} & 1 & \alpha_{34} & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & \alpha_{14} & \alpha_{24} & \alpha_{34} & 1 & & \end{array} \right)$$

The upper left 4×4 block is the correlation matrix for the first patient, the second block for the second patient, etc. In each block, α_{st} is the correlation between observations s and t . We use the different blocks to estimate these parameters. No correlation between the parameters α_{12} , α_{13} , α_{14} , α_{23} , α_{24} , and α_{34} is assumed, and they are estimated completely independently. Note that this is based on 4 observations per subject. The number of independent parameters to be estimated rapidly increases as the number of within-subject observations increases with all the attendant problems related to matrix inversion.

This is the most general correlation model and perhaps the least intuitively appealing. Essentially, all correlations between within-subject observations are

estimated independently; thus a lot more parameters need to be estimated. Because of this complexity, the GEE algorithm can break down because the correlation matrix cannot be inverted. However, it can be a useful approach if no obvious correlation structure suggests itself and can be a useful exploratory step to help arrive at a final choice of correlation structure.

Let us discuss the applicability of the unstructured correlation for the response variables in the California bird data set, owl data set, and deer data set. For the moment, we ignore that these response variables are not continuous. For the California bird data, each block of correlation in the matrix above is for a field; for the owl data each block is a nest; and for the deer data, each block is a farm. For the deer data, it does not make sense to use the unstructured correlation because there is no relationship between animals 1 and 2 at farm 1, and animals 1 and 2 at farm 2. For the California bird data, it may be an option to use this correlation structure as observations 1 and 2 in field 1, and observations 1 and 2 in field 2 both tells us something about the temporal relationship at the start of the experiment. On the down side, 10 temporal observations per field mean that we have to estimate $10 \times 9/2 = 45$ correlation parameters, which is a lot! The unstructured correlation may be an option for these data if you have hundreds of fields, but not with only 12 fields. For the owl data, it is a bit more complicated. If we just analyse the number of calls sampled at the nests without a time order, then the set up of the data is similar to that of the deer data. Hence, in this case, we cannot use the unstructured correlation. But we also know the arrival time of the parents at the nest, which unfortunately, is irregularly spaced. However, in Chapter 6, we argued that based on biology, we could assume that owl parents chose the arrival time, and therefore, from their point of view, the data are regularly spaced. Hence, if we use the unstructured correlation, then α_{12} represents the correlation between arrivals 1 and 2, α_{13} the correlation between arrivals 1 and 3, etc. This would make sense, but unfortunately, this approach requires an enormous amount of correlation parameters as some nests contain more than 50 observations. Hence, it is not practical.

Option 2: AR-1 Correlation

Another option for continuous data is to say that the correlation between two observations from the same patient, field, nest, or farm i is

$$\text{cor}(Y_{is}, Y_{it}) = \alpha^{|s-t|}$$

This type of auto-regressive correlation structure was also used in Chapter 6 (using the `corAR1` function). Autoregressive correlation is observed when correlation between within-subject observations can be modelled directly as a function of the ‘distance’ between the observations in question. Using the same example as above, the following correlation matrix is used.

$$\left(\begin{array}{cccccccccccccc} 1 & \alpha & \alpha^2 & \alpha^3 & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \alpha & 1 & \alpha & \alpha^2 & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \alpha^2 & \alpha & 1 & \alpha & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \alpha^3 & \alpha^2 & \alpha & 1 & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & \alpha & \alpha^2 & \alpha^3 & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha & 1 & \alpha & \alpha^2 & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha^2 & \alpha & 1 & \alpha & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha^3 & \alpha^2 & \alpha & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & & \ddots & & & & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & & & & & 1 & \alpha & \alpha^2 & \alpha^3 & \\ \vdots & \vdots & \vdots & \vdots & & & & & & \alpha & 1 & \alpha & \alpha^2 & \\ \vdots & \vdots & \vdots & \vdots & & & & & & \alpha^2 & \alpha & 1 & \alpha & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \alpha^3 & \alpha^2 & \alpha & 1 & \end{array} \right)$$

Again, each block refers to the same patient, field, nest or farm. The above correlation matrix assumes regular distances (or time interval) between observations. The parameterisation is rather more involved where the distances are uneven. The advantage of this correlation structure is that only one correlation parameter needs estimated, i.e. α .

For the California birds, it is a good option; the correlation between observations separated by one time unit (2 weeks) is likely to be more similar than those separated by larger time units. For the deer that, it would not make any sense as there is no time order in the sampled animals per farm. For the owl data, it only makes sense if we consider the time order in the data.

The AR-1 correlation can be used for any data set in which there is a time order, although instead of time, depth or age gradients can also be used. This means that it can be used for the California bird data and for the owl data (using the arrival time). There are several books and papers that discuss how to use GEE for spatial data; see for example Diggle and Ribeiro (2007) and especially Pebesma et al. (2000).

Option 3: Exchangeable Correlation

This is the most easily understood and most easily estimated form of within-subject correlation. The correlation between two observations from the same field i is assumed to be

$$\text{cor}(Y_{is}, Y_{it}) = \alpha$$

If, for example, we take body weights of a batch of roe deer (say 4) from 5 different sites across the country, it is probably sufficient to just say that bodyweights from a given site are correlated. We do not need to consider temporal or sequential correlation (we ignore here the potential issue of within-site spatial correlation which is

deliberately vague in this example). But it is reasonable to expect that bodyweights from a given site will be more similar, on average, than those from other sites (availability of resources, genetic similarity, etc). It can be imagined that the correlation between bodyweights Y_{ij} , where i denotes the area and j the animal within the area, may take the form

$$\begin{pmatrix} 1 & \alpha & \alpha & \alpha & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \alpha & 1 & \alpha & \alpha & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \alpha & \alpha & 1 & \alpha & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \alpha & \alpha & \alpha & 1 & 0 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & \alpha & \alpha & \alpha & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha & 1 & \alpha & \alpha & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha & \alpha & 1 & \alpha & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha & \alpha & \alpha & 1 & & & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & & \ddots & & & & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & & & & & 1 & \alpha & \alpha & \alpha \\ \vdots & \vdots & \vdots & \vdots & & & & & & \alpha & 1 & \alpha & \alpha \\ \vdots & \vdots & \vdots & \vdots & & & & & & \alpha & \alpha & 1 & \alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & \alpha & \alpha & \alpha & \alpha & 1 \end{pmatrix}$$

So we now have a new term α which expresses the correlation between bodyweights from animals in the same area. In the context of GEE, this is referred to as exchangeable correlation, but in other settings, it is often referred to as ‘compound-symmetry’. We have also seen this correlation structure in Chapters 5 and 6 with the linear mixed effects model. There, we had the compound symmetry correlation due to a random intercept, but saw a similar correlation structure in the time series chapter. In the first case, the correlation is always positive; see also Pinheiro and Bates (2000, pp. 227–228).

Option 4: Another Correlation Structure – Stationary Correlation

One interesting case is where within-subject correlation exists up to a given distance and then stops completely. Although this is not an obvious choice of correlation structure, we may happen to know this is a good model in advance or it may be indicated from an exploratory analysis of the correlation structure. If we imagine again the situation of four consecutive blood readings taken, once per hour, as in the hypothesised scenario for autoregressive correlation. Under a model where correlation is autoregressive up to a time lag of 2 hours, but ceases thereafter, the correlation matrix will take this general form

$$\left(\begin{array}{cccccccccccccc} 1 & \alpha & \alpha^2 & 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \alpha & 1 & \alpha & \alpha^2 & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \alpha^2 & \alpha & 1 & \alpha & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \alpha^2 & \alpha & 1 & 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & \alpha & \alpha^2 & 0 & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha & 1 & \alpha & \alpha^2 & & & & & & 0 \\ 0 & 0 & 0 & 0 & \alpha^2 & \alpha & 1 & \alpha & & & & & & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^2 & \alpha & 1 & & & & & & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & & \ddots & & & & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & & & & & 1 & \alpha & \alpha^2 & 0 & \\ \vdots & \vdots & \vdots & \vdots & & & & & & \alpha & 1 & \alpha & \alpha^2 & \\ \vdots & \vdots & \vdots & \vdots & & & & & & \alpha^2 & \alpha & 1 & \alpha & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \alpha^2 & \alpha & \alpha & 1 \end{array} \right)$$

Note that unlike the other correlation structures described, we see cases of zero correlation within-subject.

This gives a flavour of the various correlation structures available. There are several others (e.g. non-stationary auto-correlation and ante-dependence), and you can impose correlation estimates a priori if this information is known in advance.

The number of unknown parameters in the auto-regressive and compound symmetric correlation structures was only 1, but with the unstructured correlation we have $t \times (t - 1)/2$ parameters. This is potentially difficult to estimate, especially if we have a relatively large number of observations over time; 10 longitudinal observations means we already need to estimate 45 association parameters!

12.3 Why All the Fuss?

We saw in Chapters 5, 6, and 7 how data from the same source (e.g. all readings taken from the same beach) can be correlated and the implications this has for the variance-covariance structure, which in turn informs the error associated with the parameter estimates. In the simplest scenario, we can imagine a situation where there is no within-subject correlation and the correlation matrix for the data Y_{ij} is simply diagonal.

$$\left(\begin{array}{ccccc} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & & & \ddots & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array} \right)$$

Indeed, we no longer need to consider the problem in terms of Y_{ij} and i subjects, rather just Y_j with no subject index. This is the correlation structure adopted implicitly in GLM.

In Chapters 9 and 10, we discussed the mathematical background of GLMs. Recall that we started with a distribution, specified a likelihood function L , and then found the parameters that maximised the likelihood function. The matrix of standard errors is essential as the basis of statistical inference, and typically, this is estimated as the inverse of the matrix of second derivatives of the GLM log-likelihood L such that:

$$V_H(\hat{\beta}) = \left\{ \left(-\frac{\partial^2 L}{\partial \beta_u \partial \beta_v} \right) \right\}_{p \times p}^{-1}$$

A different approach, which is asymptotically equivalent, (i.e. tends towards an equivalent solution with increasing sample size) is based on the *expectation* of the second derivative which comes from the result

$$E \left(\frac{\partial^2 L}{\partial \beta_u \partial \beta_v} \right) = -\frac{\partial L}{\partial \beta_u} \times \frac{\partial L}{\partial \beta_v}$$

This second approach based on the expectation of the second derivatives is usually referred to as Fisher scoring. Although these two approaches will tend towards the same solution, discrepancies can occur, particularly where the sample size is small. Statistical tests are then based on the recognised t -test formulation, i.e.

$$\frac{\hat{\beta}}{s.e.(\hat{\beta})}$$

The standard errors are taken from the diagonal elements of the V_H or the Fisher information matrix. The problem with this approach is that the underlying statistical theory assumes that the observations are independent. And this is where GEE provides a solution to cases where we might be violating the independence assumption.

Basically, GEE uses the same equations as generalised least squares (GLS) and GLM, but instead of using a diagonal matrix for the covariance matrix (implying independence), we replace it by an association matrix, as defined in the previous section. If you are not familiar with the regression, GLS and GLM maths, you can skip a couple of paragraphs, as it is not essential for *using* GEE. We stress that we only present the principle; the reader interested in full mathematical details is advised to read Liang and Zeger (1986).

12.3.1 A Bit of Maths

In linear regression, the following criteria (which is the residual sum of squares) is minimised to find the optimal regression parameters.

$$\sum_{i=1}^N (\mathbf{Y}_i - \mathbf{X}_i \times \boldsymbol{\beta})' \times (\mathbf{Y}_i - \mathbf{X}_i \times \boldsymbol{\beta})$$

In the context of the California bird data, i is the index for fields, $N = 11$, \mathbf{Y}_i contains all the longitudinal data from field i , and \mathbf{X}_i denotes the associated explanatory variables. The regression parameters in $\boldsymbol{\beta}$ are obtained by minimising this expression by taking derivatives with respect to $\boldsymbol{\beta}$, setting them to 0, and solving the resulting equations. In GLS, we use a similar optimisation criterion, namely,

$$\sum_{i=1}^N (\mathbf{Y}_i - \mathbf{X}_i \times \boldsymbol{\beta})' \times \Sigma_i^{-1} \times (\mathbf{Y}_i - \mathbf{X}_i \times \boldsymbol{\beta})$$

The matrix Σ_i is a covariance matrix which can either have different diagonal elements (to model heterogeneity) or non-zero off-diagonal elements to allow for temporal or spatial correlation. In Chapters 5, 6, and 7, we used Σ_i to describe the within-field correlation structure. Taking derivatives of this optimisation criterion with respect to $\boldsymbol{\beta}$ and setting them to 0 give

$$\sum_{i=1}^N \mathbf{X}_i \times \Sigma_i^{-1} \times (\mathbf{Y}_i - \mathbf{X}_i \times \boldsymbol{\beta}) = \mathbf{0}$$

It is also common notation to replace $\mathbf{X}_i \boldsymbol{\beta}$ by $\boldsymbol{\mu}_i$. For a GEE, we follow the same procedure, and the starting point is again

$$\sum_{i=1}^N (\mathbf{Y}_i - \boldsymbol{\mu}_i)' \times \Sigma_i^{-1} \times (\mathbf{Y}_i - \boldsymbol{\mu}_i)$$

Again, the optimal regression parameter are obtained by taking derivatives and solving the generalised estimation equations

$$\sum_{i=1}^N \mathbf{D}_i \times \Sigma_i^{-1} \times (\mathbf{Y}_i - \boldsymbol{\mu}_i) = \mathbf{0} \quad (12.1)$$

The matrix \mathbf{D}_i contains first-order derivatives of the $\boldsymbol{\mu}_i$ with respect to the regression parameters. Σ_i is the covariance matrix, and it can be written as

$$\Sigma_i = \mathbf{A}_i^{\frac{1}{2}} \times \text{cor}(\mathbf{Y}_i) \times \mathbf{A}_i^{\frac{1}{2}}$$

This looks complicated, but the matrices \mathbf{A}_i are diagonal matrices containing the variances. So, basically this is just matrix notation for the definition of the covariance: Correlation multiplied with the square root of the variances. Again, in ordinary GLMs, both Σ_i and $\text{cor}(\mathbf{Y}_i)$ are diagonal matrices, because we assume independence.

The problem is that in reality we have to estimate the covariance matrix Σ_i , and this can be quite expensive in terms of numbers of parameters. GEE applies a clever trick by replacing the inner part, $\text{cor}(\mathbf{Y}_i)$, by an estimate correlation matrix $\mathbf{R}(\alpha)$ so that we get

$$\mathbf{V}_i = \phi \times \mathbf{A}_i^{\frac{1}{2}} \times \mathbf{R}_i(\alpha) \times \mathbf{A}_i^{\frac{1}{2}}$$

The ϕ allows for extra variation such as in a quasi-Poisson model. The \mathbf{V}_i is an estimate of $\boldsymbol{\Sigma}_i$. The better you choose the correlation structure, the closer the estimated covariance matrix \mathbf{V}_i (also called the working covariance matrix) is to the real covariance matrix. This means that we have to determine what form $\mathbf{R}(\alpha)$ takes, or more precisely, to choose a correlation structure that closely describes what is observed in the response data. This can be any of the correlation structures discussed in the previous section.

But we have still not answered the question in the title of this section. Well, here it comes. We want to estimate the values of the β s and their confidence intervals and then apply statistical tests. To estimate the β s, an iterative algorithm is applied that consists of the following steps:

1. For given ϕ and α (and therefore \mathbf{V}_i), obtain an estimate for the regression parameters.
2. Given the regression parameters, update ϕ and α (and therefore \mathbf{V}_i). Pearson residuals are used for this.
3. Iterate between steps 1 and 2 until convergence.

At convergence, the estimated regression parameters are consistent² and asymptotically normally distributed with mean $\boldsymbol{\beta}$ and covariance matrix: $\mathbf{B}^{-1} \times \mathbf{M} \times \mathbf{B}^{-1}$, where

$$\begin{aligned}\mathbf{B} &= \sum_{i=1}^N \mathbf{D}_i \times \boldsymbol{\Sigma}_i^{-1} \times \mathbf{D}_i \\ \mathbf{M} &= \sum_{i=1}^N \mathbf{D}_i \times \boldsymbol{\Sigma}_i^{-1} \times \text{cov}(\mathbf{Y}_i) \times \boldsymbol{\Sigma}_i^{-1} \times \mathbf{D}_i\end{aligned}$$

This statement also holds true, even if your specification of the correlation structure is not correct. We used the same notation as Fitzmaurice et al. (2004). And once we have calculated the covariance matrix, we can use its diagonal elements to obtain standard errors and confidence intervals. Hence, the last thing we have to do is explain how to get the \mathbf{B} and \mathbf{M} . This is a matter of replacing $\boldsymbol{\Sigma}_i$ by its estimate \mathbf{V}_i and $\text{cov}(\mathbf{Y}_i)$ by the covariance matrix $(\mathbf{Y}_i - \boldsymbol{\mu}_i) \times (\mathbf{Y}_i - \boldsymbol{\mu}_i)'$. Your chosen correlation $\mathbf{R}(\alpha)$ structure is then used in the covariance term in the inner part of the matrix \mathbf{M} , resulting in the so-called sandwich estimator. GEE is robust against misspecification of the correlation structure (it still provides valid standard errors). This does not mean you do not have to bother about choosing a good correlation structure; the better your choice, the better the standard errors. And, it is only a characteristic of large sample sizes.

²Consistent means that estimated parameters are nearly equal to the population parameters.

All the complications involved in choosing the within-subject correlation structure are essentially a means to an end. Usually we are interested in the significance of covariates or so-called fixed effects. In a sense, the correlation structure can be seen as an inconvenience that needs to be accounted for before making meaningful inferences about the parameters we are primarily interested in.

In summary, to answer the title of this section, GEE incorporates a correlation structure on the data from the same field, and as a result, we obtain consistent estimators.

In the second step of the two-step algorithm described above, for given regression parameters update ϕ , α , and \mathbf{V}_i , things are more complicated. Depending on the type of correlation structure you use, e.g. AR1, unstructured or exchangeable, the software will use different expressions for these two parameters.

12.4 Association for Binary Data

We can easily extend the idea above to deal with similar count data problems such as, for example, the number of ticks found on the same deer measured for body-weight. Alternatively, in Chapter 22, we use a case study where the response variable is the presence or absence of badger activity at farms: A binary variable. The same holds for the deer data. Various statistical textbooks contain phrases like: ‘the correlation is modelled at the level of the linear predictor rather than at the scale of the raw data’. The underlying idea is that for binary data, the correlation coefficient is not the most natural tool to define association. Using some basic definitions like $P(A \text{ and } B) = P(B) \times P(A | B)$, the definition of the expectation of discrete random variable, the mean and variance of a binary variable, and the definition of the correlation and covariance, we can easily show that the correlation between two binary variables with means μ_1 and μ_2 ($\mu_1 \geq \mu_2$) is smaller than $\sqrt{(\mu_2 - \mu_1\mu_2)/(\mu_1 - \mu_1\mu_2)}$. If, for example, $E(Y_1) = \mu_1 = 0.7$ and $E(Y_2) = \mu_2 = 0.3$, then the correlation between Y_1 and Y_2 is smaller than 0.49. To overcome this, Fitzmaurice et al. (2004) used odds ratios to define the (unstructured) association as $\log(\text{OR}(Y_{is}, Y_{ik})) = \alpha_{sk}$, where

$$\text{OR}(Y_{is}, Y_{ik}) = \frac{\Pr(Y_{is} = 1 \text{ and } Y_{ik} = 1) \times \Pr(Y_{is} = 0 \text{ and } Y_{ik} = 0)}{\Pr(Y_{is} = 1 \text{ and } Y_{ik} = 0) \times \Pr(Y_{is} = 0 \text{ and } Y_{ik} = 1)}$$

However, this association parameterisation is not available in the main GEE packages in R (it is in SAS), but you could program it yourself using the option for the user specified correlation structure in the GEE functions. In the GEE functions, we use in R in the next section, and in Chapter 22, we specify a correlation structure at the level of the raw data.

12.5 Examples of GEE

12.5.1 A GEE for the California Birds

In this section we revisit the California bird data, and apply GEE. The first two steps of GEE were presented in Section 12.1, but are repeated here. In the first step, we specify the relationship between the mean μ_{is} and the covariates:

$$E(Y_{is}) = \mu_{is} = e^{\alpha + \beta_1 \times \text{Depth}_{is} + \beta_2 \times \text{Depth}_{is}^2 + \beta_3 \times \text{Sptreat}_{is}}$$

In the second step, we specify the variance of the observed data:

$$\text{Var}(Y_{is}) = \phi \times \mu_{is}$$

Hence, we use $v(\mu_{is}) = \phi\mu_{is}$, which is in line with the characteristics of a (quasi-) Poisson GLM, but keep in mind we do not specify any distribution here. In the third step, we need to specify a correlation structure. One option is to use biological knowledge and argue that the number of birds in a field i at time s depends on those measured at time $s - 1$, and also, although less strong, on $s - 2$, etc., in the same field. Accepting this approach suggests using an auto-regressive correlation structure. We could also make an auto-correlation function for the data of each field, and investigate whether there is a significant auto-correlation. And if this shows no correlation, then we can apply a GLM.

The alternative option of a compound correlation is unlikely to be appropriate here. Why would bird numbers separated by 2 weeks (1 sampling unit) have the same correlation as those separated by 20 weeks (10 sampling units)?

There are various packages for GEE in R, but we only use the `geeglm` function from the `geepack` package in this book. The `gee` function from the `gee` package is also useful and so is the package `yags`. These packages are not part of the base installation of R; so you will need to download and install them. We use the `geepack` package as it is slightly more advanced than the others, e.g. it allows for a Wald test via the `anova` command.

The following code loads the `geepack` package (assuming it has been downloaded and installed) and applies the GEE (you also need to run the code from Section 12.1 for the data preparation).

```
> library(geepack)
> M.gee1 <- geeglm(Richness ~ offset(LA) + DEPTH +
+ DEPTH2 + fSptreat, data = RFBirds,
+ family = poisson, id = fField, corstr = "ar1")
> summary(M.gee1)
```

Note that this function wants us to specify a distribution with the `family` option, even though we are not assuming any distribution directly.

The grouping structure is given by the `id` option; this specifies which bird observations form a block of data. The `corstr` option specifies the type of correlation. This correlation is applied on each block of data. We argued above that the AR-1 auto-correlation structure should be used; hence `corstr = "ar1"`. Alternatives are unstructured (multiple α s), exchangeable (one α), independence (this gives the same results as the ordinary GLM), and userdefined (for the braves; you can program your own correlation structure). Our data does not contain missing values and were sorted along time within a field. If this is not the case, you need to use the `waves` option; see also the `geeglm` help file. This option ensures that R does not mess up the order of the observations. The `summary` command gives the following output.

```
Coefficients:
            Estimate      Std.err      Wald      p (>W)
(Intercept) -0.678203399 0.3337043786 4.130438 0.04211845
fSptreatrlfld -0.522313667 0.2450125672 4.544499 0.03302468
DEPTH         0.049823774 0.0287951864 2.993874 0.08358002
DEPTH2        -0.001141096 0.0008060641 2.004033 0.15688129

Estimated Scale Parameters:
            Estimate      Std.err
(Intercept) 2.333533 0.3069735

Correlation: Structure = ar1 Link = identity
Estimated Correlation Parameters:
            Estimate      Std.err
alpha 0.4215071 0.1133636
Number of clusters: 11 Maximum cluster size: 10
```

The correlation between two sequential observations in the same field is 0.42; if the time lag is two units (4 weeks), the correlation is $0.42^2 = 0.177$, between observations separated by three units (6 weeks), it is $0.42^3 = 0.075$, etc. The scale parameter is 2.333, which is similar to the over-dispersion parameter of the quasi-Poisson model applied on the same data in Section 12.1. There is a weak but significant treatment effect of the straw. Hence, the following model was fitted on the bird data.

$$\begin{aligned}E[Y_{is}] &= \mu_{is} = e^{-0.678+0.049 \times \text{Depth}_{is}-0.001 \times \text{Depth}_{is}^2-0.522 \times \text{Sptreat}_{is}} \\ \text{var}(Y_{is}) &= 2.333 \times \mu_{is} \\ \text{cor}(Y_{is}, Y_{it}) &= 0.421^{|s-t|}\end{aligned}$$

This relationship is not conditional on random effects, only on the explanatory variables. For this reason, it is called a marginal model. Hardin and Hilbe (2002) called it the population average GEE, abbreviated as PA-GEE.

Note that in the GLM in Section 12.1 both the straw management variable and the depth variables are significant. In the GEE, which takes into account temporal correlation, only the straw management variable is significant!

The nice thing of the geepack package is that it allows for a Wald test, which can be used to test the significance of nominal variables with more than two levels. This is not the case here, but for illustrative purposes, we show how it can be used to decide whether we need any of the depth terms. The code below fits a GEE without any of the depth terms and applies a Wald test using the anova command. The output suggests that we only need fSptreat.

```
> M.gee2 <- geeglm(Richness ~ offset(LA) + fSptreat,
+                     data = RFBirds, family = poisson, id = FIELD,
+                     corstr = "ar1")

> anova(M.gee1, M.gee2)

Analysis of 'Wald statistic' Table
Model 1 Richness ~ offset(LA) + DEPTH + DEPTH2 + fSptreat
Model 2 Richness ~ offset(LA) + fSptreat
      Df    X2 P(>|Chi|)
1 2 3.9350 0.1398
```

12.5.2 A GEE for the Owls

So, what is an appropriate correlation structure for the owl data? We could use the compound correlation structure, which is called ‘exchangeable’ within the context of the GEE. This assumes that all observations from the nest are correlation with the value of α . Code to do this is given by

```
> library(geepack)
> Form <- formula(NCalls ~ offset(LBroodSize) +
+                   SexParent * FoodTreatment +
+                   SexParent * ArrivalTime)
> O4 <- geeglm(Form, data = Owls, family = poisson,
+                id = Nest, corstr = "exchangeable")
```

The results of the summary(O4) command are not given here, but show that the estimated value of α is 0.058, which is rather small.

In Chapters 5 and 6, we analysed the *average* sibling negotiation. Recall that we have multiple observations from the same nest, but that these were obtained during two nights. The food treatment was swapped during the second night. In Chapter 5, the compound correlation was imposed by using nest as a random intercept. In Chapter 6, we continued the analysis by arguing that there may be auto-regressive correlation between the observations made in the same night from the same nest. The only thing is arrival times of the birds are not regularly spaced in

time, but we argued that from the owls' point of view, time may be regularly spaced (this was a biological assumption). We can do the same here, except that we use the number of calls.

The problem is that the data file does not contain a column that identifies the group of observations from the same night and nest; hence we have to make it.

```
> N <- length(Owls$Nest)
> NLev <- c(paste(unique(Owls$Nest), ".Dep", sep = ""),
           paste(unique(Owls$Nest), ".Sat", sep = ""))
> Owls$NestNight <- factor(levels = NLev)
> for (i in 1:N) {
  if (Owls$FoodTreatment[i] == "Deprived") {
    Owls$NestNight[i] <-
      paste(Owls$Nest[i], ".Dep", sep = "") }
  if (Owls$FoodTreatment[i] == "Satiated") {
    Owls$NestNight[i] <-
      paste(Owls$Nest[i], ".Sat", sep = "") } }
```

This is a bit of tedious programming, and instead of explaining it in detail, let us show the results of the code:

```
> Owls[1 : 10, c(1, 2, 4, 10)]
   Nest FoodTreatment ArrivalTime     NestNight
1 AutavauxTV     Deprived    22.25 AutavauxTV.Dep
2 AutavauxTV     Satiated    22.38 AutavauxTV.Sat
3 AutavauxTV     Deprived    22.53 AutavauxTV.Dep
4 AutavauxTV     Deprived    22.56 AutavauxTV.Dep
5 AutavauxTV     Deprived    22.61 AutavauxTV.Dep
6 AutavauxTV     Deprived    22.65 AutavauxTV.Dep
7 AutavauxTV     Deprived    22.76 AutavauxTV.Dep
8 AutavauxTV     Satiated    22.90 AutavauxTV.Sat
9 AutavauxTV     Deprived    22.98 AutavauxTV.Dep
10 AutavauxTV    Satiated    23.07 AutavauxTV.Sat
```

The variable `NestNight` tells us which observations are from the same night and same nest. The column `ArrivalTime` shows at what time an observation was made, but as we already discussed, we will consider the arrivals as regularly spaced in time. So, the `for` loop with the `if` statement was only used to make the variable `NestNight`. You could also have done this in Excel. As always in R, things can be done in multiple ways. Here is an alternative piece of R code to obtain exactly the same `NestNight`.

```
> Owls$NestNight <- factor(
  ifelse(Owls$FoodTreatment == "Deprived",
         paste(Owls$Nest, ".Dep", sep = ""),
         paste(Owls$Nest, ".Sat", sep = "")))
```

The `ifelse` executes the first paste command if an observation is food deprived, and as the name already suggests, the second paste command otherwise. No need for a loop. Elegant, but it takes a bit more time to see what it does.

Applying the GEE is now simple:

```
> O3 <- geeglm(Form, data = Owls, family = poisson,
                 id = NestNight, corstr = "ar1")
```

To figure out whether we need the two two-way interactions, we can drop each of them in turn, apply the Wald test, and remove the least significant variable:

```
> O3.A <- geeglm(NCalls ~ off-set(LBroodSize) +
                  SexParent + FoodTreatment +
                  SexParent * ArrivalTime, data = Owls,
                  family = poisson, id = NestNight, corstr = "ar1")
> O3.B <- geeglm(NCalls ~ off-set(LBroodSize) +
                  SexParent * FoodTreatment +
                  SexParent + ArrivalTime, data = Owls,
                  family = poisson, id = NestNight, corstr = "ar1")
> anova(O3, O3.A)

Analysis of 'Wald statistic' Table
Model 1 NCalls ~ offset(LBroodSize) + SexParent * Food-Treatment +
               SexParent * ArrivalTime
Model 2 NCalls ~ offset(LBroodSize) + SexParent + FoodTreatment +
               SexParent * ArrivalTime
      Df      X2  P(>|Chi|)
1  1 0.23867  0.62517

> anova(O3, O3.B)

Analysis of 'Wald statistic' Table
Model 1 NCalls ~ offset(LBroodSize) + SexParent * Food-Treatment +
               SexParent * ArrivalTime
Model 2 NCalls ~ offset(LBroodSize) + SexParent * Food-Treatment +
               SexParent + ArrivalTime
      Df      X2  P(>|Chi|)
1  1 0.40269  0.52570
```

The sex of the parent and food treatment interaction is the least significant term and was dropped. This process can then be repeated a couple of times until all terms in the model are significant. The final model and its output are given by:

```
> O6 <- geeglm(NCalls ~ off-set(LBroodSize) +
                  FoodTreatment + ArrivalTime, data = Owls,
                  family = poisson, id = NestNight, corstr = "ar1")
> summary(O6)

Call:
geeglm(formula = NCalls ~ offset(LBroodSize) + FoodTreatment +
       ArrivalTime, family = poisson, data = Owls, id = NestNight,
       corstr = "ar1")
```

```

Coefficients:
              Estimate     Std.err      Wald      p(>W)
(Intercept)    3.5927875  0.67421928 28.39623 9.885749e-08
FoodTreatmentSatiated -0.5780999  0.11507976 25.23527 5.074576e-07
ArrivalTime     -0.1217358  0.02725415 19.95133 7.943886e-06

Estimated Scale Parameters:
              Estimate     Std.err
(Intercept) 6.639577  0.5234689

Correlation: Structure = ar1 Link = identity

Estimated Correlation Parameters:
              Estimate     Std.err
alpha 0.5167197  0.06830255

Number of clusters: 277 Maximum cluster size: 18

```

The correlation of the calls between two sequential arrivals is 0.51, which is relatively high. The overdispersion is 6.6, which is similar to that of the quasi-Poisson GLM. The estimated regression parameters are similar to those of the quasi-Poisson GLM, but the *p*-values are considerably larger (at least for the slopes). However, the biological conclusions are the same; there is a food treatment effect (lower number of calls from food satiated observations) and later the night, the less calls. The final GEE is given by

$$E(\text{NCalls}_{is}) = \mu_{is} \quad \text{and} \quad \text{var}(\text{NCalls}_{is}) = 6.6 \times \mu_{is}$$

$$\text{cor}(\text{NCalls}_{is}, \text{NCalls}_{it}) = 0.51^{|t-s|}$$

12.5.3 A GEE for the Deer Data

The required correlation structure for the deer data is obvious; it has to be the compound correlation, alias the exchangeable correlation because there is no specific (e.g. time) order between the observations from the same farm. The code and numerical output to fit this model is as follows. The exchangeable correlation is selected using the `corstr = "exchangeable"` bit, and `id = Farm` tells the `geeglm` function which observations are from the same farm.

```

> library(geepack)
> DE.gee <- geeglm(Ecervi.01 ~ CLength * fSex,
+                      data = DeerEcervi, family = binomial,
+                      id = Farm, corstr = "exchangeable")
> summary(DE.gee)

Call:
geeglm(formula = Ecervi.01 ~ CLength * fSex, family = binomial,
       data = DeerEcervi, id = Farm, corstr = "exchangeable")

Coefficients:
              Estimate     Std.err      Wald      p(> W)
(Intercept) 0.73338099  0.280987616  6.812162 9.053910e-03

```

```

CLength      0.03016867 0.006983758 18.660962 1.561469e-05
fSex2        0.47624445 0.217822972 4.780271 2.878759e-02
CLength:fSex2 0.02728028 0.014510259 3.534658 6.009874e-02

Estimated Scale Parameters:
    Estimate   Std.err
(Intercept) 1.145337 0.4108975

Correlation: Structure = exchangeable Link = identity
Estimated Correlation Parameters:
    Estimate   Std.err
alpha 0.3304893 0.04672826
Number of clusters: 24 Maximum cluster size: 209

```

Note that a scale parameter is used. For a fair comparison with the binomial GLM (which does not contain a dispersion parameter), you can use the option `scale.fix = TRUE` in the `geeglm` command. Because the estimated dispersion parameter is only 1.14, we did not do this here. The correlation parameter is 0.33, which is moderate. The two-way interaction term is not significant ($p = 0.06$) at the 5% level, where in the binomial GLM it was! Hence, by including the compound correlation, the biological conclusions have changed! Perhaps we should re-phrase the last sentence a little bit as it suggests that both models are valid. The GLM without the correlation structure is potentially flawed as it ignores the correlation structure in the data. Therefore, only the GEE should be used for biological interpretation!

12.6 Concluding Remarks

GLS is a special case of GEE if we specify a Normal distribution and the identity link function. But we do not recommend running the GLS with GEE software as most existing GEE functions in R are less flexible in the sense of allowing for multiple variances ϕ for modelling heterogeneity.

For longitudinal data, GEE is useful if you have many fields or nest and relatively few longitudinal observations per field or nest i . If it is the other way around, standard errors produced by the sandwich estimator are less good.

Hardin and Hilbe (2002) used an AIC-type criterion to compare models with different correlation structures. It is called *quasilikelihood under the independence model information criterion* (QIC) after a paper from Pan (2001). A similar criterion is also used for selection explanatory variables. The `geeglm` function does not produce the QIC; hence, you have to program this yourself. The appendix in Hardin and Hilbe (2002) gives Stata code for this. The R package `yags` does produce the QIC. It is open code, which means that you can easily see how the programmer of `yags` implemented it. The problem that you may encounter with the QIC is that not every referee may have heard of it or agree with it.

We have not mentioned the word model validation yet. Hardin and Hilbe (2002) dedicate a full chapter to this; they present a couple of tests to detect patterns in residuals, and also graphical model validation tools. The graphical validation uses

Pearson residuals and follows the model validation steps of GLM; see also Chapters 9 and 10. We strongly suggest that after reading this chapter, you consult Hardin and Hilbe (2002). However, you have to either use Stata to follow their examples or read over the Stata code and use any of the R packages to do the same.

Chapter 13

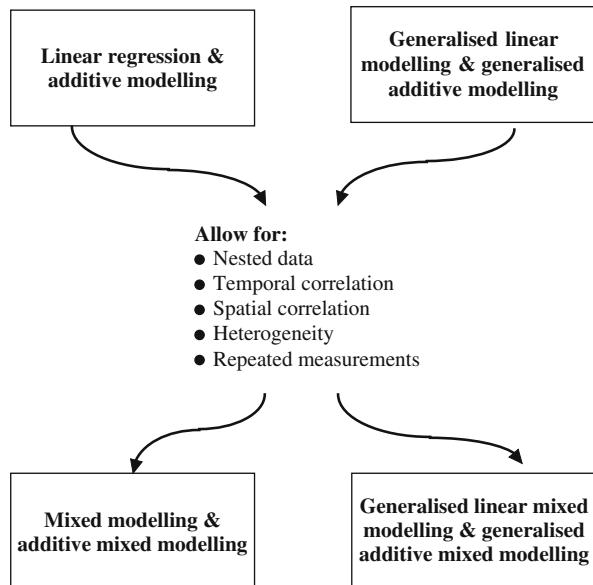
GLMM and GAMM

In Chapters 2 and 3, we reviewed linear regression and additive modelling techniques. In Chapters 4–7, we showed how to extend these methods to allow for heterogeneity, nested data, and temporal or spatial correlation structures. The resulting methods were called linear mixed modelling and additive mixed modelling (see the left hand pathway of Fig. 13.1). In Chapter 9, we introduced generalised linear modelling (GLM) and generalised additive modelling (GAM), and applied them to absence–presence data, proportional data, and count data. We used the Bernoulli and binomial distributions for 0–1 data (the 0 stands for absence and the 1 for presence), and proportional data (Y successes out of n independent trials), and we used the Poisson distribution for count data. However, one of the underlying assumptions of these approaches (GLM and GAM) is that the data are independent, which is not always the case. In this chapter, we take this into account and extend the GLM and GAM models to allow for correlation between the observations, and nested data structures. It should come as no surprise that these methods are called generalised linear mixed modelling (GLMM) and generalised additive mixed modelling (GAMM); see the right hand pathway of Fig. 13.1.

The good news is that these extensions follow similar steps we used in mixed modelling. For example, the inclusion of a random intercept in a GLM is imposing the compound symmetrical correlation structure, just as it did in the linear mixed model. In fact, just as the linear regression model is a GLM with a Gaussian distribution, so is the linear mixed model a GLMM with a Gaussian distribution.

When there is good news, there is often some bad news. And the bad news is that GLMM and GAMM are on the frontier of statistical research. This means that available documentation is rather technical, and there are only a few, if any, textbooks aimed at ecologists. There are multiple approaches for obtaining estimated parameters, and there are at least four packages in R that can be used for GLMM. Sometimes these give the same results, but sometimes they give different results. Some of these methods produce a deviance and AIC; others do not. This makes the model selection in GLMM more of an art than a science. The main message is that when applying GLMM or GAMM, try to keep the models simple or you may get numerical estimation problems.

Fig. 13.1 Relationship between linear regression, additive modelling, mixed modelling, additive modelling, GLM, GAM, GLMM, and GAMM. The Generalised Estimation Equations is an alternative technique for the *lower right box*



The literature that we consulted for writing this chapter were almost exclusively written for medical, economical, and social science. We strongly recommend Snijders and Bosker (1999), Raudenbush and Bryk (2002), Goldstein (2003), Fitzmaurice et al. (2004), Brown and Prescott (2006), and for the GAMM Rupert et al. (2003) and Wood (2006). With some effort, you should be able to work your way through these books after reading this chapter. Luke (2004) is reasonably non-technical and can be read as an introduction. If you have good mathematical skills, we recommend McCulloch and Searle (2001) or Jiang (2007). The good news is that publications using GLMM or GAMM are now appearing more frequently in the ecological literature, e.g. Vicente et al. (2006) and Pierce et al. (2007) among others.

13.1 Setting the Scene for Binomial GLMM

In Chapter 12, we used data from Vicente et al. (2005), who looked at the distribution and faecal shedding patterns of the first-stage larvae (L1) of *Elaphostrongylus cervi* in red deer across Spain. In this chapter, we focus on the relationship between the presence and absence of *E. cervi* L1 in deer and the explanatory variables length and sex of the animal and farm identity. Because the response variable is of the form 0–1, we are immediately in the world of a binomial GLM. The following model is applied on these data:

$$\text{logit}(p_{ij}) = \alpha + \beta_1 \times \text{Length}_{ij} + \beta_2 \times \text{Sex}_{ij} + \beta_3 \times \text{Length}_{ij} \times \text{Sex}_{ij} + \beta_4 \times \text{Farm}_i$$

The notation logit stands for the logistic link (Chapter 10), p_{ij} is the probability that animal j on farm i has the parasite, Length_{ij} is the length of the deer, Sex_{ij} tells us whether it is male or female, and Farm_i identifies the farm. Because of the large number of farms, we did not include an interaction term involving the variable farm.

The following code accesses the data, defines the nominal variables as nominal, and centres length. In Chapter 12, we gave a justification for centring length.

```
> library(AED); data(DeerEcervi)
> DeerEcervi$Ecervi.01 <- DeerEcervi$Ecervi
> DeerEcervi$Ecervi.01[DeerEcervi$Ecervi>0] <-1
> DeerEcervi$fSex <- factor(DeerEcervi$Sex)
> DeerEcervi$CLength <- DeerEcervi$Length -
  mean(DeerEcervi$Length)
> DeerEcervi$fFarm <- factor(DeerEcervi$Farm)
```

The code below applies a GLM on the data, drops each allowable term in turn from the model, and applies a likelihood ratio test that is Chi-square distributed. Note that because the interaction between length and sex is included, we cannot drop the main terms CLength and fSex. The `drop1` function compares the deviance of the specified model with that of nested models. The difference between these two deviances is Chi-square distributed. The GLM model includes a farm effect, a length effect, a sex effect, and an interaction between length and sex.

```
> DE.glm<-glm(Ecervi.01 ~ CLength * fSex+fFarm,
                 data = DeerEcervi, family = binomial)
> drop1(DE.glm, test = "Chi")
Single term deletions.

Model: Ecervi.01 ~ CLength * fSex + fFarm
      Df Deviance     AIC      LRT    Pr(Chi)
<none>       745.50  799.50
fFarm        23   1003.72 1011.72   258.22 < 2.2e-16
CLength:fSex  1    755.48  807.48     9.98  0.001579
```

The first line shows the deviance for the model in which no term is dropped. Its AIC is 799.50. By dropping the nominal variable farm from this model, the deviance increases to 1011.72, a change of 258.22. The change in deviance for a binomial GLM is Chi-square distributed with 23 degrees of freedom and has a p -value that is smaller than 0.001, which means that the term is highly significant. By dropping only the interaction term, the change in deviance is 9.98, which is also significant.

To obtain insight in what the model is doing, we want to visualise the predicted values. Because the model contains a length effect, sex effect, farm effect, and an interaction between length and sex, visualisation is not trivial. The easiest option is to choose a particular farm and sex and then plot the predicted probabilities versus length. We arbitrary decided to choose sex = 1 (female). The R code below first

creates a graph in which the observed presence and absence values of the parasite *E. cervi* L1 in deer are present. This is the `plot` command. The remaining code predicts the probability of the presence of the parasite for a range of length values at particular farms. This is done with a loop; each iteration represents a farm. The `order` command is used to avoid spaghetti plots.

```
> plot(DeerEcervi$CLength, DeerEcervi$Ecervi.01,
      xlab = "Length", ylab = "Probability of \
presence of E. cervi L1", main = "Male data")
> I1 <- order(DeerEcervi$CLength)
> AllFarms <- unique(DeerEcervi$Farm)
> for (j in AllFarms){
  mydata <- data.frame(
    CLength = DeerEcervi$CLength,
    fSex = "1",
    fFarm = AllFarms[j])
  n <- dim(mydata)[1]
  if (n > 10){
    P.DE2 <- predict(DE.glm, mydata,
                      type = "response")
    lines(mydata$CLength[I1], P.DE2[I1])}}
```

The predicted values for the female data are presented in Fig. 13.2. To create this graph, we chose a particular farm and sex (see code above) and then calculated the probabilities as a (logistic) function of different length values and the chosen farm and sex. Doing this for different farms produces the multiple lines in the figure.

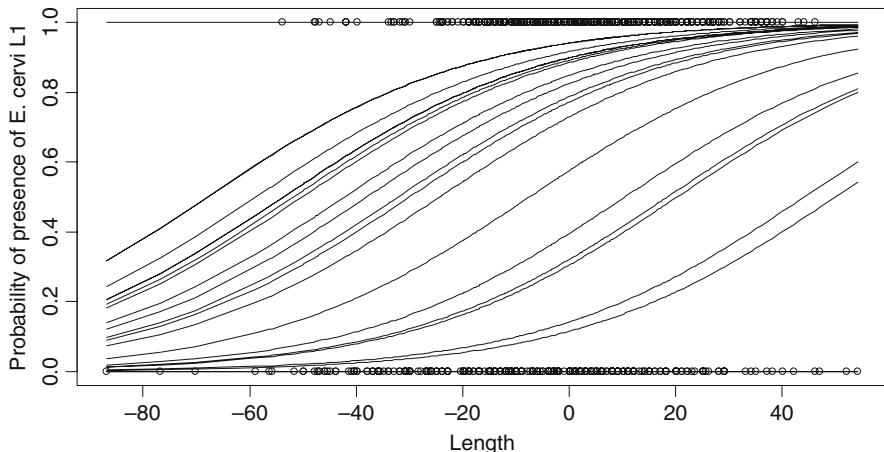


Fig. 13.2 Predicted probabilities of parasitic infection along (centred) deer length for females at all farms. Each line represents a farm

It is easy to do the same for the male data, and the graph looks similar (and is not presented here). Instead of using the `for` loop, it is also possible to do the prediction for all data at once (not farm by farm) and use the function `matlines` to do the plotting of individual curves; see its help file for examples.

The problem with this model is that the explanatory variable `farm` consumes 23 degrees of freedom and we are not even interested in knowing that there is a farm effect. We cannot drop it neither as it is highly significant. It is also possible that there is a length – farm interaction, costing another 23 parameters. The other problem is how we predict from this model. We can choose a value for `length` in Fig. 13.2, and then read off the probability for the presence of the parasite at a certain farm. But we can only do this exercise for our 24 farms. The model does not allow us to make a statement for farms in general.

This discussion should sound familiar to you, as it is identical to the discussion we had with the beach effect for the RIKZ data in Chapter 5. There, we had nine beaches, and on each beach, we had five observations. We replaced the nine parameters from beach by one random intercept and called the model a random intercept model. Now, we have 24 farms and multiple observations per farm. We can do exactly the same in a GLM, and this is discussed in the next section.

13.2 GLMM and GAMM for Binomial and Poisson Data

In this section, we apply GLMM and GAMM on two data sets; these were also used in Chapter 12. We start with the deer data we used above, followed by owl data (counts).

13.2.1 Deer Data

In Section 13.1, we applied a GLM on the deer data. We encountered two problems: The explanatory variable `farm` is using up a large number of degrees of freedom, and we can only make predictions for the current set of farms. We now use the same extension as we did for linear regression and random intercepts (Chapter 5) and work towards the GLM equivalent of a mixed model. Instead of using `farm` as a fixed effect with 24 levels, we use it as a random effect and the model becomes

$$\begin{aligned} Y_{ij} &\sim \text{Bin}(1, p_{ij}) \\ \text{logit}(p_{ij}) &= \alpha + \beta_1 \times \text{Length}_{ij} + \beta_2 \times \text{Sex}_{ij} + \beta_3 \times \text{Length}_{ij} \times \text{Sex}_{ij} + a_i \\ a_i &\sim N(0, \sigma_a^2) \end{aligned}$$

Y_{ij} is 1 if animal j on farm i has *E. cervi* L1 and 0 otherwise. The random intercept a_i is assumed to be normally distributed with mean 0 and variance σ_a^2 . If this variance is small, then the contribution from a_i is also rather small and all farms will have a similar logistic curve. On the other hand, if σ_a^2 is relatively large, then

each farm will have very different intercepts. This approach reduces the number of parameters considerably compared to using Farm as a fixed effect.

Using farm as a random intercept has another major advantage. Just as in linear mixed modelling, a random intercept model is implicitly introducing the compound symmetrical correlation structure. This implies that the probability of a deer carrying the parasite is correlated to other deer on the same farm.

There are various functions in R that can be used for GLMM; the main ones are `glmmPQL` from the MASS package, `lmer` from the `lme4` package, and `glmmML` from the `glmmML` package. Later in this section, we compare the output from all these models, but first we concentrate on the `glmmPQL` method. The following R code applies the GLMM model described above.

```
> library(MASS)
> DE.PQL <- glmmPQL(Ecervi.01 ~ CLength * fSex,
+                      random = ~ 1 | fFarm, family = binomial,
+                      data = DeerEcervi)
> summary(DE.PQL)
```

We used the object name `DE.PQL` because it reminds us of `DEer` and which tool was used (PQL, which will be discussed later in this chapter). The function `glmmPQL` is in the MASS package from Venables and Ripley (2002), and we first need to load this package. The random effect is specified in a similar way as we did for linear mixed models in Chapter 5. In fact, the only new code is the `family = binomial` option. The probability of presence of the parasite is modelled as a function of length, sex, and their interaction. The random effect farm is adding a random term to the intercept. The results of the `summary` command are given below.

```
Linear mixed-effects model fit by maximum likelihood
Data: DeerEcervi
      AIC      BIC   logLik
     NA      NA      NA

Random effects:
Formula: ~1 | fFarm
          (Intercept) Residual
StdDev:    1.462108  0.9620576

Variance function:
Structure: fixed weights
Formula: ~invwt

Fixed effects: Ecervi.01 ~ CLength * fSex
                Value Std.Error DF t-value p-value
(Intercept) 0.8883697 0.3373283 799 2.633547 0.0086
CLength     0.0378608 0.0065269 799 5.800768 0.0000
fSex2       0.6104570 0.2137293 799 2.856216 0.0044
CLength:fSex2 0.0350666 0.0108558 799 3.230228 0.0013
```

Number of Observations: 826

Number of Groups: 24

The random intercept a_i has a standard error of 1.462, and the residual standard error is 0.962. The residual standard error is for the working residuals, which are used internally and are less useful than, for example, Pearson residuals. The AIC and BIC are not defined, and we explain later why not. The interaction term is significant at the 5% level, and this means that we have to include the main terms as well. We now discuss how to interpret this output. For a female deer ($fSex = '1'$), the probability that a deer has the parasite *E. cervi* L1 is given by

$$\text{logit}(p_{ij}) = 0.888 + 0.037 \times \text{Length}_{ij} + a_i \quad a_i \sim N(0, 1.462^2)$$

The first level of the variable Sex is used as baseline; hence, the contribution from the Sex and the interaction are 0. For a male deer ($\text{Sex} = 2$), the formula is given by

$$\text{logit}(p_{ij}) = 1.498 + 0.072 \times \text{Length}_{ij} + a_i \quad a_i \sim N(0, 1.462^2)$$

The value of 1.498 is obtained by adding the contribution from the main term $fSex$ to the intercept and 0.072 is the correction for the intercept for the male species ($= 0.037 + 0.035$). Just as before, we will only visualise the results for the female deer.

The random intercept a_i is assumed to be normally distributed with mean 0 and variance 1.462^2 . This means that the majority of the values (95% to be more exact) of a_i are between -1.96×1.462 and 1.96×1.462 . Figure 13.3 shows three lines.

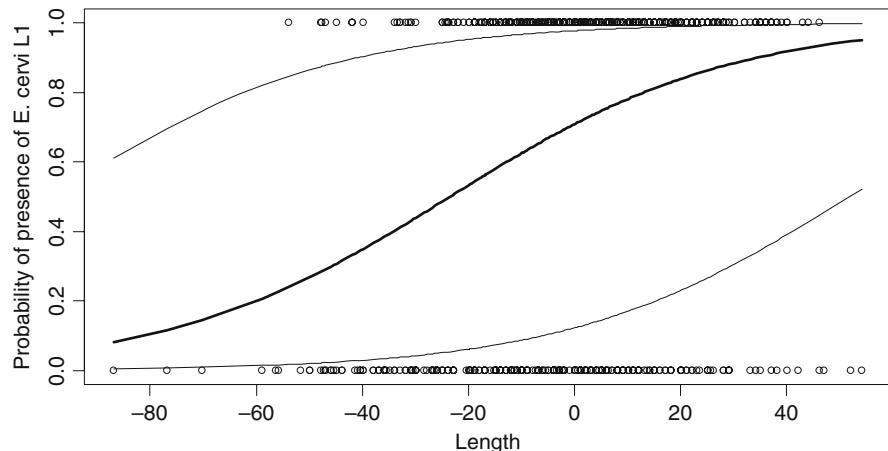


Fig. 13.3 GLMM predicted probabilities of parasitic infection along (centred) deer length for females at all farms. The *thick line* in the *middle* represents the predicted values for the ‘population of farms’, and the other two lines are obtained by adding and subtracting 1.95×1.462 for the random intercept to the predictor function. The space between these *two curves* shows the variation between the predicted values per farm

The thick line in the middle shows the estimated probabilities for a range of length values for the female data. These are predicted probabilities for a typical farm. Typical means that in this case $a_i = 0$. The other two lines are obtained by adding 1.96×1.462 to the predictor function and subtracting 1.96×1.462 from the predictor function. Hence, 95% of the farms have logistic curves between these two extremes. The interpretation of the graph is as follows. Go to a typical farm and sample a deer of average length (Length = 0). It has a probability of approximately 0.7 of having the parasite (this value is taken from the curve for the population). However, depending on which particular farm we visit, for the majority of farms this probability can be anything between 0.1 and 0.9! So, there is considerable between-farm variation. At this stage, it should be emphasised that the model can still be improved.

The code to produce the graph is as follows.

```
> g <- 0.8883697 + 0.0378608 * DeerEcervi$CLength
> p.averageFarm1 <- exp(g) / (1 + exp(g))
> I1 <- order(DeerEcervi$CLength) #Avoid spaghetti plot
> plot(DeerEcervi$CLength, DeerEcervi$Ecervi.01,
       ylab = "Probability of presence of E. cervi L1",
       xlab = "Length")
> lines(DeerEcervi$CLength[I1], p.averageFarm1[I1], lwd=3)
> p.Upp<-exp(g+1.96*1.462108)/(1+exp(g+1.96*1.462108))
> p.Low<-exp(g-1.96*1.462108)/(1+exp(g-1.96*1.462108))
> lines(DeerEcervi$CLength[I1], p.Upp[I1])
> lines(DeerEcervi$CLength[I1], p.Low[I1])
```

The first two lines calculate the predicted probabilities for the curve in the middle. Instead of using some complex programming, we calculated these manually. The `order` command is used to avoid a spaghetti plot. The rest of the code calculates the probabilities for the other two curves and superimposes the lines.

We mentioned earlier in this section that the GLMM can be run in at least two other libraries, and we now briefly discuss the code and the output. The mathematical details and the reason why we have different functions are discussed in Section 13.4.

The second function you can use for GLMM is the `lmer` function from the package `lme4`. The following code runs exactly the same model as before.

```
> library(lme4)
> DE.lme4 <- lmer(Ecervi.01 ~ CLength * fSex +
                     (1 | fFarm), family = binomial,
                     data = DeerEcervi)
> summary(DE.lme4)
```

The random effect is now specified by $(1 | fFarm)$. We only present the results and compare it with the `glmmPQL` results towards the end of this section.

```

Generalized linear mixed model fit using Laplace
Formula: Ecervi.01 ~ CLength * fSex + (1 | fFarm)
Data: DeerEcervi. Family: binomial(logit link)

      AIC    BIC   logLik  deviance
832.6 856.1 -411.3     822.6

Random effects:
Groups Name        Variance Std.Dev.
fFarm  (Intercept) 2.3859   1.5446
number of obs: 826, groups: fFarm, 24

Estimated scale (compare to 1 ) 0.9684129

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.941504  0.354827  2.653  0.00797
CLength      0.038975  0.006815  5.719 1.07e-08
fSex2        0.624665  0.222848  2.803  0.00506
CLength:fSex2 0.035866  0.011348  3.161  0.00157

```

The standard error of the random intercepts a_i is now 1.54. The main difference between lmer and glmmPQL is that the lmer gives an AIC, BIC, log likelihood value, and a deviance. This makes model comparison with lmer easier. Standard errors, z -values, and p -values obtained by both methods are similar.

The last option we discuss is the glmmML function in the package with the same name. This package is extensively used in Chapter 21, where the presence and absence of koalas are analysed using a binomial GLMM. The following R code can be used.

```

> library(glmmML)
> DE.glmmML <- glmmML(Ecervi.01 ~ CLength * fSex,
+                         cluster = fFarm, family = binomial,
+                         data = DeerEcervi)
> summary(DE.glmmML)

```

In this function, the random intercept is specified with the option `cluster = fFarm`. Its output is given below. Again, we get an AIC and estimated values are similar to the other two functions, except for the residual standard error.

```

Call: glmmML(formula = Ecervi.01 ~ CLength * fSex,
family = binomial, data = DeerEcervi, cluster = fFarm)

            coef se(coef)      z Pr(>|z|)
(Intercept) 0.93968 0.357915 2.625 8.65e-03
CLength      0.03898 0.006956 5.604 2.10e-08
fSex2        0.62451 0.224251 2.785 5.35e-03
CLength:fSex2 0.03586 0.011437 3.135 1.72e-03

```

```
Standard deviation in mixing distribution: 1.547
Std. Error: 0.2975

Residual deviance: 822.6 on 821 degrees of freedom
AIC: 832.6
```

13.2.1.1 Comparison of Results

Let us now compare the results from the functions `glmmPQL`, `lmer`, and `glmmML`. For convenience, we have reproduced all estimated regression parameters and standard errors in Table 13.1. We have also added the binomial GLM and GEE results.

Note that the `lmer` and `glmmML` results are nearly the same. The `glmmPQL` method also gives very similar results. As can be expected, the GLM obtained without any correlation structure gives slightly different results; note the different sex estimate. Except for the intercept, the GEE results are also similar to the GLMM results. Further comments comparing GEEs with GLMMs can be found on p. 300 of Venables and Ripley (2002). They also mentioned the package `glme`, which apparently can do a GLMM and fix the overdispersion to a pre-set value (`glmmPQL` automatically estimates overdispersion, also if you do not want this).

Finally, we comment on the different interpretation of the parameters in a GLMM and GEE. In the GLMM in Fig. 13.3, the thick line is the length effect of a *typical* farm. Hence, the regression parameters in the GLMM are with respect to an individual farm due to the random intercept a_i . For the GEE, the regression parameters represent the effect of the population.

Table 13.1 Estimated regression parameters and standard errors obtained by `glm`, `glmPQL`, `lmer`, `glmmML`, and GEE. Note that further differences can be obtained by changing the estimation methods within a function

	Estimates	SE		Estimates	SE
Glm					
Intercept	0.652	0.109	lmer	0.941	0.354
Length	0.025	0.005	Intercept	0.038	0.006
Sex	0.163	0.174	Length	0.624	0.222
Length × Sex	0.020	0.009	Sex	0.035	0.011
glmmPQL					
Intercept	0.888	0.337	glmmML	0.939	0.357
Length	0.037	0.006	Intercept	0.038	0.006
Sex	0.610	0.213	Length	0.624	0.224
Length × Sex	0.035	0.010	Sex	0.035	0.011
GEE					
Intercept	0.773	0.280			
Length	0.030	0.006			
Sex	0.476	0.217			
Length × Sex	0.027	0.014			

13.2.2 The Owl Data Revisited

In Chapters 5, 6, and 12, we used a data set from Roulin and Bersier (2007), who analysed the begging behaviour of nestling barn owls. In Chapters 5 and 6, we analysed the response variable sibling negotiation, which is defined as the number of calls just before arrival of a parent at a nest divided by the number of siblings per nest. The data were log-transformed and a Gaussian linear mixed effects model was applied, and also an additive mixed effects model with arrival time as smoother. In Chapter 5, we used nest as random effect, and in Chapter 6 an auto-regressive correlation structure was implemented. In Chapter 12, we analysed the number of calls using a GLM with a Poisson distribution (number of calls is a count) and the log-transformed number of siblings per nest was used as an offset variable in the linear predictor function. Two GEE models were applied: a GEE with the compound correlation structure between all observations from the same nest and one GEE with an auto-regressive correlation between sequential observations from the same nest per night. Here, we will analyse these data in yet another way, namely, with a GLMM using the Poisson distribution (number of calls is a count) and also with a GAMM.

The Poisson GLMM for these data is given by the following:

$$\begin{aligned} \text{NCalls}_{is} &\sim \text{Poisson}(\mu_{is}) \Rightarrow E(\text{NCalls}_{is}) \sim \mu_{is} \\ \eta_{is} &= \text{offset(LBroodSize}_{is}) + \beta_1 \times \text{SexParent}_{is} + \beta_2 \times \text{FoodTreatment}_{is} \\ &\quad + \beta_3 \times \text{ArrivalTime}_{is} + \beta_4 \times \text{SexParent}_{is} \times \text{FoodTreatment}_{is} \\ &\quad + \beta_5 \times \text{SexParent}_{is} \times \text{ArrivalTime}_{is} + a_i \\ a_i &\sim N(0, \sigma_a^2) \\ \log(\mu_{is}) &= \eta_{is} \end{aligned}$$

The first line states that the number of calls for observation s at nest i , NCalls_{is} , is Poisson distributed with mean μ_{is} . The linear predictor function looks similar to that of an ordinary Poisson GLM, except that we use the log transformed broodsize as an offset (Chapter 9), and there is the a_i bit at the end. Its purpose is exactly the same as the random intercept for farm in Section 13.2.1; it allows for a different intercept for each nest. We assume that it is normally distributed with mean 0 and variance σ_a^2 . We use `lmer` to fit the model. The same model in terms of explanatory variables is used as in Chapters 5, 6, and 12. The following code was used.

```
> library(AED) ; data(Owls)
> library(nlme)
> Owls$NCalls <- Owls$SiblingNegotiation
> Owls$LBroodSize <- log(Owls$BroodSize)
> Owls$fNest <- factor(Owls$Nest)
> O1.lmer <- lmer(NCalls ~ offset(LBroodSize) +
+ SexParent * FoodTreatment +
```

```

SexParent * ArrivalTime + (1 | fNest),
  data = Owls, family = poisson)
> summary(O1.lmer)

Generalized linear mixed model fit using Laplace
Formula: NCalls ~ offset(LBroodSize) + SexParent *
FoodTreatment + SexParent * ArrivalTime + (1 | fNest)
Data: Owls
Family: poisson(log link)
AIC  BIC logLik deviance
3329 3359 -1657     3315

Random effects:
Groups Name      Variance Std.Dev.
fNest  (Intercept) 0.20980  0.45803
number of obs: 599, groups: fNest, 27

Estimated scale (compare to 1) 2.332117

Fixed effects:
Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.58145  0.36262  9.877 <2e-16
SexParentMale 0.38785  0.44861  0.865  0.3873
FoodTreatmentSatiated -0.66680  0.05610 -11.886 <2e-16
ArrivalTime   -0.11948  0.01440  -8.298 <2e-16
SexParentMale:FoodTreatment.Sat 0.13239  0.07044  1.880  0.0602
SexParentMale:ArrivalTime    -0.01647  0.01836  -0.897  0.3697

Correlation of Fixed Effects:
(Intr) SxPrnM FdTrtS ArrvlT SPM:FT
SexParentMl -0.739
FdTrtmntStt -0.077  0.062
ArrivalTime -0.964  0.759  0.017
SxPrntM:FTS  0.055 -0.073 -0.767 -0.010
SxPrntMl:AT  0.737 -0.995 -0.012 -0.765  0.012

```

The correlation between the intercept and the slope for arrival time is rather large (-0.964). This is because arrival time was not centred. In case of numerical problems, centring continuous variables may help. The model can be further simplified because the interaction between sex of the parent and arrival time is not significant. You can reach the same conclusion by dropping this interaction, refitting the model, and comparing the change in likelihood.

```

> O2.lmer <- lmer(NCalls ~ offset(LBroodSize) +
  SexParent * FoodTreatment +
  ArrivalTime + (1 | fNest), data = Owls,
  family = poisson)
> anova(O1.lmer, O2.lmer)

Models:
O2.lmer: NCalls ~ offset(LBroodSize) + SexParent * FoodTreatment +
  SexParent + ArrivalTime + (1 | fNest)

```

```
O2.lmer: NCalls ~ offset(LBroodSize) + SexParent * FoodTreatment +
          SexParent * ArrivalTime + (1 | fNest)

      Df      AIC      BIC  logLik   Chisq Chi Df Pr(>Chisq)
O2.lmer  6  3327.4  3353.7 -1657.7
O1.lmer  7  3328.6  3359.3 -1657.3  0.8029       1     0.3702
```

You can repeat this process and drop the second two-way interaction as it is not significant neither and the same holds for the main term sex of the parent. This means that we end up with a GLMM that only contains the two main terms arrival time and food treatment, and nest as random effect. The code and relevant numerical output is given below.

```
> O3.lmer <- lmer(NCalls ~ offset(LBroodSize) +
                     FoodTreatment + ArrivalTime + (1 | fNest),
                     data = Owls, family = poisson)
> summary(O3.lmer)

Generalized linear mixed model fit using Laplace
Formula: NCalls ~ offset(LBroodSize) + FoodTreatment + ArrivalTime +
(1 | fNest)
Data: Owls
Family: poisson(log link)
AIC  BIC logLik deviance
3328 3346 -1660     3320

Random effects:
Groups Name        Variance Std.Dev.
fNest  (Intercept) 0.20854  0.45666

number of obs: 599, groups: fNest, 27
Estimated scale (compare to 1) 2.331403

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.802867  0.243846  15.60  <2e-16
FoodTreatmentSatiated -0.589608  0.035941  -16.41  <2e-16
ArrivalTime   -0.128840  0.009258  -13.92  <2e-16
```

You can continue the analysis by trying to add a random slope for arrival time or even a generalised additive mixed model in which arrival time is fitted as a smoother. The latter model is specified by

$$\begin{aligned} \text{NCalls}_{is} &\sim \text{Poisson}(\mu_{is}) \Rightarrow E(\text{NCalls}_{is}) \sim \mu_{is} \\ \eta_{is} &= \text{offset}(L\text{BroodSize}_{is}) + \beta_1 \times \text{FoodTreatment}_{is} + s(\text{ArrivalTime}_{is}) + a_i \\ \log(\mu_{is}) &= \eta_{is} \\ a_i &\sim N(0, \sigma_a^2) \end{aligned}$$

Arrival time is now fitted with a smoother. To implement this model in R, we need the gamm function from the mgcv package.

```
> library(mgcv)
> O4.gamm <- gamm(NCalls ~ offset(LBroodSize) +
+ FoodTreatment + s(ArrivalTime),
+ random = list(fNest =~ 1), data = Owls,
+ family = poisson)
```

The object `O4.gamm` has two items, a `$gam` and a `$lme` bit. Using the words from the `gamm` help files, some of the output in the `$lme` looks rather bizarre. Let us start easy with the `$gam` part. We can use the following commands:

```
> summary(O4.gamm$gam, cor = FALSE)
> anova(O4.gamm$gam)
> plot(O4.gamm$gam)
```

We only present the output of the first command as the second one shows merely a condensed version of it (it is useful if you have nominal variables with more than two levels).

```
Family: poisson. Link function: log
Formula: NCalls ~ offset(LBroodSize) + FoodTreatment + s(ArrivalTime)

Parametric coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.60731 0.07716 7.870 1.70e-14
FoodTreatmentSatiated -0.57624 0.07949 -7.249 1.32e-12

Approximate significance of smooth terms:
edf Est.rank F p-value
s(ArrivalTime) 6.781 9 9.724 6.23e-14

R-sq.(adj) = 0.211 Scale est. = 5.1031 n = 599
```

The scale estimator is the variance of the working residuals inside the algorithm. The information on the parametric coefficients tells us that the food treatment is significantly different from 0 at the 5% level. To be more specific, observations that received the satiated treatment had an intercept that is 0.57 lower than for food-deprived nests. The arrival time smoother had 6.7 degrees of freedom, and is significant. The `plot` command presents this smoother, see Fig. 13.4. Note that the shape of the smoother is very similar to the one in Fig. 5.8! In order to get the fitted values for a *typical* observation, we need to add the intercept (0.607), the food treatment effect (-0.576 for satiated observations), and the offset.

Finally, let us focus on the `$lme` part of the output; it is a little intimidating though! This reason for this is that `gamm` is repeatedly calling `glmmPQL` if a non-Gaussian distribution, or non-identity link function, is used. For Gaussian distributions with the identity link, it calls `lme`. Now here is the confusing bit: It's possible to show that the smooth terms of the GAMM can be presented in the mixed-effects form (Wood, 2006, p. 317), namely, $\mathbf{X}_F \times \boldsymbol{\beta}_F + \mathbf{Z} \times \mathbf{b}$, where \mathbf{X}_F is a matrix

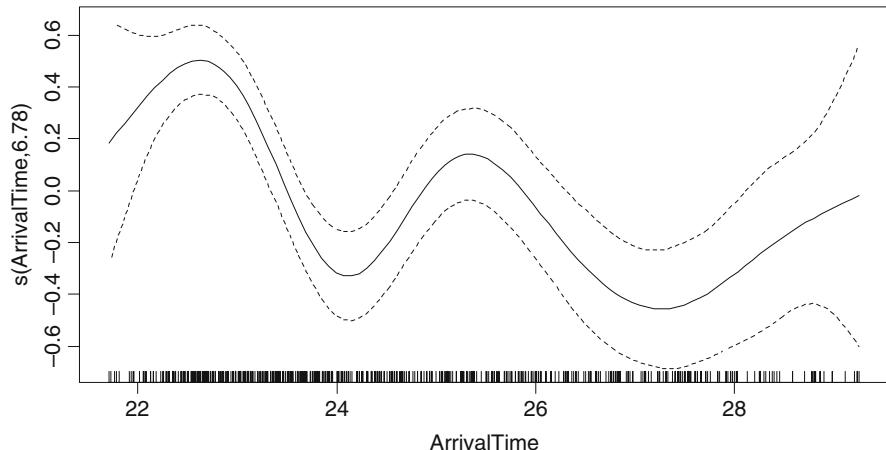


Fig. 13.4 Estimated smoother for the GAMM. Note that the smoother is centred around zero. To get fitted values, you need to add the intercept, food treatment effect, the offset, and the contribution from the random effect for a nest. The smoother shows two bumps: one at 22.30 and one at about 01.30 (in real time). An explanation can be sought in the biology, but before you do this, you need to exclude the possibility that there is still somehow a nest effect in here. Perhaps the bumps are due to activity at only a group of nests during parts of the night. The random intercept will take care of changes in mean values of the number of calls per nest, but not of changes *in the relationship* between arrival time and calls at different nests. Make boxplots of nest activity during the night (are owls active during the entire night or only part of the night), and inspect the residuals from a random intercept and slope GLMM for any patterns

containing the smoother basis; see Chapter 3. \mathbf{Z} is a matrix containing the random effects (Chapter 5) derived from the smoother basis and penalty matrix (presenting the penalty as a quadratic form) and \mathbf{b} are the random effects, which are assumed to be normally distributed with mean 0 and variance \mathbf{I}/λ . Hence, the GAMM is written in parametric terms and the penalty λ , also called the wiggly component in Wood (2006), is used in the random component. This makes the `lme` summary part rather bizarre; see below.

```
> summary(O4.gamm$lme)

Linear mixed-effects model fit by maximum likelihood
Data: strip.offset(mf)
      AIC      BIC logLik
      NA      NA      NA

Random effects:
Formula: ~Xr.1 - 1 | g.1
Structure: pdIdnot
          Xr.11    Xr.12    Xr.13    Xr.14    Xr.15    Xr.16
Xr.17    Xr.18
StdDev: 19.57691 19.57691 19.57691 19.57691 19.57691 19.57691
19.57691 19.57691
```

```

Formula: ~1 | fNest %in% g.1
          (Intercept) Residual
StdDev:   0.2935566 2.259006

Variance function:
Structure: fixed weights
Formula: ~invwt

Fixed effects: y ~ X - 1 + offset(LBroodSize)
                Value Std.Error DF t-value p-value
X(Intercept)      0.6073122 0.0771576 570 7.871062 0.0000
XFoodTreatmentSatiated -0.5762376 0.0795368 570 -7.244919 0.0000
Xs(ArrivalTime)Fx1      0.6378219 0.5925502 570 1.076401 0.2822

Correlation:
            X(Int) XFdTrS
XFoodTreatmentSatiated -0.365
Xs(ArrivalTime)Fx1      -0.050  0.058

Standardized Within-Group Residuals:
      Min        Q1        Med        Q3        Max
-1.5701868 -0.7615271 -0.2231992  0.5589554  4.9173689

Number of Observations: 599
Number of Groups:
g.1 fNest %in% g.1
      1             27

```

The interesting bit from this output is the variance of the random intercept for nests; it is equal to 0.293². The residual standard deviation (of the working residuals) was also presented earlier using `summary(O4.gamm$gam)`, except that it was presented as a variance. Because the `glmmPQL` routine is used, no AIC is given. The random effects part gives information on I/λ . It is probably easier to obtain this via

```
> intervals(O4.gamm$lme, which = "var-cov")
```

```

Approximate 95% confidence intervals
Random Effects:
Level: g.1
      lower     est.     upper
sd(Xr.1 - 1) 98.3855 383.2554 1465.705
Level: fNest
      lower     est.     upper
sd((Intercept)) 0.1911345 0.2935566 0.4508631

Within-group standard error:
      lower     est.     upper
2.130732 2.259006 2.395004

```

The 383.255 is the square of 19.57², which we already met in the `lme` summary output. To be precise, 383.255 is equal to σ^2/λ , where σ^2 is the variance of the (working) residuals. This gives $\lambda = 2.259^2/383.255 = 0.013$. However, we already know the amount of smoothing from the `anova(O4.gamm$gam)` command; hence, this is probably not worthwhile to mention in a report or paper, unless you want to focus on the *approximate* confidence intervals.

The information in the summary `lme` output on the fixed effects bit is not interesting either; just use the `anova(O4.gamm$gam)` command for clearer information on the significance on individual terms. Further details can be found in Sections 6.5–6.7 in Wood (2006). He also presented residual plots, where the residuals were obtained from the `$lme` bit. For our model, these are obtained via

```
> E4 <- resid(O4.gamm$lme, type = "normalized")
```

These take into account the random effects. You can plot these residuals versus arrival time, and see whether there is any auto-correlation structure left in the data. If there is, try adding an auto-regressive correlation within a specific nest and day combination; see also Chapter 6. Other interesting validation tools are to plot square-root-transformed fitted values versus square-root-transformed observed values (should be a straight line), Pearson residuals versus square-root-transformed fitted values (should form a band with no patterns), and raw residuals (observed versus fitted values) versus square-root-transformed fitted values (should show a clear cone); see also Fig. 6.11 in Wood (2006) and associated code.

13.2.3 A Word of Warning

Although the analyses presented in the previous two subsections look relatively simple, you should not be too enthusiastic with all the *p*-values, AICs, and nested model comparisons. All these values are rather approximate! Furthermore, at the time of writing, the `lme4` package was under development with no support for the `correlation` argument in Poisson GLMMs, and the `resid` function did not give residuals for a Poisson GLMM. Type: `resid(O3.lmer)`; it gives: `Error: 'resid' is not implemented yet`. This does not mean that a package that does give residuals for a Poisson GLMM is any better; it is just that it is not trivial to calculate them.

Summarising, you should be very careful with *p*-values close to magic 5% borderline in GLMMs and GAMMs, even more careful as in ordinary GLMs and GAMs.

13.3 The Underlying Mathematics in GLMM

This section may be skipped by readers not interested in the underlying mathematical details. In this section, we first explain the difference between conditional

and unconditional distributions, then present the likelihood function for the GLMM models, and finally discuss how it is calculated.

The difference between conditional and unconditional distributions is best explained within a Gaussian context. Recall from Chapter 5 that the linear mixed model is given by

$$Y_{ij} = \alpha + X_i \times \beta + Z_i \times b_i + a_i + \varepsilon_{ij}$$

Note that the random term b_i is assumed to be normally distributed with a mean of 0 and covariance matrix \mathbf{D} (actually, in this case, \mathbf{D} just contains one element, but it is easier to use matrix notation as it is more general). The same holds for the second random term ε_{ij} . Its covariance matrix is given by $\boldsymbol{\Sigma}_i$. The mean value of Y_{ij} , where the mean is taken over all observations i , is given by

$$E(Y_{ij}) = X_i \times \beta.$$

We now introduce a new concept: the conditional mean of Y_{ij} . It is the mean value of Y_{ij} for given b_i . So, we pretend we know the value of b_i . The mathematical notation for this is $E(Y_{ij}|b_i)$. The vertical line followed by b_i means that it is ‘conditional on b_i ’. Its value is given by

$$E(Y_{ij}|b_i) = X_i \times \beta + Z_i \times b_i$$

We can do the same for the variance of Y_{ij} . The conditional variance of Y_{ij} is given by $\text{cov}(Y_{ij}|b_i) = \boldsymbol{\Sigma}_i$, and the unconditional variance is

$$\text{cov}(Y_{ij}) = \mathbf{Z}_i \times \mathbf{D} \times \mathbf{Z}'_i + \sum$$

The principle of conditional mean and variances can be extended to distributions. Hence, we can specify a Poisson or Binomial distribution conditional on b_i . This allows us to define a Poisson GLMM with the following three steps.

1. Conditional on the random effects b_i , the counts Y_{ij} are assumed to be Poisson distributed with mean $\mu_{ij}|b_i$. As a consequence, we have the following relationship between the mean and the variance of Y_{ij} : $E(Y_{ij}|b_i) = \text{var}(Y_{ij}|b_i)$.
2. The relationship between the conditional mean and the explanatory variables is determined by the log link $\log(\mu|b_i) = \alpha + X_i \times \beta + Z_i \times b_i$.
3. The random effects b_i are assumed to be normally distributed with mean 0 and covariance matrix \mathbf{D} .

The only difference with an ordinary Poisson GLM model is the specification of a conditional distribution, and the inclusion of the random term. The Binomial GLMM can be defined in a similar way, namely,

1. Conditional on the random effects b_i , the presence and absence data Y_{ij} are assumed to be binomial distributed with probability $p_{ij}|b_i$. As a consequence,

we have the following relationship between the mean and the variance of Y_{ij} :

$$\text{E}(Y_{ij}|b_i) = p_{ij}|b_i \text{ and } \text{var}(Y_{ij}|b_i) = p_{ij}|b_i \times (1 - p_{ij}|b_i)$$

2. The relationship between the conditional mean and the explanatory variables is determined by the logistic link $\text{logit}(p_{ij}|b_i) = \alpha + X_i \times \beta + Z_i \times b_i$.
3. The random effects b_i are assumed to be normally distributed with mean 0 and covariance matrix D.

It is also possible to formulate the GLMM in an abstract formulation using the same formula as presented in Chapter 9. See, for example, Fitzmaurice et al. (2004). We do not do that here. So far, the mathematics is not that intimidating. However, we have reached the point where the problems begin as we now look at the formulation of the likelihood function. In ordinary GLM models, the maximum likelihood function is specified and derivates with respect to parameters are calculated and set to 0. The parameters that maximise the likelihood are then found by solving these equations. As output, we get estimated parameters, standard errors, a deviance, and an AIC among other information. In GLMM, this is a considerably more complicated process, and the output may not contain a deviance and AIC. The reason for this is that the likelihood function for the GLMM has the following form

$$L(\beta, D) = \prod_i \int f(Y_{ij}|b_i) \times f(b_i) db_i$$

The symbol that looks like two vertical roman pillars with one horizontal pillar on top of it represents a multiplication operator. The second one is the integral. The terms $f(Y_{ij})$ and $f(b_i)$ are distribution functions. As explained above, in ordinary GLM models, we take the derivative of $L()$ with respect to the parameters, and after some basic algebra and an iterative algorithm, we end up with the parameters. This process does not work well for the GLMM and there are no simple solutions for the parameter estimates.

One option is to use numerical integration techniques and replace the integral by a summation. This is called Gaussian quadrature. But various choices have to be made in this process, and the higher the requested accuracy of the solutions, the higher the computational burden. For complicated models, it may not converge at all. Chapter 10 in McCulloch and Searle (2001) describes a series methods for getting parameter estimates, for example, numerical quadrature (numerical integration of the integral) with various flavours like Markov chain Monte Carlo algorithms, stochastic approximation algorithms, simulated maximum likelihood, and penalised quasi-likelihood (PQL) methods. Key concepts in the last approach are Laplace's approximation and Taylor series expansions. To fully understand what these different methods are doing requires a high degree of mathematical knowledge.

The main message to take away from this is the difficulty in obtaining parameter estimates in GLMM. It depends on the package and method used. Some packages do not produce a deviance and AIC; hence, model selection is based on standard errors and Wald statistics. A few packages do produce a deviance and AIC, but interpretation should still be done with care.

A different approach to GLMM is given in Chapter 23, where we discuss Bayesian approaches.

Chapter 14

Estimating Trends for Antarctic Birds in Relation to Climate Change

A.F. Zuur, C. Barbraud, E.N. Ieno, H. Weimerskirch, G.M. Smith,
and N.J. Walker

14.1 Introduction

The earth's climate is changing rapidly and these changes are expected to affect the structure and functioning of ecosystems. It is now clearly established that recent climate changes have impacted on living organisms. Several studies have demonstrated changes in population abundance, geographic distribution, and even microevolutionary changes in relation to climatic fluctuations (Parmesan, 2006).

Perhaps the best documented and most spectacular responses of living organisms to climate change are changes in phenology, which is the timing of seasonal activities of biological events such as the sprouting of plants. The vast majority of studies from the Northern Hemisphere that have analysed the relationships between long-term phenological and climate data sets have reported an advance in spring activities. For example, the earlier arrival and reproduction of migratory birds or earlier breaking of leaf buds since the mid-20th century in response to increasing temperatures. Some studies have also reported early onset of autumn activities such as grape-harvesting dates. However, due to the scarcity of long-term data sets, phenological changes are poorly documented in the Southern Hemisphere, particularly in Antarctica. Nevertheless, it is crucial to know whether, and to what extent, phenological changes have also occurred in the Southern Hemisphere for at least two reasons: (i) climatic changes between both hemispheres are different and (ii) we need to understand and eventually predict the impact of future climatic changes on species and ecosystems.

Permanent human occupation of the Antarctic continent is very recent compared to the other continents, and the landmark for scientific studies in Antarctica is the International Polar Year 1957–58 when most of the existing permanent research stations were built. In Terre Adélie, East Antarctica, the Dumont d'Urville research station was established during the mid 1950s, and since then, ornithologists have over wintered almost every year recording arrival and laying dates of Antarctic seabirds as part of long-term studies on Antarctic marine top predators (Barbraud

A.F. Zuur (✉)

Highland Statistics Ltd., Newburgh, AB41 6FN, United Kingdom

Fig. 14.1 Emperor Penguin. The photograph was taken by C. Barbraud



and Weimerskirch, 2006). Fortunately, all but one of the Antarctic seabird species breed close to the research station and records of phenological data have been collected over a 50-year period with quasi-annual frequency.

Here, we use arrival and laying dates of three of these bird species to estimate trends and determine the effects of possible explanatory variables.

The Emperor Penguin *Aptenodytes forsteri* is the largest of the existing penguins (males weigh up to 45 kg) and breeds in winter on solid sea ice (Fig. 14.1). Males and females arrive on the breeding area from mid to late March. During the next two months, pairs form and the female lays the single egg to her male partner, who will incubate during the next two months in the heart of winter. Then, as with most seabirds, males and females alternate foraging trips at sea to feed, and to bring back food for the growing chick at the colony. The chicks leave the colony in early December at the onset of summer.

The Adelie Penguin *Pygoscelis adeliae* is a medium-sized penguin (c. 4.5 kg), breeding during the austral summer on rocky islands or on coastal nunataks (ice-free areas of the Antarctic continent). Adelies arrive and start building their nest just after mid-October and lay their eggs in mid November. The chicks leave the colonies in early February, just before the winter.

The Cape Petrel *Daption capense* is a small (c. 400 g) Procellariiform species and breeds during the austral summer on rocky islands. The breeding period is relatively short because birds arrive in mid October, lay their egg in late November, and the chicks are fledged in early March.

During the breeding period, the three species feed directly on krill or on fish that heavily depend on krill. Krill abundance and distribution are closely related to sea ice. After winters with extensive sea ice, adult krill survival and krill recruitment is high; therefore, krill abundance is higher after winters with extensive sea ice compared to winters with poor sea ice.

14.1.1 Explanatory Variables

During the breeding period, satellite tracking and diet studies have shown that all three species are more or less associated with sea ice. Both penguin species use sea

ice floes as resting platforms and forage within the pack ice. And although Cape Petrels do not forage directly in sea ice habitats, they feed in areas of open water that are covered by sea ice during winter. Consequently, you might hypothesise that sea ice extent can affect the breeding ecology of these species, either indirectly through an impact on the abundance of their food resources or directly through food resources availability. Therefore, we used sea ice extent as a candidate explanatory variable for trends in arrival and laying dates. Because our phenological data starts in the early 1950s, and sea ice extent data derived from satellite observations are only available from the early 1970s, we used a proxy of sea ice extent recently developed for East Antarctica. Methanesulphonic acid (MSA) is a product of biological activity in surface ocean water whose production is heavily influenced by the presence of sea ice in the Southern Ocean. An ice core from East Antarctica has reported a significant correlation between MSA and satellite-derived sea ice extent, and this calibration applied to longer term MSA data has permitted to reconstruct sea ice extent since the mid-nineteenth century.

The other candidate variable considered here is the Southern Oscillation Index (SOI), which represents the El Niño Southern Oscillation conditions. High, positive values of SOI indicate La Niña conditions and low negative values indicate El Niño conditions. Many studies have shown that the El Niño Southern Oscillation (and therefore SOI) impacts on demographic rates and food resources of many animals, including seabirds. In addition, contrary to the proxy of sea ice extent, SOI is a large scale climate index that may affect seabirds, both during the breeding and non-breeding season.

At present, very little is known about the at sea distribution during the non breeding period of the three seabird species considered here. Anecdotal observations suggest that both penguin species migrate north of their colonies, but remain within Antarctic waters close to the pack ice, and that Cape Petrels migrate in sub-Antarctic and sub-tropical waters. During the breeding period, both penguin species forage within the pack ice up to 150 km from the colonies but nothing is known for the Cape Petrel.

The aim of this case study is to (i) estimate trends in the arrival and laying dates in the three bird species, (ii) analyse the differences between arrival and laying dates, and (iii) determine the effects of possible explanatory variables (e.g. ice cover and the Southern Oscillation Index).

14.2 Data Exploration

Figure 14.2 contains an `xypplot` from the lattice package, showing the patterns over time in arriving and laying dates for the three bird species. To aid visual interpretation, we added a LOESS smoothing curve (Chapter 3) with a span width of 0.5 in each panel. The question addressed in this chapter is whether there is a significant trend in each series. The shape of the LOESS smoothers suggests that something is going on, but there are two main problems for these data. In principle, we have time series data; the timing of arrival in a certain year may depend on the timing in the previous year. The same holds for laying dates. This means that we should take

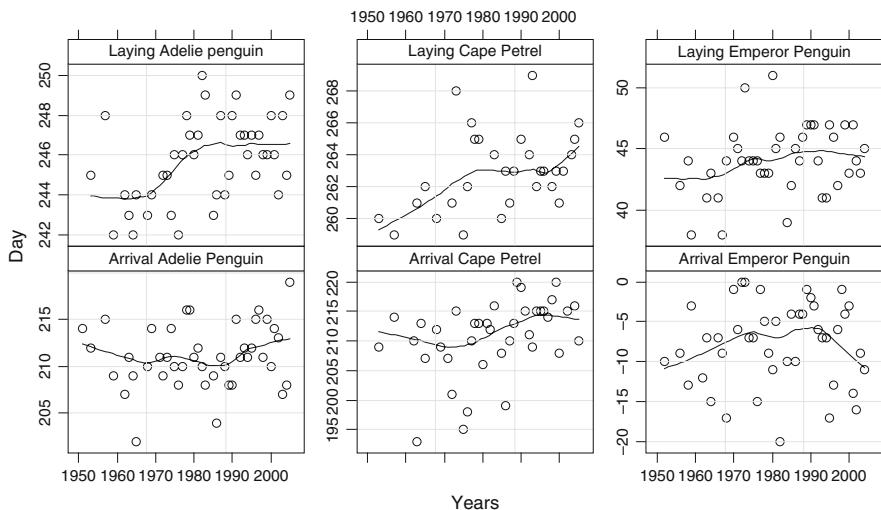


Fig. 14.2 Time series of arrival dates, laying dates, and the difference between laying and arrival dates of three bird species. A LOESS smoother with a span width of 0.5 was added to aid visual interpretation

into account the auto-correlation in the data. Preliminary graphs using the auto-correlation function (Chapter 6) showed that some time series have a significant, albeit weak, auto-correlation with a lag of 1 year. One example is given in Fig. 14.3, which shows that laying dates of the Adelie Penguin in year s is weakly related to those in year $s - 1$ (the auto-correlation with time lag 1 is significantly different from 0 at the 5% level). The other issue with the LOESS smoother is the span width (Chapter 3). If we increase it, we get a less smooth curve and decreasing the span width means that we end up with a more rapidly changing trend.

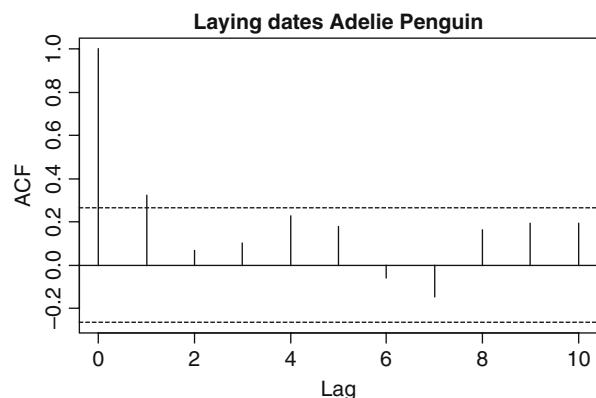


Fig. 14.3 Auto-correlation function of the laying dates of the Adelie Penguin. The horizontal axis shows the time lags and the vertical axis the correlation. The dotted line represents the 95% confidence bands

The smoothers in Fig. 14.2 suggest that the laying dates of the Adelie Penguin and Cape Petrel have increased since the mid-1970s. The question is now whether this is indeed the case or whether the smoother is misleading. As explained in Chapter 3, the smoother can be misleading in two ways: (i) by using the wrong amount of smoothing and (ii) by ignoring potential auto-correlation structures. The aim of this chapter is to eliminate these problems.

The following R code was used to generate Fig. 14.2.

```
> library(AED); data(Antarcticbirds)
> ABirds <- Antarcticbirds #saves some space
> library(lattice)
> Birds <- c(ABirds$ArrivalAP, ABirds$LayingAP,
  ABirds$ArrivalCP, ABirds$LayingCP,
  ABirds$ArrivalEP, ABirds$LayingEP)
> AllYears <- rep(ABirds$Year, 6)
> MyNames<-c("Arrival Adelie Penguin",
  "Laying Adelie penguin", "Arrival Cape Petrel",
  "Laying Cape Petrel", "Arrival Emperor Penguin",
  "Laying Emperor Penguin")
> ID1 <- factor(rep(MyNames, each=length(ABirds$Year)),
  levels = c(MyNames[1], MyNames[3], MyNames[5],
  MyNames[2], MyNames[4], MyNames[6]))
> xyplot(Birds ~ AllYears | ID1, xlab="Years",
  ylab = "Day", layout = c(3, 2), data = ABirds,
  strip = function(bg = 'white', ...),
  strip.default(bg = 'white', ...),
  scales = list(alternating = TRUE,
    x = list(relation = "same"),
    y = list(relation = "free")),
  panel = function(x, y){
    panel.xyplot(x, y, col = 1)
    panel.loess(x, y, col = 1, span = 0.5)
    panel.grid(h = -1, v = 2)})
```

This is a rather intimidating piece of code, but the results in Fig. 14.2 make it worthwhile. Let's go over it step by step. The library and data commands were discussed in Chapter 2. The ASCII file with the data contains seven columns of data: the year and the arrival and laying times for the three species. Each series contains 55 observations. To make the `xyplot` from the lattice package, we need to store the arrival and laying dates in a single vector of length $6 \times 55 = 330$. We could have done this data editing in a spreadsheet program like Excel, but it is much easier to do this in R. We also need a vector of length 330 that contains the year of each observation. Again, we could have copied and pasted the column year six times under each other in the spreadsheet, but it is much easier in R with the `rep` command.

So, now we have the original six blocks of data in a single column. To let the `xyplot` function know the identity of the blocks, we made a nominal variable ID1 that contains the names of the variables in the six blocks. The `levels` option in the `factor` command was then used to ensure that each time series of the same birds were under each other in the graph. Again, we made use of the `rep` function. The rest of the code is the same as we used in Chapter 2: we called the `xyplot` function and specified what should be plotted along the *x* and *y* axes in each panel, labels, etc. The `scales` option ensured that each panel has a different range along the *y*-axis. Although intimidating, it is all code that we have used before.

Making the auto-correlation function in Fig. 14.3 is actually much easier. The only thing to take care of is the `na.action` option. Details of the `acf` function were discussed in Chapter 6.

```
> L.AP <- acf(ABirds$LayingAP, lag.max = 10,
               na.action = na.pass,
               main = "Laying dates Adelie Penguin")
```

Another useful data exploration tool is the `pairplot` (Fig. 14.4). This addresses whether changes in the arrival and laying dates of the same birds and of different birds are similar. The following code was used to create it.

```
> pairs(ABirds[,2:7], upper.panel = panel.smooth,
       lower.panel = panel.cor)
```

The arguments `upper.panel` and `lower.panel` in the `pairs` command call functions. We took these from the `pairs` help file in R; see `?pairs`. We needed to make some small modifications to the R functions `panel.smooth` and `panel.cor`, because we had missing values and preferred black lines for the smoothing lines. Our modified panel functions are available in the AED package. Each panel above the diagonal in Fig. 14.4 shows a scatterplot of two variables. A LOESS smoothing curve with a span of 0.66 was added. The panels below the diagonal contain the (Pearson) correlations coefficients between two variables.

The font size of a correlation is proportional to its value. The panels in the `pairplot` show that there is no strong correlation between arrival and laying dates of the same and different birds. Pairplots are also useful for identifying outliers and extreme observations, which are not present in these data. So we can avoid transformations.

With the ongoing debates on climate change in mind, it is useful to look at the difference between arrival and laying dates for each bird and its relation to the explanatory variables MSA and SOI. Figure 14.5 shows these differences in arrival and laying for the three bird species plotted against time. We have added the two explanatory variables MSA and SOI. The problem with MSA is that it has a clear trend over time, and it has a relative large number of missing values towards the end of the 1990s. To avoid collinearity problems, it is perhaps better not to use year and MSA as explanatory variables in the same model (the correlation between them

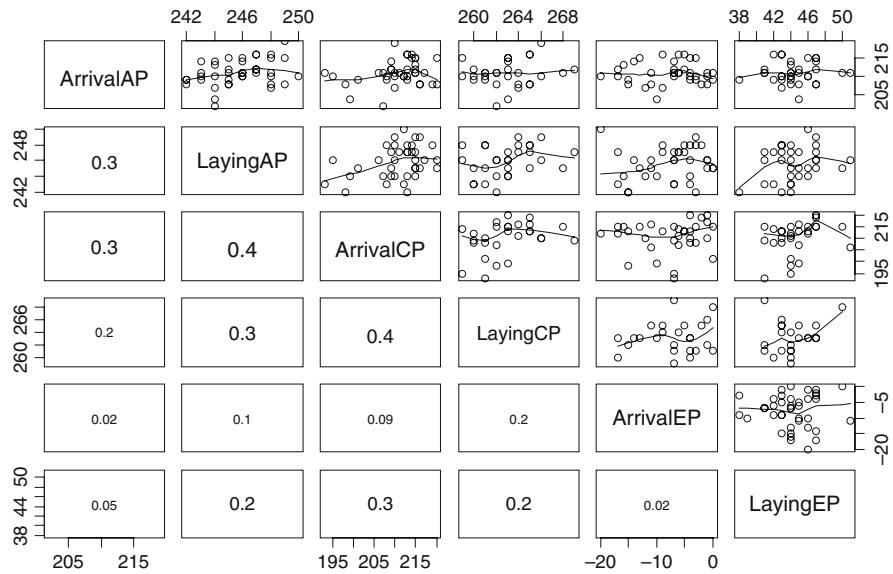


Fig. 14.4 Pairplot for the arrival and laying dates. The *lower panels* show the Pearson correlation coefficients, and the font of the correlation is proportional to its value. The *upper panels* show pair-wise scatterplot and a smoothing curve (LOESS) was added

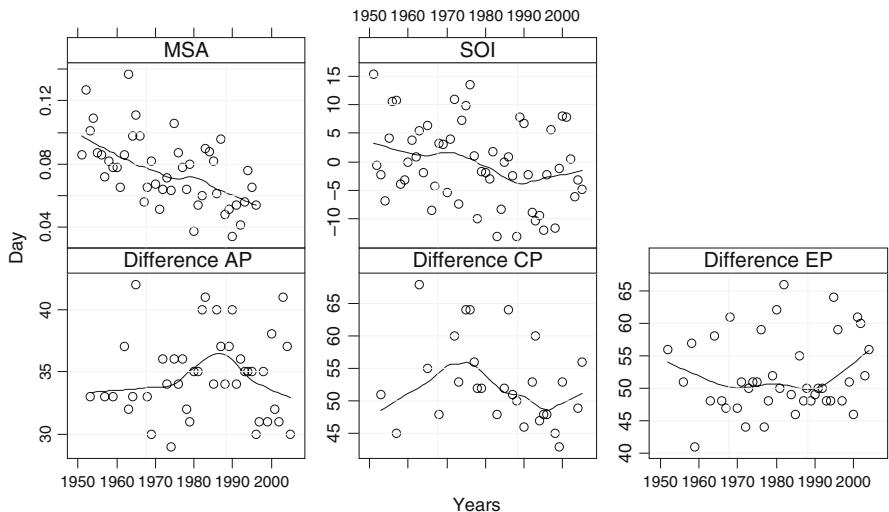


Fig. 14.5 Differences between arrival and laying dates against time for each species. AP, CP, and EP stand for Adelie Penguin, Cape Petrel and Emperor Penguin respectively. The *upper two panels* show the MSA and SOI time series; both are potential explanatory variables

is -0.57). The same holds for MSA and SOI, but now the motivation for not using them together is that the variations in MSA are driven by SOI. There is no clear trend over time in the ‘difference time series’ for the species.

The R code used to create Fig. 14.5 is given below. It follows the same steps as for Fig. 14.2.

```
> ABirds$DifAP <- ABirds$LayingAP - ABirds$ArrivalAP
> ABirds$DifCP <- ABirds$LayingCP - ABirds$ArrivalCP
> ABirds$DifEP <- ABirds$LayingEP - ABirds$ArrivalEP
> AllDif <- c(ABirds$DifAP, ABirds$DifCP,
   ABirds$DifEP, ABirds$MSA, ABirds$SOI)
> AllYear <- rep(ABirds$Year, 5)
> IDDif <- rep(c("Difference AP", "Difference CP",
   "Difference EP", "MSA", "SOI"), each = 55)
> xyplot(AllDif ~ AllYear | IDDif, xlab = "Years",
   ylab = "Day", layout = c(3, 2),
   strip = function(bg = 'white', ...),
   strip.default(bg = 'white', ...),
   scales = list(alternating = TRUE,
     x = list(relation = "same"),
     y = list(relation = "free")),
   panel = function(x, y){
     panel.xyplot(x, y, col = 1)
     panel.loess(x, y, col = 1, span = 0.5)
     panel.grid(h = -1, v = 2)})
```

14.3 Trends and Auto-correlation

The smoothing curves in Fig. 14.2 indicate the presence of long-term trends in some of the arrival and laying time series. However, we quite arbitrarily chose a span width of 0.5 for the LOESS smoother, and choosing a different value may give a different message. To estimate the optimal amount of smoothing we can use cross-validation (Wood, 2006; Chapter 3) and we can also allow for auto-correlation (Chapter 6). We now compare the models with and without auto-correlation and test which one is better. The reason for investigating whether we need a residual auto-correlation structure is that if we falsely omit it, p -values may be seriously inflated. Zuur et al. (2007; Chapter 23) used a bird data set in which the model with and without auto-correlation nearly resulted in different conclusions on the importance of different management variables.

Cross-validation and/or adding a residual auto-correlation structure to a smoothing model requires the use of the gamm function in the mgcv package in R. The model we apply on each time series is

$$\begin{aligned}
 Y_s &= \alpha + f(\text{Year}_s) + \varepsilon_s \\
 \varepsilon_s &\sim N(0, \sigma^2) \\
 \text{cor}(\varepsilon_s, \varepsilon_t) &= h(\rho, d(\text{Year}_s, \text{Year}_t))
 \end{aligned} \tag{14.1}$$

The first part specifies that the time series is modelled as an intercept plus a smoothing function plus residuals. These residuals are assumed to be normally distributed, but not independent of each other. We allow for a certain dependence structure using the function $h()$, which depends on an unknown parameter ρ and a function $d()$ which is a function of time (or better: the difference between time points). The trick is now to find an appropriate structure for $h()$ and we can compare different forms using the AIC or likelihood ratio tests. As discussed in Chapters 6 and 7, we have a series of options to model the function $h()$. The one of interest here is the auto-regressive moving average (ARMA) serial correlation structure. We could also apply correlation structures from spatial data analysis methods (Chapter 7), which is especially useful if the time series are irregular spaced. However, the time series are regular spaced, albeit with missing values, and therefore we do not need to use any spatial correlation structures.

If the trend is linear, then we can use a model of the form

$$\begin{aligned}
 Y_s &= \alpha + \beta \times \text{Year}_s + \varepsilon_s \\
 \varepsilon_s &\sim N(0, \sigma^2) \\
 \text{cor}(\varepsilon_s, \varepsilon_t) &= h(\rho, d(\text{Year}_s, \text{Year}_t))
 \end{aligned} \tag{14.2}$$

The first two parts of the equation are the familiar linear regression model. And, if we assume independence of the residuals, it is a linear regression model. But we have not made this assumption here, and the last part of Equation (14.2) allows for residual dependence. On the other hand, the cross-validation will help decide whether we need the model in Equation (14.1) or (14.2). Indeed Equation (14.2) is a special case of Equation (14.1), so we might as well focus in first instance only on the model in Equation (14.1). This is because a straight line (linear regression) is a special case of a smoothing curve (Fox, 2000). The model in Equation (14.1) can be fitted in R with the `gamm` function using a Gaussian distribution and identity link (Chapter 3), and Equation (14.2) is fitted using the `gls` function in the `nlme` package. The following R code fits the additive model with an ARMA error structure (Chapter 6) on the arrival dates of the Adelie Penguins.

```

> library(mgcv)
> library(nlme)
> B1 <- gamm(ArrivalAP ~ s(Year), data = ABirds,
+               correlation = corARMA(form =~ Year, p = 1, q = 0))
> AIC(B1$lme)

```

The option `corARMA (form =~ Year, p = 1, q = 0)` specifies the auto-regressive residual ARMA structure of order (p, q) . The notation for this is

$\text{ARMA}(p, q)$. The AIC of this model is 237.83. To choose the optimal ARMA structure, we used all combinations for p and q from 0 to 2. For the combination $p = q = 0$, you need to omit the correlation option. Hence, this is just an ordinary GAM without a correlation structure. To assess which combination of p and q results in the ‘best’ model, we used the AIC. The lower the AIC, the better the model.

The notation $s(\text{Year})$ means that a smoother is applied on Year and cross-validation is used to estimate the optimal amount of smoothing.

This modelling approach was applied on all six arrivals and laying date time series. All six time series gave results where the optimal residual error structure was a $\text{ARMA}(0,0)$, meaning that no correlation structure was needed. This means that we are back to using ordinary smoothing (or regression). For all six time series, the amount of smoothing was 1 degree of freedom, meaning that each trend is a straight line. This allows us to apply the linear regression model in Equation (14.2) without the auto-correlation structure. The slope of the trend was only significantly different from 0 for the laying time series of the Adelie Penguin ($p = 0.003$) and for both arrival ($p = 0.009$) and laying ($p = 0.029$) Cape Petrel time series. For the other three series, the slope was not significantly different from zero.

14.4 Using Ice Extent as an Explanatory Variable

In this section, we consider models of the form

$$\begin{aligned} Y_s &= \alpha + f(\text{MSA}_s) + \varepsilon_s \\ \varepsilon_s &\sim N(0, \sigma^2) \\ \text{cor}(\varepsilon_s, \varepsilon_t) &= h(\rho, d(\text{Year}_s, \text{Year}_t)) \end{aligned} \quad (14.3)$$

Y_s is the arrival or laying date in year s and MSA_s is the Methanesulfonic acid concentration (μM) in year s , representing the sea ice extent. Again, we can use cross-validation to estimate the amount of smoothing, and if it turns out that the estimated degrees of freedom is equal to one, we will end up with the model

$$\begin{aligned} Y_s &= \alpha + \beta \times \text{MSA}_s + \varepsilon_s \\ \varepsilon_s &\sim N(0, \sigma^2) \\ \text{cor}(\varepsilon_s, \varepsilon_t) &= h(\rho, d(\text{Year}_s, \text{Year}_t)) \end{aligned} \quad (14.4)$$

As in the previous section, different residual correlation structures can be applied using the correlations option in `gamm` and `gls` and the AIC is used to compare them. For all six arrival and laying time series, the optimal residual correlation structure was $\text{ARMA}(0,0)$, which means that no correlation structure is needed. Dropping the correlation structure means we are back in the world of ordinary additive modelling or linear regression, depending on the amount of smoothing. The cross-validation method gave 1 degree of freedom for each series, indicating that we can use the linear regression model in Equation (14.4).

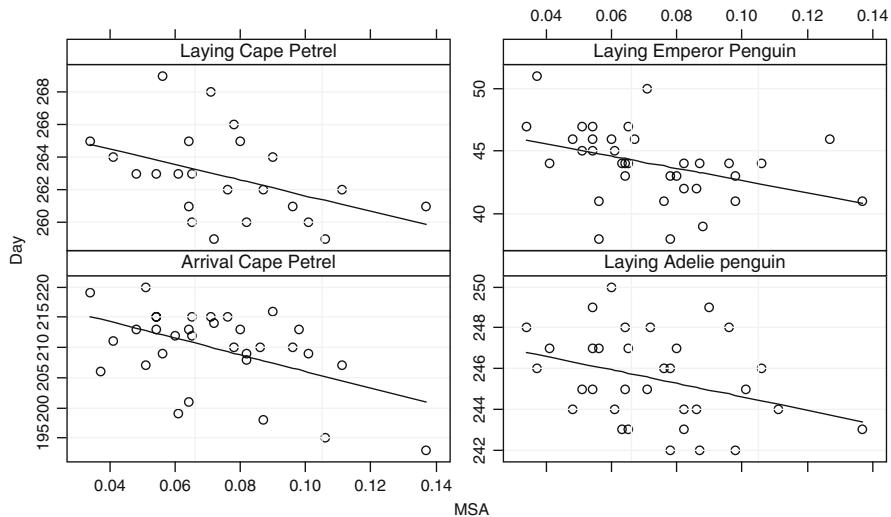


Fig. 14.6 Fitted values obtained by linear regression. Only the time series with a significant slope for MSA are shown. The R^2 for the four series are 12% (Laying Adelie penguin), 19% (Laying Cape Petrel), 15% (Laying Emperor Penguin), and 24% (Arrival Cape Petrel)

The linear regression model showed that MSA has a negative effect on laying dates of all three birds (Adelie Penguin, $p = 0.053$; Cape Petrel, $p = 0.039$; and Emperor Penguin, $p = 0.039$), and also on the arrival date of Cape Petrel ($p = 0.004$). The observed data and fitted lines for these four time series are presented in Fig. 14.6.

The following R code was used for the linear regression models.

```
> M1 <- lm(LayingAP ~ MSA, data = ABirds)
> M2 <- lm(LayingCP ~ MSA, data = ABirds)
> M3 <- lm(LayingEP ~ MSA, data = ABirds)
> M4 <- lm(ArrivalCP ~ MSA, data = ABirds)
> summary(M1); summary(M2)
> summary(M3); summary(M4)
```

This code is just the familiar linear regression and summary commands from Chapter 2 that gives the estimated values, R^2 , F -statistic, t -values, and p -values. We have not reproduced all the numerical output from the summary commands here. The model fits are presented in the lattice graph (Fig. 14.6) using the following R code.

```
> Bird4 <- c(ABirds$LayingAP, ABirds$LayingCP,
           ABirds$LayingEP, ABirds$ArrivalCP)
> MSA4 <- rep(ABirds$MSA, 4)
```

```

> ID4 <- rep(c("Laying Adelie penguin",
+ "Laying Cape Petrel",
+ "Laying Emperor Penguin",
+ "Arrival Cape Petrel"), each = 55)
> xyplot(Bird4 ~ MSA4 | ID4, xlab = "MSA",
+ ylab = "Day", layout = c(2, 2),
+ strip = function(bg = 'white', ...),
+ strip.default(bg = 'white', ...),
+ scales = list(alternating = TRUE,
+               x = list(relation = "same"),
+               y = list(relation = "free")),
+ panel = function(x, y, subscripts, ...){
+   panel.xyplot(x, y, col = 1)
+   panel.grid(h = -1, v = 2)
+   I1 <- !is.na(y) & !is.na(x)
+   tmp <- lm(y[I1] ~ x[I1])
+   x1 <- x[I1]
+   y1 <- fitted(tmp)
+   I2 <- order(x1)
+   panel.lines(x1[I2], y1[I2], col = 1, span = 1)})

```

The first three commands create three variables containing the stacked observed data, names, and MSA values for the four series. The code for the `xyplot` should now look familiar, except perhaps applying the linear regression within the `xyplot` function. The only thing to watch for is ensuring that we deal correctly with the missing values in the data. The command `I1 <- !is.na(y) & !is.na(x)` identifies the observations for which we have an observation for both the response and explanatory variables. The linear regression is applied on these data, and the `order` command is used to avoid a spaghetti plot (Chapter 2).

The main problem using MSA as an explanatory variable is that we lose 16% of the data due to missing values. Recall from Chapter 2 that the *entire* row of data is omitted, even if only one variable has a missing value for that observation.

14.5 SOI and Differences Between Arrival and Laying Dates

For the last analysis in this chapter, we use SOI as an explanatory variable. A slightly different statistical approach is followed and the arrival and laying dates for a bird are analysed simultaneously. Using an interaction term, we can use this approach to make a statement on the difference of the SOI effect on arrival and laying dates. This approach is potentially invalid, but we discuss at the end of this section how to correct for this.

A simple boxplot (not presented here) shows that the variation in arrival dates is considerably larger than for laying dates for all three bird species. This means that an ordinary linear regression model (or additive model) applied on the combined

arrival and laying dates is likely to violate the homogeneity assumption. On top of this, there may be auto-correlation. An additive model applied on the individual time series using SOI as an explanatory variable showed that all trends were linear, and therefore, we will work with a linear regression model of the form:

$$\begin{aligned} Y_{sj} &= \alpha + \beta_1 \times \text{SOI}_s + \beta_2 \times \text{ID}_j + \beta_3 \times \text{SOI}_s \text{ID}_j + \varepsilon_s \\ \varepsilon_s &\sim N(0, \sigma_j^2) \\ \text{cor}(\varepsilon_s, \varepsilon_t) &= h(\rho, d(\text{Year}_s, \text{Year}_t)) \end{aligned} \quad (14.5)$$

Y_{sj} is the arrival ($j = 1$) or laying ($j = 2$) date of a particular species in year s . In R, we stack the arrival and laying dates into one vector of length 110. This vector is then modelled as an intercept plus a function of SOI, a nominal variable ID (arrival or laying), and an interaction between SOI and ID. In matrix notation we have

$$\begin{pmatrix} \text{Arrival}_1 \\ \vdots \\ \text{Arrival}_{55} \\ \text{Laying}_1 \\ \vdots \\ \text{Laying}_{55} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} \alpha + \begin{pmatrix} \text{SOI}_1 \\ \vdots \\ \text{SOI}_{55} \\ \text{SOI}_1 \\ \vdots \\ \text{SOI}_{55} \end{pmatrix} \beta_1 + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \beta_2 + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \text{SOI}_1 \\ \vdots \\ \text{SOI}_{55} \end{pmatrix} \beta_3 + \begin{pmatrix} \varepsilon_{1,1} \\ \vdots \\ \varepsilon_{1,55} \\ \varepsilon_{2,1} \\ \vdots \\ \varepsilon_{2,55} \end{pmatrix}$$

If we assume independence and homogeneity of the residuals, then this is the ordinary linear regression model with one continuous variable, one nominal variable and the interaction between them (also known as analysis of covariance, abbreviated as ANCOVA). However, we do not assume independence or homogeneity. And, as discussed in Chapter 4, we can use GLS estimation to estimate multiple variance terms, in this case σ_1^2 for the arrival dates and σ_2^2 for the laying dates. The question is then whether $\sigma_1 = \sigma_2$, or whether we indeed need two different variances.

As to the auto-correlation structure, for simplicity, we only consider the ARMA(1,0) structure. This means that the residual correlation structure takes the form (Pinheiro and Bates, 2000; Chapter 6):

$$\text{cor}(\varepsilon_s, \varepsilon_t) = \rho^{|s-t|} \quad (14.6)$$

The parameter ρ is between -1 and 1. This auto-correlation structure dictates that the larger the time period between two years, the smaller the dependence between them. The following R code applies the model in Equation (14.5) with the auto-correlation in Equation (14.6).

```
> AP <- c(ABirds$ArrivalAP, ABirds$LayingAP)
> SOI2 <- c(ABirds$SOI, ABirds$SOI)
> Y2 <- c(ABirds$Year, ABirds$Year)
> ID <- factor(rep(c("Arrival", "Laying"), each = 55))
```

```
> library(nlme)
> vf2 <- varIdent(form =~ 1 | ID)
> M5 <- gls(AP ~ SOI2 + ID + SOI2:ID, weights = vf2,
             na.action = na.omit)
> M6 <- gls(AP ~ SOI2 + ID + SOI2:ID, weights = vf2,
             na.action = na.omit,
             correlation = corAR1(form =~ Y2 | ID))
> anova(M5, M6)
```

The first command concatenates the arrival and laying dates time series of the Adelie Penguin. The second command creates a vector with corresponding SOI values, and Y2 contains the years in which an observation was taken. Finally, ID is a nominal variable identifying the two time series. The `library` command ensures we can access the `gls` function, which is needed to fit the model in Equation (14.5) in R. The `varIdent` function was discussed in Chapter 4 and allows for a different variance for each of the bird time series. We first call the `gls` function and fit a model without auto-correlation. Then we fit a model with the auto-correlation structure as specified in Equation (14.6) and store its results in M6. The `anova` (M5, M6) command applies a likelihood ratio test and gives

Model	df	AIC	BIC	logLik	L.Ratio	p-value
M5	6	427.82	442.26	-207.91		
M6	7	426.07	442.92	-206.03	3.744	0.05

These results show that there is a weak residual auto-correlation structure ($p = 0.05$), and we should keep it in the model. We can also compare a model with one variance and a model with two variances. The likelihood ratio test (not presented here) indicates that we should use two variances ($p = 0.002$). The results of the following code show that we do not need the interaction term between ID and SOI.

```
> M7 <- gls(AP ~ SOI2 + ID + SOI2:ID, weights = vf2,
             na.action = na.omit, method = "ML",
             correlation = corAR1(form =~ Y2 | ID))
> M8 <- gls(AP ~ SOI2 + ID, weights = vf2,
             na.action = na.omit, method = "ML",
             correlation = corAR1(form =~ Y2 | ID))
> anova(M7, M8)
```

As discussed in Chapter 4, if we compare two models with the same random structure, but with different fixed effect, we need to use the maximum likelihood estimation method instead of REML. The resulting test statistic is obtained by the `anova` (M7, M8) command, and it gave a test statistic $L = 0.16$ ($df = 1, p = 0.68$). This indicates that we can drop the interaction term as it is not significant.

In the model with SOI2 and ID as explanatory variables (M8), the `summary` (M8) command shows that only ID is significant, meaning that there is no SOI effect on Adelie Penguins. Hence, the optimal model is given by

```
> M9 <- gls(AP ~ ID, weights = vf2, method = "ML",
  na.action = na.omit,
  correlation = corAR1(form =~ Y2 | ID))
> summary(M9)
```

Its numerical output is given by

```
Correlation Structure: ARMA(1,0)
Formula: ~Y2 | ID
Parameter estimate(s):
  Phil
0.26
Variance function:
Structure: Different standard deviations per stratum
Formula: ~1 | ID
Parameter estimates:
  Arrival    Laying
  1.00      0.61
Coefficients:
            Value Std.Error t-value p-value
(Intercept) 211.11     0.64     329.26   <0.001
IDLaying     34.56     0.75      45.63   <0.001

Residual standard error: 3.361275
Degrees of freedom: 86 total; 84 residual
```

This output shows that the predicted arrival time for Adelie Penguin is day 211 (rounded), and the laying date is $211 + 34 = 245$. There is no effect of SOI on either arrival or laying dates. The residual standard error for the arrival dates is 3.36, but for the laying dates it is 0.6 smaller. There is also a small amount of auto-correlation as $\rho = 0.26$. This means that the residual auto-correlation between two sequential years is equal to $0.26^1 = 0.26$, and for time points that are separated by 2 years, this correlation is $0.26^2 = 0.07$.

The same analysis was carried out on the Cape Petrel time series. Using the same code, but with the first line replaced by `CP <- c(ABirds$ArrivalCP, ABirds$LayingCP)` and consequently AP by CP. For the Cape Petrel, we found that different variances per arrival and laying series are needed, but there was no need for an auto-correlation structure. The interaction term had a *p*-value of 0.09, but we decided to keep it in as it was close to the ‘magic’ significance level of 0.05. The SOI effect and the nominal variable ID were highly significant. The following R code was used.

```
> CP <- c(ABirds$ArrivalCP, ABirds$LayingCP)
> SOI2 <- c(ABirds$SOI, ABirds$SOI)
> Y2 <- c(ABirds$Year, ABirds$Year)
> ID <- factor(rep(c("Arrival", "Laying"), each = 55))
```

```
> vf2 <- varIdent(form= ~ 1 | ID)
> M10<-gls(CP ~ SOI2 + ID + SOI2:ID, weights = vf2,
   na.action = na.omit, method = "ML")
```

The results below obtained by the `summary(M10)` command. Note that the spread for the laying dates is considerably lower!

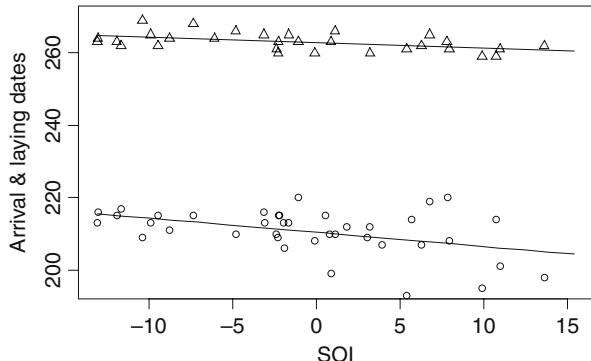
```
Structure: Different standard deviations per stratum
Formula: ~1 | ID
Parameter estimates:
  Arrival     Laying
  1.00      0.37
Coefficients:
            Value    Std.Error  t-value p-value
(Intercept) 210.41    0.92      226.62  0.00
SOI2        -0.39    0.13      -2.99    0.00
IDLaying     52.37    1.01      51.76    0.00
SOI2:IDLaying 0.24    0.14      1.71    0.09
```

The lines of R code below produce a scatterplot of the data, and the fitted lines obtained by the optimal model; see Fig. 14.7.

```
> plot(ABirds$SOI, ABirds$ArrivalCP,
      ylim = c(195, 270), type = "n",
      ylab = "Arrival & laying dates")
> points(ABirds$SOI, ABirds$ArrivalCP, pch = 1)
> points(ABirds$SOI, ABirds$LayingCP, pch = 2)
> MyX <- data.frame(SOI2 = seq(from = min(ABirds$SOI),
      to = max(ABirds$SOI), length = 20),
      ID = "Arrival")
> Pred1 <- predict(M10, newdata = MyX)
> lines(MyX$SOI2, Pred1)
> MyX <- data.frame(SOI2 = seq(from = min(ABirds$SOI),
      to = max(ABirds$SOI),
      length = 20),
      ID = "Laying")
> Pred2 <- predict(M10, newdata = MyX)
> lines(MyX$SOI2, Pred2)
```

For the emperor penguin, we found strong evidence to use two variance terms ($p < 0.001$) and weak evidence for an auto-correlation structure ($p = 0.07$). However, neither the interaction nor the SOI was significant. The output of the final model is given on the next page and shows there is some auto-correlation, the residual variation in laying dates is nearly half of the arrival time residual variation, and the difference between arrival and laying dates is 52 days.

Fig. 14.7 Arrival and laying dates for the Cape Petrel. The *triangles* are the laying dates, and the *dots* the arrival dates. Due to the weak interaction, the *lines* are nearly parallel indicating that there are no strong differences between the SOI-date relationship for arrival and laying



Correlation Structure: ARMA(1, 0)

Formula: ~Y2 | ID

Parameter estimate(s) :

Phil

0.20

Variance function:

Structure: Different standard deviations per stratum

Formula: ~1 | ID

Parameter estimates:

Arrival Laying

1.00 0.53

Coefficients:

	Value	Std.Error	t-value	p-value
(Intercept)	-7.91	0.91	-8.62	<0.001
IDLaying	51.94	1.04	49.58	<0.001

At the start of this section, we mentioned that the analysis presented in this section is potentially invalid. This would happen if the residuals of the laying time series are correlated to the residuals of the arrival time series. We allowed for correlation within a series, but not between a series. In Chapter 6, we applied a similar analysis on a bird time series from Hawaii. However, in that example, because sampling was done during the breeding season on different islands, the between-series independence assumption is more plausible than it is for the time series used in this chapter. The easiest way to verify the independence assumption is to calculate the correlation between the two residual time series per species. If the correlation is not significant, we are lucky and the approach in this section is valid. In this case, the correlation is 0.29 and the associated *p*-value is 0.06. At the 5% level, it is not significant, but we are still not that happy with a *p*-value so close to the magic 5% level. Obtaining this correlation coefficient requires some rather tedious R programming (due to missing values), and the code is on the book's website.

14.6 Discussion

Wrongly ignoring dependence structures in data means a greater chance of type I errors. So, with time series data, we should always check for residual auto-correlation, and only where there is no significant auto-correlation, use methods that ignore auto-correlation. For the arrival and laying time series, there was no strong residual auto-correlation; hence, one can proceed with methods that do not include an auto-correlation structure (e.g. linear regression, additive modelling).

For all bird time series, linear regression was favoured above smoothing techniques. As well as smoothing methods, we also tried other models that allow for non-linear trends (e.g. quadratic models using Year and Year² as explanatory variables), but they confirmed the cross-validation results.

A detailed model validation consisting of plots of (normalised) residuals versus fitted values, auto-correlation plots, histograms, and plotting residuals versus explanatory variables was applied in each section. For nearly all models, there were no problems. We also tried models that contained both year and SOI, but these did not improve the models.

The ecological interpretation of these results shows there is a positive trend in the laying time series of the Adelie Penguin and the arrival and laying of the Cape Petrel series. This may be related to the MSA time series, but unfortunately using this variable means we lose 16% of the data. It should be noted that MSA itself is negatively related to time; there is a clear decreasing trend in MSA (Fig. 14.5). The negative relationships between MSA, laying, and arrival dates indicate that birds arrive and lay earlier when sea ice extent increases. This fits well with our knowledge of the reproductive ecology of these species because they need to build up body reserves (fat) before breeding. So, in years with extensive sea ice, food might be more abundant allowing birds to build up fat reserves quicker than years with less sea ice. Consequently, the negative trend in MSA may explain part of the positive trends in arrival and laying dates, although MSA explained at most 24% of the variability in arrival and laying dates. Other factors such as the duration of the sea season or individual characteristics such as age or experience may explain part of the remaining variability. The analysis using the combined arrival and laying dates for a species and SOI as explanatory variable showed there was a large difference in spread in arrival and laying dates for all species and that SOI has a negative effect on arrival and laying dates of Cape Petrel. We also applied the same analysis using year as the explanatory variable instead of SOI. We have not presented the results here, but they showed that the Adelie Penguin had a weak auto-correlation and large spread in arrival and laying dates (the ratio between the standard errors was 0.57) and a significant ID and year effect, but no interaction. This means that arrival and laying dates have increased over time and at the same rate. We found similar results for the Cape Petrel, except there was no auto-correlation. For the Emperor Penguin, there was only an ID effect and weak auto-correlation, but no year effect.

It is not surprising to find a large difference in spread in arrival and laying dates across the species. Unlike arrival dates, laying is closely synchronised in Antarctic seabird populations, meaning that within a population all individuals lay their egg in

a short time window every year. Therefore, laying date is probably less affected by factors such as age, experience, sex or meteorological factors than arrival date. The fact that arrival and laying dates have increased over time at the same rate for the Adelie Penguin and the Cape Petrel over such a long period suggests that the time interval separating those phenological events is relatively inflexible. This probably reflects the invariance in the timing of physiological mechanisms involved during the egg development process and to a lesser extent the time needed for birds to build their nest, courtship and pair.

Finally, an interesting result is the negative effect of SOI on arrival and laying dates of the Cape Petrel. The negative slope indicates that El Niño conditions (negative SOI) delay arrival and laying of Cape Petrels. Because Cape Petrels spend the non-breeding season at more northerly latitudes than Adelie and Emperor Penguins, they might be more affected by El Niño conditions. This result is also in accordance with previous studies on seabirds that have demonstrated that El Niño conditions usually cause a decrease in oceanic productivity and of seabird demographic parameters.

14.7 What to Report in a Paper

If we were to write a paper based on the analysis presented in this chapter, we would present an introduction describing the questions and data. We would then continue presenting the data in a multiple panel graph (e.g. Fig. 14.2), present the models (additive model and linear regression) and put emphasise on the potential auto-correlation problem. The fact that the ARMA(0,0) error structure was the most optimal structure can be presented without too much numerical output, and the same holds for the cross-validation. However, you cannot omit this information as they justify the application of the linear regression or smoothing model without auto-correlation.

Describing the approach and summarising the results that justify the linear regression model with an independent error structure does not have to be any longer than two paragraphs. The results of the linear models should be presented in a table showing the estimated parameters, t -values, p -values, and R^2 and F -statistic for all time series. You should also comment on the results of a model validation for the linear regression. (Did the residuals show any patterns in terms of homogeneity, normality, and residuals values versus year, and residuals values versus explanatory variables?) As the linear regression model was preferred over the additive model, state that non-linear patterns are unlikely to occur. The model formulation for the combined arrival and laying time series and the analysis followed may be confusing for the reader, and you may want to explain this aspect in more detail. As to what these results tell us about climate change is left for you to decide.

Chapter 15

Large-Scale Impacts of Land-Use Change in a Scottish Farming Catchment

A.F. Zuur, D. Raffaelli, A.A. Saveliev, N.J. Walker, E.N. Ieno, and G.M. Smith

15.1 Introduction

A catchment is an area of land defined by the origins and discharges of all tributary streams feeding large rivers flowing into the sea. It is therefore a natural bio-physical unit distinct from adjacent catchments and forms the obvious basis for integrated environmental management policies. In Europe, river catchments tend to be dominated by agriculture, at least at lower altitudes. In the case of the Ythan catchment (Fig. 15.1), Aberdeenshire, Scotland, where the river rises at only a few hundred metres, more than 90% of the land area is now under agricultural production. Much of this is arable crops like wheat, barley, and oil-seed rape, which demand high inputs of chemical nitrogen. The Ythan catchment also hosts large numbers of pigs and other livestock (and also some of the authors of this book).

Whilst the Ythan catchment has always been prime agricultural land, there have been major changes in land-use over the past 40 years because of market trends and drivers such as the Common Agriculture Policy. This policy encouraged growing of crops through subsidies not previously available for crops such as barley and wheat at the expense of less profitable crops such as oats. The conversion of grassland to cereals, increased application of nitrogen, and increase in animal manures and slurries over the past 40 years have inevitably affected water quality, specifically elevated levels of nitrate. These levels were so high in the 1990s that the Ythan catchment had the distinction of being the first in the UK to be designated a Nitrogen Vulnerable Zone under the European Community Nitrates Directive.

Staff at Culterty Field Station, University of Aberdeen, were able to document and describe trends in this process in great detail through a series of monitoring programmes, data analyses, and field experiments. Data on land-use were obtained from ‘parish returns’ – records of amounts of land under different crops and numbers of animals held for each farm in the parish that are returned to the Scottish Records Office annually. These data were extracted for all parishes (community

A.F. Zuur (✉)
Highland Statistics Ltd., Newburgh, AB41 6FN, United Kingdom

Fig. 15.1 Small part of the Ythan estuary. The photograph was taken by Alain Zuur



administrative areas) within the catchment for land under oats, wheat, oil-seed rape, barley, and for numbers of pigs, cattle and sheep for the period 1960s to 1990s. Levels of nitrates (only small amounts can be attributed to sewage) were extracted from databases held by the North-East River Purification Board and supplemented by the field station's own observations. The environmental impact of high levels of nitrates is expected to be seen as blooms of algae in rivers and estuaries, where they form extensive green mats that strip the oxygen from the underlying mudflats, reducing the invertebrates available to feeding shorebirds.

Counts of shorebirds have been made every month for the past 40 years by staff and students at the field station and most of these data make up the database held by the British trust for Ornithology for this estuary. Mean counts for the winter months November–February were calculated for the most abundant waders: oyster-catcher *Haematopus ostralegus*, redshank *Tringa totanus*, dunlin *Calidris alpina*, knot *Calidris canutus*, turnstone *Arenaria interpres*, bar-tailed godwit *Limosa lapponica* and curlew *Numenius arquata*.

Using these data, we were able to trace possible connections from agricultural policy and land-use change through to ecological impacts on species of high conservation importance, the shorebirds. The data sets are interesting because they are typical of those available for detecting historical trends in variables that may be linked to a current environmental impact. The time series is unusually long for ecological data, but the data were not originally collected with this specific analysis in mind (linking agriculture change with shorebird numbers) and they are imperfect in many respects, as we shall see. All too often the ecologist has to work with whatever data are available rather than what would be ideal. Unfortunately, it is impossible to collect data retrospectively, unless one has a time machine.

We analysed these data to try and answer the following questions:

1. Are there any trends in the bird time series?
2. Is there a simple and obvious relationship between agricultural change and shorebird numbers?

3. Is the relationship different for different species of shorebird?
4. Which aspects of land-use best account for changes in water quality and therefore need to be targeted for restoration programmes?

To answer these questions, we split the analysis into three stages. In the first stage, we applied a data exploration, and in the second stage, we focussed on the first question: are there any trends in the bird time series? We used GAMs (Chapters 3 and 6) for this. The reason we used smoothing techniques will become clear after we have looked at the data exploration. The ‘mixed’ bit is needed because the data are time series and, as always in ecology, there is heterogeneity. In the last step of the analysis, we included the information on land-use and agricultural changes.

15.2 Data Exploration

The dataset consists of average winter values for seven bird species (Oystercatcher, Turnstone, Curlew, Bar-tailed Godwit, Redshank, Knot, and Dunlin) and seven potential explanatory variables (wheat, barley, oats, cattle, sheep, pigs, and nitrate). The best way to visualise a dataset with up to 20–25 time series is a multiple panel graph made with the `xypplot` function from the lattice package (Fig. 15.2). The data file contains the variables in columns and the years in rows. To create the multiple panel graph for these data, we need to create three columns. The first column should contain all the variables we wish to plot along the vertical axes. Because there are 14 variables, the second column should contain 14 times year (we need

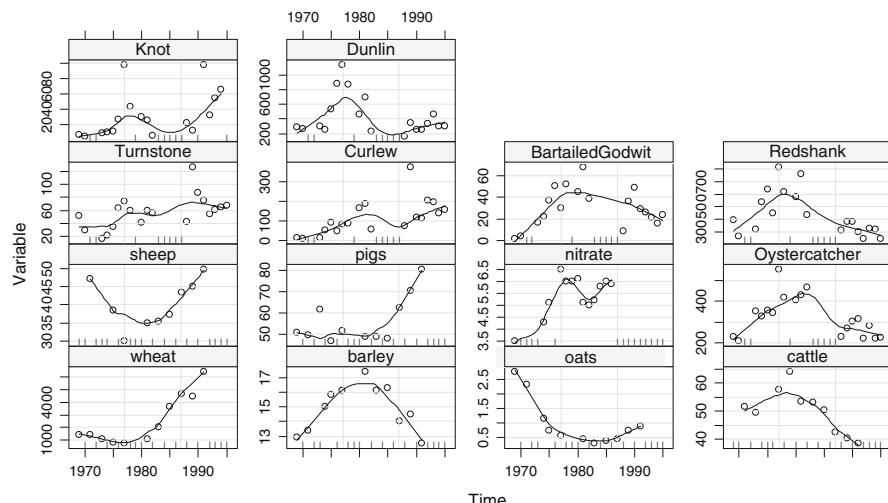


Fig. 15.2 Plot of the seven bird species and the potential explanatory variables. A LOESS smoother (with default amount of smoothing) was added to enhance visual interpretation

to concatenate Year 14 times). Finally, ID14 is the variable that tells the `xyplot` function which elements belong to the same variable. In the R code below, we used the `levels` option to ensure that the `xyplot` function places the panels of the birds next to each other.

```
> data(AED); data(Ythan); library(lattice)
> Birds <- as.vector(as.matrix(Ythan[, 2:8]))
> X <- as.vector(as.matrix(Ythan[, 9:15]))
> YX14 <- c(Birds, X)
> Year14 <- rep(Ythan$Year, 14)
> N <- length(Ythan$Year)
> ID14 <- factor(rep(names(Ythan[,2:15]), each = N),
  levels = c("wheat", "barley", "oats", "cattle",
  "sheep", "pigs", "nitrate", "Oystercatcher",
  "Turnstone", "Curlew", "BartailedGodwit",
  "Redshank", "Knot", "Dunlin"))
```

The code below produces Fig. 15.2.

```
> xyplot(YX14 ~ Year14 | ID14, xlab = "Time",
  ylab = "Variable", layout = c(4, 4),
  scales = list(alternating = TRUE,
    x = list(relation = "same"),
    y = list(relation = "free")),
  panel = function(x, y){
    panel.xyplot(x, y, col = 1)
    panel.grid(h = -1, v = 2)
    panel.loess(x, y, col = 1, span = 0.5)
    I2 <- is.na(y)
    panel.text(x[I2], min(y, na.rm = TRUE), '|', cex = 0.5)})
```

The new bit of code is the `panel.text` function. It plots the symbol ‘|’ wherever there is a missing value. Although it takes a lot of complicated R code to make the `xyplot` graph, the results are impressive. The panels for the explanatory variables indicate serious collinearity and a large number of missing values. Most bird time series seem to have a peak around 1980, and redshanks, oystercatcher, dunlin, and bar-tailed godwit seem to follow a similar pattern over time. Some similarity between this pattern and some of the explanatory variables can also be detected. Although more difficult to see, the different ranges of the y-axis for the bird panels indicate a potential problem (heterogeneity) if we analyse all birds simultaneously.

Heterogeneity by different bird species is to be expected and can be dealt with using (i) a data transformation, (ii) standardisation, or (iii) using the `varIdent` residual variance structure as discussed in Chapter 4. However, as we discussed in

Chapter 2, a data transformation will be avoided as much as possible. We will return to this point later.

The data exploration indicates that we can expect problems with homogeneity and that although there may be effects on the bird numbers related to the explanatory variables, due to the large number of missing values, it may be difficult to fit a model that contains both the bird numbers and the explanatory variables. In fact, if anything comes out of the analyses of these data, we should be very happy!

15.3 Estimation of Trends for the Bird Data

The shape of the trends for the birds in Fig. 15.2 suggests using a model of the form

$$\text{Birds}_{is} = \alpha_i + f_i(\text{Year}_s) + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma^2) \quad (15.1)$$

Birds_{is} is the average number of birds species i in the winter of year s , α_i and $f_i(\text{Year}_s)$ are the intercept and smoother for bird species I , respectively, and ε_{is} is normally distributed noise with mean 0 and variance σ^2 . If all the birds follow the same pattern over time, we can drop the index i from the smoother $f_i(\text{Year}_s)$. However, the shape of the smoothers in Fig. 15.2 clearly indicates that this is not the case. The ranges of the vertical axes in the same figure are rather different and suggest using

$$\text{Birds}_{is} = \alpha_i + f_i(\text{Year}_s) + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma_i^2) \quad (15.2)$$

The only difference with the previous formula is the index i attached to the variance; it allows for heterogeneity between bird species. It makes sense to allow for this form of heterogeneity as some bird species are only ever present in low numbers while other bird species are normally found in very large numbers. This is a common problem in ecology as species of interest often occur at very different levels of abundance leading to lower and higher variances.

The following R code sets up the data for the additive mixed model in Equation (15.2) with multiple smoothers.

```
> Birds7 <- as.vector(as.matrix(Ythan[, 2:8]))
> BirdNames <- c("Oystercatcher", "Turnstone",
+                  "Curlew", "BartailedGodwit",
+                  "Redshank", "Knot", "Dunlin")
> ID7 <- factor(rep(BirdNames, each = N),
+                 levels = BirdNames)
> Year7 <- rep(Ythan$Year, 7)
> Oyst.01 <- as.numeric(ID7 == "Oystercatcher")
> Turn.01 <- as.numeric(ID7 == "Turnstone")
> Curl.01 <- as.numeric(ID7 == "Curlew")
> Bart.01 <- as.numeric(ID7 == "BartailedGodwit")
```

```
> Reds.01 <- as.numeric(ID7 == "Redshank")
> Knot.01 <- as.numeric(ID7 == "Knot")
> Dunl.01 <- as.numeric(ID7 == "Dunlin")
> f7 <- formula(Birds7 ~ ID7 +
+   s(Year7, by = Oyst.01, bs = "cr") +
+   s(Year7, by = Turn.01, bs = "cr") +
+   s(Year7, by = Curl.01, bs = "cr") +
+   s(Year7, by = Bart.01, bs = "cr") +
+   s(Year7, by = Reds.01, bs = "cr") +
+   s(Year7, by = Knot.01, bs = "cr") +
+   s(Year7, by = Dunl.01, bs = "cr"))
```

The vector `Birds7` contains all bird data in a long vector. We also need a vector `ID7` that tells R which part of `Birds7` belongs to a certain species (N is the number of years that sampling took place). And obviously, we also need to copy and paste the variable `Year` seven times. The variables `Oyst.01`, `Turn.01`, etc., are vectors consisting of zeros and ones. For example, an element of `Oyst.01` is equal to 1 if the corresponding observation is an oystercatcher. These can be used in a GAM together with the `by` option to model interaction between year and species identity (which is `ID7`). As a result, a GAM gives 7 smoothers, one for each species. To reduce computing time (in some of the model that will be used later), we decided to use a cubic regression spline (`bs = cr` in R). The GAM itself is implemented with the code.¹

```
> library(mgcv); library(nlme)
> lmc <- lmeControl(niterEM = 5000, msMaxIter = 1000)
> M0 <- gamm(f7, control = lmc, method = "REML",
+             weights = varIdent(form =~ 1 | ID7))
```

As you can see, it takes more effort to prepare the data than to do the actual GAM command. REML estimation is used because we first want to find the optimal random component (Chapter 4). The `weights = varIdent(form =~ 1 | ID7)` implements the different variances per species and the `by` command in the smoother ensures that we have one smoother for each bird species i . The option `control = lmc` was used to ensure convergence.

15.3.1 Model Validation

The first validation plot we should make is residuals (normalised) versus fitted values (Chapter 4). The normalised residuals are corrected for the different variances per species. We can either plot residuals and fitted values for all species in one graph

¹We used R version 2.6 and mgcv version 1.3–27. More recent versions of R and mgcv require a small modification to the code; see the book website (www.highstat.com) for updated code.

(and use for example seven different symbols or colours) or draw them in a multiple panel plot with the `xypplot` function. We will do both. The `nlme` package has some handy tools to plot residuals versus fitted values. Obviously, we can use

```
> E0 <- resid(M0$lme, type = "normalized")
> F0 <- fitted(M0$lme)
```

and then plot residuals `E0` versus fitted values `F0` using the `plot` command or the `xypplot` function, but R can do this much faster. The following three commands each plot (normalised) residuals versus fitted values or time (Fig. 15.3).

```
> plot(M0$lme, resid(., type = "n") ~ fitted(.),
       abline = 0, col = 1)
> plot(M0$lme, resid(., type = "n") ~ Year7,
       abline = 0, col = 1, xlab = "Year")
> plot(M0$lme, resid(., type = "n") ~ fitted(.) | ID7,
       abline = 0, col = 1,
       par.strip.text = list(cex = 0.75))
```

The problem with this code is that it actually uses the lattice package and draws fancy multipanel graphs, and therefore, the `par(mfrow = c(2, 2))` tool to plot multiple graphs on the same window does not work. So, how did we create Fig. 15.3? The answer is in Sarkar (2008): Store each graph in an object, and use the `print.trellis` command (see: `?print.trellis`) to place the panels on a grid.

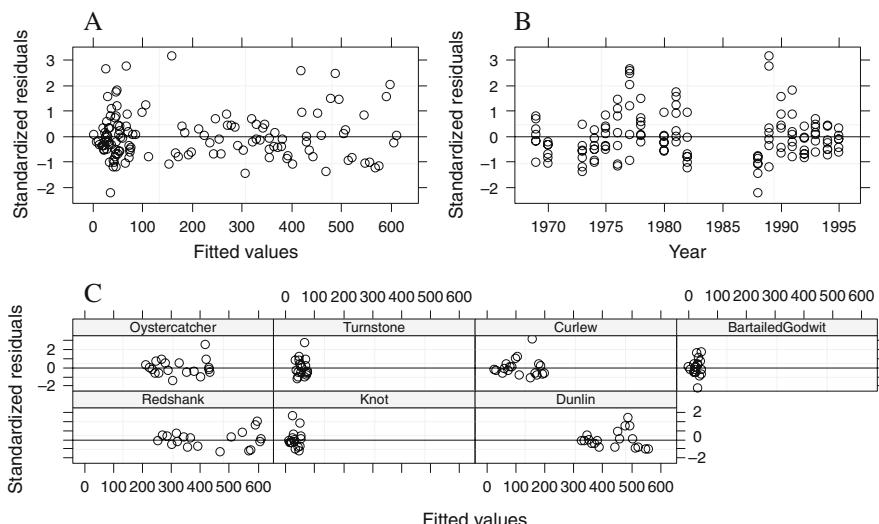


Fig. 15.3 Graphical validation of the model in Equation (15.2). **A:** Residuals versus fitted values. **B:** Residuals versus year. **C:** Residuals versus fitted values per species

```

> p1 <- plot(M0$lme, resid(., type = "n") ~ fitted(.),
              abline = 0, col = 1)
> p2 <- plot(M0$lme, resid(., type = "n") ~ Year7,
              abline = 0, col = 1, xlab = "Year")
> p3 <- plot(M0$lme, resid(., type = "n") ~ fitted(.) |
              ID7, abline = 0, col = 1,
              par.strip.text = list(cex = 0.75))
> print(p1, position = c(0, 0, 1, 1),
        split = c(1, 1, 2, 2), more = TRUE)
> print(p2, position = c(0, 0, 1, 1),
        split = c(2, 1, 2, 2), more = TRUE)
> print(p3, position = c(0, 0, 2, 1),
        split = c(1, 2, 2, 2), more = FALSE)

```

The `split` option in the `print` command tells R to divide the graphical window in a 2-by-2 grid (as determined by the last two numbers) and places each graph in a particular grid (as determined by the first two coordinates). Panel C is stretched over two grids because the `location` option specifies that `xmax = 2` (instead of 1). This is quite complicated R stuff (you could have done the same in Word with a table), but it can be handy to know. Sarkar (2008) is an excellent reference for lattice package.

What does it all tell us in terms of biology? Are we willing to assume homogeneity of variance based on Fig. 15.3A? We are hesitating a little bit as the residuals in the middle (between 100 and 400) seem to have slightly less spread. This could be a sample size issue as only a few birds have values in this range, see Fig. 15.3C. We can also argue that it looks homogeneous as by chance alone, 5% of the data can be outside the -2 to 2 interval. We also plotted residuals versus time (Fig. 15.4). Note there is an increase in residual spread for larger fitted values for some species (e.g. redshanks, curlew, and dunlin), but not for all! One option is to use a Poisson distribution, but because the data are winter *averages* and not counts, this is not the best option. Note that if we apply a generalised linear or additive model with a Poisson distribution, the average winter values are rounded to the nearest integer.

In Section 4.1, we introduced several approaches to model heterogeneity in a squid data set. The response variable was testis weight and the explanatory variable mantel length. In some months, variation in weight increased for larger length, but not in every month. We used the `varPower`, `varExp`, and `varConstPower` functions to allow for different spread along the variance covariate length per month. It seems we need a similar mechanism here to model the (potential) heterogeneity of variance. The only problem is that while we were able to use length as variance covariate for the squid data, here we do not have such a variable as the only available explanatory variables have many missing values. So instead we can use the fitted values as variance covariate. All that is needed is to adjust the `weights` option in the `gamm` function:

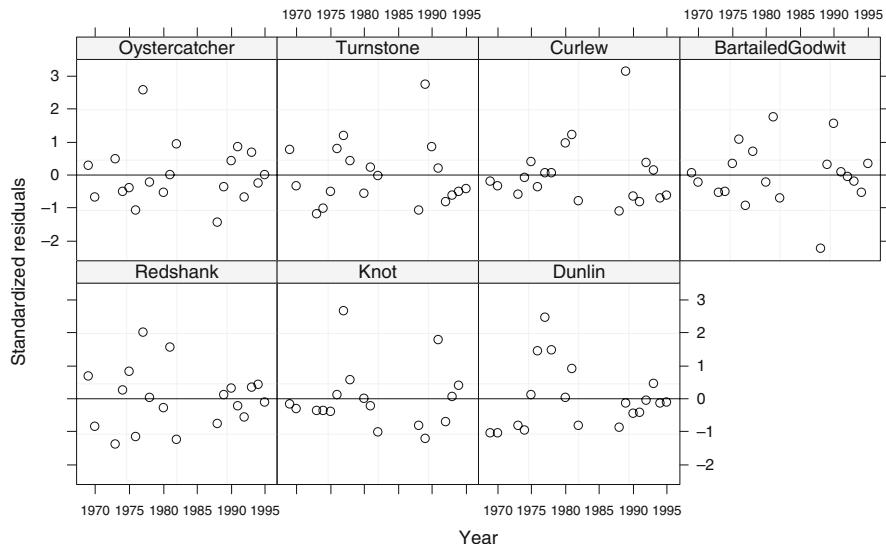


Fig. 15.4 Graphical validation of the model in Equation (15.2). Residuals versus time for each species

```
> M1<-gamm(f7, control = lmc, method = "REML",
  weights = varComb(varIdent(form = ~1 | ID7),
  varPower(form =~ fitted(.) | ID7)))
```

The variance structure creates the following additive mixed model.

$$\begin{aligned} \text{Birds}_{is} &= \alpha_i + f_i(\text{Year}_s) + \varepsilon_{is} \\ \varepsilon_{is} &\sim N(0, \sigma_i^2 |\hat{\alpha}_i + \hat{f}_i(\text{Year}_s)|^{\delta_i}) \end{aligned} \quad (15.3)$$

The variance structure for the noise ε_{is} looks rather complicated, but it is not. If δ_i is equal to 0 for all bird species i , we obtain exactly the same model as in Equation (15.2). The ‘hats’ above α and f_i indicate that these are estimates. If δ_i is larger than 0, the variance is proportional to the fitted values. So, this is a variance structure that allows for heterogeneity within a bird time series. The underlying principle reminds us of the Poisson distribution, where the mean equals the variance, but this is a normal distribution. As well as the within-bird-time series heterogeneity, we still allow for a different spread per bird using the index i attached to σ^2 .

The problem with the model in Equation (15.3) is the lack of convergence. This comes as no surprise as the model contains 7 smoothers with cross-validation applied on each smoother, 7 variances σ_i^2 , and 7 δ_i s. And even more relevant, the data contains many gaps due to the missing values and time series are relatively short. We tried several options to deal with this and actually, all failed

(in terms of numerical convergence) or were considered not particularly helpful. However, we believe you can learn just as much from unsuccessful approaches as you can from successful ones. So, we now discuss some of approaches that failed.

15.3.2 Failed Approach 1

To reduce numerical computing complexity, we initially set the degrees of freedom for each smoother to 4, leaving it to decide later whether we need to increase or decrease this number. This requires modifying the `s` function:

```
s(Year7, by = Oyst.01, bs = "cr", fx = TRUE, k = 5).
```

This was done for each smoother in the model. However, this caused a new problem; the `gamm` function of the `mgcv` package needs either a random component or at least one smoother on which it can apply a cross-validation. Adding a random intercept has the advantage that it also allows us to automatically model the temporal correlation within the time series. For example, consider the following model:

$$\begin{aligned} \text{Birds}_{is} &= \alpha + f_i(\text{Year}_s) + a_i + \varepsilon_{is} \\ \varepsilon_{is} &\sim N(0, \sigma_i^2 \times |\hat{\alpha}_i + \hat{f}_i(\text{Year}_s)|^{\delta_i}) \\ a_i &\sim N(0, \sigma_a^2) \end{aligned} \quad (15.4)$$

Note that the intercept α no longer has an index i . Instead, there is now a random intercept a_i that is normally distributed with mean 0 and variance σ_a^2 . This model is the smoothing equivalent of the random intercept mixed effects model discussed in Chapter 5. Recall that such a model induces the compound symmetry correlation on the time series. At this stage, it is useful to discuss this correlation structure. The model in Equation (15.4) is not fundamentally different from $\mathbf{Y}_i = \mathbf{X}_i \times \boldsymbol{\beta} + \mathbf{Z}_i \times \mathbf{b}_i + \boldsymbol{\varepsilon}_i$, which is the hierarchical mixed model discussed in Chapter 5. Or perhaps we should write it as $\mathbf{Birds}_i = \mathbf{X}_i \times \boldsymbol{\beta} + \mathbf{Z}_i \times \mathbf{b}_i + \boldsymbol{\varepsilon}_i$. The $\mathbf{X}_i \times \boldsymbol{\beta}$ component is the equivalent of the intercept and the smoothing curve, $\mathbf{Z}_i \times \mathbf{b}_i$ contains the random intercept and $\boldsymbol{\varepsilon}_i$ the residuals. We used a vector notation: $\mathbf{Birds}_i = (\text{Birds}_{i1}, \dots, \text{Birds}_{i27})'$ which contains the bird data of species i for all years. Just as in Chapter 5, we can write that the marginal distribution for \mathbf{Birds}_i is normally distributed with mean $\mathbf{X}_i \times \boldsymbol{\beta}$ and covariance \mathbf{V}_i . Equation (15.4) implies the following structure for \mathbf{V}_i .

$$\begin{pmatrix} \sigma_a^2 + \sigma_i^2 |\hat{\alpha} + \hat{f}_i(\text{Year}_1)|^{\delta_i} & \sigma_a^2 & \cdots & \sigma_a^2 \\ \sigma_a^2 & \sigma_a^2 + \sigma_i^2 \times |\hat{\alpha} + \hat{f}_i(\text{Year}_2)|^{\delta_i} & \cdots & \sigma_a^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_a^2 & \sigma_a^2 & \cdots & \sigma_a^2 + \sigma_i^2 \times |\hat{\alpha} + \hat{f}_i(\text{Year}_{27})|^{\delta_i} \end{pmatrix}$$

This matrix has a dimension of 19×19 as each bird species was measured over 19 years (the other years contain missing values). The covariance between two observations of the same bird species i is σ_a^2 , whatever the time lag between the two observations. The variance of bird species i depends on a bird-specific variance σ_i^2 and the fitted values in year s . (The parameter δ_i shows how strong the variance depends on the fitted values for species i .) The problem with this model is that computing time on an average computer is about 15 min and convergence problems arise. So we have to ask whether we are really interested in modelling heterogeneity between bird species as well as within each time series. The main underlying questions are related to effects of agricultural use on birds. Therefore, the first form of heterogeneity may not be of interest to this study. The easiest way to remove between-bird heterogeneity is by standardising each time series, e.g. by subtracting the mean of each time series and dividing by its standard deviation. The second form of heterogeneity requires a bit more thought.

15.3.3 Failed Approach 2

We have already established that we may not be interested in between species heterogeneity. Then why should we use seven variances to model it? Standardisation; subtracting the mean of each time series and dividing it by its standard deviation ensures that each time series is scaled in the same range. This allows us to drop the `varIdent` code to model different variances. The following R code standardises the data.

```
> Birds7 <- as.vector(as.matrix(scale(Ythan[,2:8])))
```

Note that this is nearly the same code as before, except that the `scale` function standardises each column in the selected part of the data matrix `Ythan`. We could have used:

```
> Birds7 <- c(scale(Ythan$Oystercatcher) ,
   scale(Ythan$Turnstone) ,
   scale(Ythan$Curlew) ,
   scale(Ythan$BartailedGodwit) ,
   scale(Ythan$Redshank) ,
   scale(Ythan $Knot) ,
   scale(Ythan$Dunlin))
```

Yet, a third option is to use `tapply`. We applied models (15.2) and (15.3) again, but this time we dropped the index i from σ_i^2 and α as all the time series are standardised (all have the same variance, and a mean of 0). R code for this is

```
f7 <- formula(Birds7 ~ 1+
 s(Year7, by = Oyst.01, bs = "cr") +
```

```

s(Year7, by = Turn.01, bs = "cr") +
s(Year7, by = Cirl.01, bs = "cr") +
s(Year7, by = Bart.01, bs = "cr") +
s(Year7, by = Reds.01, bs = "cr") +
s(Year7, by = Knot.01, bs = "cr") +
s(Year7, by = Dunl.01, bs = "cr"))
M2 <- gamm(f7, method = "REML", control = lmc,
weights = varPower(form =~ fitted(.) | ID7))

```

Again, the model with heterogeneity within the time series did not converge.

15.3.4 Assume Homogeneity?

Having tried everything we can think of, it is now time to acknowledge that for these data, we cannot easily model the heterogeneity within a bird time series. Probably, the data are just too short for this. There are now three options: (i) give up, (ii) transform the data and make statements on the transformed data, or (iii) assume homogeneity over time in Fig. 15.4. We decided to go for option 3. If the heterogeneity was more obvious, we would go for option (ii). The problem with option (ii) is that for other data sets, we saw that a transformation changed the shape of the trends. We readdress this issue in Section 15.5.

Before we can go into a discussion what graphs tell us in terms of biology, there is one last issue to discuss: independence over time.

15.4 Dealing with Independence

We return to the un-standardised data. We still have the potential problem of independence. The model in Equation (15.2) assumes that the residuals for bird species i in year s are independent of year $s - 1$, $s - 2$, etc. One way to verify this is the auto-correlation plot. However, due to the large number of missing values, a variogram may be a better tool to assess temporal dependence. The following R code extracts the residuals from the object M0 (the model in Equation (15.2)) and calculates a (robust) variogram.

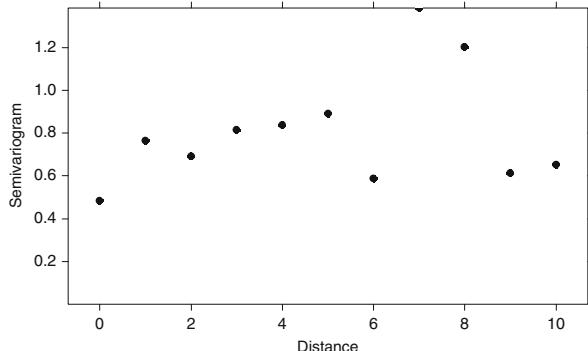
```

> plot(Variogram(M0$lme, form =~ Year7 | ID7,
                  maxDist = 10, robust = TRUE),
      pch = 16, smooth = FALSE, cex = 1.2)

```

The function `Variogram` is part of the `nlme` package. We used a maximum distance of 10 years as it is unlikely that birds in year s are affected by birds in year $s - 10$ (or over longer time lags). The `form` option specifies that the time series structure is within a bird species. If the points are scattered along a horizontal

Fig. 15.5 Variogram for the residuals obtained by the additive model in Equation (15.2). The horizontal axis represents distance between years and the vertical axis the value of the variogram



line in the variogram, independence of the residuals may be assumed. Figure 15.5 indicates that this may be a valid assumption.

A more formal way of assessing dependence over time is to include a time series correlation structure in the model and then test with the likelihood ratio test (if the models are nested) or compare the models with a tool like the AIC or BIC.

Adding a temporal correlation to the model in Equation (15.2) is relatively easy.

```
> M0A <- gamm(f7, method = "REML",
  control = lmc, weights = varIdent(form=~1|ID7),
  correlation = corSpher(form =~ Year7 | ID7,
  nugget = TRUE, fixed = FALSE))
```

The only new code is the correlation bit. It implements a spherical correlation structure as discussed in Chapter 7. In fact, we can try any of the following correlation options: No correlation, `corSpher`, `corRatio`, `corLin`, `corGaus`, `corExp`, and `corAR1`. R code for these models is given on the book website. The AIC value for the model without the temporal correlation was 1342.48, and the AICs of the models with a correlation structure were 1344.61 (`corSpher`), 1344.61 (`corLin`), 1343.77 (`CorRatio`), 1343.76 (`CorExp`), 1343.61 (`CorGaus`), and 1342.55 (`corAR1`). This means that adding a residual auto-correlation structure does not improve the model. Hence, our ‘optimal’ model is still the one in Equation (15.2).

We now have a look at the numerical output of M0. The estimated degrees of freedom, *F*-statistics, and *p*-values for the smoothers are obtained using the `anova(M0$gam)` command and are as follows.

Parametric Terms:

df	F	p-value
ID7	6	146.4 <2e-16

Approximate significance of smooth terms:

	edf	Est.rank	F	p-value
s(Year7):Oyst.01	3.626	8.000	6.101	1.72e-06
s(Year7):Turn.01	1.001	1.000	7.331	0.007861
s(Year7):Curl.01	1.001	2.000	6.839	0.001587
s(Year7):Bart.01	3.171	7.000	4.018	0.000594
s(Year7):Reds.01	3.259	7.000	4.882	7.98e-05
s(Year7):Knot.01	1.000	1.000	4.413	0.037946
s(Year7):Dunl.01	1.000	1.000	1.501	0.223209

All smoothers are highly significant, except for the Knot and Dunlin smoothers. The fitted values (= smoother plus intercept) are given in Fig. 15.6. Three species (oystercatcher, redshanks, and godwit) have high values around 1980 followed by a decrease. Knot, curlew, and turnstone show a nearly linear increase since the early 1970s. Confidence bands around the smoother for dunlin are rather larger and you should avoid drawing any conclusions for this species.

The following R code was used:

```
> P0 <- predict(M0$gam, se = TRUE)
> Isna <- is.na(Birds7)
> F <- P0$fit
> Fup <- P0$fit + 1.96 * P0$se.fit
> Flow <- P0$fit - 1.96 * P0$se.fit
> xyplot(F + Fup + Flow ~ Year7[!Isna] | ID7[!Isna],
  xlab = "Time", ylab = "Fitted values",
  lty=c(1, 2, 2), col = 1, type = c("l", "l", "l"),
  scales = list(alternating = TRUE,
    x = list(relation = "same"),
    y = list(relation = "free")))
```

Section 5.3.1 in Sarkar (2008) contains full details on the second part of this R code. First, we predict values from model M0. Because there are missing values, we have to remove them from the Year7 and ID7 vectors inside the `xyplot`. The variables F, Fup, and Flow contain the fitted values, upper confidence band, and lower confidence band, respectively. The `F + Fup + Flow ~ Year7` bit means that each vector is plotted versus Year7; it is not adding them up! Information on line type (`lty`) and type options are given in Sarkar (2008). Alternatively, just change the values and see what happens.

The shape of the trends in Fig. 15.6 makes one wonder whether we could summarise the oystercatcher, redshanks, and bar-tailed godwit with one trend and the turnstone, curlew and knot with another trend. To verify whether this is indeed the case, we can fit a model with seven trends and a model with three trends, use ML estimation in both models, and compare them using the AIC. The following code fits the model with three trends and with seven trends and compares them.

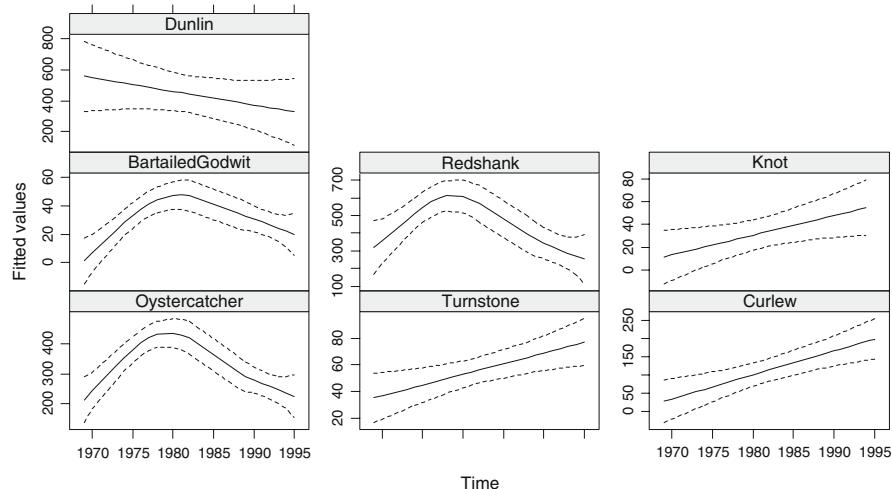


Fig. 15.6 Smoothing curves (solid line) obtained by the model in Equation (15.2). Dotted lines are 95% point-wise confidence bands

```

> ORB.01 <- as.numeric(ID7 == "Oystercatcher" |
+                         ID7 == "Redshank" |
+                         ID7 == "BartaileGodwit")
> TCK.01 <- as.numeric(ID7 == "Turnstone" |
+                         ID7 == "Curlew" |
+                         ID7 == "Knot")
> D.01 <- as.numeric(ID7 == "Dunlin")
> M0.3 <- gamm(Birds7 ~ 1 +
+               s(Year7, by = ORB.01, bs = "cr") +
+               s(Year7, by = TCK.01, bs = "cr") +
+               s(Year7, by = D.01, bs = "cr"),
+               method = "ML", control = lmc)
> M0.7 <- gamm(f7, control = lmc, method = "ML",
+               weights = varIdent(form =~ 1 | ID7))
> AIC(M0.7$lme, M0.3$lme)
      df      AIC
M0.7$lme 28 1452.157
M0.3$lme 20 1460.026

```

The AIC shows that the model with seven trends is better than the model with three trends. Perhaps, we were fooled in Fig. 15.6 by the different ranges along the vertical axes. If they are all the same, the smoothers look rather different from each other.

15.5 To Transform or Not to Transform

Initially, we were frustrated with the analysis of these data and after a couple of failed approaches, we convinced ourselves that we were not interested in the heterogeneity within a time series. We tried several transformations, and by trial and error, we found that the square root transformation stabilised the within-bird-time series variance. The problem is that a transformation not only removes heterogeneity, but it may also changes the shape of the trends (and therefore the conclusions). Applying the transformation is simple, just use

```
> Birds7 <- as.vector(as.matrix(sqrt(Ythan[, 2:8])))
```

The rest of the code is identical. The `varIdent` variance structure was needed and adding a residual auto-correlation did not improve the models. The predicted trends for these data are given in Fig. 15.7. Except for dunlin, the shapes of the trends are similar compared to those in Fig. 15.6. The only differences are that for the square-root-transformed data, we can safely assume homogeneity, but the smoothers are on the square root scale.

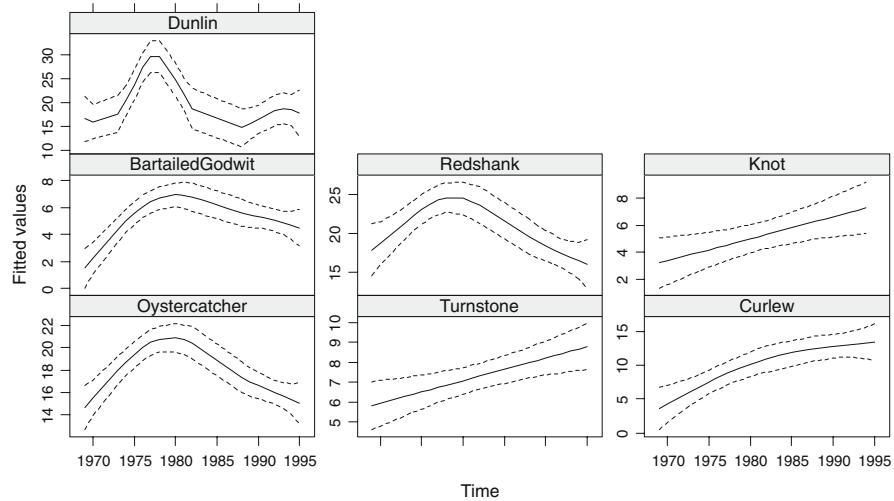


Fig. 15.7 Smoothing curves (solid line) obtained by the model in Equation (15.2). Square-root-transformed bird data were used. Dotted lines are 95% point-wise confidence bands

15.6 Birds and Explanatory Variables

In the previous section, we applied additive mixed models and found that 6 of the 7 birds could be divided into two groups. The oystercatcher, redshanks, and bar-tailed godwit follow a non-linear pattern over time with the highest values around

1980. The turnstone, curlew, and knot follow a linear and increasing pattern over time. Note that this is a visual observation; the actual trends are different from each other. No pattern for dunlin could be found. The question is now how to link the explanatory variables to either of the bird time series. The problem is that there are only 6 years in which both the birds and the explanatory variables were measured. The algorithm for additive modelling will use only these six years! So, there is no way we can add the explanatory variables into the additive mixed models used in the previous section. The other problem is that the shape of the smoothers in Fig. 15.2 indicate serious collinearity between nearly all explanatory variables. One of the few things that we can do is to plot the smoothers for the explanatory variables and the smoothers for the birds in one graph and see which ones are similar (Fig. 15.8). Because the explanatory variables had missing values, we predicted these values to avoid an erratic curve. But we did not predict values before the first year or beyond the last year of observation per explanatory variable. We put the curves that looked similar close to each other.

The concluding question is to decide which explanatory variable is best related to the two bird trends. Or perhaps we need to rephrase the question to: Which one is not related to the oystercatcher, redshanks, and bar-tailed godwit trend? The pigs, cattle, and sheep trends are remarkably similar to the oystercatcher, redshanks and bar-tailed godwit trend. Note that pigs and wheat follow similar patterns over time, but none of these clearly match any of the bird trends. Unfortunately, there are not enough observations in nitrate to say anything sensible, except that when the nitrate

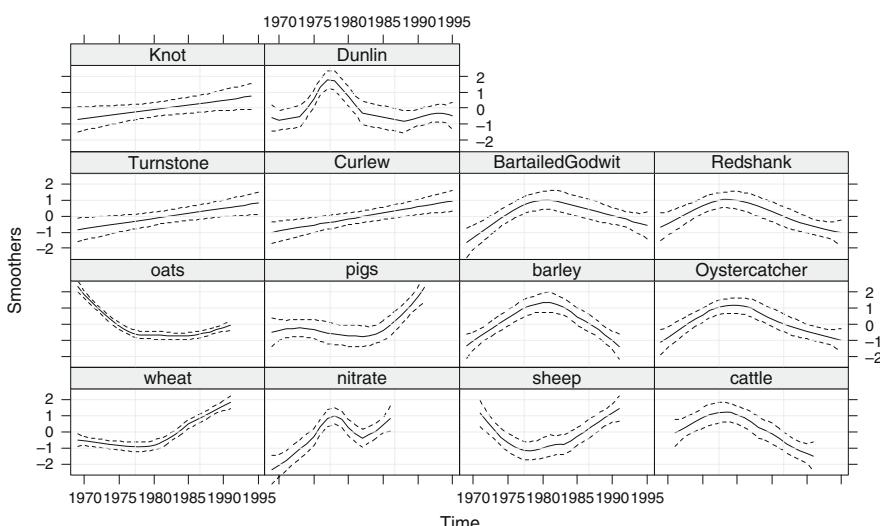


Fig. 15.8 Smoothing curves for each explanatory variable and the estimated smoothers from Fig. 15.6. The R code for this graph is presented on the book's website

trend went down in the late 1970s, the oystercatcher, redshanks, and bar-tailed godwit trends followed. However, this is rather speculative and based on a subjective observation.

15.7 Conclusions

The main statistical conclusions are that the seven bird time series seem to follow two patterns. The oystercatcher, redshanks and bar-tailed godwit all follow a similar (though not identical!) non-linear pattern over time, with the highest values around 1980. The turnstone, curlew, and knot all follow a linear (increasing) pattern over time. No pattern for dunlin could be found. The oystercatcher, redshank, and bar-tailed godwit trend seems to match the pigs, cattle, and sheep trends. When nitrate patterns changed, this bird trend changed as well. However, the data on the explanatory variables are too sparse to go beyond giving a nice multipanel graph where the bird trends and the trends for the explanatory variables are plotted.

These outcomes are interesting both ecologically and statistically. There were two main groups of trends over time in the bird data; one hump backed species group and one monotonic species group indicate that there are different ecological processes at work in this system: higher levels of nutrient enrichment seems good for some species, but bad for others. This may be at least in part explained by the ways in the elevated levels of nitrate are known to affect the invertebrates the birds feeding on. As nitrate levels increase in this system, the growth of mats of fast growing green seaweeds (henceforth termed ‘alal mats’) is stimulated, but the spatial distribution of these mats is very patchy. Underneath these patches, few of the invertebrates on which birds feed survive, but between the patches of weed, the same species of invertebrate thrive in the enriched conditions. So at low nutrient levels, the overall productivity of the estuary will increase, even though invertebrates are excluded from the patchy algal mats and the estuary can ‘carry’ more birds. At high levels of nutrients, however, the enriching effects on invertebrate numbers and biomass are markedly reduced as the algal mats spread into previously unaffected and enriched areas. Under this extensive cover of algal mats, the invertebrates on which birds feed virtually disappear over much of the estuary and shorebirds decline. One of the few species which is not reduced by the algal mats is the tiny mud snail *Hydrobia ulvae*, whose numbers may even increase within the mats. However, the tangled filamentous structure of the mats substantially reduces the foraging efficiency of the shorebirds so that there is no compensation for the loss of other invertebrates. One would therefore expect a non-monotonic trend in shorebirds over time with increasing nutrient run off as shown by oystercatcher, redshank, and bar-tailed godwit. Our analysis indicates that other factors are at work with respect to curlew, turnstone, and knot: Either they do not respond to algal mats in the same way as the other species or the continually increasing numbers on the estuary are a reflection of demographic processes happening outside the system, perhaps on the breeding grounds many thousands of kilometres distant. The statistical approaches used here

have crystallised this in a way which was not apparent in previous analyses, such as the likely different causes of changes in different shorebird species and allowed the framing of new research questions that can be explored in this system.

15.8 What to Write in a Paper

The first question we have to ask is as follows: Can we write a paper about the results presented in this chapter? The data analysed cover the life span of a scientific career, yet the data are too sparse to analyse bird data and agricultural variables in the same model. Having said this, fancy methods are not essential to compare trends in birds with agricultural variables. The link between nitrate and the oystercatcher, redshanks, and bar-tailed godwit trend is completely speculative and requires far more study before anything sensible can be said. This is something that should be made clear in the discussion of the paper! However, if you were to submit this chapter for publication, we would include the following.

1. An introduction describing the questions.
2. A Data and Methods section explaining how the data were collected and a few paragraphs on additive mixed modelling. Because this is a relatively new statistical method, half a page may be needed. You should explain the need for trying to add residual temporal correlation and heterogeneity structures. The referee may ask why you needed additive modelling, rather than just applying a transformation. Also justify why you used the Gaussian model and not the Poisson GLM or GAM.
3. It is tempting to present Fig. 15.2, but there will be a certain repetition with the graphs showing the final results.
4. You then need to summarise Section 15.3. Present the starting model, intermediate models, and the final model, (if these were not already presented in the Methods section), give AIC tables and likelihood ratio tests, and validate the optimal model. Present the F -values and p -values for the smoothers of the optimal model. Include the smoothers (or fitted values) for the optimal model (this is Fig. 15.6).
5. Make clear that due to the sparseness of the data, it is not possible to analyse bird data and agricultural variables in the same model. The only sensible thing to do is to present Fig. 15.8.
6. In the discussion, be sure not to say that sheep, barley or cattle are *driving* the mean winter values of oystercatcher, redshanks, and bar-tailed godwit. If anything, sheep, barley, or cattle are a measure of farming intensity, and increase use of fertilisers together with waste from livestock may drive nitrate concentrations in the Ythan. From this point onwards, the story becomes speculative, but interesting!

Chapter 16

Negative Binomial GAM and GAMM to Analyse Amphibian Roadkills

A.F. Zuur, A. Mira, F. Carvalho, E.N. Ieno, A.A. Saveliev, G.M. Smith,
and N.J. Walker

16.1 Introduction

This chapter analyses amphibian fatalities along a road in Portugal. The data are counts of kills making a Gaussian distribution unlikely; restricting our choice of techniques. We began with generalised linear models (GLM) and generalised additive models (GAM) with a Poisson distribution, but these models were overdispersed. To solve this, you can either apply a quasi-Poisson GLM or GAM, or use the negative binomial distribution (Chapter 9). In this particular example, either approaches can be applied as the overdispersion was fairly small (around 5), but with many ecological data sets it can be considerably larger, in which case the negative binomial GLM (or GAM) is the natural choice. As many textbooks give examples using quasi-Poisson GAMs and GLMs and only a few using the negative binomial, we decided to use the negative binomial distribution.

We chose GAM because the relationships between roadkills and explanatory variables were non-linear. We address issues like collinearity, residual patterns, and spatial correlations.

16.1.1 Roadkills

Since the second part of the twentieth century, roads have become a common feature in contemporary landscapes. For example, in North America alone, the road network has reached eight million kilometres and road construction is still increasing. Roads provide people and goods mobility, and are a central element in society (Forman et al., 2002). However, their impact on wildlife can be harmful as they (i) fragment populations, (ii) present barriers to dispersal as well as access to food and mates, and (iii) restrict gene flow. Also a large numbers of fatalities can occur as a result of animal–vehicle collisions.

A.F. Zuur (✉)
Highland Statistics Ltd., Newburgh, AB41 6FN, United Kingdom

The life cycle of most amphibians has an aquatic phase, corresponding to reproduction and to tadpole development and metamorphosis; and a terrestrial phase, when individuals use adjacent territory for foraging, shelter, periods of dormancy or overwintering (Semlitsch and Bodie, 2003). High levels of roadkills occur when roads cross amphibian migration routes to and from spawning sites or during juvenile dispersal (Langton, 2002).

The data presented in this chapter come from a two-year study on vertebrate roadkills in a National Road of southern Portugal (IP2, stretch Portalegre-Monforte, 27 km long). The surveyed road has paved verges with two lanes and a moderate amount of traffic (less than 10,000 vehicles per day). Road surroundings are dominated by cork *Quercus suber* and holm oak *Q. rotundifolia* tree stands, named ‘montado’ and open land, including pastures, meadows, and fallows.

The road was inspected for amphibian roadkills every two weeks between March 1995 and March 1997. Surveys were made by a car slowly (10–20 km per hour) driving along the road on the hard-shoulder. Each animal found dead was identified to species level, whenever possible, and its geographic location, on UTM coordinates, was determined with help of detailed cartography (1:2000) of horizontal and vertical road profiles and aerial photographs. All carcasses were removed from the road to avoid double counting.

For data analysis purposes, the road was divided in 500 m segments. The response variable is the total number of amphibian fatalities per segment. All animals found dead on each segment were allocated to the coordinates of its middle point. Figure 16.1 shows an example of one of the species recorded.

Detailed digital maps of land use were made through interpretation of aerial photographs corrected with field observations. Explanatory variables were identified from these maps using a Geographic Information System. A list with all available explanatory variables and the abbreviations used is given in Table 16.1.



Fig. 16.1 *Pelobates cultripes*, one of the species that was used in our data. The photograph was taken by Marco Caetano

Table 16.1 List of explanatory variables and the abbreviation used in this chapter

Variable	Abbreviation
Open lands (ha)	OPEN.L
Olive grooves (ha)	OLIVE
Montado with shrubs (ha)	MONT.S
Montado without shrubs (ha)	MONT
Policulture (ha)	POLIC
Shrubs (ha)	SHRUB
Urban (ha)	URBAN
Water reservoirs (ha)	WAT.RES
Length of water courses (km)	L.WAT.C
Dirty road length (m)	L.D.ROAD
Paved road length (km)	L.P.ROAD
Distance to water reservoirs	D.WAT.RES
Distance to water courses	D.WAT.COUR
Distance to Natural Park (m)	D.PARK
Number of habitat Patches	N.PATCH
Edges perimeter	P.EDGE
Landscape Shannon diversity index	L.SDI

They include areas occupied by each land cover class, total length of roads and water courses on a 2,000 m strip centred on each road segment; landscape indexes (total number of patches; total perimeter of edges between different land cover classes; and landscape Shannon diversity index which relates to landscape heterogeneity); and distances from the segment centre to water and to the southwest limit of S. Mamede Natural Park (a mountain range NE-SW oriented that is known for its high levels of humidity and rainfall, where landscapes are particularly well preserved and are good examples of harmonious interactions between man and nature).

The underlying ecological question in this chapter is simple: is there a relationship between amphibian roadkills and any of the explanatory variables?

16.2 Data Exploration

The data were measured along the road, and the sampling positions are marked as dots in Fig. 16.2. The R code we used for this is as follows.

```
> library(AED); data(RoadKills)
> RK <- RoadKills
> library(lattice)
> xyplot(Y ~ X, aspect = "iso", col = 1, pch = 16,
  data = RK)
```

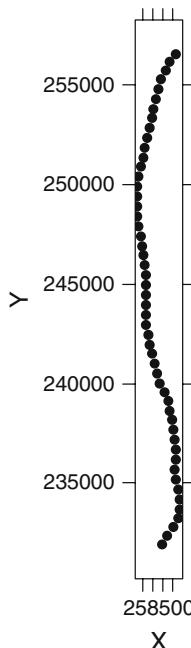


Fig. 16.2 Positions of the sampling points along the road

The first two commands access the data. We renamed the object `RoadKills` to `RK` as it is shorter. The `xyplot` command produces Fig. 16.2. The variable `D.PARK` is the distance (along the road) to the Natural Park, north of the sampling area. It therefore represents the distance (along the road) between each sampling point and the most northerly sampling site. If `D.PARK` had not been quantified, you would need to calculate the distance between each observation and the most northerly point yourself using the Pythagoras rule.

Using Cleveland dotplots (not shown here), pairplots, and initial GAM analyses, we decided to square root transform the explanatory variables `POLIC`, `WAT.RES`, `URBAN`, `OLIVE`, `L.P.ROAD`, `SHRUB`, and `D.WAT.COUR`.

There are 17 explanatory variables and only 52 observations. With such a low number of observations, we prefer not to use more than 5 or 6 explanatory variables, especially if we intend to use smoothing techniques. Furthermore, correlation coefficients between some of the explanatory variables are high. Because correlation coefficients only show pairwise correlations, we used variance inflation factors (VIF) to assess which explanatory variables are collinear and should be dropped before starting the analyses. VIFs were also used in Appendix A. We wrote our own R functions to calculate VIF values and these are part of our AED package. They are calculated with the following commands.

```

> RK$SQ.POLIC <- sqrt(RK$POLIC)
> RK$SQ.WATRES <- sqrt(RK$WAT.RES)
> RK$SQ.URBAN <- sqrt(RK$URBAN)
> RK$SQ.OLIVE <- sqrt(RK$OLIVE)
> RK$SQ.LPROAD <- sqrt(RK$L.P.ROAD)
> RK$SQ.SHRUB <- sqrt(RK$SHRUB)
> RK$SQ.DWATCOUR <- sqrt(RK$D.WAT.COUR)
> Z<-cbind(RK$OPEN.L, RK$SQ.OLIVE, RK$MONT.S, RK$MONT,
            RK$SQ.POLIC, RK$SQ.SHRUB, RK$SQ.URBAN,
            RK$SQ.WATRES, RK$L.WAT.C, RK$L.D.ROAD,
            RK$SQ.LPROAD, RK$D.WAT.RES, RK$SQ.DWATCOUR,
            RK$D.PARK, RK$N.PATCH, RK$P.EDGE, RK$L.SDI)
> corvif(Z)

```

The resulting VIF values are given in Table 16.2. As explained in Appendix A, a cut-off value of 5 or even 3 can be used to remove collinear variables; we used 3. To find a set of explanatory variables that does not contain collinearity, we removed one variable at a time, recalculated the VIF values, and repeated this process until all VIF values were smaller than 3. As a result, MONT, P.EDGE, N.PATCH, L.SDI, and SQ.URBAN were dropped. This means that we have 12 remaining explanatory variables. This is still a large number of variables!

We also present a scatterplot of all 12 selected explanatory versus the number of amphibian roadkills, see Fig. 16.3. We added a LOESS smoothing curve to help interpretation. The shape of the curves and the spread of the data around the smoothing curves do not look promising for good analysis. The only variables that seem to have a clear relationship with the roadkills are D.PARK and D.WAT.RES.

The R code for this graph is a bit a pain, but it is worth the effort. We basically need three columns of data. In the first column (`Killing12`), we copy and paste the variable containing the roadkills 12 times.

In the second column (`X12`), we concatenate the data of all 12 explanatory variables. The third column (`ID12`) needs to contain the name of the first explanatory

Table 16.2 Variance inflation factors for the full set of explanatory variables

Variable	GVIF	Variable	GVIF
OPEN.L	161.01	L.D.ROAD	4.41
SQ.OLIVE	34.44	SQ.LPROAD	3.38
MONT.S	3.96	D.WAT.RES	2.11
MONT	213.63	SQ.DWATCOUR	2.55
SQ.POLIC	3.89	D.PARK	2.91
SQ.SHRUB	3.32	N.PATCH	24.30
SQ.URBAN	14.03	P.EDGE	19.36
SQ.WATRES	1.98	L.SDI	10.02
L.WAT.C	3.64		

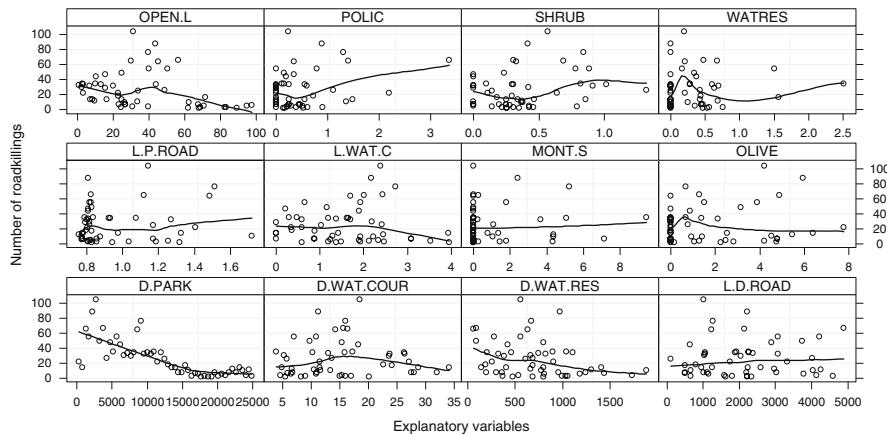


Fig. 16.3 Scatterplots of the number of amphibian roadkills (y-axis) against each of the 12 remaining explanatory variables. The heading in a panel indicates which explanatory variable is plotted along the x-axis. A smoothing (LOESS) curve was added in each panel

variable 52 times, the name of the second variable 52 times, etc. The rest is some fancy xyplot coding.

```
> X12 <- c(RK$OPEN.L, RK$SQ.OLIVE, RK$MONT.S,
  RK$SQ.POLIC, RK$SQ.SHRUB, RK$SQ.WATRES,
  RK$L.WAT.C, RK$L.D.ROAD, RK$SQ.LPROAD,
  RK$D.WAT.RES, RK$SQ.DWATCOUR, RK$D.PARK)
> Killings12 <- rep(RK$TOT.N, 12)
> I12 <- rep(c("OPEN.L", "OLIVE", "MONT.S", "POLIC",
  "SHRUB", "WATRES", "L.WAT.C", "L.D.ROAD",
  "L.P.ROAD", "D.WAT.RES", "D.WAT.COUR",
  "D.PARK"), each = 52)
> ID12 <- rep(I12, 12)
> library(lattice)
> xyplot(Killings12 ~ X12 | ID12, col = 1,
  strip = function(bg = 'white', ...),
  strip.default(bg = 'white', ...),
  scales = list(alternating = TRUE,
    x = list(relation = "free"),
    y = list(relation = "same")),
  xlab = "Explanatory variables",
  ylab = "Number of roadkillings",
  panel = function(x, y){
    panel.grid(h = -1, v = 2)
    panel.points(x, y, col = 1)
    panel.loess(x, y, col = 1, lwd = 2)}))
```

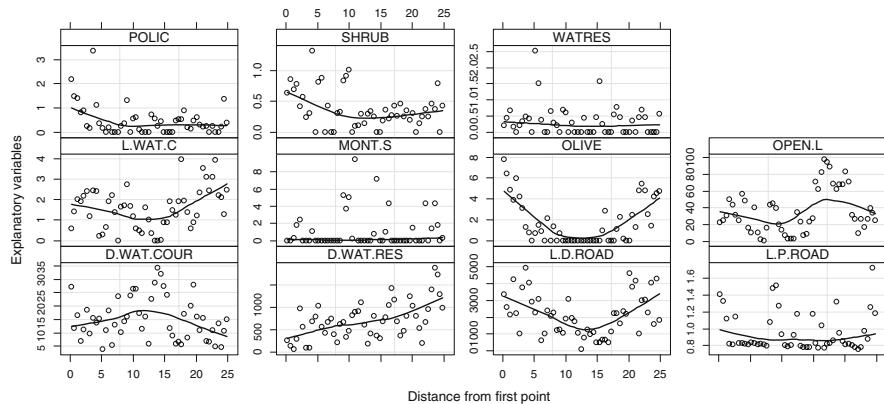


Fig. 16.4 Scatterplots of the explanatory variables (y-axis) versus the spatial variable D.PARK (distance from the first point expressed in km). The heading in a panel indicates which explanatory variable is plotted along the y-axis. A smoothing (LOESS) curve was added in each panel

The `strip` and `strip.default` options ensure that the boxes with the labels are white. The `scales` option allows for different ranges along the *x*-axes, but all *y*-axes have the same range. The `panel` function adds a grid, points, and a LOESS smoother; see also the bioluminescent case study in Chapter 17.

Now that we have this fancy R code, we would like to go one step back and focus on collinearity again. Figure 16.4 shows a similar plot as in Fig. 16.3, except that the explanatory variables are now plotted along the *y*-axis and the variable distance to the first point (D.PARK) along the *x*-axis.

We made this graph to get a feel for the spatial patterns of the explanatory variables. The variables OLIVE, D.WAT.RES, and L.D.ROAD show a clear pattern with D.PARK. Note that non-linear relationships are not picked up by the VIF. GAMs are rather sensitive to collinearity (Chapter 3), and we should not use D.PARK together with any of these three variables as they all represent the spatial position of the sampling locations. Because D.PARK has a clear ecological interpretation and it is easier to use in the independence verification later on, we decided to drop OLIVE, D.WAT.RES, and L.D.ROAD. The R code to produce Fig. 16.4 is similar to the code used for Fig. 16.3 and is not reproduced here.

16.3 GAM

The data exploration did not show any clear *linear* patterns between roadkills and the explanatory variables; so we need to move on to using a GAM. Furthermore, an initial GLM with a Poisson distribution and logarithmic link function gave an overdispersion of 5, and we therefore proceed with a GAM with a negative binomial distribution and logarithmic link function. The negative binomial distribution (Chapters 8 and 9) is useful if the variance is much larger than the mean. As it is for

this data set, where the mean number of roadkills is 25.9 and the variance is 589.3. Recall from Chapter 9 that the negative binomial GAM is given by

$$RK_i \sim NB(\mu_i, k)$$

$$E(RK_i) = \mu_i \quad \text{and} \quad \text{Var}(RK_i) = \mu_i + \frac{\mu_i^2}{k}$$

$$\mu_i = e^{\alpha + f_1(\text{OLIVE}_i) + \dots + f_{10}(\text{D.WAT.COUR}_i)}$$

RK_i is the number of amphibian roadkills at site i , where $i = 1, \dots, 52$. The notation $f_j(X)$ stands for ‘smoothing function of the explanatory variable X ’, and NB is a negative binomial distribution with mean μ_i and dispersion parameter k . The explanatory variables in the model are OPEN.L, MONT.S, SQ.POLIC, SQ.SHRUB, SQ.WATRES, L.WAT.C, SQ.LPROAD, SQ.DWATCOUR, and D.PARK. To fit the GAM, we can use the following R code.¹

```
> library(mgcv)
> library(MASS)
> M1 <- gam(TOT.N ~ s(OPEN.L) + s(MONT.S) +
+             s(SQ.POLIC) + s(SQ.SHRUB) + s(SQ.WATRES) +
+             s(L.WAT.C) + s(SQ.LPROAD) + s(SQ.DWATCOUR) +
+             s(D.PARK), family = negative.binomial(1),
+             data = RK)
```

The package MASS is needed for the negative binomial distribution. The (1) in the code `negative.binomial(1)` means that the `gam` function will estimate the optimal dispersion parameter k .

The problem here is that this model gives an error message: `Model has more coefficients than data.` Because the model is applying cross-validation, some combinations of smoothers will use more than 52 degrees of freedom and we have only 52 observations. One option is to set an upper limit to the degrees of freedom; just extend the code with `s(OPEN.L, k = 4)` and do this for all terms in the GAM.

To find the optimal model, you can use shrinkage smoothers; these will also consider 0 degrees of freedom. If the model has multiple smoothers with 0 degrees of freedom, then you can drop them simultaneously. It is a faster alternative to a step-wise backward selection using, for example, the AIC or GCV. Shrinkage smoothers are obtained by using the `bs` option inside the `s` command, and specifying one of the shrinkage smoothers, for example, `s(OPEN.L, k = 4, bs = "ts")` or `s(OPEN.L, k = 4, bs = "cs")`. You can do this for all smoothers.

The `anova` command can be used to obtain F -statistics and approximate p -values for the smoothers. Results are not shown here, but most of the smoothers are not significant at the 5% level. Dropping the least significant smoother, and refitting the model until all terms are significant, is a highly confusing exercise for

¹We used R version 2.6 and mgcv version 1.3–27. More recent versions of R and mgcv require a small modification to the code; see the book website (www.highstat.com) for updated code.

this data set. In one round, variables x and y are highly significant, and in another round, variable z is highly significant, but not x and y . This is clear evidence that there is still a certain degree of collinearity in the model. However, whichever model we applied, the variable D.PARK was always significant.

Another problem with this whole approach is that by setting an upper limit to the degrees of freedom, we may miss important variables that have a highly non-linear effect. We therefore follow a different model selection approach of forward selection. We started with a GAM that used only one explanatory variable, fitted 9 different models, and compared their AICs (obtained by the AIC command). The model with the (by far) lowest AIC was the one with D.PARK. Its AIC was 352.5, whereas the second best model (with only OPEN.L) had an AIC of 423.2. We then fitted 8 GAMS, each with two explanatory variables, one which was D.PARK. The combination D.PARK and OPEN.L had the lowest AIC (340.3). We then continued with GAMs containing 3 smoothers: D.PARK, OPEN.L, and a third variable, but no combination gave a model with a better AIC and significant smoothers. Hence, following a forward selection approach, we end up with D.PARK and OPEN.L. To run this model in R, use

```
> M2 <- gam(TOT.N ~ s(OPEN.L) + s(D.PARK) ,
              family = negative.binomial(1), data = RK)
> anova(M2)

Family: Negative Binomial(28.7654)
Link function: log
Formula:
TOT.N ~ s(OPEN.L) + s(D.PARK)

Approximate significance of smooth terms:
          edf  Est.rank    F  p-value
s(OPEN.L) 8.107     9.000  3.641  0.00282
s(D.PARK) 8.727     9.000 26.509 6.78e-13
```

The smoother for D.PARK is highly significant, and the OPEN.L smoother has a p -value of 0.003. The estimated smoothers are given in Fig. 16.5. The pattern for D.PARK can be seen in all sub-optimal models and also in the data exploration graphs. It shows a clear decrease along the gradient up to 18 km from the park and a slight increase after that distance. It is important to note that D.PARK reflects the distance to a mountain range where rainfall and humidity are higher than in surrounding flat areas. A decreasing gradient in these environmental conditions is expected to take place along the road as we move south. Moreover, both edges of the sampled road are in the boundaries of two localities (Portalegre and Monforte) with an agriculture matrix of small orchards and vegetable gardens. These are places where water availability in small ponds and channels for irrigation proposes is usually high. So the pattern found for D.PARK smoother is consistent with amphibian needs, concerning water availability. As higher amphibian abundances are expected in moist and wetter areas you would expect higher numbers of roadkills in these environments. The highest fatalities occurring in the first few kilometres of the sampled road probably also reflects the cumulative effect of water availability on

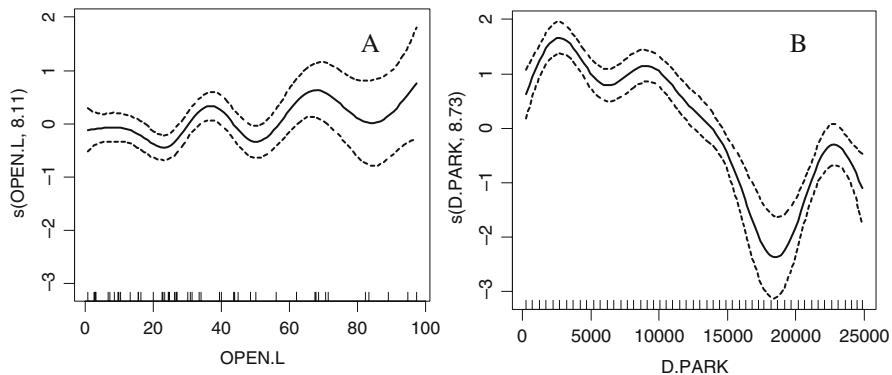


Fig. 16.5 A: Smoother for OPEN.L. B: Smoother for D.PARK. Both smoothers are from the optimal GAM model

mountain range and land use at this end of the road. At the other end of the road, only the land use is influencing the results.

The shape and interpretation of the smoother for OPEN.L is unclear. Based on the vertical ranges in both panels, the variable D.PARK contributes more to the fitted values than OPEN.L. We are rather tempted to drop OPEN.L from the model as we expect that the bumpy pattern may reflect some collinearity problems between D.PARK and OPEN.L. Figure 16.4 already indicated some sort of pattern between them, and perhaps, it was not a good idea to use them both.

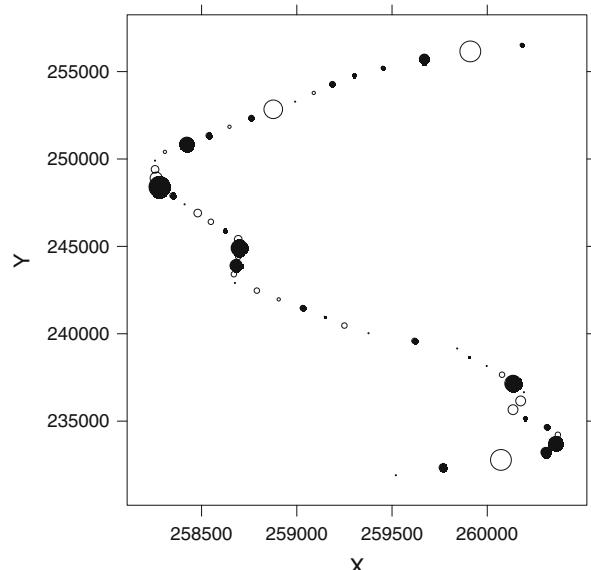
The output of the `summary` command (not shown here) shows that this model explains 93.7% of the variation, and the dispersion parameter for the negative Binomial distribution is 28.7 (see also Chapter 9).

As part of the model validation, we also need to look at independence. Sites close to each other may have similar roadkill levels. To verify this, we can plot the residuals against the spatial coordinates. The problem is that the ranges along the horizontal and vertical axes in Fig. 16.2 make it rather difficult to do this. However, we can distort the shape of the picture a little bit by omitting the `aspect` option in the `xypplot` command:

```
> E <- resid(M2, type = "pearson")
> I <- vector(length = length(E))
> I[E < 0] <- 1
> I[E >= 0] <- 16
> library(lattice)
> xypplot(Y ~ X, cex = 2 * abs(E) / max(abs(E)),
  pch = I, col = 1, data = RK)
```

The resulting graph (Fig. 16.6) shows the distorted road, but it allows for a visual inspection of the residuals. There are no immediate clear clusters of negative or positive residuals, and there are no clear clusters of large (in absolute sense) values.

Fig. 16.6 Residuals of the optimal GAM model are plotted versus the spatial coordinates of the sites. The scales of the axes were distorted to make all the dots visible. The larger the dot, the larger the residual (in absolute sense). Filled circles have a negative sign for the residuals and the open circle positive values. There should be no clustering of positive or negative residuals or clustering of large values



Another part of the model validation process consists of plotting the residuals of the optimal GAM model versus all explanatory variables; see Fig. 16.7. You should not be able to see any patterns in these graphs. In this case, there are some patterns, but they are not strong enough as judged by the AIC (adding any of these terms as a smoother to the model results in higher AIC or non-significant smoothers) to be of concern.

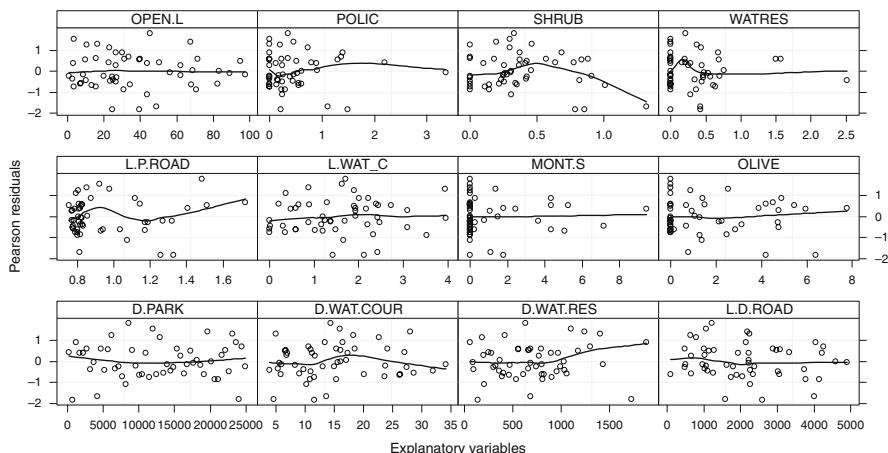


Fig. 16.7 Residuals of the optimal GAM model versus all explanatory variables. A LOESS smoothing curve was added

The R code to produce Fig. 16.7 is similar to the code used for Figs. 16.3 and 16.4 and is not presented here. Just replace the first column for the vertical axes by the residuals obtained by `residuals(M2, type = "pearson")`.

16.4 Understanding What the Negative Binomial is Doing

Before continuing with the GAMM section, we show what the negative binomial model is doing. In this discussion, it is easier to use a model that only contains D.PARK. Besides, we were not impressed with the role of OPEN.L anyway. The following code fits the negative binomial GAM with only D.PARK as a smoother.

```
> M3 <- gam(TOT.N ~ s(D.PARK), data = RK,
   family = negative.binomial(1))
```

The estimated smoother has a similar pattern as Fig. 16.5B. The estimated parameter k is 11.8. To better visualise what this model is doing, we will draw the fitted values on the real scale, add confidence bands around the fitted values, and superimpose values from a negative binomial distribution with the mean value given by the fitted GAM values and the dispersion parameter of 11.8. The following code achieves this, and the results are given in Fig. 16.8.

```
> M3Pred <- predict(M3, se = TRUE, type = "response")
> plot(RK$D.PARK, RK$TOT.N, cex = 1.1, pch = 16,
   main = "Negative binomial GAM",
   xlab = "Distance to park",
   ylab = "Number of road killings")
```

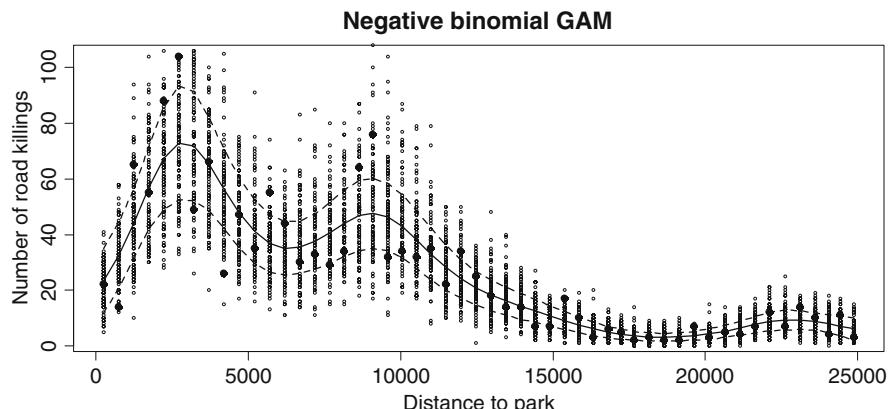


Fig. 16.8 Fitted values (solid line) and approximate 95% confidence bands (dotted lines) for the mean obtained by the negative binomial GAM. The large filled dots are observed values, and the small dots are 100 random samples per site taken from a negative binomial distribution

```

> I <- order(RK$D.PARK)
> lines(RK$D.PARK[I], M3Pred$fit[I], lwd = 2)
> lines(RK$D.PARK[I], M3Pred$fit[I] +
+        2 * M3Pred$se.fit[I], lty = 2, lwd = 2)
> lines(RK$D.PARK[I], M3Pred$fit[I] -
+        2 * M3Pred$se.fit[I], lty = 2, lwd = 2)
> for (i in 1:52) {
+   y <- rnbinom(100, size = 11.8, mu = M3Pred$fit[i])
+   points(rep(RK$D.PARK[i], 100), y, cex = 0.5)
}

```

The `predict` command takes the results from the GAM model and predicts fitted values on the response scale. The `plot` command sets up the graph with the observed values (`pch = 16` produces filled circles). The `lines` commands are used to draw the fitted values (solid thick line in the middle) and approximate pointwise 95% confidence bands (thick dotted lines) for the mean. So far, we have used no new R code; all this was used earlier in Chapter 3. The new bit comes now. The loop takes the fitted values at site i (given by $\mu = \text{M3Pred$fit}[i]$), and using a parameter of $k = 11.8$, it draws 100 values from a negative binomial distribution, which are superimposed on the graph with the `points` command at the value of `D.PARK` for each the site. It gives an impression of the likely (road killing) values at any particular site. Unfortunately, it is rather difficult to draw the negative binomial density curves on top of this graph as we did in Chapter 2 for the Normal distribution. Figure 16.9 shows density curves for different values along the `D.PARK` gradient, and you can imagine these density curves on top of the fitted values in Fig. 16.8.

Density curves from a Poisson GAM are considerably less wide.

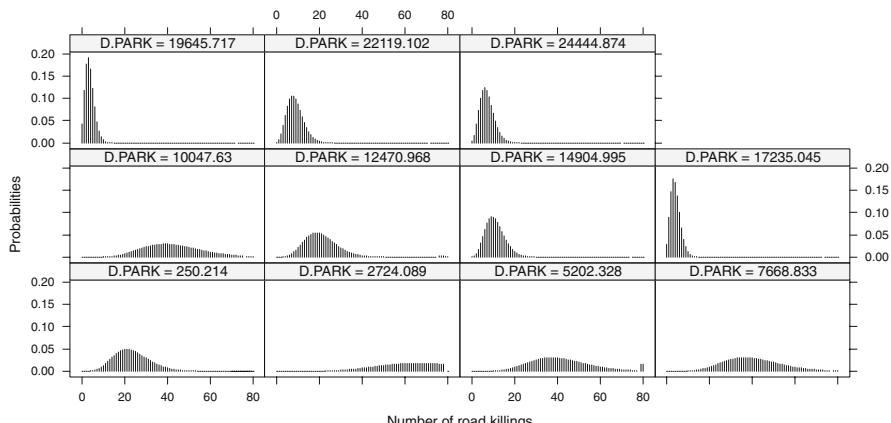


Fig. 16.9 Examples of negative binomial distributions. The density curves have a parameter of $k = 11.8$, and the mean value μ was taken from the fitted values at certain arbitrary chosen values along the `D.PARK` gradient

16.5 GAMM: Adding Spatial Correlation

In the previous section, we applied a GAM and found that the optimal model contains a smoother for D.PARK and OPEN.L. The residuals were plotted against the spatial coordinates, and we could not see any clear spatial patterns in these residuals. Instead of making this plot, we can also make a variogram of the residuals. The easiest option is to use the function `Variogram` from the `nlme` package, which is designed to work with the `gls`, `lme`, and `gamm` functions. All we need to do now is to rerun the GAM as a GAMM, just like we reran the linear regression with a GLS in Chapter 4 and use the `Variogram` function on its results. The code is given below and the resulting graph in Fig. 16.10, where there is a minor indication that points close to each other are more similar than points further separated along the road (this can be seen from a slightly increasing pattern in the variogram). However, one can equally well argue that the points form a horizontal band of points, indicating independence.

```
> library(nlme)
> RK$D.PARK.KM <- RK$D.PARK / 1000
> M4 <- gamm(TOT.N ~ s(OPEN.L) + s(D.PARK), data = RK,
  family = negative.binomial(theta = 11.8))
> M4Var <- Variogram(M4$lme, form =~ D.PARK.KM,
  nugget = TRUE, data = RK)
> plot(M4Var, col = 1, smooth = FALSE)
```

It is also possible to add a spatial correlation structure to the model and see whether it improves anything. This can easily be done by using one of the available correlation structures `corExp`, `corSpher`, `corRatio`, or `corGaus`. According to the protocol defined in Chapters 4 and 5, we should start with a model containing smoothers of all explanatory variables. However, such a model did not converge. We therefore used the optimal model from the GAM with D.PARK and OPEN.L and

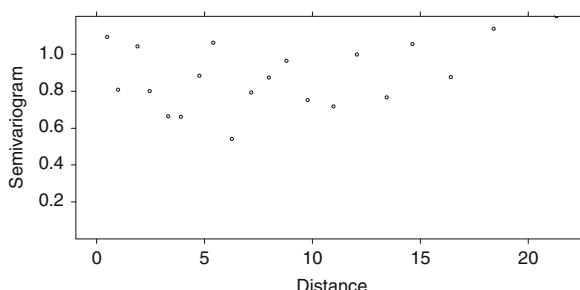


Fig. 16.10 Variogram of the residuals of the optimal GAM model with D.PARK and OPEN.L as smoothers. The variogram indicates independence as the points seem to form almost a cloud of horizontal points. Spatial correlation is present if we can see an increasing pattern up to a certain level

added a spatial correlation structure. Of all the spatial correlation structures, only the `corGaus` converged. This model is fitted by the following R code.

```
M5 <- gamm(TOT.N ~ s(OPEN.L) + s(D.PARK), data = RK,
            family = negative.binomial(theta = 11.8),
            correlation = corGaus(form =~ D.PARK.KM,
            nugget = TRUE))
```

However, the estimated range is close to 0, meaning that the chosen correlation structure makes no sense. When fitting these models without the smoother for OPEN.L, most convergence problems disappeared, but the spatial correlation functions gave rather different ranges and sill. It may be better to choose the range and sill interactively based on the residuals from the optimal GAM models in Fig. 16.10.

16.6 Discussion

This chapter provides an example of a data analysis that shows how important it is to do a good model validation and how confused you can get from a GAM if you ignore collinearity before starting the analysis. Having a large number of explanatory variables that are all linked to the spatial position of the sites (distance to water, distance to a park, etc.) does not help.

The results indicate that the variable D.PARK is the most important variable explaining amphibian roadkills. The optimal GAM model also contained OPEN.L, but the shape of the smoother is difficult to interpret and the model contained a small (tiny) amount of residual spatial correlation.

16.7 What to Write in a Paper

You need to emphasise the data exploration and problems with collinearity. You also need to discuss the interpretation of the variable D.PARK and why it was used as an explanatory variable. It is important to explain that there is no violation of independence in the model.

Chapter 17

Additive Mixed Modelling Applied on Deep-Sea Pelagic Bioluminescent Organisms

A.F. Zuur, I.G. Priede, E.N. Ieno, G.M. Smith, A.A. Saveliev, and N.J. Walker

17.1 Biological Introduction

The oceans, with a mean depth of 3,729 m and extending to a maximum depth of 11 km comprise the largest habitat on earth. The distribution of living organisms in this vast environment is far from uniform and description of this variation in space and time is challenging, both from the point of view of sampling and of statistical analysis. Most life in the oceans is dependent on primary production in the surface layers, generally in the epipelagic zone down to a depth of 200 m, where there is sufficient solar radiation to sustain photosynthesis. Microscopic algae or phytoplankton containing the pigment chlorophyll intercept solar light and use the energy to combine CO₂ and water to produce simple sugars polysaccharides, oils, proteins, and all the other constituents of the living organism. The algae and phytoplankton are either consumed by planktonic animals or dies loses buoyancy and becomes part of the downward stream of particulate organic matter (POC) falling towards the sea floor. The primary consumers themselves produce faecal pellets that enhance the POC flux and also form the basis of the food chain in the surface layers of the oceans. Predators living at greater depths also ascend at night to feed on the surface riches and then descend during the day digesting and excreting as they go. Thus, surface-derived production is exported downwards by passive and active processes sustaining life throughout the water column down to the abyssal sea floor.

There is therefore a general pattern of decrease in species abundance and biomass with depth. There are linear and non-linear components to this decline. Pressure, which increases linearly with depth, tends to disrupt biochemical reactions so that deep living organisms have acquired specially adapted protein structures. Below the photic zone, temperatures become lower, defining a cut off at the thermocline beneath which biochemical processes are slower. In this zone, biomass consumes oxygen, which in the absence of replenishment by photosynthesis can result in an oxygen minimum zone at around 1,000 m depth where life can become impossible.

A.F. Zuur (✉)
Highland Statistics Ltd., Newburgh, AB41 6FN, United Kingdom

Below this depth, sea water is cold, well-oxygenated, and ventilated by water originating from the sinking of cold water in the polar regions. Non-linearities can also be introduced by the presence of distinct water masses of different densities stacked on top of each other at different depths producing a layered effect in the ocean. Widder et al. (1999), for example, describes high animal abundances in a thin layer of less than a metre thick in the Gulf of Maine at a density discontinuity.

Most deep sea organisms are capable of emitting light in the form of bioluminescence. Usually blue light is produced either from discrete light organs or as luminescent secretions released into the water. This luminescence can be mechanically stimulated and is either the result of an alarm response by the organism or in the case of fragile animals, disintegration, and release of luminescent material into seawater. This is the mechanism that produces the so-called phosphorescent wake of ships and boats on calm nights in the open ocean. For scientific investigations, bathyphotometers are used. These work by pumping water through a chamber equipped with a light sensor, which counts the number of photons produced per litre of water. This system works well in the surface layers where organisms may occur at over $1,000 \text{ m}^{-3}$ but pumping becomes impractical at depths greater than 1,000 m where organisms are rare. For investigations in deeper waters, Priede et al. (2006) developed a free-fall vehicle with a downward looking high sensitivity ISIT (Intensified Silicon Intensified Target) video camera focussed on a 0.19 m^2 mesh screen that filled the field of view of the camera. Flashes of light produced by luminescent organisms impinging on the screen as it descends at $0.6 \text{ m}\cdot\text{s}^{-1}$ through the water column are counted to estimate the number of bioluminescent organisms per m^3 of seawater at different depths. Since over 80% of deep sea organisms are capable of luminescence, this is a novel means of producing continuous vertical profiles of marine life abundances. In practise, the ISIT profiler cannot be used at depths less than 300 m, because surface light could damage the sensitive camera and obscure bioluminescent flashes. Luminescent sources are counted over 30-s time samples, which correspond to a depth interval of 18 m between readings depending on the exact descent rate of the lander, which can vary slightly from profile to profile.

Abundance of deep-sea bioluminescent organisms is also dependent on the intensity of overlying primary production that can vary considerably in different parts of the ocean. Temperate latitudes are characterised by highly seasonal peaks of primary production in the spring followed by a fall out of POC towards the seafloor during summer, whereas in the centre of tropical gyre regions, primary production is low and uniform throughout the year (Longhurst, 1998). In addition to seasonal and regional differences, primary production can be very irregular, occurring in patches such as eddies of water spinning in the vicinity of oceanic fronts.

Bradner et al. (1987), using a free-falling photomultiplier device, concluded that in the Pacific Ocean off Hawaii bioluminescence decreases exponentially with depth. The first results from the ISIT free-fall profiler in the Tropical Atlantic Ocean indicated a monotonic decline in abundance with depth, but the relationship was not truly exponential (Priede et al., 2006). These studies, however, gave no information on seasonal changes.

17.2 The Data and Underlying Questions

The data analysed in this chapter were gathered during a series of four cruises of the *Royal Research Ship Discovery* over two years (2001 and 2002) in the temperate NE Atlantic west of Ireland (Gillibrand et al., 2006); see Fig. 17.1. The primary purpose of the cruises was to investigate deep-sea fish living on the sea floor in the Porcupine Seabight and on the Porcupine Abyssal Plain. Cruises were organised in spring and late summer to collect samples before and after the seasonal downward flux of POC that occurs in June and July (Lampitt et al., 2001). Timing of the cruises could not be precisely controlled since ship allocation is determined by conflicts between requirements of different programmes and logistic considerations. In 2001, the cruises were in April and August, and in 2002, they were in March and October. The ISIT free-fall profiler was deployed opportunistically between trawling and other sampling operations. Each location where the ship stopped to launch the profiler is termed a station. Depending on the weather conditions, the crew would then prepare to launch the instrument over the stern (back) of the ship as it moved forward slowly at approximately 1 knot. Once the equipment was streaming behind the ship, it was released and allowed to fall towards the sea floor. A timer activated the recording system after a set delay; so the depth of starting the recording and the precise location of the profile depended on the promptness of deck and crane operations by the ship's crew. This introduced inevitable variation in data collection. At the maximum depth of 4,800 m, the descent would take over two hours, greater than the maximum one-hour recording capacity of the ISIT video system. The recorder was therefore set to start and stop at intervals to ensure sampling between the surface and the sea floor. Sometimes sampling was concentrated at particular depths. When

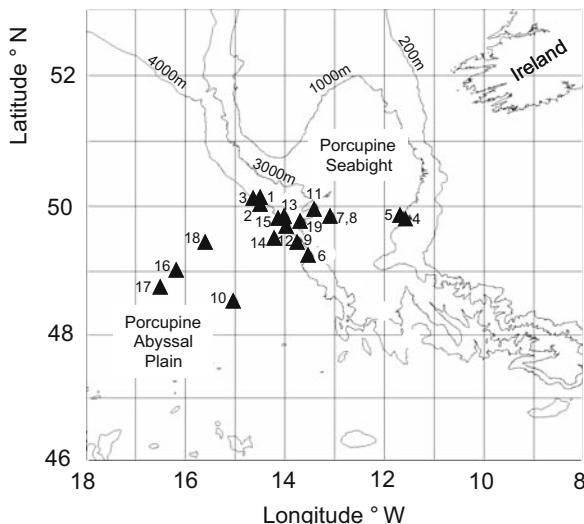


Fig. 17.1 Location of the 19 stations where measurements were taken

the vehicle had reached the sea floor an acoustic command from the ship, triggered release of ballast, and the vehicle ascended because of its positive buoyancy and was recovered back on board the *RRS Discovery*.

As there were no previous data of this kind to inform a formal sampling design, three aims influenced final sample design, which was also constrained by the ongoing ship programme: (i) to produce some replicates as close together as possible in time and space, (ii) to investigate spatial variation in waters of different depths, and (iii) to produce a balanced set of samples across the seasons. It was evident as soon as the first data were viewed that, particularly in summer and autumn, a simple paradigm of an exponential decrease with depth was inappropriate. This has resulted in the need for a sophisticated approach to the statistical analysis describing the profiles and answering questions about spatial homogeneity over the geographical sample area and about seasonal differences.

The aim of this chapter is not only to analyse the data but also to explain how to make multi-panel graphs for grouped data. We have used these graphs in nearly every chapter, but here we will use them in more detail and also make our own panel functions. A detailed explanation of these graphs can be found in Chapter 3 of Pinheiro and Bates (2000). They used specific functions from the `n1me` package to create multi-panel figures. Instead of using the Pinheiro and Bates `plot` function for grouped data, we will use the more flexible `xyplot` function from the lattice package. We show that intelligent use of graphs considerably simplifies the statistical analysis. Perhaps we should phrase this differently. Good multi-panel graphs help us to develop questions in cases when you are not 100% sure in which direction to steer the analysis. Call it a ‘hypothesis generating brainstorming session’. Therefore, we start constructing multi-panel graphs for grouped data (which can also be seen as part of the data exploration), and this will help us decide what type of statistical models to apply and how to apply them.

17.3 Construction of Multi-panel Plots for Grouped Data

Possible explanatory variables are time (of the day), month, year, station, season, latitude, longitude, and depth. Except for the last three variables, all are nominal. Figure 2.11 in Chapter 2 shows the bioluminescence sources per m^2 plotted against depth for each individual station. The graph indicates that the profiles from stations 4, 5 and 10 can be dropped, as the depth range is considerably smaller than for the other profiles. The question is now which profiles are similar and which are different. We discuss various different approaches.

17.3.1 Approach 1

Measurements took place in months 3 (March), 4 (April), 8 (August), and 10 (October). It may be the case that profiles from the same month are similar and

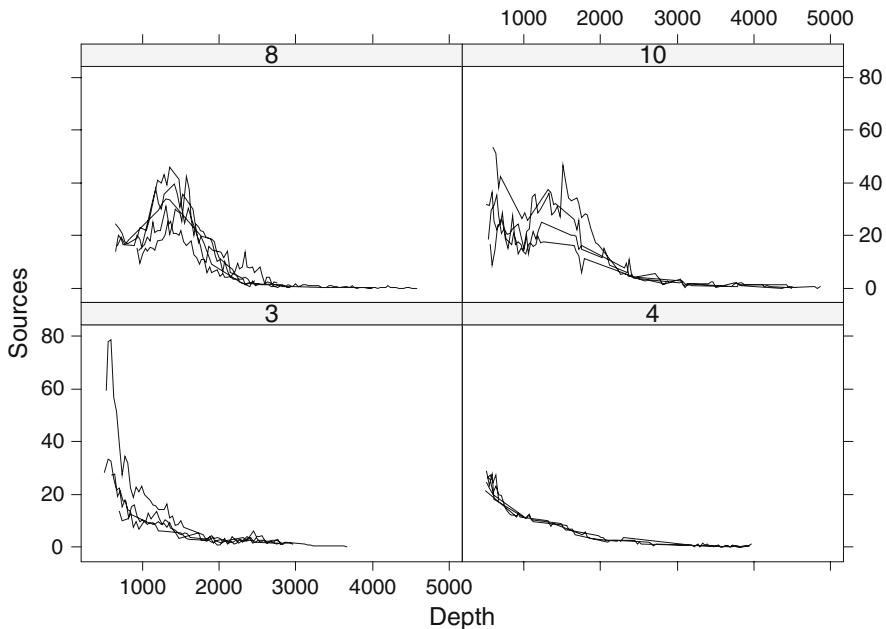


Fig. 17.2 Source–depth profiles per month. Each line represents a station. Note that the April profiles are similar. The graph may be improved by allowing for different ranges along the vertical axes

profiles from different months are dissimilar. Before applying complicated statistical methods to test this, we will draw a multi-panel graph. It has four panels. The first panel contains the profiles from month 3, the second panel from month 4, etc. The following R code accesses the data and draws the multi-panel graph with four windows (Fig. 17.2).

```
> library(AED); data(ISIT)
> ISIT$fMonth <- factor(ISIT$Month)
> ISIT$fStation <- factor(ISIT$Station)
> ISIT$fYear <- factor(ISIT$Year)
> ISIT2 <- ISIT[ISIT$fStation != "4" &
+           ISIT$fStation != "5" &
+           ISIT$fStation != "10" ,]
> library(lattice)
> MyLines <- function(xi, yi, ...){
+   I <- order(xi)
+   panel.lines(xi[I], yi[I], col = 1)}
> xyplot(Sources ~ SampleDepth | fMonth, data = ISIT2,
+         groups = fStation, xlab = "Depth", ylab = "Sources",
+         panel = panel.superpose,
+         panel.groups = MyLines)
```

The code starts by accessing the data. The object `ISIT2` is identical to `ISIT` (original data), except that stations 4, 5, and 10 are removed. It then creates a function called `MyLines`. Its task is to draw a line in the panels, while avoiding spaghetti plots. Finally, the `xyplot` creates a multi-panel figure with four windows. Each panel corresponds to a month. The option `groups` specifies that the data from the same station are grouped. The command `panel = superpose` ensures that lines for all stations (defined by `groups`) from the same month are superimposed in the same panel. Finally, the `panel.groups` option specifies which task should be carried out on the data defined by the `groups` option (station in this case). It calls our own function `MyLines`. The graph shows that the profiles in April are similar, but there are more differences between the profiles in other months. There is also more variation in the sources in the first 2,000 m compared to the deeper depths. This immediately indicates problems with heterogeneity. However, the April profiles do not seem to have this problem. This means that we may need models that allow for heterogeneity along depth in some months or stations, but not in all. Stations 1, 2, and 3 were all completed as close as possible to each other and all within a period of 90 hours, indicating that these can be considered as good replicate samples.

The measurements were taken in four months spread across two years, and the `xyplot` can easily be extended to draw a multi-panel plot with month and year information. We can also tidy up our R code. Panel labels now have a white background, the y-axes are allowed to have different ranges, and the label along the y-axis has m^{-3} , something that may take some time to work out how to do. We also divided depth by 1,000 to minimise the number of zeros in the labels along the horizontal axes. More sophisticated methods exist for this, see, for example, the `labels` option in the `xyplot` help file.

```
> xyplot(Sources ~ SampleDepth / 1000 | fMonth * fYear,
  groups = fStation, data = ISIT2,
  strip = function(bg = 'white', ...),
  strip.default(bg = 'white', ...),
  scales = list(alternating = TRUE,
    x = list(relation = "same"),
    y = list(relation = "free")),
  xlab = "Depth (km)",
  ylab = expression(paste(Sources, " m^{-3} ", "")),
  panel = panel.superpose,
  vpanel.groups = MyLines)
```

The resulting graph in Fig. 17.3 shows that measurements in April and August only took place in 2001 and the March and October sampling only in 2002.

This makes it impossible to test for a month-year interaction, and we only use month as an explanatory variable. Similar problems exist for the explanatory variable time of the day. This reduces the explanatory variables to depth, month, station, latitude, and longitude. There is also a risk with the last three variables as each station was at a unique latitude and longitude (Fig. 17.1), a certain degree of

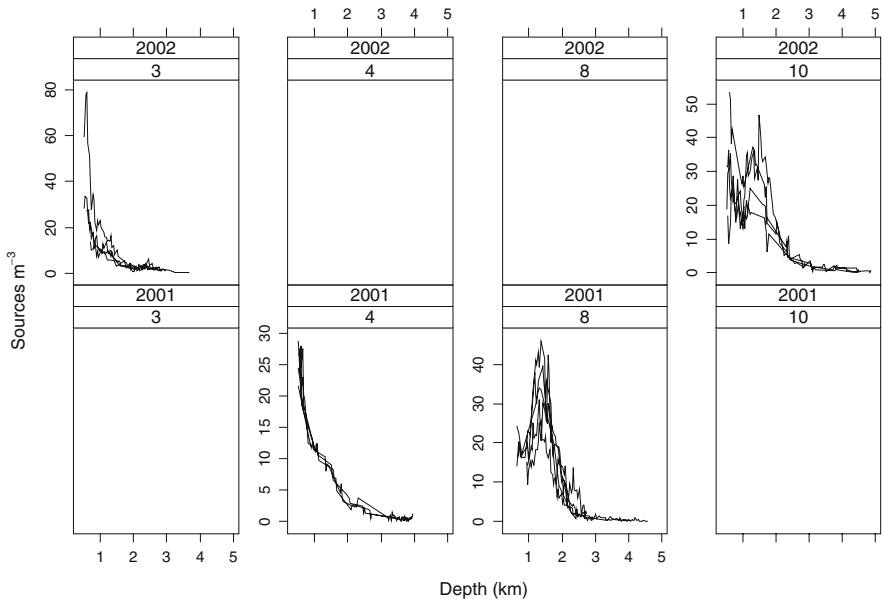


Fig. 17.3 Source–depth profiles by year and month. The *lower four panels* are for 2001 and the *upper four panels* for 2002. From *left to right* are the months

collinearity exists between station against latitude and longitude. This may become an issue if we use models that contain station as a factor, and latitude and longitude as smoothers or continuous explanatory variables.

The results so far indicate that there is a non-linear depth effect. In some months, the profiles are similar and in other months, profiles are not that similar, and there is heterogeneity between groups of profiles and within a station along depth. In the next section, we need a statistical model that describes the sources as a function of depth, station, month, latitude, and longitude. A possible model is of the form

$$S_{is} = \alpha_i + f_i(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma^2 \times |\text{Depth}|^{\delta_i}) \quad (17.1)$$

The sources at station i at depth s , S_{is} , are modelled as an intercept that differs per station, a smoothing function of depth, a month effect, and ε_{is} is the unexplained information. The smoothing function f has an index i indicating that the shape of the smoother can be different per station. This means that the source–depth relationship is allowed to differ per station. From a computing point of view, this is rather ambitious as there are 17 stations. Furthermore, the multiple panel graph in Fig. 2.11 indicates that we may expect heterogeneity along the depth gradient. Recall from Chapter 4 that there are different ways of implementing such an error structure. One option is the `varPower` method given in Equation (17.1). It models the residual spread for profile i in such a way that its variance is proportional to the variance

covariate Depth. It is also possible to attach an index i to the variance σ^2 . This allows for a different spread per station. Other alternatives exist and will be considered below. The only information that has not been used yet is spatial location. The model in Equation (17.1) assumes that residuals from different depths and stations are uncorrelated.

It may not be possible to fit the proposed model as it requires 17 smoothers with different degrees of freedom and a large number of variance components. However, the shape of the smoothers in Figs. 17.2, 17.3, and 17.4 indicate that various profiles are similar, and perhaps, we can replace these by one smoother. This means that Equation (17.1) can be changed into

$$S_{is} = \alpha_i + f_j(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma^2 \times |\text{Depth}|^{\delta_i}) \quad (17.2)$$

The only difference is the index attached to the smoother: a j instead of an i . We are now looking for groups of profiles that can be modelled by a single smoother. Hopefully, j will take only a limited number of values. Something like $j = 3, 4, 8$, and 10 referring to the four months. In this case, profiles of each month are modelled by a single smoother for each month and we end up with a model that has only four smoothers. We could also try a model with only one smoother. However, the shape of the smoothers in Figs. 17.2, 17.3, and 17.4 indicate that things will not be as simple as this. The smoothers from month 4 (station 1, 2, and 3) are very similar and may be summarised by only one smoother, but then it becomes rather difficult to

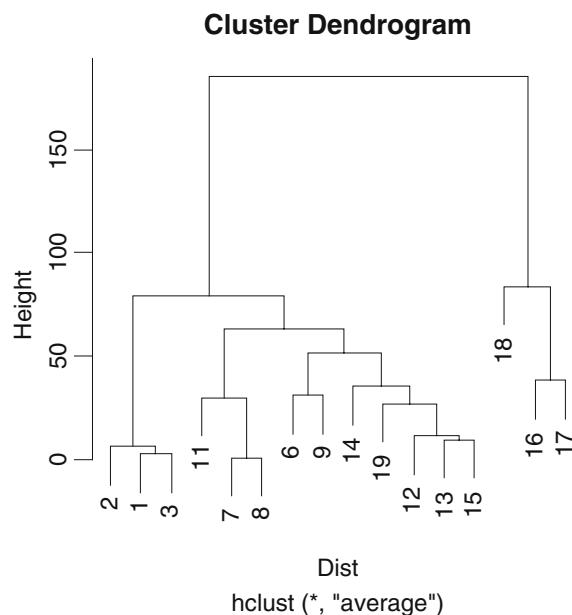


Fig. 17.4 Dendrogram representing the Euclidean distances between the geographical position of the stations. Based on the dendrogram, the following groups of stations were selected: (i) stations 1, 2, and 3, (ii) stations 6 and 9, (iii) stations 7, 8, and 11, (iv) stations 12, 13, 14, 15, and 19, (v) stations 16 and 17, and (vi) station 18

group profiles based on eyeballing. The original motivation for making the grouped multi-panel graphs was to detect groups of profiles.

17.3.2 Approach 2

Another way to group profiles is based on their geographical position. We can look at the map in Fig. 17.1 and group stations that are close to each other. A slightly less subjective approach is to calculate distances between the stations and apply clustering on the (Euclidean) distance matrix. The results can be presented in a dendrogram (Zuur et al., 2007). Judging from the dendrogram which stations should be grouped together is still subjective, but less subjective than looking at the map in Fig. 17.1. The following R code extracts the x - and y -coordinates for each station (it would have been easier to read them from a 16-by-2 ASCII file), calculates the Euclidean distances between the 16 stations, applies clustering with average linkage (Zuur et al., 2007), and presents the results in a dendrogram (Fig. 17.4). Stations that are grouped first (at the bottom) are close to each other.

```
> Xcoord <- vector(length = 16)
> Ycoord <- vector(length = 16)
> UStation <- unique(ISIT2$Station)
> for (i in 1:16) {
  Xcoord[i] <- ISIT2$Xkm[UStation[i]==ISIT2$Station][1]
  Ycoord[i] <- ISIT2$Ykm[UStation[i]==ISIT2$Station][1]
}
#Calculate a distance matrix between the 16 stations
#using Pythagoras
> D <- matrix(nrow = 16, ncol = 16)
> for (i in 1 : 16){
  for (j in 1 : 16){
    D[i,j] <- sqrt((Ycoord[i] - Ycoord[j]) ^ 2 +
      (Xcoord[i] - Xcoord[j]) ^ 2)}}
```

```
> colnames(D) <- unique(ISIT2$Station)
> rownames(D) <- unique(ISIT2$Station)
> MyNames <- unique(ISIT2$Station)
#Apply clustering
> Dist <- as.dist(D)
> hc <- hclust(Dist, "ave")
> plot(hc, labels = MyNames)
```

The dendrogram in Fig. 17.4 suggests using the following groups of stations: (i) stations 1, 2, and 3, (ii) stations 6 and 9, (iii) stations 7, 8, and 11, (iv) stations 12, 13, 14, 15, and 19, (v) stations 16 and 17, and (vi) station 18. It should be noted that this grouping is only based on the geographical position of the stations and

information on sources at these stations is not taken into account. Comparison with Fig. 17.1 shows that stations 1, 2, and 3 are very close to each other, and stations 16, 17, and 18 are the offshore stations. Between these two groups are stations within the mouth of the Porcupine Seabight.

17.3.3 Approach 3

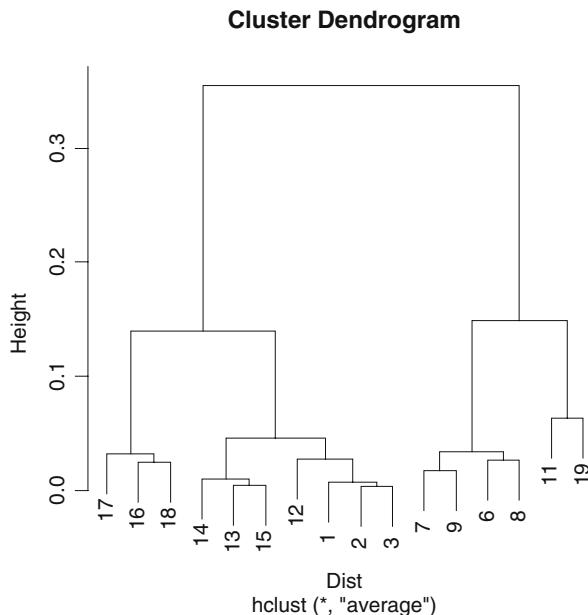
The last approach used here to group profiles is as follows. Two profiles can be labelled as similar if they are highly correlated. The problem is that we cannot calculate a (Pearson) correlation coefficient because the data are not measured at exactly the same depths. For example, the first four measurements of station 1 are at 517, 547, 582, and 614 m. For station 2, these are 501, 865, 989, and 927 m. One way to get source values at the same depth at both stations is to apply additive modelling on each profile and predict source values at predefined depth intervals. This gives us two profiles with (predicted) values at the same depth, allowing us to calculate a correlation coefficient. If we do this for all stations, we can calculate a 16-by-16 correlation matrix. To visualise the patterns in this matrix, non-metric multidimensional scaling or clustering can be used to identify groups of profiles.

The following R code was used:

```
> MyDepth <- seq(from = min(ISIT2$SampleDepth),
+                  to = max(ISIT2$SampleDepth), by = 25)
> NEWSOURCES <- matrix(nrow = 175, ncol = 16)
> NEWSOURCES[] <- NA
> library (mgcv)
> j <- 1
> for (k in MyNames) {
+   Mi <- gam(Sources ~ s(SampleDepth), data = ISIT2,
+             subset = (ISIT2$Station == k))
+   Depthi <- ISIT2$SampleDepth[ISIT2$Station == k]
+   I1 <- MyDepth > min(Depthi) & MyDepth < max(Depthi)
+   mynewXdata <- data.frame(SampleDepth = MyDepth[I1])
+   M.pred <- predict(Mi, newdata = mynewXdata)
+   NEWSOURCES[I1,j] <- M.pred
+   j <- j + 1 }
> D <- cor(NEWSOURCES, use = "pairwise.complete.obs")
> colnames(D) <- unique(ISIT2$Station)
> rownames(D) <- unique(ISIT2$Station)
> Dist <- as.dist(1 - D)
> hc <- hclust(Dist, "ave")
> plot(hc, labels = MyNames)
```

The code starts by calculating the depth gradient along which we will predict source values. The variable `MyDepth` goes from the smallest to the largest

Fig. 17.5 Dendrogram obtained by applying clustering on the correlation matrix Newsources. Average linkage was used. The dendrogram implies the following groups of stations:
 (i) stations 1, 2, 3, and 12,
 (ii) stations 13, 14, and 15,
 (iii) stations 6, 7, 8, and 9,
 (iv) stations 11 and 19, and
 (v) stations 16, 17, and 18



observed depth value in the study with steps of 25 m. This vector is of length 175. A matrix NEWSOURCES is created. It will contain the predicted source values along the variable MyDepth at the 16 stations. We then start a loop, and in each iteration, data from one station are analysed using additive modelling. A new data frame is created with depth values between the lowest and highest measured depths (with steps of 25 m). Source values along these depth ranges are predicted and stored at the appropriate place in the matrix NEWSOURCES. Once this process is carried out for each station, this matrix contains predicted source values at the same depths. The only remaining problem is that NEWSOURCES has many missing values as some profiles were measured at deeper depths at some stations, or the other way around, at the less deep stations. The option `use = "pairwise.complete.obs"` ensures that the correlation matrix between the 16 profiles does not contain missing values (unless the depth ranges between the two stations were completely different as originally happened when station 10 was included). The rest of the code is identical as above and produces the dendrogram in Fig. 17.5. Using a degree of subjectivity, we can distinguish the following groups:
 (i) stations 1, 2, 3, and 12, (ii) stations 13, 14, and 15, (iii) stations 6, 7, 8, and 9, (iv) stations 11 and 19, and (v) stations 16, 17, and 18. Inspection shows that groups (i) and (ii) are all spring (March and April) samples. Group (v) comprises the three stations over the Porcupine Abyssal plain from October 2002. Groups (iii) and (iv) are all autumn samples from August and October within the Porcupine Seabight.

In the next section, we apply GAM models with one smoother per group.

17.4 Estimating Common Patterns Using Additive Mixed Modelling

In the previous section, several potential models were discussed. It is clear that the source–depth relationship is non-linear and we should take into account heterogeneity between and within the stations. There is also the possibility of violation of independence. This may come as a surprise, but recall in Chapters 6 and 7 we checked for temporal correlation in the data. We don't have repeated measurements in time, but we do have them along depth! The depth gradient can be seen as a spatial gradient, and this means that we may need to add a spatial (depth) correlation structure to the model.

In the previous section, we mentioned that numerical problems may be expected if 16 smoothing curves are used (one for each station). An initial analysis confirmed this problem. We therefore need to reduce the number of smoothing curves and we consider the following options.

- Use one smoothing curve for all stations.
- Use one smoothing curve for each month (four smoothers in total).
- Use one smoothing curve for each group derived from Fig. 17.4 (six smoothers in total).
- Use one smoothing group for each group derived from Fig. 17.5 (five smoothers in total).

We will set the scene with the first option and then discuss how to proceed with the other models and then judge which approach is the best.

17.4.1 One Smoothing Curve for All Stations

Instead of applying the additive mixed model in Equation (17.2), we start with a simpler model to show why we need a more complex one. Obviously, you can argue that the data exploration already indicated that we need to allow for heterogeneity, but it is always worth while formally showing why a more complex approach is required.

$$S_{is} = \alpha_i + f(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma^2) \quad (17.3)$$

The model has one smoothing curve for all stations, a month effect (nominal variable), and a station effect (nominal variable, represented by α_i). The residuals are assumed to be independently, normally distributed with the same variance. The estimated smoothing curve is presented in Fig. 17.6A, and the residuals against fitted values in Fig. 17.6B. The latter graph confirms our suspicions; there is heterogeneity. So in spite of all the terms in the model being significant, we can bin it.

The following R code was used.

```
> library(mgcv); library(nlme)
> M1 <- gam(Sources ~ fStation + s(SampleDepth) +
+ fMonth, data = ISIT2)
> E <- resid(M1)
> F <- fitted(M1)
```

```
> op <- par(mfrow = c(2, 1), mar = c(5, 4, 1, 1))
> plot(M1)
> plot(F, E, xlab = "Fitted values", ylab = "Residuals")
> par(op)
```

The `mar` option in `par` modifies the white space around the graphs. The other R code has been discussed elsewhere.

To work towards a model that can cope with heterogeneity, we consider the following series of models that increase in complexity.

$$S_{is} = \alpha + a_i + f(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma^2) \quad (17.4A)$$

$$S_{is} = \alpha + a_i + f(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma_i^2) \quad (17.4B)$$

$$S_{is} = \alpha + a_i + f(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma^2 |\text{Depth}_s|^\delta) \quad (17.4C)$$

$$S_{is} = \alpha + a_i + f(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma_i^2 |\text{Depth}_s|^\delta) \quad (17.4D)$$

$$S_{is} = \alpha + a_i + f(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma_i^2 |\text{Depth}_s|^{\delta_i}) \quad (17.4E)$$

Instead of using a fixed intercept, we decided to use a random intercept. This is the equivalent of a random intercept mixed model (Chapter 4). So, in all mod-

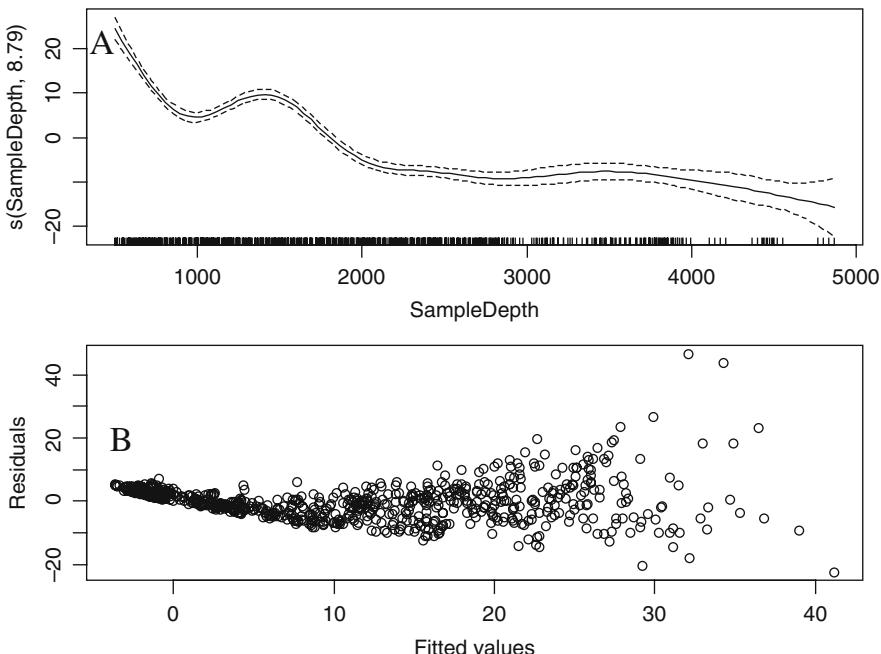


Fig. 17.6 **A:** Estimated smoothing curve for the additive model in Equation (17.3). **B:** Residuals versus fitted values showing heterogeneity. Cross-validation was used to estimate the degrees of freedom

els we assume that a_i is normally distributed with mean 0 and variance σ_a^2 . The advantage of this approach is that instead of estimating 16 intercepts, we now only need to estimate one (α) and a variance term σ_a^2 . The variance component a_i allows for random variation around the intercept. The model in Equation (17.4A) assumes homogeneity, and it was only added to provide a reference point. The model in Equation (17.4B) assumes heterogeneity per station but homogeneity within a station along depth, that in Equation (17.4C) assumes homogeneity between stations but heterogeneity within a station along depth (but the strength of the heterogeneity along the depth gradient is the same for each station), that in Equation (17.4D) allows for heterogeneity between stations and within stations along depth (same strength), and finally, the model in Equation (17.4E) implies heterogeneity between stations and heterogeneity within stations along depth. The crucial point in Equation (17.4E) is that the heterogeneity within stations along depth is allowed to differ between the stations. Hence, it is the most complete (and complicated) model in this set of models. It should be noted that the only difference between these five models are the random components. In the models, we apply later in this chapter, we use the same five random components. We refer to them as models A to E. The only difference between the models in Equations (17.4A–E) and the ones used later is the fixed effects structure (smoothers).

The following code applies models in Equations (17.4A–E) in R and compares them using the AIC and BIC criteria.¹ It was observed that the numerical algorithms performed better when we rescaled the depth so that values were between 0.5 and 5 (km) instead 500–5,000 m:

```
> lmc <- lmeControl(niterEM = 5000, msMaxIter = 1000)
> M17.4A <- gamm(f1, random = list(fStation =~ 1),
+   method = "REML", control = lmc, data = ISIT2)
> M17.4B <- gamm(f1, random = list(fStation =~ 1),
+   method = "REML", control = lmc, data = ISIT2,
+   weights = varIdent(form =~ 1 | fStation))
> M17.4C <- gamm(f1, random = list(fStation =~ 1),
+   method = "REML", control = lmc, data = ISIT2,
+   weights = varPower(form =~ Depth1000))
> M17.4D <- gamm(f1, random = list(fStation =~ 1),
+   method = "REML", control = lmc, data = ISIT2,
+   weights = varComb(varIdent(form =~ 1 | fStation),
+     varPower(form =~ Depth1000)))
> M17.E <- gamm(f1, random=list(fStation =~ 1),
+   method = "REML",control = lmc, data = ISIT2,
+   weights = varComb(varIdent(form =~ 1 | fStation),
+     varPower(form =~ Depth1000 | fStation)))
```

¹We used R version 2.6 and mgcv version 1.3–27. More recent versions of R and mgcv require a small modification to the code; see the book website (www.highstat.com) for updated code.

```
> AIC(M17.4A$lme, M17.4B$lme, M17.4C$lme, M17.4D$lme,
      M17.4E$lme)
      df      AIC
M17.4A$lme  8  4734.141
M17.4B$lme 23  4269.503
M17.4C$lme  9  4258.752
M17.4D$lme 24  3859.231
M17.4E$lme 39  3675.986
```

The only difference between the calls to the `gamm` function for these five models is the `weights` option. See Chapter 4 for a more detailed discussion. The names of the R objects correspond to the equation numbers on the previous page. The output of the `AIC` command shows that the model with heterogeneity between stations and within stations is the best model (from these five!). This is model E. The estimated smoothing curve and (normalized) residuals versus fitted values are given in Fig. 17.7. Note that all the hard work earlier did help to solve heterogeneity problems! The R code that was used to create Fig. 17.7 is similar to that for Fig. 17.6 and is not given here.

There is one thing we have ignored so far and that is spatial dependence. There are two ways we can violate the independence assumption: correlation between stations and/or correlation within (groups of) stations along the depth gradient. The first form of dependence is difficult to model within the random component structure, and it is easier to use covariates for this. We could for example use more smoothers or other explanatory variables. The second form of dependence can be checked by making a variogram (Fig. 17.8) with the following two lines of R code:

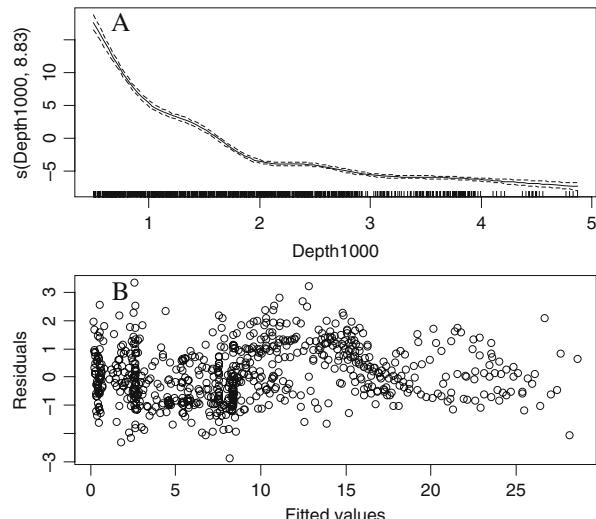
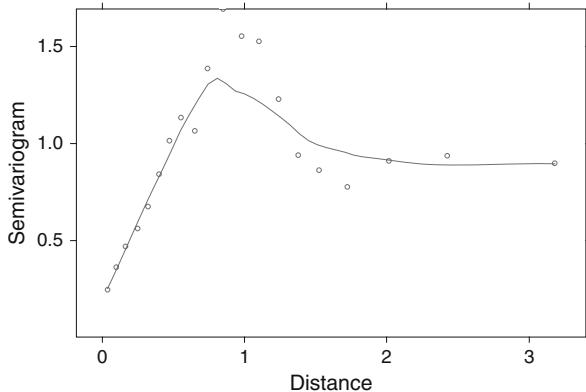


Fig. 17.7 Estimated smoothing curve (A) and normalized residuals versus fitted values (B) for the model in Equation (17.4E). Compare panel B with that of Fig. 17.6 and note how all the fancy random structures have solved the heterogeneity problem

Fig. 17.8 Variogram of the normalized residuals obtained by model 4E. The spatial correlation structure is estimated within the profiles



```
> Vario17.4E <- Variogram(M17.4E$lme, robust = TRUE,
    data = ISIT2 form =~ Depth1000 | fStation)
> plot(Vario17.4E)
```

Independence of residuals expresses itself in the variogram as a horizontal band of points. In this case, the variogram shows a sharp increase during the first 1,000 m (1 km) and a small decrease thereafter.

Thus the model implies that residuals that are within a range of 1,000 m are correlated. We specifically wrote ‘the model implies’ as the most likely explanation is that the dependence is caused by an improper fixed effects structure (meaning: not enough smoothers or missing covariates).

One option is to include a correlation structure along depth within the additive modelling structure, but a better approach (to start with) is to extend the model with more smoothers (or covariates) and see whether that solves the problem.

This is done next. If it turns out that adding more smoothers or covariates does not solve the problem, then we should consider adding a correlation on the residuals within the additive mixed model. But that is a last resort.

17.4.2 Four Smoothers; One for Each Month

To solve the independence problem discussed in the previous paragraph, we extend the fixed effects part of the model by using one smoother for all stations of the same month. Just as before, we have to take into account possible violation of heterogeneity, and therefore, we consider models with similar random error structures as before. Little is lost (as can be judged by plotting residuals versus fitted values) by using different variances per month instead of station; it saves considerable computing time!

$$S_{is} = \alpha + a_i + f_j(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma^2) \quad (17.5A)$$

$$S_{is} = \alpha + a_i + f_j(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma_j^2) \quad (17.5B)$$

$$S_{is} = \alpha + a_i + f_j(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma^2 \times |\text{Depth}_s|^\delta) \quad (17.5C)$$

$$S_{is} = \alpha + a_i + f_j(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma_j^2 \times |\text{Depth}_s|^\delta) \quad (17.5D)$$

$$S_{is} = \alpha + a_i + f_j(\text{Depth}_s) + \text{Month}_i + \varepsilon_{is} \quad \varepsilon_{is} \sim N(0, \sigma_j^2 \times |\text{Depth}_s|^{\delta_j}) \quad (17.5E)$$

The differences between the models in Equations (17.4A–E) and (17.5A–E) is the index j attached to the smoothing function f and the multiple variances per month instead of station. The index j take the values $j = 3, 4, 8$, and 10 referring to the four months. Hence, each month is allowed to have a different depth-source profile. The following R code implements the model in Equations (17.5A), (17.5B), and (17.5E).

```
> f1 <- formula(Sources ~
+   s(Depth1000, by = as.numeric(Month == 3)) +
+   s(Depth1000, by = as.numeric(Month == 4)) +
+   s(Depth1000, by = as.numeric(Month == 8)) +
+   s(Depth1000, by = as.numeric(Month == 10)) +
+   fMonth)
> M17.5A <- gamm(f1, random = list(fStation =~ 1),
+   method = "REML", control = lmc, data = ISIT2)
> M17.5B <- gamm(f1, random = list(fStation =~ 1),
+   method = "REML", control = lmc, data = ISIT2,
+   weights = varIdent(form =~ 1 | fMonth))
> #....
> M17.4E <- gamm(f1, random = list(fStation =~ 1),
+   data = ISIT2, method = "REML", control = lmc,
+   weights = varComb(varIdent(form =~ 1 | fStation),
+   varPower(form =~ Depth1000 | fStation)))
```

The other models can be implemented in the same way as before, and to save space, the R code is not shown here (it can also be found on the book website). Just as before, the AIC indicated that model E is the best. The estimated smoothing curves per month are given in Fig. 17.9. Note that we can see a clear distinction between the shapes in different months.

As part of the model validation, we also need to plot residuals versus fitted values to assess homogeneity (not shown here) and residuals versus each explanatory variable to asses independence. The plot of the (normalized) residuals versus depth (Fig. 17.10) shows that there is a problem as there are clear residual patterns. To aid visual interpretation, we added a LOESS smoother. It may also be useful to fit an

Fig. 17.9 Estimated smoothing curves and 95% point-wise confidence bands per month for model 5E. The four panels correspond to the four months. Month 3 represents stations 12–15, month 4 cruises 1–4, month 8 stations 6, 7, 8 and 11, and month 10 represents stations 16–19. The R code for this figure is presented on the book website

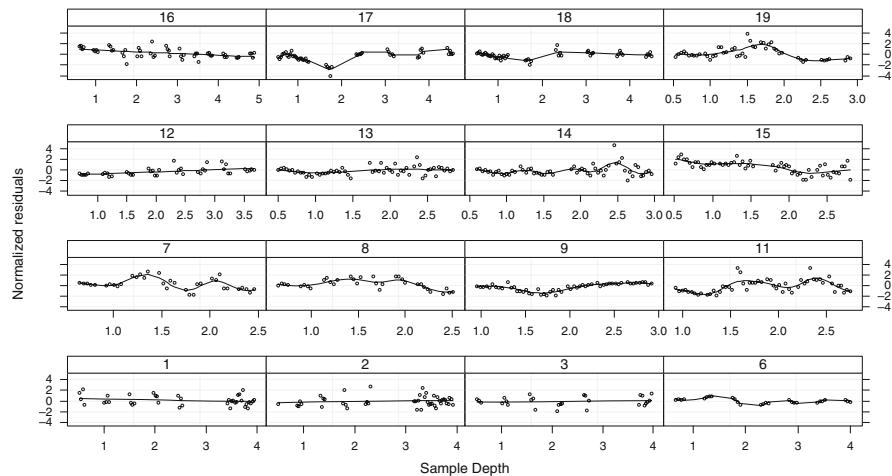
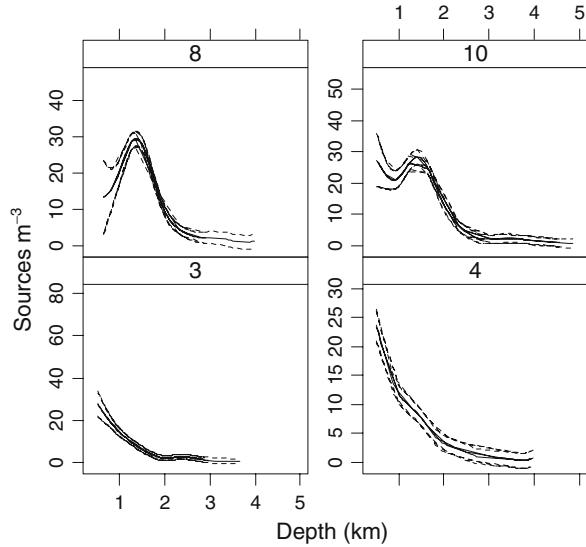


Fig. 17.10 Normalised residuals plotted versus depth for model (17.5E). Note that for some stations there is a clear residual pattern. To aid visual interpretation, LOESS curves were added. The R code for this figure is presented on the book website

additive model in each panel in Fig. 17.10 and inspect the significance levels of the smoothers. As we are fitting a smoother on residuals (as a function of depth), we should not see significant smoothers! But for stations 15 (month 3), 7, 9 and 11 (all from month 8), and 17 and 18 (month 10) we could still find a strong and significant relationship between residuals and depth.

The dependence problem is also detected if we make a variogram of the normalised residuals. It has a similar shape as in Fig. 17.8.

One option to solve this problem is to include a spatial correlation structure within the additive model, which is achieved by adding the correlation option to the gamm function:

```
correlation = corSper(form =~ Depth1000 | fStation,
                      nugget = TRUE, fixed = FALSE).
```

But just as before, the residual pattern indicates that the grouping structure by months is not optimal for all stations. So, instead of adding a complicated spatial correlation structure, we should first aim to improve the fixed effects structure. This means that we have to use more covariates or a different grouping of stations.

So, to summarise this part, the fixed effect part of the model was extended from one smoother to four (one per month). Stations 12, 13, 14, and 15 are from month 3; stations 1, 2, and 3 from month 4; stations 6, 7, 8, 9, and 11 from month 8; and stations 16–19 from month 10. But the model validation showed that especially within month 8, stations are not similar. But for month 4 (stations 1–3) and month 3 (especially stations 12–14) profiles are similar! To gain further insight, we continue with a grouping structure by geographical distances.

17.4.3 Smoothing Curves for Groups Based on Geographical Distances

In Section 17.2, we discussed how to divide the 16 stations in 5 groups based on geographical distances. Our proposed grouping of stations was (i) stations 1, 2, and 3; (ii) stations 6 and 9; (iii) stations 7, 8, and 11; (iv) stations 12, 13, 14, 15, and 19; (v) stations 16 and 17; and (vi) station 18. The R code below runs the same 5 models as in Equations (17.4) and (17.5), except that the fixed structure is adjusted to take into account our new groups.

```
> G1 <- ISIT2$Station == 1 | ISIT2$Station == 2 |
   ISIT2$Station == 3
> G2 <- ISIT2$Station == 6 | ISIT2$Station == 9
> G3 <- ISIT2$Station == 7 | ISIT2$Station == 8 |
   ISIT2$Station == 11
> G4 <- ISIT2$Station == 12 | ISIT2$Station == 13 |
   ISIT2$Station == 14 | ISIT2$Station == 15 |
   ISIT2$Station == 19
> G5 <- ISIT2$Station == 16 | ISIT2$Station == 17
> G6 <- ISIT2$Station == 18
> f1 <- formula(Sources~
   s(Depth1000, by = as.numeric(G1)) +
   s(Depth1000, by = as.numeric(G2)) +
```

```

s(Depth1000, by = as.numeric(G3)) +
s(Depth1000, by = as.numeric(G4)) +
s(Depth1000, by = as.numeric(G5)) +
s(Depth1000, by = as.numeric(G5)) + fMonth)
> M.GeoA <- gamm(f1, random = list(fStation =~ 1),
method = "REML", control = lmc, data = ISIT2)
> M.GeoB <- gamm(f1, random = list(fStation =~ 1),
method = "REML", control = lmc, data = ISIT2,
weights = varIdent(form =~ 1 | fMonth))
> # ...
> M.GeoE<-gamm(f1, random=list(fStation =~ 1),
data = ISIT2, method = "REML", control = lmc,
weights = varComb(varIdent(form =~ 1 | fMonth),
varPower(form =~ Depth1000 | fMonth)))

```

The other models can be run with similar code. The model with heterogeneity between groups and heterogeneity along depth (with differences per group) is the best, as judged by the AIC. This is model E. Just as in the previous analysis, we made a variogram of the (normalised) residuals, and we also plotted (normalised) residuals versus depth. These graphs are not shown here, but both indicated violation of independence. Hence, grouping stations based on geographical distances does not give groups in which the profiles have similar depth profiles. We also tried small modifications of the grouping structure, but this did not solve the independence problem.

17.4.4 Smoothing Curves for Groups Based on Source Correlations

In Section 17.3, we also discussed how to calculate correlations between predicted source profiles and used these to determine a grouping structure. Recall that we determined the following five groups: (i) stations 1, 2, 3, and 12; (ii) stations 6, 7, 8, and 9; (iii) stations 11 and 19; (iv) stations 13, 14, and 15; and (v) stations 16, 17, and 18. Adjusting the R code in order to implement this grouping of stations is relatively simple. All we need is the following piece of code:

```

> G1 <- ISIT2$Station == 1 | ISIT2$Station == 2 |
ISIT2$Station == 3 | ISIT2$Station == 12
> G2 <- ISIT2$Station == 6 | ISIT2$Station == 7 |
ISIT2$Station == 8 | ISIT2$Station == 9
> G3 <- ISIT2$Station == 11 | ISIT2$Station == 19
> G4 <- ISIT2$Station == 13 | ISIT2$Station == 14 |
ISIT2$Station == 15
> G5 <- ISIT2$Station == 16 | ISIT2$Station == 17 |
ISIT2$Station == 18

```

```
> f1 <- formula(Sources ~
+   s(Depth1000, by = as.numeric(G1)) +
+   s(Depth1000, by = as.numeric(G2)) +
+   s(Depth1000, by = as.numeric(G3)) +
+   s(Depth1000, by = as.numeric(G4)) +
+   s(Depth1000, by = as.numeric(G5)) + fMonth)
> M.cor4A <- gamm(f1, random = list(fStation =~ 1),
+   method = "REML", control = lmc, data = ISIT2)
> # etc...
```

Other models can be fitted by using the same code as above. Again, the AIC indicated that model E is the best. We plotted the residuals versus depth and a large number of stations contained a significant (as determined by a smoother) residual–depth pattern, especially stations 12, 7–9 (entire group 2), 11, 15, 16, and 17. This means that the chosen grouping is not a good one.

17.5 Choosing the Best Model

In the previous section, we grouped stations by month, geographical distances and based on correlations between predicted source values. None of the approaches produced a grouping of stations in which residual patterns did not show any violation of independence. Well, this is not entirely true. All analysis showed that the source–depth relationship for stations 1, 2, and 3 are similar, and the same holds for stations 12, 13, and 14. So, at least we can identify these two groups. The other profiles, however, cannot be grouped so easily. In a final attempt, we decided to fit an additive mixed model with two groups of stations (1, 2, 3 and 12, 13, 14), and we used one smoother for each group. Hence, the fixed effects structure assumes that (i) stations 1, 2, and 3 have the same source–depth relationship, (ii) stations 12, 13, and 14 have the same source–depth relationship, and (iii) all other stations have different source–depth relationships. Furthermore, we used one smoother for each of the other stations. The same five random error structures described in Equations (17.4) and (17.5) were used. Some of these models, especially E, are highly complicated, and may potentially not converge. To reduce computing time, we set the degrees of freedom for each smoother to 4, and to our surprise all five models converged.

In terms of the random structure, the model that contains all the options (heterogeneity between groups, along depth but not for all groups), model E, was the best as judged by the AIC. However, the BIC indicated model C (heterogeneity along depth).

The good news is that the variogram of the normalised residuals of model C did not show a clear violation of independence. Figure 17.11 shows the estimated smoothing curve for each station. Note that the smoothers for stations 1, 2, and 3 are identical, and the same holds for those of stations 12, 13, and 14. Based on this graph, the analysis can be taken further by grouping stations 7, 8, and 19 to see whether that improves the model. The way to proceed is to (i) adopt

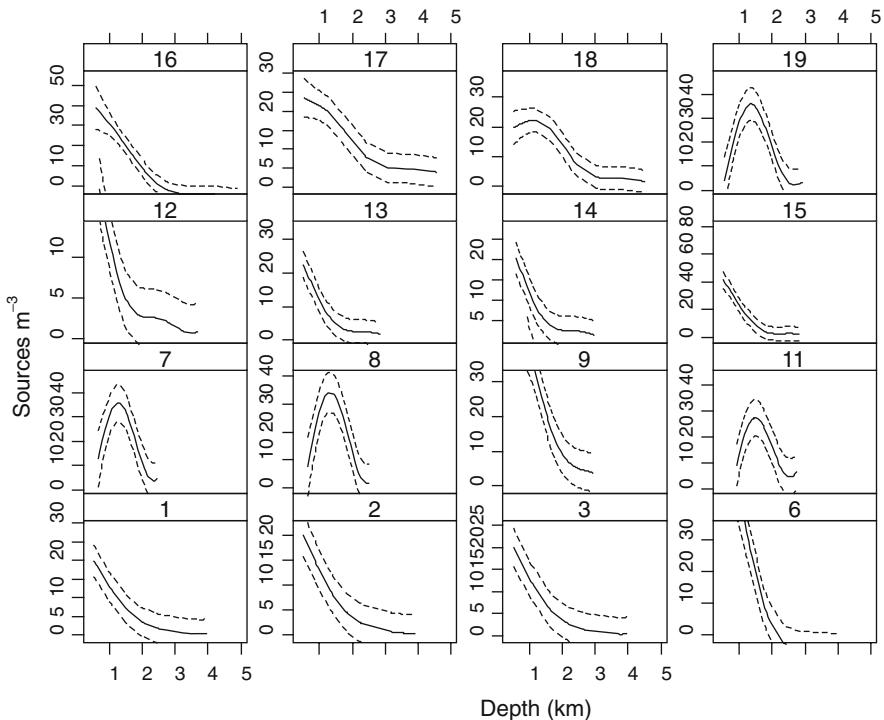


Fig. 17.11 Estimated smoothing curves for each station obtained by the model with 12 groups

the variance structure of model C, (ii) switch to maximum likelihood estimation (method = "ML"), and (iii) compare models in terms of the smoothers. We leave this as an exercise to the reader, but initial analyses indicated that there is not much to be gained by further grouping the stations.

17.6 Discussion

The data were originally published using (i) a logarithmic (\log_{10}) transformation on the sources, (ii) a random effect for station, and (iii) one smoother; see Gillibrand et al. (2007). The transformation solved a lot of trouble; the AIC still identified as optimal the most complicated model E, whilst the BIC indicated model C (only heterogeneity along depth). Even a visual inspection of the residuals of model A did not show any clear heterogeneity! Hence, a logarithmic transformation makes life much easier! However, there is still violation of independence; so we need to add more smoothers or covariates.

However, this is not a trivial exercise, as we showed in this chapter. The question is then: To transform or not to transform? Our opinion is that working with the original (untransformed) data gives more information on what is going on. We

did not even discuss the numerical output for the optimal models; the heterogeneity parameters give a wealth of information as well. In our view, only when numerical instability of the estimation routines becomes an issue, transforming and/or standar-dising an option.

In Chapter 4, we mentioned that the model selection should follow a top-down approach by starting with a fixed effects structure that contains all explanatory variables and possible interaction terms. We did not follow that approach here, simply because the full model had numerical problems. So, we started by grouping stations and trying to find the optimal grouping structure. But the price we paid for this is that from the beginning, we were facing violation of independence and only when we used a close-to-optimal model, the problem became less serious.

So, what does this exercise tell us?

Firstly the close similarity and clustering of stations 1, 2, and 3 indicates that reproducibility of results is good, and there is probably no need to expend sampling effort in replicate profiles. The ISIT system has since been adapted to fit onto a standard oceanographic CTD (Conductivity, temperature, and depth) profiler (Heger et al., 2008).

Secondly, it is evident that there is a seasonal change in profiles between spring and autumn with a post-summer peak in abundance of bioluminescent sources at about 1,200 m. Simple mathematical curves such as the exponential relationship proposed by Bradner et al. (1987) are clearly inappropriate. The estimation of smoothing curves (Fig. 17.10) is very useful for the biologist since it provides an objective means of combining sets of data and producing estimates of the depth of the peak and mean number of sources m^{-3} at different depths.

Thirdly, contrary to what was stated by Gillibrand et al. (2007), there is a difference between stations in the Porcupine Seabight compared with those offshore over the Porcupine Abyssal Plain. Examining the panels in Fig. 2.11, it seems the autumn peak below 1,000 m is less strong in the offshore stations (16, 17 and 18) than closer inshore (19, 6, 7, 8, 9).

The reasons for the deep bioluminescent layer is unclear, but is probably related to two effects. The peak almost certainly represents a seasonal increase in deep biomass fed by organic matter flux from the spring bloom in surface waters. This effect is probably accentuated by accumulation in a layer of North Atlantic intermediate water at this depth, which is derived from Mediterranean water moving northwards from Gibraltar. This effect may be stronger further inshore, where there is a northward moving shelf edge current.

17.7 What to Write in a Paper

Within the field of bioluminescent research, Gillibrand et al. (2007) and this case study are two of the first texts where advanced statistical methods have been used. If you are submitting a paper in a subject area where additive mixed mod-elling techniques are uncommon, you will face the daunting task of convincing

an entire group of scientists of the need for complicated statistical methods. The best starting point is Fig. 2.11 as it clearly shows that linear regression methods (or ANCOVA) are unsuitable. It may be an option to discuss the heterogeneity problem by showing only Fig. 17.6B. At that point, you will need to discuss why you did not apply a logarithmic transformation. Predicting values on the original scale may be a valid argument and so is the fact that a transformation changes relationships between sources and depth. In order to make the referee (and reader) of your paper happy, a non-technical explanation of additive mixed modelling and especially the variance structures is required. If you fail to do this, they will come back with the question: Why do you need all this complicated modelling?

In approach 3 (Subsection 17.3.3) we used the data to calculate Pearson correlation coefficients and applied clustering on them. We then used the same data in the GAMMs (using the results from the clustering). This approach is likely to receive (valid) criticism!

Acknowledgments We thank Dr Emma Gillibrand for permission to use data from her PhD thesis.

Chapter 18

Additive Mixed Modelling Applied on Phytoplankton Time Series Data

A.F. Zuur, M.J. Latuhihin, E.N. Ieno, J.G. Baretta-Bekker, G.M. Smith,
and N.J. Walker

18.1 Introduction

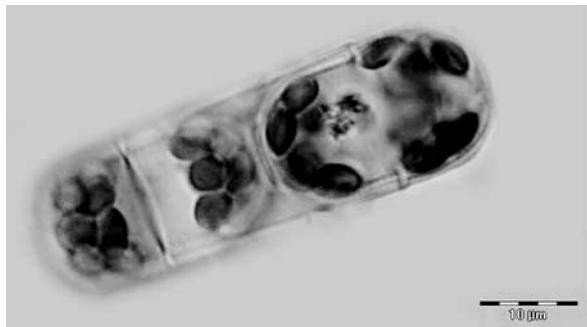
This chapter looks at a data set where our first reaction was: ‘How in heavens name are we going to analyse these data?’ The data consist of a large number of phytoplankton species measured at 31 stations in Dutch estuarine and marine waters. Measurements took place 0–4 times per month from 1990 until present (2005). Environmental data (e.g. temperature, salinity, etc.) were also measured, albeit sometimes at different sampling times! The statistical analysis of these data is complicated for several reasons:

1. Environmental variables and phytoplankton variables were not always measured at the same time.
2. There may be temporal correlation, there may be spatial correlation, and both correlation structures may be complicated.
3. The data contain a large number of species.
4. The data are irregularly spaced.
5. There may be heterogeneity over time (e.g. more variation in summer than in winter).
6. Trends over time and in space may be non-linear.
7. The phytoplankton data were counted by different laboratories.

This chapter is a spin-off from a technical report produced by the first two authors of this book for Rijkswaterstaat – Centre for Water Management, a Dutch governmental department. In that report, univariate methods were applied on aggregated phytoplankton series. The motivation to use aggregated data was to reduce the large number of zeros in the original data. An alternative statistical analysis is to apply multivariate methods like the Mantel test, BIOENV and ANOSIM; see Clarke and Warwick (1994), Legendre and Legendre (1998), and Zuur et al. (2007) for details.

A.F. Zuur (✉)
Highland Statistics Ltd., Newburgh, AB41 6FN, United Kingdom

Fig. 18.1 *Melosira nummuloides* under the microscope; one of the small diatom species of the DIAT1-group (Photo C. Brochard – Koeman en Bijkerk bv)



The problem with these multivariate methods is that the permutation methods used to assess statistical significance ignore the temporal and spatial correlation structures in the data. Here, we follow the technical report approach and focus on a group of aggregated phytoplankton species. To save space, we only use one group: small diatoms (between 0 and 1,000 μm^3). These will be denoted by DIAT1 (Fig. 18.1). Other groups are not considered in this chapter.

As from Section 18.3, we describe an analysis that, in theory, can cope with some of the problems. It should be noted, however, that different analysis strategies are possible, but may give different results and that our chosen approach can be improved on and should be considered as a first attempt. However, given the complexity of the data, any statistical method will have serious difficulties with these data.

18.1.1 Biological Background of the Project

Marine biodiversity is under significant anthropogenic pressures such as physical engineering, physical and chemical pollution, eutrophication (enrichment with nutrients), and the introduction of invasive species. Eutrophication due to anthropogenic nutrient loading has greatly impacted ecological processes in marine waters, and therefore, a lot of effort has been put into reducing nitrogen and phosphorus discharges. To detect the effectiveness of such policy actions in the Netherlands, the Dutch national monitoring programme aims to provide the required information. In addition to a physical and chemical monitoring programme that had been running for several decades, Rijkswaterstaat began a biological monitoring programme for surface waters in the early 1990s. The primary goal of this programme is to provide biological information, especially in relation to long-term changes. The marine biological monitoring programme has been designed to assess ecosystem functioning and food-web relationships determine the structure of this system. Phytoplankton, the free-living, drifting, and mainly photosynthetic organisms in aquatic systems, is the major producer and forms the basis of the marine food web. Higher organisms such as benthic fauna, fish, and sea birds are all indirectly

dependent on phytoplankton. Hence, information about the status of the phytoplankton community is essential to assess ecosystem functioning. In general, the growth of phytoplankton is regulated by underwater light and nutrient availability. Species composition and abundance of phytoplankton vary from season to season. The growth season usually starts with a bloom of diatoms in (early) spring followed by the blooming of *Phaeocystis* sp and in summer, blooms of (dino-)flagellates. Moreover, distinct differences can be detected between the various water bodies (in this chapter we call them areas), both in terms of species composition and abundance. The nutrient regime of the Dutch estuaries, the Delta in the south and the Wadden Sea and Ems estuary in the north, is mainly influenced by freshwater discharges with strongly elevated levels of nitrogen (N) and phosphorus (P) originating from farmland. For the North Sea ecosystem on the contrary, the much more oligotrophic Atlantic Ocean is the main source of nutrients. It seems reasonable to assume there is still some influence of riverine water in the coastal zone, but this rapidly decreases when going to the open sea. In general, increasing salinity goes hand in hand with decreasing nutrient concentrations. Nutrient enrichment usually results in an increase of phytoplankton biomass and often coincides with shifts in phytoplankton species composition. This latter phenomenon is due to different characteristics between individual algal species which have different storage capacities, nutrient uptake kinetics, etc. For example, silicon is an essential nutrient for diatoms, which is a major group of algae. But concentrations of this element seem unaffected by human activities. This implies that, due to eutrophication with N and P, it is likely that the species composition will change in the direction of increased abundance of algal species not dependent on silicon for their growth.

The mechanisms and implications of eutrophication for freshwater systems are reasonably well understood, but this is not the case for marine ecosystems, and the response of marine ecosystems to eutrophication is less predictable. It is suggested (Cloern, 2001) that the interaction between all the parameters characterizing a marine ecosystem – e.g. tidal regime, turbidity, depth, and biomass of benthic suspension feeders – play an important role. More precisely, the complex interaction of all physical and biological attributes operating together seems to act as a filter to modulate the response of an ecosystem to nutrient enrichment. As a result, some estuarine-coastal ecosystems appear to be highly sensitive to change in nutrient inputs, while others appear to be more resistant.

The main underlying question in this study is whether there are trends visible in the phytoplankton community, and if any, what trend, and whether there is a relationship with environmental variables. The rest of this chapter now follows the structure of the original technical report. Since April 1990, the species composition of phytoplankton has been monitored at 31 stations, which have been aggregated into ten different areas. Figure 18.2 presents the locations of the stations and defines the areas.

Water samples were collected from each sampling station and preserved with Lugol's solution, while at a limited number of stations a duplicate series was also counted live to improve identification. Samples were counted using an inverted microscope, and densities were subsequently calculated as number per liter. The

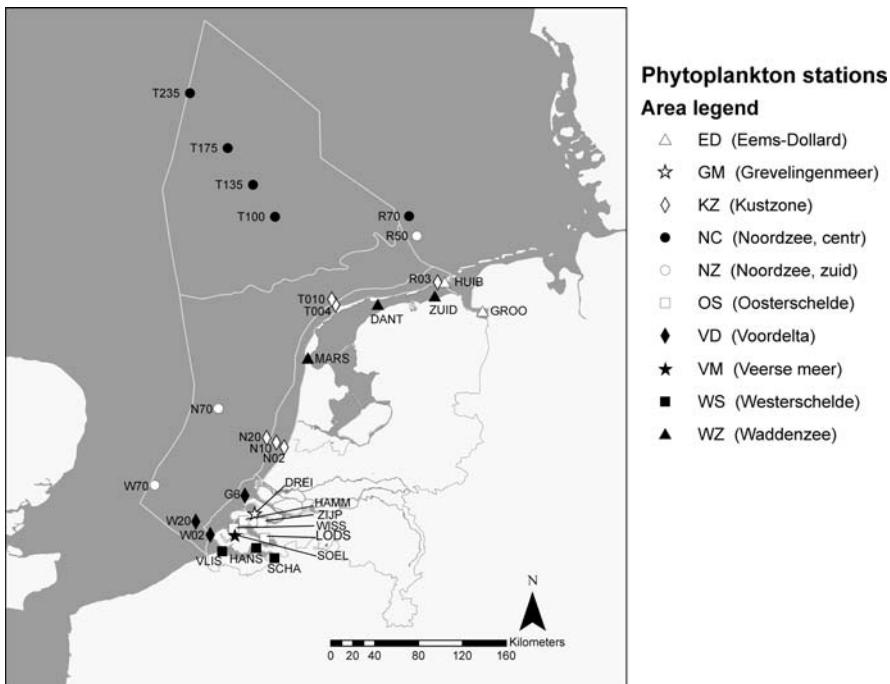


Fig. 18.2 Station locations and the area boundaries

sampling frequency depended on the season: monthly during winter and fortnightly during summer. All stations were sampled just below the water surface. When the water column is stratified, and this usually occurs on some, mainly offshore, stations in the summer, then samples were also taken at the thermocline and a few metres above the bottom with a Rosette sampler. This study is based on the Lugol-preserved samples taken close to the water surface.

To improve consistency in the phytoplankton data over time, the first year of the time series, 1990, has been skipped because the phytoplankton monitoring only started in April that year. Moreover, some taxa were left out because they were not consistently counted over time, and many taxa that can be individually identified microscopically today, were lumped together in the early years. Thus, the initial number of about 600 different taxa was reduced to a dataset that contains 175 taxa from 1991 on. As explained above, in this chapter we only focus on an even more aggregated group of small diatoms.

The environmental variables that were used in the technical report are dissolved inorganic nitrogen (DIN = ammonium, nitrite plus nitrate), dissolved inorganic phosphorus (DIP), silicon, total nitrogen, total phosphorus (all in $\mu\text{mol/l}$), salinity, temperature (Celsius), Secchi depth (dm), and suspended matter (mg/l). Here, we only use DIN and temperature for illustrative purposes.

18.2 Data Exploration

Instead of starting with a discussion of the statistical modelling approach, we first apply a data exploration as it spreads some light on the type of data we are working with. We arbitrarily chose DIN for this. Standard data exploration tools for multiple time series are a `xyplot` and `bwplot` (both from the `lattice` package), box-plots, and Cleveland dotplots. Pairplots are less useful for this particular example, because the DIAT1, DIN and temperature data were not sampled at the same time. Figure 18.3 shows a graph of log-transformed DIN values versus time, for each station. The following code was used to make the graph.

```
> library(AED); data(RIKZDATAEnv); library(lattice)
> RIKZ2 <- RIKZDATAEnv #Saves space
> RIKZ1 <- RIKZ2[RIKZ2$Year > 1990, ]
> I <- !is.na(RIKZ1$DIN)
> RIKZ <- RIKZ1[I, ]
> RIKZ$LDIN <- log(RIKZ$DIN)
> RIKZ$fStation <- factor(RIKZ$Station)
> RIKZ$MyTime <- RIKZ$Year + RIKZ$dDay3 / 365
> xyplot(LDIN ~ MyTime | Station, data = RIKZ,
         xlab = "Time", col = 1, type = "h",
         strip = function (bg = 'white', ...)
         strip.default(bg = 'white', ...))
```

The first series of commands are used to access the data, discard data from 1990, remove rows with missing values (it makes the model validation easier), and it applies a logarithmic transformation. The variable `MyTime` is used to provide sensible axis labels. We used the option `type = "h"` to ensure that observations

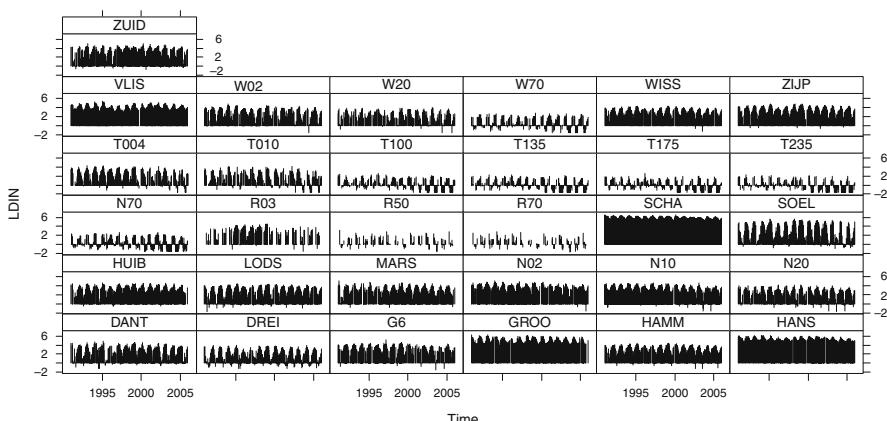


Fig. 18.3 Graph of log-transformed DIN (vertical axes) versus time (horizontal axes) per station. Vertical lines are used to show values

are not presented as points (in which case the graph becomes one big cloud of observations) or lines (missing values are not shown properly).

The graph shows there are differences per station in terms of absolute values, variation and number of missing values. The same graph for the untransformed data showed even more differences in absolute values per station. The question is now, what are we going to do with this? One option is to use log-transformed data, and another option is to standardise each time series. The latter option means that the data at stations like R50, with low values, become equally important as stations like ZUID with much higher values. Because DIN is a measure of available nutrients, we prefer *not* to make all series equally important, hence our choice for the log-transformation. The fact that we use a logarithmic transformation, and not a square root, is based on the range of the data.

For the same reason, a logarithmic transformation was applied on DIAT1. There was no need to transform temperature.

Figure 18.4 shows a conditional boxplot of the log transformed DIN values conditional on station. It shows that there are considerable differences between the stations, indicating that whatever model we apply, the term station has to be included.

We can either do this as a fixed term and pay the price of 30 regression parameters or use it as a random effect. The latter option makes more sense as it will allow us to make a general statement. If station is used as a fixed term, our statements and conclusions only hold for these particular 31 stations. Other advantages of using station as a random effect is that it saves a large number of parameters (one variance term versus 30 regression parameters) and it introduces a correlation structure between the observations at the same station (albeit it is the rather basic, compound symmetrical correlation structure; see Chapter 6).

Another aspect we need to take into account is seasonality. Figure 18.5 shows boxplots of log-transformed DIN values for each area conditional on month. Note

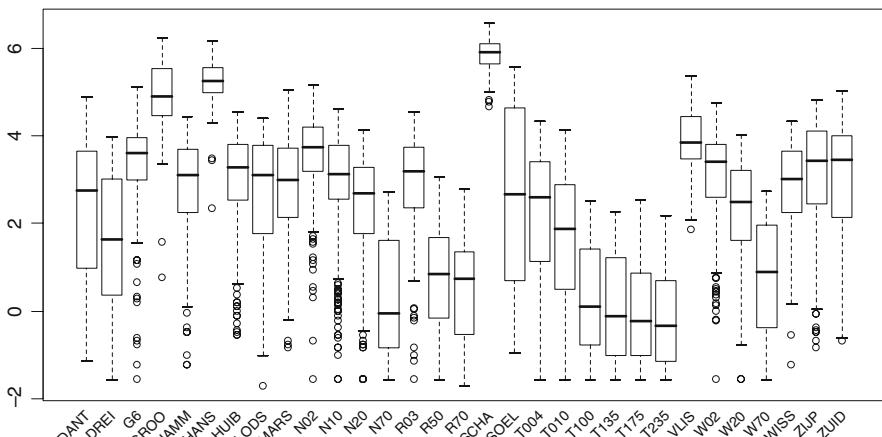


Fig. 18.4 Boxplot of log-transformed DIN conditional on Station. Mean values differ considerably per station, indicating that the term station has to be used in the model

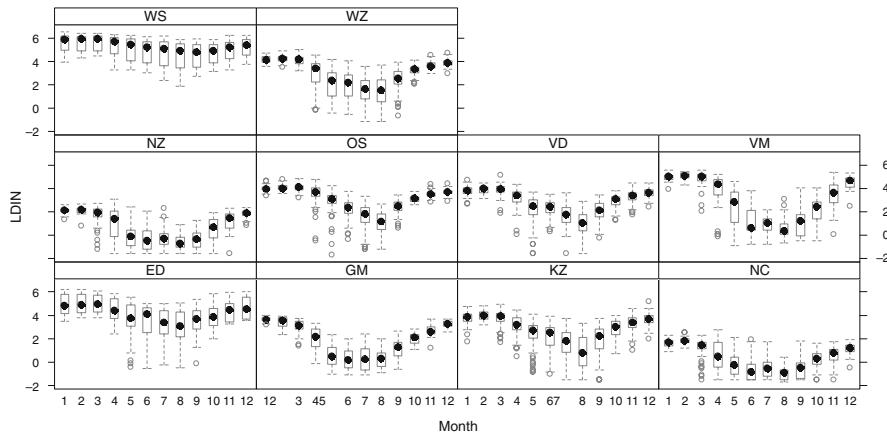


Fig. 18.5 Boxplots of log-transformed DIN versus month per area. Each area consists of various stations. There is a clear seasonal pattern that differs per area

there is a clear seasonal pattern at all areas, but not all patterns are identical. This means that we may have to include an interaction term between seasonal effects and area in the model. Note that for some months, in some areas, there is a group of observations outside the boxplot. For untransformed data, we saw similar patterns. We will discuss these points later.

The R code to produce Figs. 18.4 and 18.5 is given below. The code is self-explanatory, and in case of any doubts, consult the help files.

```
> #Figure 18.4:
> RIKE$Month <- factor(RIKE$Month)
> bwplot(LDIN ~ Month | Area, data=RIKE, xlab="Month",
  strip = function(bg = 'white', ...)
  strip.default(bg = 'white', ...), col = 1,
  scales = list(rot = 45, cex = .6))

> #Figure 18.5
> boxplot(LDIN~fStation, data = RIKE, xaxt = "n")
> text(1:31, par("usr")[3] - 0.25, srt = 45, adj = 1,
  labels = levels(RIKE$fStation), xpd = TRUE,
  cex = 0.75)
```

18.3 A Statistical Data Analysis Strategy for DIN

If environmental and phytoplankton data had been measured at the same time, the following model could be our starting point.

$$\text{Phytoplankton data}_s = f(\text{environmental data}_s) + \text{noise}_s$$

The notation $f()$ stands for ‘*is a function of*’, well at least for the moment. The index s represents the sampling time (e.g. day). However, most statistical software routines will drop each observation where at least one of the variables is missing. This means that if the response and environmental variables are not sampled at the same time, you may end up with no data at all. Hence, conventional methods like linear regression, generalised linear modelling (GLM), or generalised additive modelling (GAM) cannot easily be used to model the function f . So, the first item from our list of problems, given in Section 18.1, is already causing a major headache. Our solution is to use a different model with the form:

$$\text{Variable} = f(\text{Time}) + \text{noise}$$

This means that each variable, either environmental or phytoplankton, is modelled as a function f of time, which represents the trend. We will apply this model on each variable, and compare the estimated trends. The advantage of this approach is that we do not have to compare the environmental and phytoplankton data directly, but just their temporal trends. These trends are smoothing functions over time and have values at the same time points. This allows us to compare the trends of different variables. We should note that our prime aim is to compare long-term trends and not the short-term (or, within-year) variation.

However, the bad news is this model is still complex. We still need to be able to deal with heterogeneity (more variation in summer months than in winter months), spatial and temporal correlation, non-linear trends, etc. A method that can potentially cope with this complexity is mixed modelling or if we allow for non-linear (or better: non-parametric) trends: additive mixed modelling.

In linear mixed modelling and additive mixed modelling, the model selection approach should follow a protocol that roughly contains the following steps (Fitzmaurice et al., 2004; Diggle et al., 2002; Chapters 5 and 6):

1. Start with a model that is as good as you can get it in terms of the fixed explanatory variables.
2. Using the fixed terms from step 1, find the optimal random structure. This means that for the noise component, we have to try different options (e.g. random effects, temporal correlation, different variances, etc.).
3. For the optimal random structure found in step 2, find the optimal fixed structure.

This is a scheme that works well for linear mixed models applied on relatively small data sets, but for a large and complicated data set like ours, we have to be a bit more creative. In the remaining part of this section, we show how we sequentially develop our models and finally end up with something that seems to do the job. The starting point is again a model of the form

$$\text{LDIN} = \text{intercept} + f(\text{Time}) + \varepsilon$$

The ε represents the noise or unexplained bit and LDIN the log-transformed DIN data. First we need to add some indices. We have 31 stations, and measurements which are taken over time. This gives

$$\text{LDIN}_{is} = \text{intercept}_i + f_i(\text{Time}_s) + \varepsilon_{is}$$

where i is the station index from 1 to 31 and s represents the time units. The noise term ε_{is} is assumed to be normally and independently distributed with mean 0 and variance σ^2 . The model above allows for a different trend at each station (the function f has a subscript i). If smoothing techniques are applied to model the function f , then it is almost impossible to fit a model with 31 trends on a data set of this size. If we are lucky, only one trend will be needed for all stations, and the subscript i can be dropped from the function f . From the data exploration section, we know that the model needs a long-term trend, station effect, and a seasonal component; so a possible starting point is

$$\text{LDIN}_{is} = \text{intercept} + \text{factor}(\text{Station}_i) + f(\text{Time}_s) + \text{factor}(\text{Month}_s) + \varepsilon_{is} \quad (18.1)$$

The function f is now a smoothing function over time and is typically modelled with a spline. We have seen the notation *factor* in various other chapters; it is used to tell R that the corresponding variable is categorical. Used here, it indicates the variables Station and Month are considered as categorical variables in Equation (18.1). The costs are 11 parameters for Month and 30 for Station. We will return to the 30 parameters for Station in a moment. For the seasonal component, we have multiple options. Instead of using a categorical variable Month, you can also use sinus or cosines functions (Pinheiro and Bates, 2000) or a smoother $f(\text{DayInTheYear}_s)$, where the variable DayInTheYear_s takes values between 1 and 365 (Wood, 2006). Based on initial runs, the latter option performs the best as judged by the AIC. Note that we are not too fussy about leap years.

Regarding the argument Time_s in the function $f(\text{Time}_s)$, we have two options. We can use the day of sampling expressed as the number of days since the first sampling day of the experiment (or since 1 January 1991). But you then need to ensure that sampling day for all variables is expressed relative to the same starting date! The second option is to use $f(\text{Year}_s)$, where Year_s has integer values between 1991 and 2005. It takes less computing time and is slightly easier for comparing trends of different variables, and this is the approach we use here.

Up to now, the model contains components for trends over time and trends within a year (the seasonal pattern). However, sampling took place at 31 stations and we also have the spatial coordinates for each station (denoted by X_i and Y_i). In the same way as temporal trends were added, we can include a spatial trend $f(X_i, Y_i)$, which is a 2-dimensional smoother. This gives the following model:

$$\begin{aligned} \text{LDIN}_{is} = & \text{intercept} + \text{factor}(\text{Station}_i) + f(\text{Year}_s) + f(\text{DayInTheYear}_s) \\ & + f(X_i, Y_i) + \varepsilon_{is} \end{aligned} \quad (18.2)$$

The problem with this model is that we are paying the penalty of 30 regression parameters for the station effect. This is in fact the same discussion that we had when random effects were introduced in Chapter 5. Are we really interested in knowing which stations have higher values than others? Do we want to make statements for only these 30 stations? In this case, the answer to both questions is no, and this is a typical example of using a random intercept for station. It allows us to make statements for all similar stations along the Dutch coast and saves several degrees of freedom. Therefore, model (18.2) becomes

$$\text{LDIN}_{is} = \text{intercept} + f(\text{Year}_s) + f(\text{DayInTheYear}_s) + f(X_i, Y_i) + a_i + \varepsilon_{is} \quad (18.3)$$

The random intercept a_i is assumed to be normally distributed with mean 0 and variance $\sigma_{\text{station}}^2$. So far, adding terms was based on common sense and some initial analyses. At this stage, it is perhaps useful to apply the model and see where it fails. This will guide further improvements, if needed.

The advantage of the model in Equation (18.3) is that we have decomposed the time series into long-term trends and short-term trends. Each of these components can be extracted and compared with other environmental long-term and short-term trends, or with the phytoplankton short-term and long-term trends.

The following R code is used to implement the model in Equation (18.3).

```
> RIKZ$X <- RIKZ$X31UE_ED50 #spatial coordinates
> RIKZ$Y <- RIKZ$X31UN_ED50 #spatial coordinates
> library(mgcv)
> M1 <- gamm(LDIN ~ s(Year) + s(dDay3) + s(X, Y),
  random = list(fStation =~ 1), data = RIKZ)
```

The variable `dDay3` contains the coding of the sampling day in a year, expressed as a number between 1 and 365. The results of this model are given in Fig. 18.6, which was produced with the following R code.

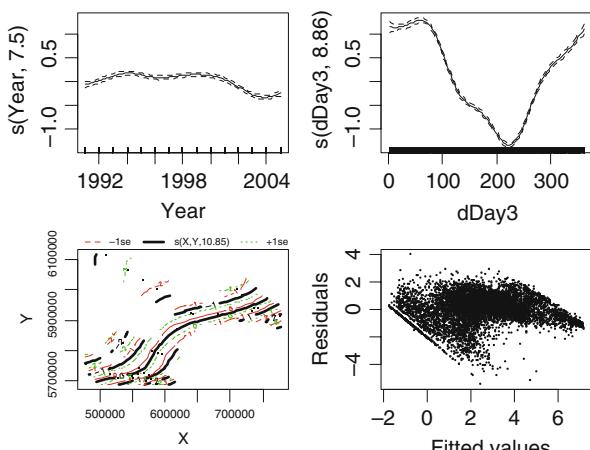


Fig. 18.6 Results for the model in Equation (18.3): *Upper left*: Smoothing component $f(\text{Year}_s)$. *Upper right*: Smoothing component $f(\text{DayInTheYear}_s)$. *Lower left*: Smoothing component $f(X_i, Y_i)$. *Lower right*: Normalised residuals versus fitted values showing heterogeneity

```

> op <- par(mfrow = c(2, 2))
> plot(M1$gam, select = c(1))
> plot(M1$gam, select = c(2))
> plot(M1$gam, select = c(3))
> E <- resid(M1$lme, type = "normalized")
> F <- fitted(M1$lme)
> plot(x = F, y = E, xlab = "Fitted values",
       ylab = "Residuals", cex = 0.3)
> par(op)

```

There are various problems with the model in Equation (18.3) with both heterogeneity and patterns in the residuals. The latter problem is probably due to using only one smoother for long-term trends at all stations and using one seasonal component for all stations. The data exploration had already indicated that these patterns differ per station. Hence, a natural extension is to use multiple long-term trends and multiple seasonal smoothers. To find a balance between what is needed and what can be done with current software and the numerical capacity of computers, we introduce an interaction term between some of the smoothers and area. If we use one long-term smoother per area and one seasonal pattern per area, the model becomes

$$\text{LDIN}_{is} = \text{intercept} + f_{\text{area}}(\text{Year}_s) + f_{\text{area}}(\text{DayInTheYear}_s) + f(\mathbf{X}_i, \mathbf{Y}_i) + a_i + \varepsilon_{is} \quad (18.4)$$

The term $f_{\text{area}}(\text{Year}_s)$ is the long-term smoother for a particular area (each area consists of multiple stations), and the same holds for the within-year pattern $f_{\text{area}}(\text{DayInTheYear}_s)$. Recall that there are 10 areas, meaning the model has $10 + 10 + 1 = 21$ smoothers. Instead of the notation $f_{\text{area}}(\text{Year}_s)$, you can also use $f_a(\text{Year}_s)$ or even $f(\text{Year}_s):\text{Area}$. The choice of notation depends on your own preference or the style of the journal you are aiming for. The R code to fit the model in Equation (18.4) is given by¹

```

> M2 <- gamm(LDIN ~
  s(Year, by = as.numeric(Area == "WZ"), bs = "cr") +
  s(Year, by = as.numeric(Area == "GM"), bs = "cr") +
  s(Year, by = as.numeric(Area == "VD"), bs = "cr") +
  s(Year, by = as.numeric(Area == "ED"), bs = "cr") +
  s(Year, by = as.numeric(Area == "OS"), bs = "cr") +
  s(Year, by = as.numeric(Area == "WS"), bs = "cr") +
  s(Year, by = as.numeric(Area == "KZ"), bs = "cr") +
  s(Year, by = as.numeric(Area == "NZ"), bs = "cr") +
  s(Year, by = as.numeric(Area == "NC"), bs = "cr") +
  s(Year, by = as.numeric(Area == "VM"), bs = "cr") +
  s(dDay3, by = as.numeric(Area == "WZ"), bs = "cr") +

```

¹We used R version 2.6 and mgcv version 1.3–27. More recent versions of R and mgcv require a small modification to the code; see the book website (www.highstat.com) for updated code.

```

s(dDay3, by = as.numeric(Area == "GM"), bs = "cr") +
s(dDay3, by = as.numeric(Area == "VD"), bs = "cr") +
s(dDay3, by = as.numeric(Area == "ED"), bs = "cr") +
s(dDay3, by = as.numeric(Area == "OS"), bs = "cr") +
s(dDay3, by = as.numeric(Area == "WS"), bs = "cr") +
s(dDay3, by = as.numeric(Area == "KZ"), bs = "cr") +
s(dDay3, by = as.numeric(Area == "NZ"), bs = "cr") +
s(dDay3, by = as.numeric(Area == "NC"), bs = "cr") +
s(dDay3, by = as.numeric(Area == "VM"), bs = "cr") +
s(X, Y), random = list(fStation =~ 1), data = RIKZ)

```

It looks intimidating, but it is only a simple extension of the model in Equation (18.3). The `by` option is used to ensure that the particular smoother is only applied on a subset of the data where the argument of the `by` option is equal to 1. The `as.numeric()` is used to convert the value TRUE to a 1 and FALSE to 0. The AIC of models (18.3) and (18.4) is 18366.07 and 16648.41, respectively. You can also try intermediate models with multiple long-term smoothers and a single seasonal smoother, or the other way around, but their AICs are all larger than 16648.41. We used cubic regression splines (`bs = "cr"`) for the temporal trends because with large data sets these have shorter computing times than the default thin spline smoother.

The estimated long-term and seasonal smoothers obtained by this model are given in Figs. 18.7 and 18.8. The code to make these graphs is complex and given on the book website. Several long-term smoothers have similar patterns, e.g. WZ, ED, and GM. In fact, most trends have two peaks; one in the early 1990s and one towards the end of the 1990s. The trend for VM shows a strong decrease since 1998. As to the seasonal patterns per area, you can see that some areas (e.g. WS, ED) have a less strong seasonal pattern. The other areas have all slightly different shapes.

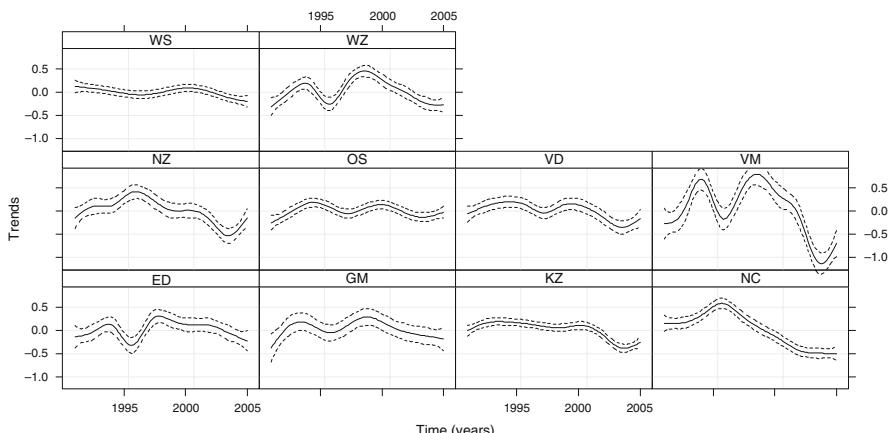


Fig. 18.7 Estimated long-term smoother for each area obtained by the model in Equation (18.4)

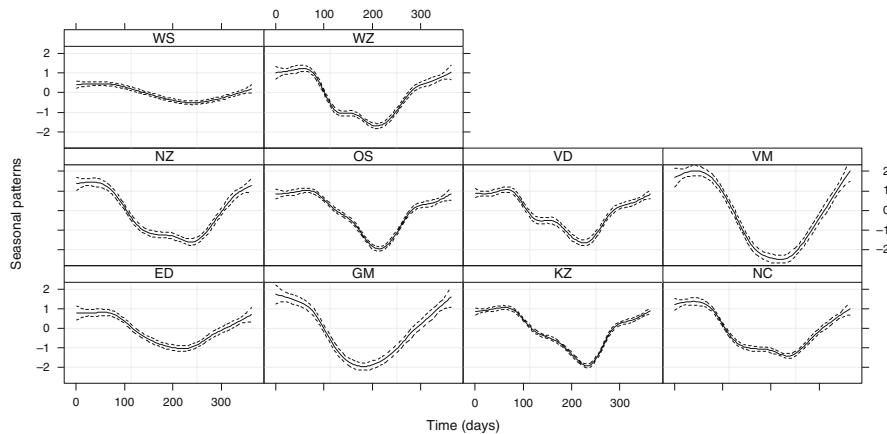


Fig. 18.8 Estimated seasonal pattern per area obtained by the model in Equation (18.4). The x-axis contains the days from 1 to 365

As part of the model validation process, we plotted normalised residuals versus month (for all stations), see Fig. 18.9. If you wonder why we did this, then the answer is ‘common sense’. In most ecological systems, the spread in the data differs between months or seasons. You can get the same message by redrawing the lower right panel in Fig. 18.6 and use different colours per month or season. Figure 18.9 shows that there is more variation in spring and summer than in autumn and winter, which violates the homogeneity assumption. The figure was created with the following R code.

```
> E2 <- resid(M2$lme, type = "n")
> plot(E2 ~ RIKZ$fMonth, xlab = "Month",
      ylab = "Normalised residuals")
```

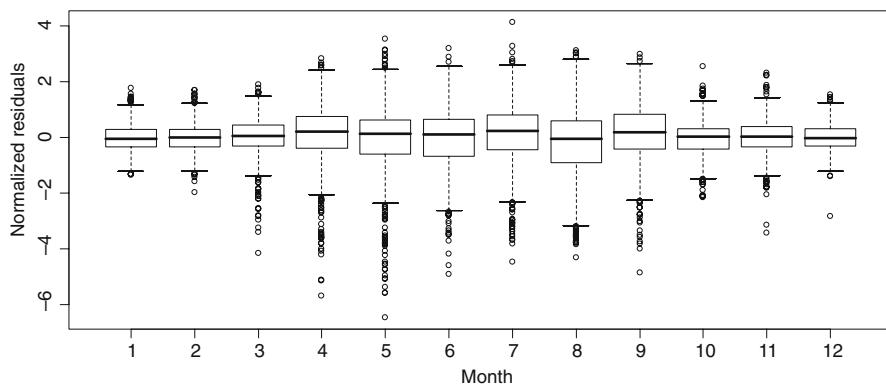


Fig. 18.9 Normalised residuals plotted versus month obtained by model (18.4)

A solution for the heterogeneity problem is to relax the assumption that the residuals ε_{is} are normally distributed with mean 0 and variance σ^2 . Instead, we can use a Normal distribution with mean 0 and variance σ_m^2 , where m stands for month. Hence, the residuals are allowed to have a different spread per month. The problem is that computing time for such a model for these data can be long (hours on a modern computer), and therefore, it may be a more realistic option to use a different variance per season (four variances) or per 6-month period (two variances). We decided to go for four variances and define the seasons as months 1–3, 4–6, 7–9, and 10–12. However, further fine-tuning of the model can still be achieved. The R code for the model with four variances is a simple extension of the previous R code and is not reproduced here. We only have to define a variable defining the four seasons:

```
> n <- length(RIKZ$Month)
> RIKZ$M14 <- vector(length = n)
> RIKZ$M14[1:n] <- 0
> RIKZ$M14[RIKZ$Month >= 1 & RIKZ$Month <= 3] <- 1
> RIKZ$M14[RIKZ$Month >= 4 & RIKZ$Month <= 6] <- 2
> RIKZ$M14[RIKZ$Month >= 7 & RIKZ$Month <= 9] <- 3
> RIKZ$M14[RIKZ$Month >= 10 & RIKZ$Month <= 12] <- 4
> RIKZ$fm14 <- factor(RIKZ$M14)
```

Allowing for different variances is done with the `weights` option and the `varIdent` structure; see also Chapter 5. All we have to do is add the code `weights = varIdent(form = ~1 | fm14)` to the `gamm` function presented above.

Unfortunately, this model did not converge. Using the `varIdent` function with two variances (two seasons) neither converged. However, using 10 long-term smoothers and one seasonal pattern for all smoothers plus four variances for the seasons (and a random intercept) did not cause any numerical problems. If this sort of numerical trouble happens, it can be quite a challenge to sort out. One option is to increase the number of iterations in the `gamm` routine or reduce the convergence criteria, see the help file of `gamm` how to do this. Other options are to fix the degrees of freedom (and not use cross-validation) or set the four variances to fixed values (e.g. based on the residual variation of previous models). Unbalanced data, missing values, etc., can also cause convergence problems. We tried all of this, but without success. However, replacing the 10 seasonal smoothers by a `fMonth` × `Area` component in the model in Equation (18.4) and re-running the code did not give any converge problems. The estimated long-term smoothers obtained by this model had nearly identical shapes as those in Fig. 18.7. Furthermore, extending this model with four residual variances did not cause any numerical problems. Again, its estimated long-term trends are similar as to those in Fig. 18.7 and are therefore not presented again.

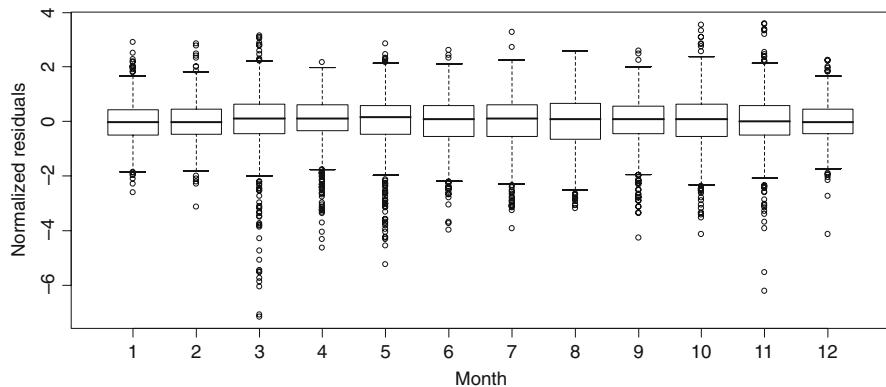


Fig. 18.10 Normalised residuals of the model with 10 long-term smoothers, seasonal components modelled by $f\text{Month} \times \text{Area}$, a spatial trend, a random intercept for stations, and four variances. Note that Area is automatically a factor due to its coding. Residuals are grouped per month

However, the model validation did show some problems. Although the residual spread is approximately the same in all months, we still have more negative residuals in the spring and summer than in the autumn and winter (Fig. 18.10). We had already seen this behaviour in the data exploration section. By plotting the residuals versus time for each station (Fig. 18.11), we can see that there is no clear pattern in these large values. One option to deal with this is to include a nested (within station) random intercept for month. This allows for random variation around the seasonal pattern, and this variation can be different per month. However, this would only

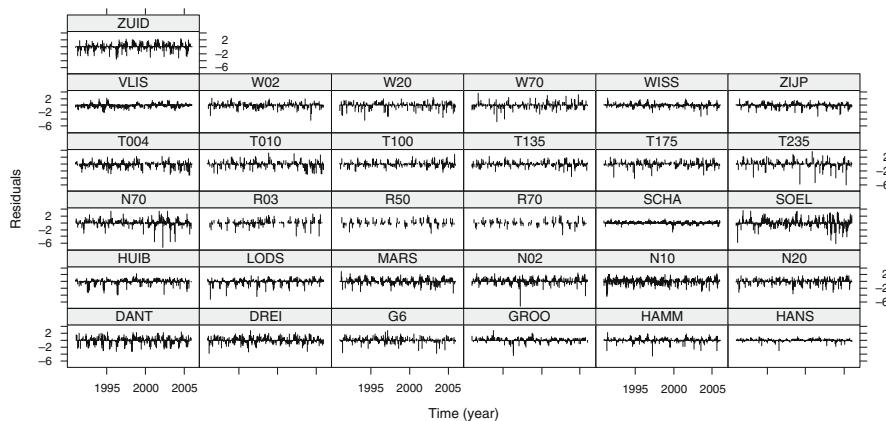


Fig. 18.11 Normalised residuals of the model with 10 long-term smoothers, seasonal components modelled by $f\text{Month} \times \text{Area}$, a spatial trend, a random intercept for stations, and four variances. Residuals plotted versus year for each station

hide the fact that we have large observations in some months and stations. This may well be a sampling issue; not all stations are sampled on the same day due to practicalities of travel arrangements (some stations are separated by 250 km). If DIN values are high (for a short period) in a certain region, then you may measure it at one station, but values may have dropped already by the time you reach the next station. So, instead of hiding it in random effects, we will leave it as it is. The R code to produce Figs. 18.10 and 18.11 is not reproduced here as it closely follows earlier code.

Plotting normalised residuals versus fitted values showed that there is still a certain degree of heterogeneity in the residuals. This is because some stations have less variation in DIN values. This can also be seen in the data exploration section and even in Fig. 18.11. Another way of spotting this is to plot fitted values against residuals and use a different colour per station. It is difficult to solve this. It is not practical to use the varIdent structure and 31 different variances as computing time would drastically increase. A better option is to scale each time series, for example, by using: $\text{LDIN}_{is}/\max(\text{LDIN}_{is})$. Such a standardisation ensures all the time series have similar *variation*, but the average values can still be different (this is contrary to centring and dividing by the standard deviation).

The last aspect we look at (as part of the model validation process) is spatial patterns in residuals. We made a bubble plot of averaged (per station) residuals (Fig. 18.12), and there seems to be no clear clustering of positive (or negative)

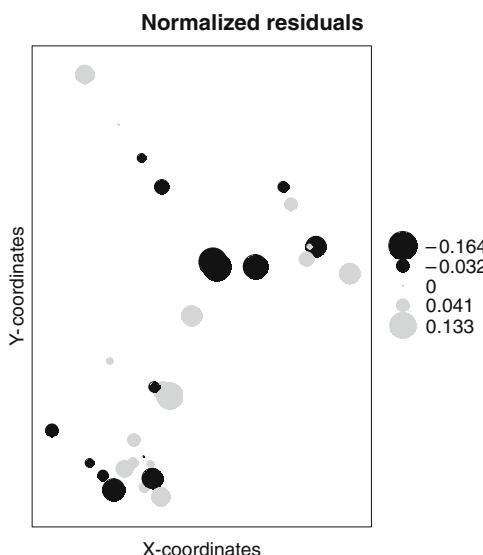


Fig. 18.12 Bubble plot for averaged residuals per station. Large dots represent large residuals with black dots for negative residuals and grey dots for positive residuals. The R code to produce this graph is available from the book website

residuals. It is also possible to make this graph for data of each year or each season. Alternatively, variograms can be made of residuals per station or per year. It would be a nice challenge to make an `xypplot` with multiple variograms in it, but we will leave this as an exercise for the reader.

To allow for spatial or temporal correlation in the residuals, you can attempt to add the correlation option to the `gamm` function, but our initial attempts resulted in convergence problems due to the large sample size. So, we are pushing things a little bit too far with current software and hardware.

As to the numerical output, all trends were highly significant. However, we advise being cautious with these p -values as there is considerable residual information left in the model. It may be an option to allow for more smoothers for the series or analyse these time series separately.

18.4 Results for Temperature

The same modelling strategy was applied on the (untransformed) temperature data. We started the model selection process from scratch. The data exploration showed that the patterns over time show less variation compared to the DIN data. Figure 18.13 shows the boxplots of temperature per month for each area. It will be interesting to test whether the seasonal pattern change per area or not.

The same strategy used for the DIN analysis was followed. The AIC showed that the model with 10 long-term smoothers, 10 seasonal smoothers, a spatial trend, and a random intercept was the best model. There was only minor (visual) evidence of heterogeneity and therefore no real need to use multiple variances per season. The estimated long-term smoothers are shown in Fig. 18.14. It may be an option

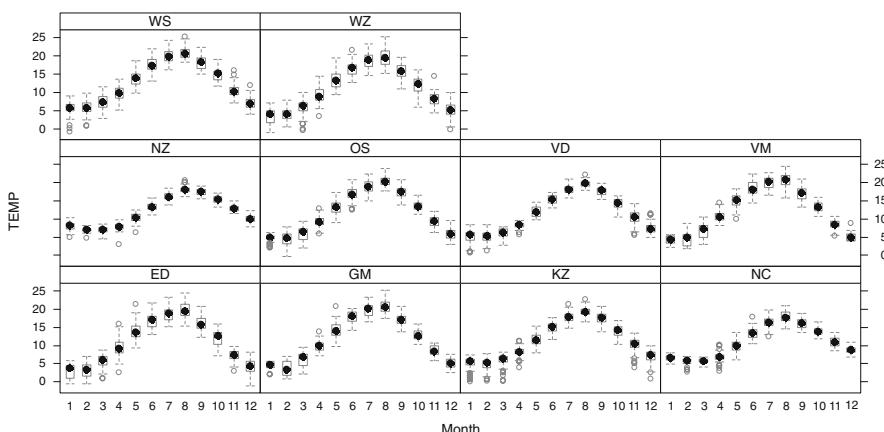


Fig. 18.13 Temperature per month for each area

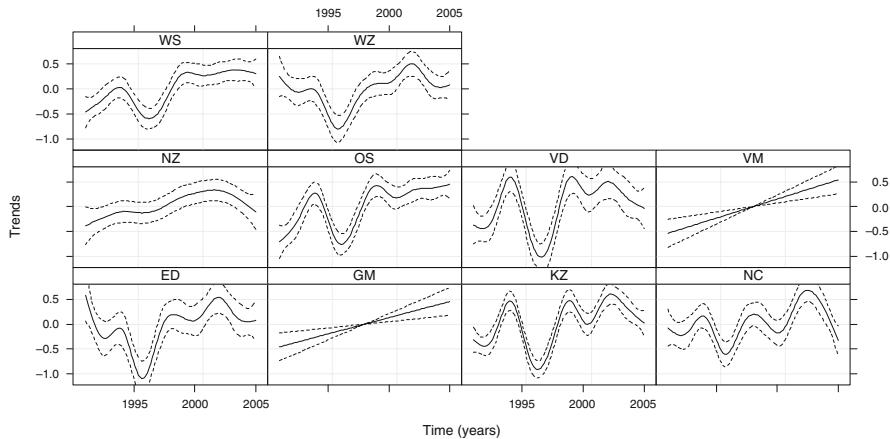


Fig. 18.14 Long-term trends for temperature by area. The *solid line* is the smoother and the *dotted lines* are 95% confidence bands

to group some of the areas and use only one smoother for them, but this makes the comparison with the phytoplankton data, presented later, more difficult. The 10 seasonal components are given in Fig. 18.15; note that NZ, NC, VD, and KZ trends are slightly different from the others. The shape of these curves also shows why a sinus function would not work; the patterns are not symmetrically shaped during the year. The spatial trend $f(X, Y)$ is presented in Fig. 18.16.

The R code for the temperature data analysis is identical to the code used in the previous section and is not presented again.

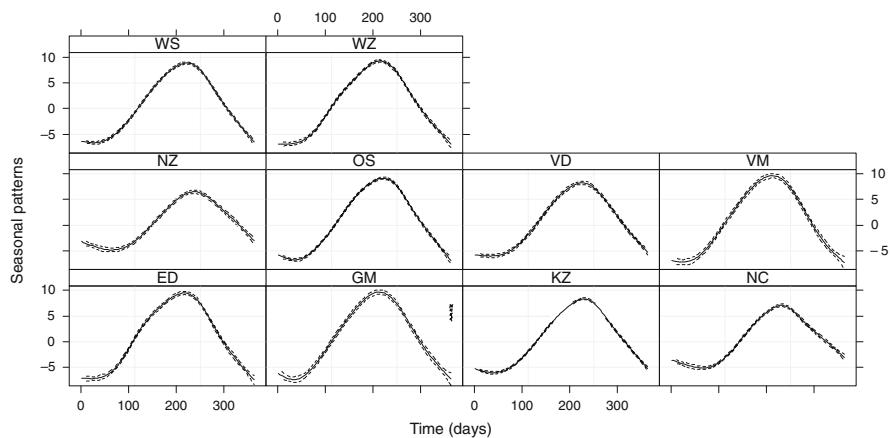


Fig. 18.15 Seasonal components for temperature by area. The *solid line* is the smoother and the *dotted lines* are 95% confidence bands

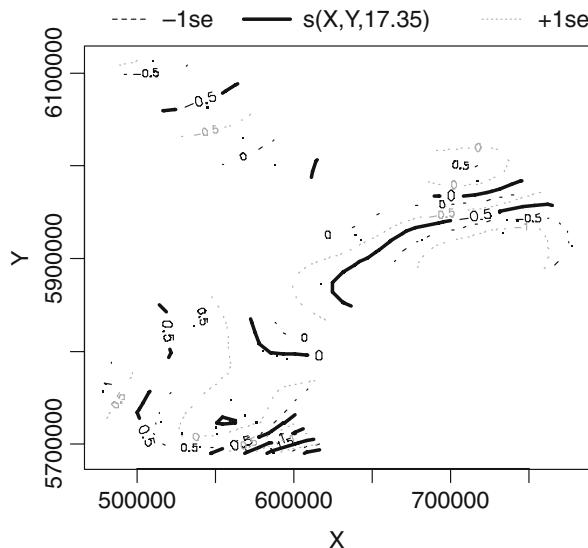


Fig. 18.16 Spatial trend for the temperature data. The *solid line* represents the *contour lines* and the *dotted lines* are confidence bands. It is also possible to plot this graph as a 3-dimensional picture

18.5 Results for DIAT1

In this section, the aggregated DIAT1 (diatoms between 0 and 1,000 μm^3) phytoplankton series are analysed. An initial data exploration was carried out, and this indicated that a log-transformation was needed.

The main difference with the environmental data and these data is that during the 15 years of sampling, four different laboratories were successively involved with the counting of the phytoplankton. There was no overlap between the laboratories and there is a clear ‘laboratory’ effect that can be seen using a simple boxplot or a more advanced boxplot produced by the `bwplot` function from the lattice package, see Fig. 18.17. At most areas, values taken by laboratory D are the highest. However, this was also the laboratory that took the most recent samples. If we include the term `factor(Laboratory)` in model (18.4) and replace LDIN by LDIAT1, the categorical variable laboratory is highly significant. In fact, the estimated trends for the model with and without the laboratory effect are only slightly different, especially during the period when laboratory D was in charge of counting. The question then rises, whether there is indeed a laboratory effect or whether abundances have increased during the period when laboratory D counted. Unfortunately, there is no way we can distinguish between the two. The only thing that we can say is that the estimated laboratory effect (as measured by estimated parameters for each level of the categorical variable) is larger than you would expect based on common sense approach to our existing ecological understanding. We have found similar changes of abundance between other years, not corresponding with

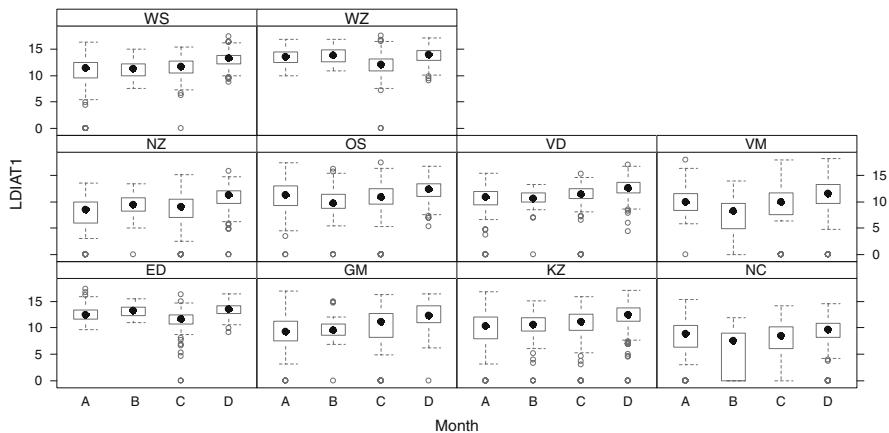


Fig. 18.17 Boxplot of log-transformed DIAT1 conditional on laboratory (represented by A, B, C, and D) per area

factor Lab. As well as looking further into this particular DIAT1-group either on the level of species or per station, the Lab-pattern appeared not to be of a structural kind. Summarising, we cannot say whether any changes over time in abundances are due to a laboratory effect or whether it represents a real change. We therefore concluded that the laboratory effect is small compared to observed changes and ignored it.

The modelling approach followed similar lines used for the environmental variables. Note that most long-term trends in Fig. 18.18 seem to increase up to about 2001. Seasonal patterns are rather different per area (Fig. 18.19). Some areas show a clear diatom blooming in early spring followed by a smaller bloom in late autumn. The spatial pattern is given in Fig. 18.20.

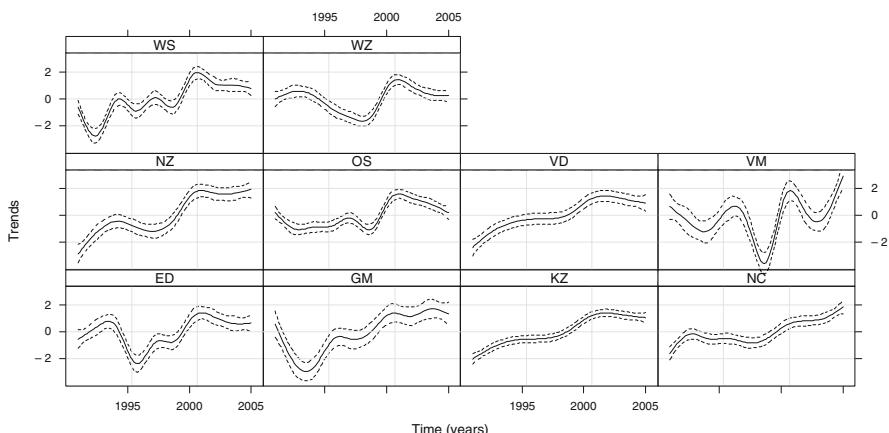


Fig. 18.18 Estimated trends for log-transformed diatoms (DIAT1). The model did not contain a laboratory effect

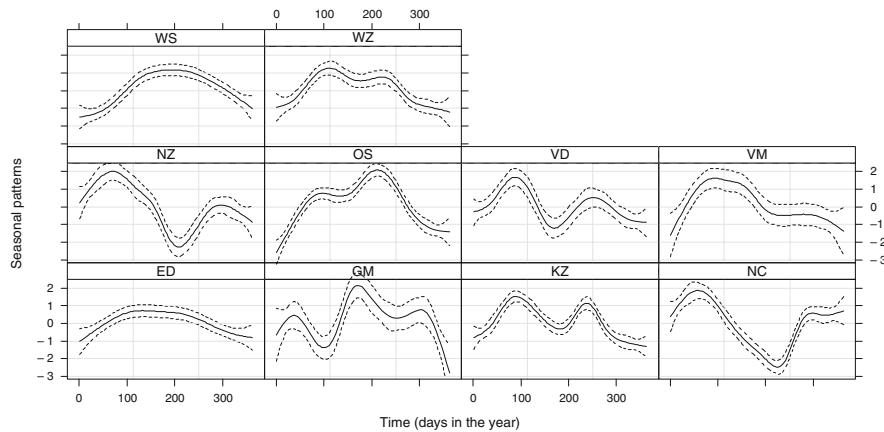
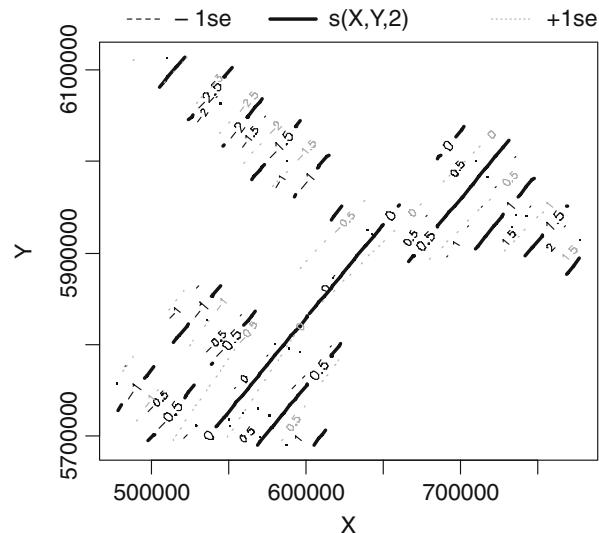


Fig. 18.19 Estimated seasonal patterns for log-transformed diatoms (DIAT1). The model did not contain a laboratory effect

Fig. 18.20 Estimated spatial patterns for log-transformed diatoms (DIAT1). The *solid* lines are the *contour lines* and the *dotted* lines are 95% confidence bands. The model did not contain a laboratory effect



18.6 Comparing Phytoplankton and Environmental Trends

In the previous three sections, we applied a Gaussian GAMM on multiple times series for DIN, temperature, and DIAT1. For each variable, we have 10 long-term trends. The question now is whether there is any relationship between the DIAT1 trends and the DIN and temperature trends. One may be tempted to consider DIAT1 as a response variable and DIN and temperature as explanatory variables. However, the original data set had approximately 8–10 explanatory variables, and there was

considerable collinearity between these variables. This makes it rather difficult to pinpoint any of the variables as *the* driving variable. An extra interpretation problem is caused by the seasonal patterns in the original data, as this may cause the high correlations between the explanatory variables. We therefore go for the simple approach comparing the long-term DIAT1 trends with the DIN and temperature trends using Pearson correlation coefficients. There is no point in comparing the ED DIAT1 trend with the WS DIN trend as these areas are 250 km apart. Hence, it makes more sense to compare the DIAT1 and environmental trends per area.

A word of caution is also needed. Long-term trends tend to be smooth functions by definition, and the Pearson correlation coefficient between two smooth functions tends to be high. Furthermore, we are going to calculate 20 correlation coefficients, which means that there are potential problems with multiple testing. Our view on this is to just calculate the correlations, present them graphically, see which combinations have the highest correlations, and refrain from interpreting *p*-values. The estimated Pearson correlations are given in Table 18.1 and a graphical presentation of these correlations in Fig. 18.21. The graphical presentation may look like overkill, but it is useful if more environmental variables are used. Another way to present the estimated correlations is presented in Fig. 18.22; the correlations between the DIAT1 and environmental variables are presented in two panels, the font size of the labels is proportional to the (absolute) estimated correlations. The advantage of this graph is that you have a better overview where (spatially) the areas with high correlations are.

The R code to calculate the correlation between the trends and to produce Figs. 18.21 and 18.22 is rather complicated and is given on the book website. The main problem in the R code is to access the estimated smoothers. By default, the `plot.gam` function is creating smoothers of length 100; hence, the smoothers in for example Fig. 18.18 are interpolated curves. Here we used long-term smoothers of length 15 (because there are 15 years).

Table 18.1 Estimated Pearson correlation coefficients between the DIAT1 trends and the corresponding (i.e. same area) DIN and temperature trends

	DIN	TEMP
DIAT1WZ	-0.29	0.62
DIAT1GM	-0.33	0.78
DIAT1VD	-0.36	0.48
DIAT1ED	-0.3	0.68
DIAT1OS	-0.2	0.37
DIAT1WS	-0.39	0.66
DIAT1KZ	-0.61	0.58
DIAT1NZ	-0.54	0.79
DIAT1NC	-0.85	0.3
DIAT1VM	-0.51	0.32

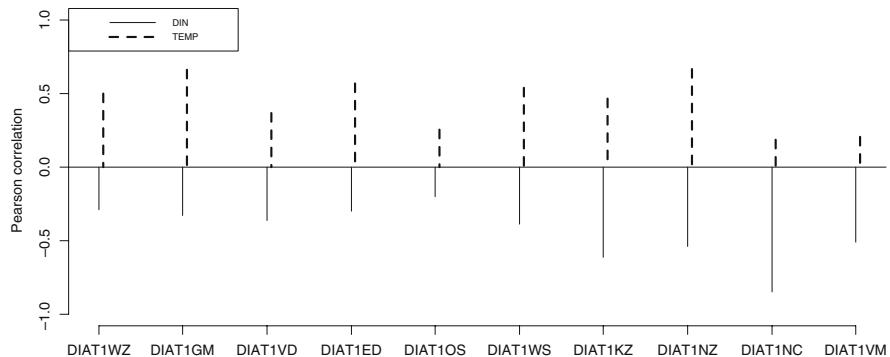
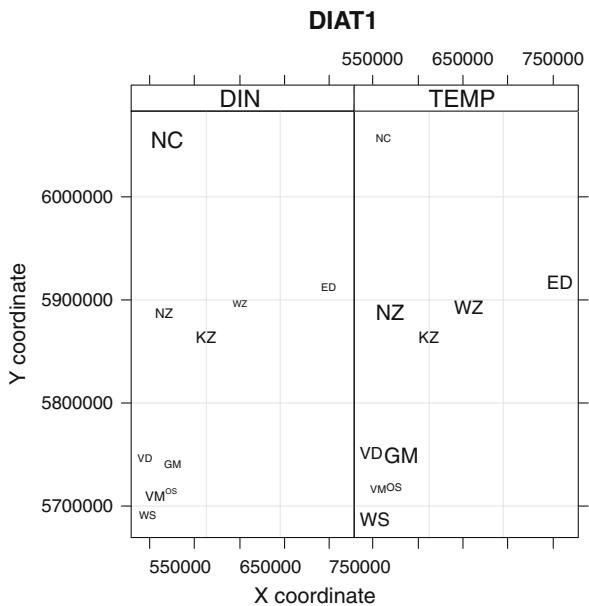


Fig. 18.21 Graphical presentation of the estimated Pearson correlation coefficients in Table 18.1. The endpoint of a line gives the value (along the y-axis) of a DIAT1 trend with the corresponding environmental variable (for the same area)

Fig. 18.22 Graphical presentation of the estimated Pearson correlation coefficients. The font size of the labels for an area is proportional to the absolute value of the estimated correlation



18.7 Conclusions

The analysis of the Rijkswaterstaat time series was one of the more challenging exercises in this book. However, it is a type of data set you are very likely to come across if you work with ecological or environmental monitoring data. By no means is this a finalised analysis. It would, for example, be good to add temporal

residual correlation structures using, for example, the option `correlation = corAR1(form =~ dDay1 | fStation)` within the `gamm` function. At the time of writing, our (new) computer (with a Windows operating system) was not able to carry out such analyses for this data set (and due to the complex mathematical calculations, it is unlikely to run neither on a Mac, LINUX, or UNIX operating system). It would be even better to use a spatio-temporal residual correlation structure, which you would have to program yourself. Before making any attempts to include a correlation structure, you should make an experimental variogram of the normalised residuals per station and plot the experimental variograms in a lattice plot. If these suggest there is no temporal correlation, then there is no point trying to add a temporal correlation structure inside the model.

The data presented here is merely an illustration how to deal with data of this type and is a spin-off from a technical report. The original report used more environmental variables and more phytoplankton groups. Because we only used a small part of the data here, we will not go into a biological discussion of the results.

The technical aspects of the analysis of multiple phytoplankton species are simple; just apply the same methodology on the most important species and use good visualisation tools to present the results.

18.8 What to Write in a Paper

If this chapter was your work, you are faced with a dilemma. The residuals of the estimated models still show patterns. So you either have to present this as a paper with preliminary results and make it clear that further work is going on or you can argue that this is as much as can be done with current hardware and software, and because all terms in the model are highly significant, the results are reasonable robust. Whichever route you go, you have to be very careful with the interpretation of the results due to the remaining patterns. However, in the technical report we analysed the data slightly differently, but the estimated long-term trends were nearly identical to the ones presented here. Perhaps, some simulation studies to assess sensitivity would be a useful addition to convince the referees.

Chapter 19

Mixed Effects Modelling Applied on American Foulbrood Affecting Honey Bees Larvae

A.F. Zuur, L.B. Gende, E.N. Ieno, N.J. Fernández, M.J. Egularas, R. Fritz, N.J. Walker, A.A. Saveliev, and G.M. Smith

19.1 Introduction

In this chapter, we apply mixed modelling to honeybee data. The data are considered nested because multiple observations were taken from the same hive. A total of 24 hives were sampled.

American Foulbrood (AFB) is an infectious disease affecting the larval stage of honeybees (*Apis mellifera*) and is the most widespread and destructive of the brood diseases (Shimanuki, 1997). The causative agent is *Paenibacillus larvae* (Genersch et al., 2006) and the spore forming bacterium infects queen, drone, and worker larvae. Only the spore stage of the bacterium (Fig. 19.1) is infectious to honey bee larvae. The spores germinate into the vegetative stage soon after they enter the larval gut and continue to multiply until larval death. The spores are extremely infective and resilient, and one dead larva may contain billions of spores (Hansen and Brødsgaard, 1999).

Although adult bees are not directly affected by AFB, some of the tasks carried out by workers might have an impact on the transmission of AFB spores within the colony and on the transmission of spores between colonies. When a bee hatches from its cell, its first task is to clean the surrounding cells, and its next task is tending and feeding of larvae. Here, the risk of transmitting AFB spores is particularly great if larvae that succumbed to AFB are cleaned prior to feeding susceptible larvae (Lindstrom, 2006).

Because AFB is extremely contagious, hard to cure, and lethal at the colony level, it is of importance to detect outbreaks, before they spread and become difficult to control (Lindstrom, 2006). Reliable detection methods are also important for studies of pathogen transmission within and between colonies. Of the available methods, sampling adult bees has been shown the most effective (Nordström et al., 2002). Hornitzky and Karlovskis (1989) introduced the method of culturing adult honey bees for AFB, and demonstrated that spores can be detected from colonies without

A.F. Zuur (✉)
Highland Statistics Ltd., Newburgh, AB41 6FN, United Kingdom

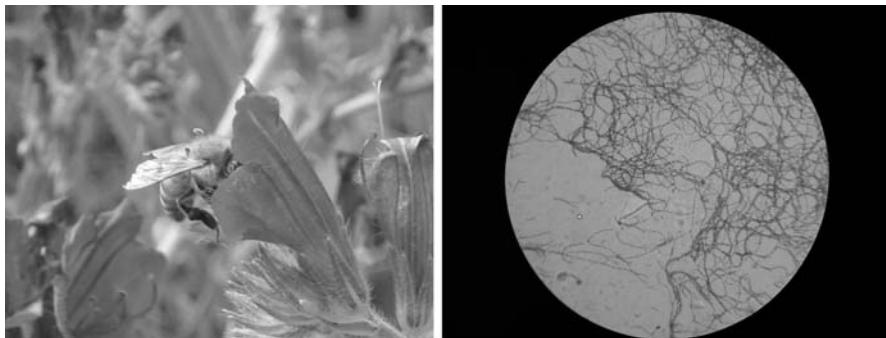


Fig. 19.1 *Left:* Honeybee. *Right:* Vegetative stage of the bacteria at microscopic level

clinical symptoms. Recently, culturing of *P. larvae* from adult honey bee samples has been shown to be a more sensitive tool for AFB screening compared to culturing of honey samples (Nordström et al., 2002). When samples of adult bees are used, the detection level of *P. larvae* is closely linked to the distribution of spores among the bees. For this reason, we will model the density of *P. larvae* with the potential explanatory variables as number of bees in the hive, presence or absence of AFB, and hive identity. Technical details on how spores were counted can be found in Hornitzky and Karlovskis (1989).

19.2 Data Exploration

There are three observations per hive, with a total of 24 hives. Figure 19.2A shows a Cleveland dotplot for the spores (density) conditional on hives. Recall from Chapter 2 that this graph groups the observations from the same hive along the vertical axis, and the values of the spores can be read from the horizontal axis. Two hives have considerably higher values than the others, indicating that serious problems with homogeneity can be expected if linear regression or mixed effects modelling is applied. One option is to use different variances per hive (Chapter 4), but this would result in 24 extra variances. This might make the estimation process for multiple variances with generalised least squares (GLS) unstable. We therefore prefer to transform the data using a logarithmic transformation. A square root transformation was also tried, but was considered too weak to ensure homogeneity. Because some observations have the value of 0, a $\log_{10}(Y_{ij} + 1)$ transformation was applied, where Y_{ij} is the density of spores in observation j in hive i , with $j = 1, \dots, 3$, and $i = 1, \dots, 24$. The transformed data are shown in Fig. 19.2B. The R code to access the data, transform the spore data, and make the two Cleveland dotplots, is given below. The first two commands are used to access the data. The `par` command sets up the graphical window and the `mar` option controls the amount of white space around the individual panels. The `dotchart` command was discussed in Chapter 2.

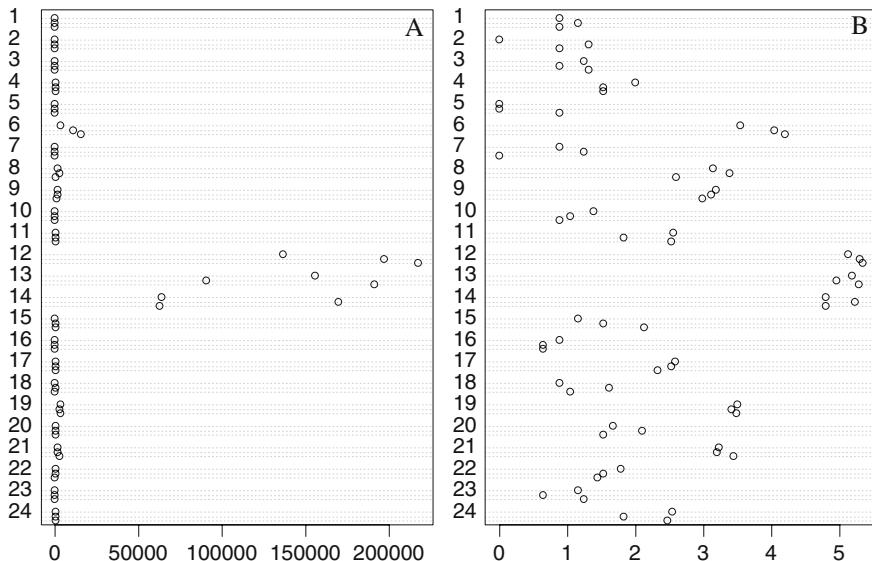


Fig. 19.2 **A:** Cleveland dotplot for the untransformed spores (densities) data. The data are grouped by hives. **B:** Cleveland dotplot for the log₁₀-transformed data. The vertical axes show the three observations per hive and the horizontal axes the values of the spores data

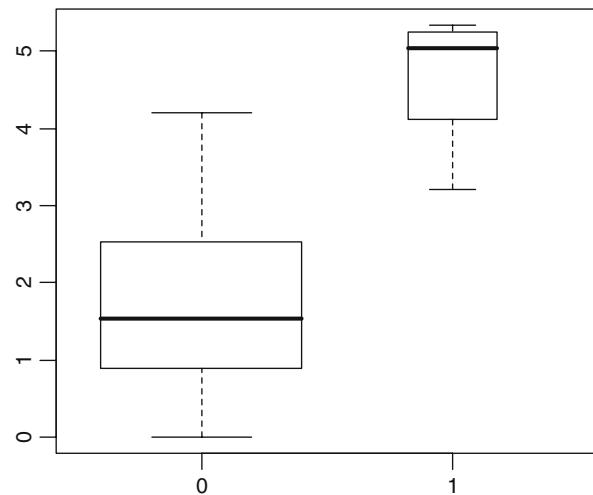
```
> library(AED); data(Bees)
> Bees$fhive <- factor(Bees$fhive)
> Bees$Lspobee <- log10(Bees$spobee + 1)
> op<- par(mfrow = c(1, 2), mar = c(3, 4, 1, 1))
> dotchart(Bees$spobee, groups = Bees$fhive)
> dotchart(Bees$Lspobee, groups = Bees$fhive)
> par(op)
```

Instead of using the Cleveland dotplot, we could have used a conditional boxplot. However, with only three values per hive, this would have been less useful.

The explanatory variable `Infection` quantifies the degree of infection (AFB), with values 0 (none), 1 (minor), 2 (moderate), and 3 (major). Although mixed effects modelling can cope with a certain degree of unbalanced data, in this case it may be better to convert the variable `Infection` in 0 (no infection) and 1 (infection is present) as there are only a few observations that have the value 2 or 3 for this variable. The R code to do this is

```
> Bees$Infection01 <- Bees$Infection
> Bees$Infection01[Bees$Infection01 > 0] <- 1
> Bees$fInfection01 <- factor(Bees$Infection01)
```

Fig. 19.3 Boxplot of log-transformed spores densities conditional on the variable fInfection01 (AFB). Note that there are considerably more observations with fInfection01 equal to 0. The width of a boxplot is proportional to sample size



All observations for the variable Infection that are larger than 0 are set to 1. After this transformation, 17% of its values are equal to 1 and 73% are 0.

A boxplot of spores conditional on Infection01 shows clear differences between the two levels (Fig. 19.3). The boxplot was made with the command

```
> boxplot(LSpobee ~ fInfection01, data = Bees,
           varwidth = TRUE)
```

Other graphical validation tools were also applied, for example, the coplot and xyplot, but no clear patterns were found. These graphs and R code are not presented here.

19.3 Analysis of the Data

The response variable is the log-transformed density of spores and the explanatory variables are infection (nominal with two classes) and number of bees. To investigate whether there is a hive effect, we first applied a linear regression model on the data. As explanatory variables we used infection and number of bees together with their interaction. The standardised residuals from this model were plotted against hive (Fig. 19.4) and show a clear pattern. In this graph, we want to see residuals that are scattered around zero, but in this case, we have various hives where all three residuals are above the zero line or all are below the zero line. This indicates there is within-hive correlation.

An option is to include hive as an explanatory variable. However, if we do this as a fixed term, paying the price of losing 23 degrees of freedom is rather high! And on top of that, the resulting model would only hold for these 24 hives. A logical

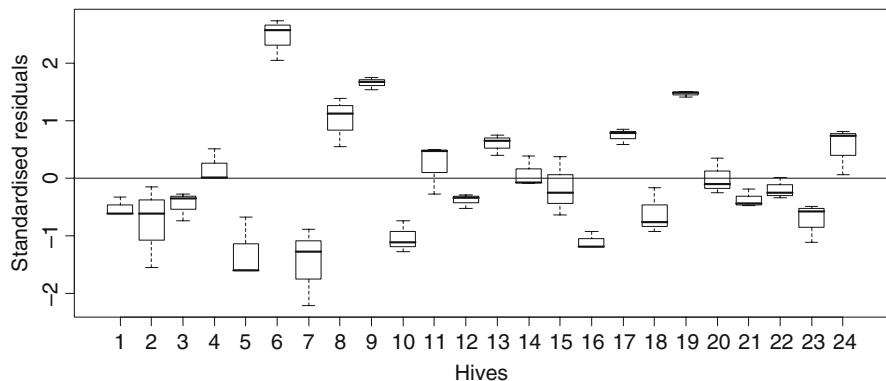


Fig. 19.4 Standardised residuals from the linear regression model where log transformed spores are modelled as a function of infection, number of bees, and their interaction

solution is to proceed with a random intercept model (Chapter 5). The advantages of such an approach are (i) it only requires one extra parameter (the variance for the random intercept), compared to the linear regression model that required 23 extra parameters; (ii) we can make a statement for hives in general and not only these 24; and (iii) as an extra bonus, it introduces a correlation structure between observations of the same hive.

The following R code was used to apply the linear regression model, extract the normalised residuals and produce the boxplot in Fig. 19.4. The `abline` command adds the horizontal line at zero.

```
> M1 <- lm(LSpobee ~ fInfection01 * BeesN, data = Bees)
> E1 <- rstandard(M1)
> plot(E1 ~ Bees$fHive, xlab = "Hives",
       ylab = "Standardised residuals")
> abline(0, 0)
```

Recall from Chapters 4 and 5 that the selection approach for linear mixed effects models should broadly follow a protocol consisting of 10 steps. In step 1, we start with a model that has as many explanatory variables as possible (in the fixed part of the model), then we find the optimal random structure (steps 2–6), the optimal fixed structure (steps 7–8), present the results of the optimal model using REML estimation (step 9), and finally, give an interpretation (step 10). We follow these same steps here.

Step 1 of the Protocol

Earlier in this chapter, we started with a model that contained all the explanatory variables and their interaction in the fixed part of the model. In this case, there are only two fixed explanatory variables.

Steps 2–6 of the Protocol

Starting with a random intercept model, we have

$$\begin{aligned} \text{LSpabee}_{ij} = & \alpha + \beta_1 \times \text{BeesN}_{ij} + \beta_2 \times \text{fInfection01}_{ij} \\ & + \beta_3 \times \text{BeesN}_{ij} \times \text{fInfection01}_{ij} + a_i + \varepsilon_{ij} \end{aligned}$$

In words, the log-transformed spores are modelled as an intercept (α), plus a linear ‘number of bees per hive’ effect (BeesN), an infection effect (fInfection01), the interaction between these two terms, a random intercept a_i that is assumed to be normally distributed with mean 0 and variance σ_a^2 , and something that is ‘real’ noise (ε_{ij}). The index i refers to hives ($i = 1, \dots, 24$) and j to the observation within a hive ($j = 1, \dots, 3$). The term ε_{ij} is the within-hive variation, and is assumed to be independently normally distributed with mean 0 and variance σ^2 .

We use the function `lme` from the R package `nlme` to fit the random intercept model in Equation (19.1). To assess whether the mixed effects model is better than the ordinary linear regression model, we need to refit the latter one using the `gls` function without the random intercept. The `anova` function can then be used to compare AICs or apply a likelihood ratio test. The required R code and output of the `anova` command are given below.

```
> library(nlme)
> M2<-gls(LSpabee ~ fInfection01 * BeesN, data = Bees)
> M3<-lme(LSpabee ~ fInfection01 * BeesN,
           random =~ 1 | fHive, data = Bees)
> anova(M2,M3)

Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M2     1 5 251.5938 262.6914 -120.79692
M3     2 6 175.0129 188.3299 -81.50643 1 vs 2 78.58097 <.0001
```

We can either use the AIC to select the optimal model or apply the likelihood ratio test. The AIC values indicate that the mixed model is preferred. The problem with the likelihood ratio test is that we are testing on the boundary (Chapter 5). The correct p -value is obtained by typing

```
> 0.5 * (1 - pchisq(78.58097, 1))
```

This is still smaller than 0.001; so both approaches favour the mixed model.

There are a few ways to extend the random part of the model. We can try a random intercept and slope model, and we can try using multiple variances. As to the first option, the BeesN effect may be different per hive and the same may hold for the fInfection01 effect. However, both options gave higher AICs. The R code for these models and model comparisons are given below.

```
> M4 <- lme(LSpabee ~ fInfection01 * BeesN,
           random =~ 1 + BeesN | fHive, data = Bees)
```

```
> M5 <- lme(LSpobee ~ fInfection01 * BeesN,
  random = ~ 1 + fInfection01 | fHive, data = Bees)
> anova(M2, M3, M4, M5)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
M2	1	5	251.5938	262.6914	-120.79692			
M3	2	6	175.0129	188.3299	-81.50643	1 vs 2	78.58097	<.0001
M4	3	8	178.8460	196.6020	-81.42299	2 vs 3	0.16689	0.9199
M5	4	8	177.7606	195.5167	-80.88032			

As extending the model with random slopes gives no improvement, we can look at an alternative of adding multiple variances for the residuals ε_{ij} . One option is to fit the model with and without multiple variances and compare them with the AIC or the likelihood ratio test. Another option is to plot the residuals of the model that is the best so far, the random intercept model in Equation (19.1), and see whether anything is wrong. We chose the second approach. The command `plot(M3, col = 1)` produces a plot of the residuals against fitted values for the random intercept model (Fig. 19.5). Note that there is some evidence of heterogeneity as the residual spread is slightly smaller for larger fitted values. These are actually the observations for which `Infection01` is equal to 1 (this can be seen by using colours or different symbols), which suggests extending the random intercept model in Equation (19.1) from $\varepsilon_{ij} \sim N(0, \sigma^2)$ to $\varepsilon_{ij} \sim N(0, \sigma_k^2)$, where $k = 1, 2$.

This means that we use a variance for the observations that have no infection and a different variance for the observations that have `Infection01 = 1`. Technically, the `varIdent` variance structure is used for this; see also Chapter 4. The AIC of this model (171.65) is slightly better than the random intercept model (175.01) in

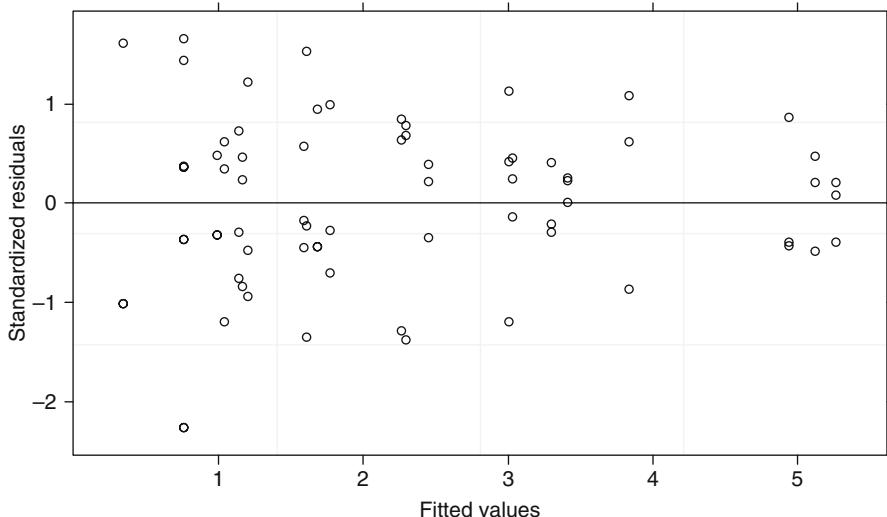


Fig. 19.5 Residuals versus fitted values for the mixed model

Equation (19.1), and the likelihood ratio test gives a p -value of 0.02, indicating that we have weak evidence to reject the null hypothesis that both variances are the same. The normalised residuals (not shown here) now look better.

The R code below fits the new model, and compares it with the random intercept model.

```
> M6 <- lme(LSpabee ~ fInfection01 * BeesN,
   random =~ 1 | fHive, data = Bees,
   weights = varIdent(form =~ 1 | fInfection01))
> anova(M3, M6)

Model df      AIC      BIC    logLik   Test L.Ratio p-value
M3     1  6 175.0129 188.3299 -81.50643
M6     2  7 171.6587 187.1952 -78.82933 1 vs 2  5.3542  0.0207
```

Steps 7 and 8 of the Protocol

We now continue with the seventh and eighth step of the protocol to find the optimal fixed structure for the selected random structure. This means that using our optimal random structure (random intercept plus two variances for ε_{ij}), we need to look at the optimal fixed structure. As discussed in Chapters 4 and 5, we can either do this using the t -statistics from the summary command, sequential F -tests using the anova command, or likelihood ratio tests of nested models. The first two approaches require REML estimation with the third approach needing ML estimation. We will use the last approach as the first two approaches can easily be carried out by the reader, and there is a higher degree for ‘confusion’ with the third approach.

In the first step, we need to apply the model with all terms and a model without the interaction. Note that we cannot drop any of the main terms yet. The update command is used to fit the model without the interaction term; see also Chapters 4 and 5.

```
> M7full <- lme(LSpabee ~ fInfection01 * BeesN,
   random =~ 1 | fHive, method = "ML", data = Bees
   weights = varIdent(form =~ 1 | fInfection01))
> M7sub <- update(M7full, .~. -fInfection01 : BeesN)
> anova(M7full, M7sub)

Model df      AIC      BIC    logLik   Test L.Ratio p-value
M7full     1  7 129.8792 145.8159 -57.93962
M7sub      2  6 128.4452 142.1052 -58.22262 1 vs 2  0.5660039  0.4519
```

The anova command gives $L = 0.56$ ($df = 1$) with $p = 0.45$, allowing us to drop the interaction term to give a model with two main terms. We can now either switch to approach one and use the t -statistics to assess the significance of these two main terms or we can be consistent and go on with the likelihood ratio testing approach. We prefer consistency. The following code reapplies the model, drops each of the main terms in turn, and then applies the likelihood ratio test.

```
> M8full <- lme(LSpobee ~ fInfection01 + BeesN,
  random =~ 1 | fHive, method = "ML", data = Bees,
  weights = varIdent(form =~ 1 | fInfection01))
> M8sub1 <- update(M8full, .~. -fInfection01)
> M8sub2 <- update(M8full, .~. -BeesN)
> anova(M8full, M8sub1)

      Model df      AIC      BIC    logLik   Test L.Ratio p-value
M8full     1  6 128.4452 142.1052  -58.22262
M8sub1     2  5 144.6700 156.0533  -67.33497 1 vs 2 18.22471  <.0001

> anova (M8full,M8sub2)

      Model df      AIC      BIC    logLik   Test L.Ratio p-value
M8full     1  6 128.4452 142.1052  -58.22262
M8sub2     2  5 129.3882 140.7715  -59.69408 1 vs 2 2.942923  0.0863
```

The two anova commands give $p < 0.001$ and $p = 0.08$, making the term beesN the least significant, and we continue without it. This leaves us with one final model comparison of the models with and without the term fInfection01. The following R code is used:

```
> M9full <- lme(LSpobee ~ fInfection01,
  random =~ 1 | fHive, method = "ML", data = Bees,
  weights = varIdent(form =~ 1 | fInfection01))
> M9sub1 <- update(M9full, .~. -fInfection01)
> anova (M9full, M9sub1)

      Model df      AIC      BIC    logLik   Test L.Ratio p-value
M9full     1  5 129.3882 140.7715  -59.69408
M9sub1     2  4 147.0532 156.1599  -69.52661 1 vs 2 19.66507  <.0001
```

The last anova command gives $L = 19.66$ ($df = 1, p < 0.0001$), indicating that infection is highly significant. So after a considerably amount of R coding, we end up with a model where only one fixed explanatory variable, infection, is highly significant.

Step 9 of the Protocol

In the last two steps of the protocol (9 and 10), we have to refit the model with REML, further validate and present the results, and then explain what it all means. The last part is the difficult bit and will be done in the discussion. The first part is easy:

```
> Mfinal <- lme(LSpobee ~ fInfection01,
  random =~ 1 | fHive, data = Bees, method="REML",
  weights = varIdent(form =~ 1 | fInfection01))
> summary(Mfinal)
```

```

Linear mixed-effects model fit by REML
Data: Bees
      AIC      BIC      logLik
 130.1747 141.4171 -60.08733

Random effects:
Formula: ~1 | fHive
          (Intercept) Residual
StdDev:  0.9892908 0.3615819

Variance function:
Structure: Different standard deviations per stratum
Formula: ~1 | fInfection01
Parameter estimates:
          0         1
1.000000 0.473795

Fixed effects: LSpobee ~ fInfection01
                Value Std.Error DF t-value p-value
(Intercept) 1.757273 0.2260837 48 7.772666     0
fInfection01 2.902090 0.5461078 22 5.314135     0

Correlation:
          (Intr)
fInfection01 -0.414

Standardized Within-Group Residuals:
      Min        Q1        Med        Q3        Max
-2.1548732 -0.6068385  0.2019003  0.5621671  1.6855583

Number of Observations: 72
Number of Groups: 24

```

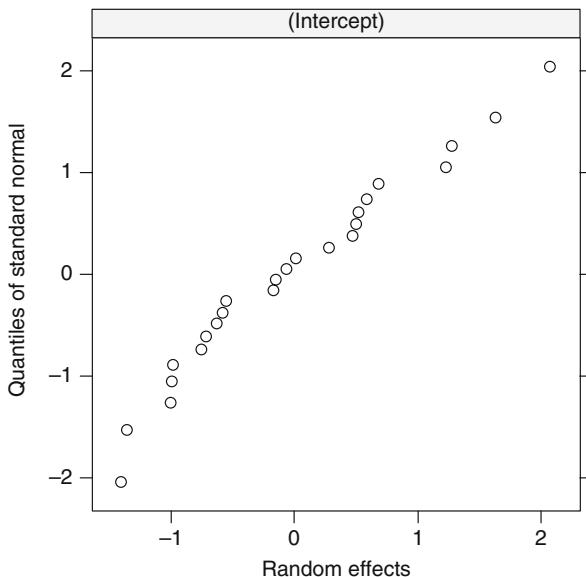
Let us to summarise all this information. The optimal model is given by

$$LSpobee_{ij} = 1.75 + 2.90 \times fInfection01_{ij} + a_i + \varepsilon_{ij}$$

where $a_i \sim N(0, 0.98^2)$. For the within-hive residuals, we have $\varepsilon_{ij} \sim N(0, 0.36^2)$ if the observation has no disease ($Infection01 = 0$) and $\varepsilon_{ij} \sim N(0, 0.36^2 \times 0.47^2)$ if it has a disease ($Infection01 = 1$). If an observation has no diseases, then the expected density of spores is 1.75 on the logarithmic scale. If it has a disease, then the expected density is $1.75 + 2.90 = 4.65$. Depending on the hive, there is a random variation on both expected values. This is due to the random intercept, and 95% of its values are between -1.96×0.36 and 1.96×0.36 .

Finally, we inspect the residuals of the optimal model. This should actually be done in steps 7 and 8, but because we want to do this for the REML estimates, we do it here. We need to inspect the optimal model for homogeneity of the residuals ε_{ij} .

Fig. 19.6 QQ-plot of the mixed effects model
Mfinal



We have already discussed how to do this using the command `plot(Mfinal)`. Results are not presented here, but we can safely say they indicate homogeneity. We can also assume normality of these residuals. This can be verified with `qqnorm(Mfinal)`. It produces a QQ-plot of the normalised residuals. Results are not presented here, but normality is a reasonable conclusion in this case. Finally, we need to verify the normality assumption for the random effects. Use the R command `qqnorm(Mfinal, ~ranef(.) , col = 1)`, and again, normality seems a reasonable conclusion (Fig. 19.6).

Another useful command is `intervals(Mfinal)`. It shows the approximate 95% confidence bands of the parameters and random variances.

Approximate 95% confidence intervals

Fixed effects:

	lower	est.	upper
(Intercept)	1.302701	1.757273	2.211845
fInfection011	1.769532	2.902090	4.034648
attr(,"label")			
[1] "Fixed effects:"			

Random Effects:

	lower	est.	upper
sd((Intercept))	0.7259948	0.9892908	1.348076

```
Variance function:
      lower     est.     upper
1 0.2770579 0.473795 0.8102339
attr(,"label")
[1] "Variance function:

Within-group standard error:
      lower     est.     upper
0.2904009 0.3615819 0.4502102
```

We have now finished steps 1–9 of the protocol and we discuss the interpretation of the model in the next section.

19.4 Discussion

In this chapter, we applied linear mixed effects modelling because the data are nested (three observations per hive). The model showed that there is a significant disease effect on the spore density data. The intraclass correlation is $0.98^2/(0.98^2 + 0.36^2) = 0.88$ if a hive has no disease and $0.98^2/(0.98^2 + 0.36^2 \times 0.47^2) = 0.97$ if a hive has the disease. This is rather high, and means that the effective sample size is considerably smaller than $3 \times 24 = 72$ (Chapter 5). We might as well take one sample per hive and sample more hives.

If the number of spores are analysed instead of density, we can use generalised estimation equations with a Poisson distribution (Chapter 9) or generalised linear mixed modelling with a Poisson distribution (Chapter 13).

19.5 What to Write in a Paper

A paper based on the results presented in this chapter should include a short description of the problem (introduction) and the set up of the experiment (methods). It will need to justify the use of the logarithmic transformation on spores densities and the use of mixed effects modelling. You should also outline the protocol for model selection, and in the results section, mention how you got to the final model. There is no need to present all the R code or results of intermediate models. You may want to include one graph showing homogeneity of the residuals. You should also present the estimated parameters, standard errors, *t*-values, and *p*-values of the optimal model. Warn the reader that the data are unbalanced (not many observations with a disease); so care is needed with the interpretation.

Acknowledgments We would like to thank Fernando Rodriguez, beekeeper from Buenos Aires Province and Sergio Ruffinengo for his collaboration in this project and also MalenaSabatino for the honey bee photography.

Chapter 20

Three-Way Nested Data for Age Determination Techniques Applied to Cetaceans

E.N. Ieno, P.L. Luque, G.J. Pierce, A.F. Zuur, M.B. Santos, N.J. Walker,
A.A. Saveliev, and G.M. Smith

20.1 Introduction

In the previous case study, we showed how multiple samples from bacteria in honey bees from the same hive gave a nested data structure, and mixed modelling techniques were applied to allow for correlations between observations from the same hive. The bee data provided an example of two-way nested data, and the underlying theory for this was discussed in Chapter 5. In this chapter, we go one step further and use three-way nested data, which extends the two-way approach discussed in Chapter 5. The underlying theory builds on the approach used for two-way data, and we recommend reading Chapter 5 before starting this chapter as we assume familiarity with the theory, model selection, and R code for two-way nested data.

We use a subset of the data analysed in Luque (2008), who compared the results from three staining methods to determine the age of cetaceans stranded in Spain and Scotland. The data are nested in the sense that samples derive from multiple species, and from each species, we have various specimens (individual animals). From each specimen, several teeth were sectioned and tooth sections were stained using three staining methods (the Mayer Haematoxylin, Ehrlrich Haematoxylin, and Toluidine Blue methods), giving three age estimates from each tooth. A diagram of the nested structure is given in Fig. 20.1. The three age observations per specimen (obtained by the three staining methods) are likely to be correlated, but we may also expect correlation between age readings within the same species (if, for example, different species have different lifespans and/or different age classes tend to become stranded and thus become the source of samples). The response variable is the estimated age of the animal. Available explanatory variables are sex (male or female), location of stranding (Scotland or Spain), and stain (Mayer Haematoxylin, Ehrlrich Haematoxylin, and Toluidine Blue).

In Chapter 4 of West et al. (2006), a three-way nested data set on mathematic scores for students within multiple classes and multiple schools is analysed. From a

E.N. Ieno (✉)
Highland Statistics Ltd., Newburgh, AB41 6FN, United Kingdom

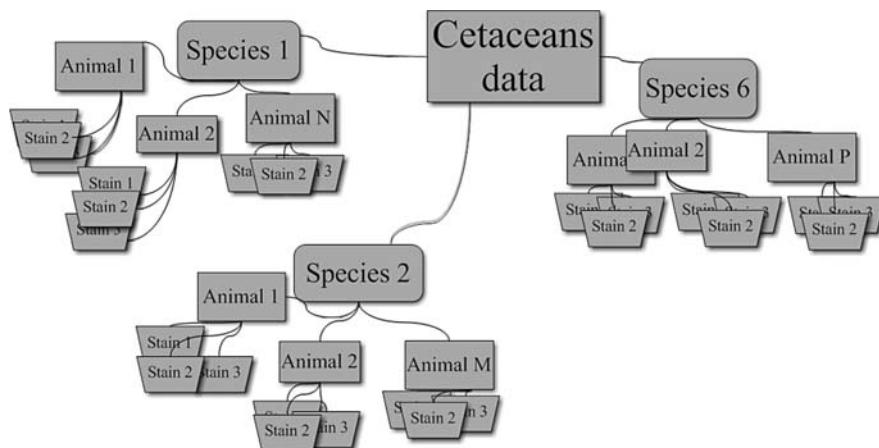


Fig. 20.1 Sketch of the nested structure of the data. Six cetacean species were sampled. These were *Delphinus delphis*, *Lagenorhynchus acutus*, *Phocoena phocoena*, *Stenella coeruleoalba*, *Stenella frontalis*, and *Tursiops truncatus*. For each species, various specimens (animals) were available. The number of specimens per species range between 3 and 25. From each specimen, three estimated age readings were obtained by the three staining methods (labeled as 1, 2, and 3 in the graph)

statistical point of view, there are not many differences between their classroom example and our cetacean data set. In fact, we will closely follow their steps. The only difference is that West et al. (2006) used two different model selection approaches; (i) the step-down approach, which was presented as our protocol with steps 1–10, and (ii) a step-up approach. The classroom data are analysed with the step-up approach. For the cetacean data, we will follow our familiar step-down approach.

20.2 Data Exploration

The first question to ask with nested data is how much variation is there between specimens and between species? Figure 20.2 shows a boxplot of age conditional on specimen. Recall that we have three observations per specimen. The graph shows that we have a large between-specimen variation, which means we probably need to use ‘animal’ as a random effect. The same graph was made for species (Fig. 20.3) and shows there is considerably less between-species variation. This should not be too surprising as each animal has only one true age, but the samples of stranded animals for each species should include the range of age classes present in the populations. Even if one species tends to be longer-lived than another, there will inevitably be a considerable overlap in ages present.

We also made boxplots of age conditional on sex, age conditional on location, and age conditional on staining method. Results are not shown here, but they

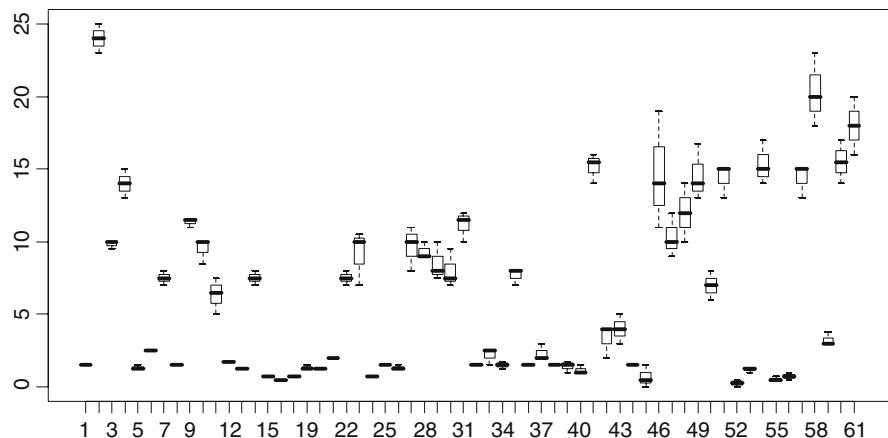


Fig. 20.2 Boxplot of age conditional on animal. Each boxplot consists of three observations from the same animal. Not all numbers are plotted along the horizontal axis due to limited space

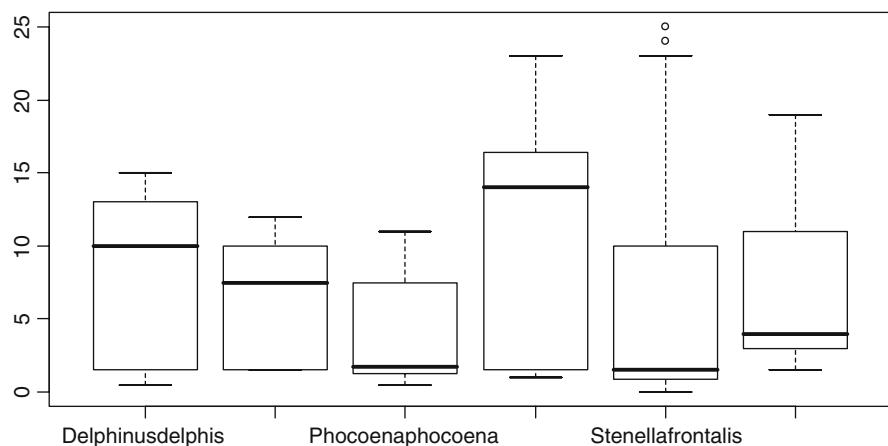


Fig. 20.3 Boxplot of age conditional of species. There is considerably less between-species variation compared to between-animal variation

indicate that the variation in ages recorded in Scotland is considerably less than Spain, indicating we may need to use different variances per country. There are also three observations with undetermined sex, and to allow for interactions between sex, location, and age determination, we removed these observations.

The following R code was used to access the data, make the two boxplots, and remove the three observations where sex was undetermined. By now, you should be familiar with this code. The object *Cetaceans2* contains the male and female data (the class *unknown* was dropped).

```
> library(AED); data(Cetaceans)
> Cetaceans$fSpecies <- factor(Cetaceans$Species)
> Cetaceans$fDolphinID <- factor(Cetaceans$DolphinID)
> boxplot(Age ~ fSpecies, data = Cetaceans)
> boxplot(Age ~ fDolphinID), data = Cetaceans)
> I <- Cetaceans$Sex==0
> Cetaceans2 <- Cetaceans[!I, ]
```

20.3 Data Analysis

The starting point for the analysis is a model of the form

$$\text{Age}_{ijk} = \text{fixed part}_{ijk} + \text{random part}_{ijk} \quad (20.1)$$

The variable Age_{ijk} is the age of observation i in animal j of species k . The index k runs from 1 to 6 and i from 1 to 3. The number of animals per species differs. We start discussing the fixed part of the model and then the random part. Recall from Chapters 4 and 5 that the protocol dictates that we start with a model that contains as many fixed explanatory variables as possible. In this case, we have three nominal explanatory variables. We therefore start with a model containing sex, stain, and location as main terms, all two-way interactions, and the three-way interaction. Hence, the fixed part consists of

$$\begin{aligned} & \text{Sex}_{ijk} + \text{Stain}_{ijk} + \text{Location}_{ijk} + \text{Sex}_{ijk} \times \text{Stain}_{ijk} + \text{Sex}_{ijk} \times \text{Location}_{ijk} + \\ & \text{Stain}_{ijk} \times \text{Location}_{ijk} + \text{Sex}_{ijk} \times \text{Stain}_{ijk} \times \text{Location}_{ijk} \end{aligned}$$

This model is fitted with the `gls` function to serve as a reference model. The following code was used for this.

```
> library(nlme)
> Cetaceans2$fSex <- factor(Cetaceans2$Sex)
> Cetaceans2$fLocation <- factor(Cetaceans2$Location)
> Cetaceans2$fStain <- factor(Cetaceans2$Stain)
> f1 <- formula(Age ~ fSex * fStain * fLocation)
> M1 <- gls(f1, method = "REML", data = Cetaceans2)
```

We can now go to step 2 of the analysis. The random effect ‘animal’ is nested within the random effect ‘species’. Just as West et al. (2006), we argue that if the random effect ‘animal’ is included in the model, then the random effect ‘species’ should also be included in the model. Making our starting point for the random part,

$$a_k + a_{j|k} + \varepsilon_{ijk}$$

The term ε_{ijk} is the unexplained error and represents the within-animal variation. It is assumed to be normally distributed with mean 0 and variance σ^2 . However, the data exploration indicated that there may be different spread per location and we should be prepared at some stage to test whether multiple variances are needed per location. But to avoid too many steps at once, we will wait until we reach steps 3–5 of the analysis before considering this in any detail.

Recall that the index k refers to species k . We assume that a_k is normally distributed with mean 0 and variance $\sigma_{\text{species}}^2$. The term a_k is a random intercept and allows for variation between the species. The amount of variation is determined by $\sigma_{\text{species}}^2$. The term a_{jk} looks intimidating, but represents the variation between animals (index j) of the same species (index k). We assume it is normally distributed with mean 0 and variance σ_{animal}^2 . Summarising, a_k allows for variation between the species and a_{jk} for the variation between animals within the same species.

Therefore, our starting model contains a sex, location, and stain effect as well as all their interactions, and we also use random intercepts that model between-species variation and between-animal variation within the species.

As part of this analysis (step 2), the first model comparison is between the model with the two random effects a_k and a_{jk} and a model without them. Recall that these random effects are nested. If the between-animal variation is important, then we should use both random effects. So, we will not test whether the random effect a_k on its own is important. The following code applies the model with both random effects and compares the model with and without the random effects using the `anova` command.

```
> M2 <- lme(f1, random = ~1 | fSpecies / fDolphinID,
             data = Cetaceans2, method = "REML")
> anova(M1, M2)
```

It is important that you define the variables `fSpecies` and `fDolphinID` as factors before the `lme` command or R will give an error message. The output of the `anova` command is as follows:

Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
M1	1 13	1101.4488	1141.8261	-537.7244			
M2	2 15	740.3277	786.9168	-355.1638	1 vs 2	365.1212	<.0001

The AIC indicates that the model with the two random effects is considerably better. The likelihood ratio statistic is $L = 365.12$, and the cited p -value indicates that we can reject the null hypothesis $H_0: \sigma_{\text{animal}} = 0$ in favour of the alternative $H_1: \sigma_{\text{animal}} > 0$. However, note that we are testing on the boundary, and therefore, the cited critical p -value should be multiplied with 0.5; see also Chapter 5 or Chapter 4 in West et al. (2006). Even after applying this correction, we still come to the same conclusion that we need the two random intercepts.

We now have two options. We can either apply a model validation, check for homogeneity (especially plotting residuals versus location), or extend the model by allowing for multiple variance based on location and see whether it improves the model. The motivation for the last approach is because the data exploration showed a clear difference in spread per location. Recall from Chapter 4 that adding such a variance structure extends the model to

$$\varepsilon_{ijk} \sim N(0, \sigma_s^2)$$

The index s refers to the two locations, allowing the residuals from the two locations to have a different spread. Based on the data exploration, we decided to include the multiple variance structure and see whether it improved the model. Some might argue that this variance structure should have been used in the starting model, but there are two reasons for not doing this; firstly because we prefer to start as simple as possible and secondly the explanatory variables could have explained the differences in spread. The following R code was used to extend the model.

```
> M3 <- lme(f1, random =~ 1 | fSpecies / fDolphinID,
            weights = varIdent(form =~ 1 | fLocation),
            data = Cetaceans2)
```

The only new bit of code is the weights option with the `varIdent` variance structure (see also Chapter 4). The `anova` command shows that the AIC of this model is 733.01 and the likelihood ratio statistic is $L = 9.30$ ($df = 1, p = 0.002$), making this the best model so far. We can now proceed to steps 7–9 of the analysis to find the optimal fixed structure for our selected random structure.

To find the optimal model in terms of the fixed explanatory variables, we can either use the t -statistics, sequential F -tests, or likelihood ratio tests. In this instance, as we have factors with more than two levels (stain), we decided to use the third option. This part of the analysis was described earlier in Chapter 5, and we assume the reader is familiar with the tedious repetitive process of fitting a full model dropping all allowable terms in turn, applying likelihood ratio tests of nested models dropping the least significant term, and repeating the whole process until all terms are significant.

We first fitted a model with all terms (main terms: all two-way interactions and the three-way interaction) and then a model without the three-way interaction. Both models were fitted with maximum likelihood estimation (ML). The likelihood ratio test indicated that we could drop the three-way interaction ($L = 5.05, df = 2, p = 0.07$). The process then continued by dropping each of the three two-way interaction terms in turn and identified the least significant with the likelihood ratio test. The first interaction term to go out was the sex \times location term ($L = 0.35, df = 1, p = 0.55$), followed by the sex \times stain interaction ($L = 1.5, df = 2, p = 0.46$), and finally, sex as a main term was dropped ($L = 0.68, df = 1, p = 0.40$) as it was not included in the remaining two-way interaction term. At this point, the fixed part of the model contained stain, location, and the stain \times location interaction. Dropping

the interaction gave $L = 19.14$ ($df = 2, p < 0.001$), giving the optimal model in terms of fixed terms. In words, it is given by

$$\text{Age}_{ijk} = \text{Stain}_{ijk} + \text{Location}_{ijk} + \text{Stain}_{ijk} \times \text{Location}_{ijk} + a_k + a_{j|k} + \varepsilon_{ijk}$$

We refitted the model with REML and applied a model validation. There are no problems with homogeneity. The numerical output of the model is obtained with the `summary` command:

```
> options(digits=4)
> summary(M3)

Linear mixed-effects model fit by REML
Data: Cetaceans2
AIC   BIC logLik
734.7 766.1 -357.4

Random effects:
Formula: ~1 | fSpecies
          (Intercept)
StdDev:      1.285

Formula: ~1 | fDolphinID %in% fSpecies
          (Intercept) Residual
StdDev:      5.503   0.6496

Variance function:
Structure: Different standard deviations per stratum
Formula: ~1 | fLocation
Parameter estimates:
Scotland   Spain
1.000     1.596
Fixed effects: list(f1)
                Value Std.Error DF t-value p-value
(Intercept) 4.050    1.3502 114  3.000  0.0033
fStainMayer 0.398    0.1624 114  2.454  0.0157
fStainToluidine 0.227    0.1624 114  1.395  0.1657
fLocationSpain 3.928    1.8085  52  2.172  0.0345
fStainMayer:fLocationSpain 1.481    0.3255 114  4.551  0.0000
fStainToluidine:fLocationSpain 0.672    0.3255 114  2.063  0.0414

Correlation:
              (Intr) fStnMy fStnTl fLctns fsm:LS
fStainMayer   -0.060
fStainToluidine -0.060  0.500
fLocationSpain -0.718  0.045  0.045
fStainMayer:fLocationSpain  0.030 -0.499 -0.249 -0.090
fStainToluidine:fLocationSpain  0.030 -0.249 -0.499 -0.090  0.500

Standardized Within-Group Residuals:
Min       Q1       Med      Q3      Max
-2.97802 -0.31902 -0.04765  0.30647  3.33243
```

Number of Observations: 177

Number of Groups:

fSpecies	fDolphinID	%in%	fSpecies
6			59

The Mayer staining method when applied to samples from Spain is significantly different from the Ehrlrich (baseline) method when applied to sample from Scotland (baseline). The Toluidine method is also significantly different from the Ehrlrich method, but a p -value of 0.04 is not really that impressive. It may be an option to change the baseline and see whether the Mayer and Toluidine methods differ from each other. Note that the main term location makes a major contribution for Spain to the fitted values.

The summary command also gives information on the random terms. The estimated values for σ , σ_{animal} , and σ_{species} are 0.64, 5.50, and 1.18, respectively. The multiplication factors for the different standard deviations per stratum for location are 1 for Scotland and 1.59 for Spain. The residual spread in Spain is therefore considerably larger than it is in Scotland. This means that more of the age variation is explained by the available explanatory variables in Scotland than in Spain.

20.3.1 Intraclass Correlations

We now discuss the interpretation of the random intercepts. The output above shows that the random effect a_k , representing the between species variation, is $N(0, 1.28^2)$, the random effect $a_{j|k}$, representing the between animal variation in the same species, is $N(0, 5.50^2)$ for observations from Scotland, the random noise ε_{ijk} is $N(0, 0.64^2)$, and for observations from Spain, the random noise ε_{ijk} is $N(0, 1.02^2)$. The value of 1.02 is obtained by multiplying 0.64 and 1.59. The values can be used to calculate the intraclass correlation at the species level (ICC_{species}) and at the animal level (ICC_{animal}). The formulae were taken from West et al. (2006) and are as follows:

$$ICC_{\text{species}} = \frac{\sigma_{\text{species}}^2}{\sigma_{\text{species}}^2 + \sigma_{\text{animal}}^2 + \sigma^2}$$

$$ICC_{\text{animal}} = \frac{\sigma_{\text{species}}^2 + \sigma_{\text{animal}}^2}{\sigma_{\text{species}}^2 + \sigma_{\text{animal}}^2 + \sigma^2}$$

The only difference is that we need to calculate these ICCs for both Scotland and for Spain as we have two σ s. The actual calculations are just a matter of filling in the values and give the ICCs for Scotland: $ICC_{\text{species}} = 0.05$ and $ICC_{\text{animal}} = 0.98$ and for Spain: $ICC_{\text{species}} = 0.05$ and $ICC_{\text{animal}} = 0.96$. Hence, there is massive correlation between the three observations of the same animal for data of both countries, as we would of course expect for this dataset. The correlation between animals of the same species is low. Just as West et al. (2006), we can show the implications of these ICC

values. Suppose we have three animals from the same species. The model we fitted implies the following marginal correlation structure for the Spanish data.

	1	2	3	4	5	6	7	8	9
1	1	0.96	0.96	0.05	0.05	0.05	0.05	0.05	0.05
2		1	0.96	0.05	0.05	0.05	0.05	0.05	0.05
3			1	0.05	0.05	0.05	0.05	0.05	0.05
4				1	0.96	0.96	0.05	0.05	0.05
5					1	0.96	0.05	0.05	0.05
6						1	0.05	0.05	0.05
7							1	0.96	0.96
8								1	0.96
9									1

The values 1–3 are teeth from the same animal. The correlation between these three observations is very high (0.96). The same holds for observations 4–6; these are also from the same, albeit a different, animal. These are also highly correlated. And the same holds for observations 7–9, which are all from a third animal. However, the correlation between two age observations from different animals, say 1 and 4, is low (0.05). The lower part of the correlation matrix is identical to the upper part.

20.4 Discussion

Some of the results displayed above are obvious from the nature of the data. We expect the three staining methods to give similar results on age for the same animal, and we would expect a fairly wide overlap in the ages of animals available for different species. However the overlap between age ranges for the different species is not complete, as for example, common dolphins live longer than harbour porpoises.

It is less obvious what the ‘country’ effect means (and why ages should be more variable in one country than another) as all the teeth were prepared and assessed by the same team. There was a different range of species among strandings in the two countries and although we restricted the analysis to the three most common species, their relative abundance differs between countries. So there could be some confounding of country and species effects. There may also have been differences in the effectiveness of staining due to different storage procedures for teeth used by the local sampling programmes (in Scotland, teeth are normally stored in alcohol, whereas the Spanish samples had been stored frozen).

However, one important effect we have ignored is that the variation in age readings probably depends on age: teeth of older animals are more difficult to interpret because the later incremental growth layers are closer together. Spain had a higher proportion of common dolphins in the sample (as compared to dominance of porpoises in the Scottish sample) meaning the Spanish sample was biased towards

older, larger animals. Thus, if we had included length (highly correlated with age but independent of the age measurements) as an explanatory variable, the country effect may have disappeared.

The objective of the original study was to compare the efficacy of several staining methods to prepare dolphin teeth used to determine age. The heterogeneity of the available teeth samples presented challenges that could not be easily overcome without the availability of mixed effects modelling. The availability of mixed effects modelling should not be considered a replacement for good sampling design, but it does offer a solution to problems created by opportunistic sampling, where these are the only data available.

20.5 What to Write in a Paper

We can be very short: The same as we suggested in Chapter 19. Present boxplots to emphasise the large between-animal variation, discuss the need for mixed effects modelling, explain the model selection approach and present the results, discuss the model validation, and explain what it means in terms of biology.

Chapter 21

GLMM Applied on the Spatial Distribution of Koalas in a Fragmented Landscape

J.R. Rhodes, C.A. McAlpine, A.F. Zuur, G.M. Smith, and E.N. Ieno

21.1 Introduction

Predicting the spatial distribution of wildlife populations is an important component of the development of management strategies for their conservation. Landscape structure and composition are important determinants of where species occur and the viability of their populations. In particular, the amount of suitable habitat and its level of fragmentation (i.e. how broken apart it is) in a landscape can be important determinants of the distribution and abundance of biological populations (Hanski, 1998; Fahrig, 2003). In addition to the role of habitat, anthropogenic impacts, such as wildlife mortality on roads or direct wildlife-human conflict, can also have large impacts on the distribution and abundance of a species (Fahrig et al., 1995; Woodroffe and Ginsberg, 1998; Naves et al., 2003). Therefore, if we are to manage landscapes to successfully conserve wildlife, it is important that we understand the role of these landscape processes in determining their distributions.

In this chapter, we will model the impact of landscape pattern on the distribution of koalas (*Phascolarctos cinereus*, Fig. 21.1) in a landscape in eastern Australia. Koalas are folivorous arboreal marsupials restricted to the eucalypt forests of eastern and southeastern Australia. Across their geographic range, they feed on a wide range of tree species from the genus *Eucalyptus*, but mostly prefer only a few species in any particular area (Hindell and Lee, 1987; Phillips and Callaghan, 2000; Phillips et al., 2000). Koala habitat generally consists of forest associations containing their preferred tree species, although other factors, such as tree size, water availability, and nutrient status, can also be important determinants of habitat quality (Moore et al., 2004; Matthews et al., 2007). Since European settlement, koalas have suffered declines in their abundance and distribution due to clearing and degradation of eucalypt forests, together with historical hunting, disease, bushfire, drought, and urbanisation (ANZECC, 1998; Melzer et al., 2000; Phillips, 2000).

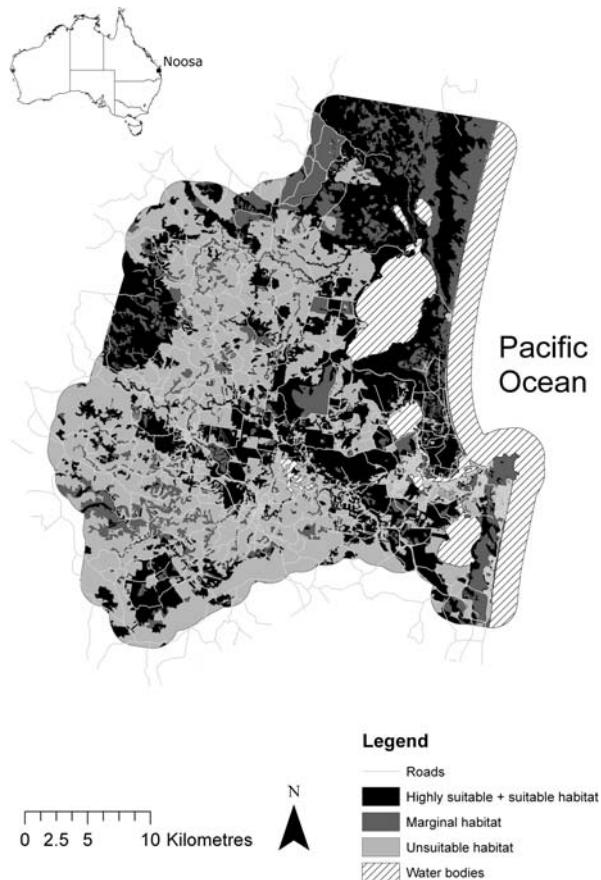
J.R. Rhodes (✉)

The University of Queensland, School of Geography, Planning and Architecture, Brisbane, QLD 4072, Australia

Fig. 21.1 Young koala (photo by Dick Marks, Australian Koala Foundation. www.savethekoala.com)



Fig. 21.2 Map of the study area (Noosa Local Government Area) showing the distribution of koala habitat and the location of roads (Australian Koala Foundation unpublished data)



The study area we consider for this chapter is the Noosa Local Government Area (LGA) in southeast Queensland, Australia (Fig. 21.2). Noosa has a subtropical coastal climate with native vegetation ranging from coastal heath to wet and dry eucalypt forests and subtropical rainforests. Over 50% of the original eucalypt forests have been cleared for farming and urban development (Seabrook et al., 2003). Koalas are, therefore, threatened by the loss and fragmentation of their habitat and by threats associated with urbanisation, such as cars and domestic dogs. To allow successful management strategies to be developed, it is important for conservation planners to be able to quantify the impact of these threats on koala distributions in the area.

We will use generalised linear mixed effects models (GLMM) to model the distribution of koalas using data on their presence and absence at sites located across the study area. We also take a multi-scale approach in the sense that our explanatory variables will be landscape characteristics measured at different landscape extents. The different landscape extents will be chosen to represent those scales thought to be most relevant for koala population dynamics, and hence their distributions. The chapter concentrates on dealing with collinearity and spatial auto-correlation for these types of landscape models. In addition, we present an information-theoretic approach to model selection, which allows us to assess both model and parameter uncertainty. We finish with a discussion on the implications of the results for koala conservation and what should be included in a scientific paper.

21.2 The Data

The data presented are based on surveys that were conducted to determine koala presence or absence at 300 locations in Noosa. This formed part of a larger study investigating the role of landscape change on koala distributions across eastern Australia (McAlpine et al., 2006; Rhodes et al., 2006). Using a form of stratified random sampling (McKay et al., 1979; Thompson, 1992) 100 sites were first located across the Noosa LGA. Then, within each site, three subsites were located 100 m apart. At each subsite, the presence or absence of koalas was then determined using standardised searches for koala faecal pellets around the bases of trees (as in Phillips and Callaghan, 2000). Previous work has identified the koala's preferred tree species in Noosa and these have been classified into primary, secondary, and supplementary species (Australian Koala Foundation (AKF) unpublished data). At each subsite, the percentage of trees that were primary and secondary species was recorded. Finally, the distribution of koala habitat (classified into highly suitable, suitable, marginal, and unsuitable habitat) and the location of paved roads were mapped in a geographical information system (AKF unpublished data, Fig. 21.2). The data on the presence/absence of koalas will be the response variable for our analysis, while the data on preferred tree species, habitat and roads will form the basis of the explanatory variables.

The data set can be accessed in R using the following code:

```
> library (AED); data (Koalas)
```

The resulting data frame contains a row for each subsite. The first two columns are the site and subsite ID numbers, the next two columns are the eastings and northings of the location of each subsite (in AMG ADG 1966 coordinates), the fifth column indicates whether koala pellets were found at the subsite (= 1) or not found at the subsite (= 0), and the remaining columns are the explanatory variables associated with each subsite (Table 21.1).

The explanatory variables were chosen to represent characteristics of the landscape considered likely to be important determinants of the distribution of koalas. The variables can be split into those characterising habitat at the site-scale, and those characterising habitat and human impacts at broader landscape-scales (i.e., within 1, 2.5, or 5 km buffers around each subsite). The site-scale habitat variables (`pprim_sssite` and `psec_sssite`) measure the percentage of primary and secondary tree species at each subsite and reflect resource availability at this scale. Two of the landscape-scale variables (`pss` and `pm`) measure the percentage of the landscape, within each buffer, that is highly suitable plus suitable habitat and marginal

Table 21.1 Description of the explanatory variables

Variable name	Description	Detail description
<code>pprim_sssite</code>	Resources available at site-scale	Percentage of trees in each subsite that are primary tree species
<code>psec_sssite</code>	Resources available at site-scale	Percentage of trees in each subsite that are secondary tree species
<code>phss_1km</code> <code>phss_2.5km</code> <code>phss_5km</code>	Habitat available at landscape-scale	Percentage of the landscape within 1, 2.5, and 5km, respectively, of each subsite that is highly suitable plus suitable habitat
<code>pm_1km</code> <code>pm_2.5km</code> <code>pm_5km</code>	Habitat available at landscape-scale	Percentage of the landscape within 1, 2.5, and 5 km, respectively, of each subsite that is marginal habitat
<code>pdens_1km</code> <code>pdens_2.5km</code> <code>pdens_5km</code>	Landscape fragmentation	Density (patches/100 ha) of habitat patches, consisting of highly suitable plus suitable plus marginal habitat, in the landscape within 1, 2.5, and 5 km, respectively, of each subsite
<code>edens_1km</code> <code>edens_2.5km</code> <code>edens_5km</code>	Landscape fragmentation	Density (m/ha) of habitat patch edges, consisting of highly suitable plus suitable plus marginal habitat, in the landscape within 1, 2.5, and 5 km, respectively, of each subsite
<code>rdens_1km</code> <code>rdens_2.5km</code> <code>rdens_5km</code>	Human impact at landscape-scale	Density (m/ha) of paved roads within 1, 2.5, and 5 km, respectively, of each subsite

habitat, respectively. These variables represent the amount of habitat resources available at the landscape-scale. Two of the landscape-scale variables (`pdens` and `edens`) measure the density of habitat patches and the density of habitat edges within each buffer, respectively. These variables represent the level of landscape fragmentation; patch density and edge density both tend to increase as habitat becomes more fragmented. Finally, one of the landscape-scale variables (`rdens`) measures the density of roads in each buffer and represents the level of human impact due to koala mortality of roads and general urbanisation.

21.3 Data Exploration and Preliminary Analysis

Two important issues to consider before building regression models of species' distributions are whether there is high collinearity between the explanatory variables and whether spatial auto-correlation between data points is likely to be an important factor. High collinearity can result in coefficient estimates that are difficult to interpret as independent effects and/or have high standard errors (Neter et al., 1990; Graham, 2003). Positive spatial auto-correlation violates the usual assumption of independence between data points and leads to the underestimation of standard errors, and elevated type I errors, if not accounted for (Legendre, 1993). Collinearity between explanatory variables and spatial auto-correlation are commonly encountered when using observational data to construct regression models of species' distributions. For both these issues, we examine whether they are likely to be a problem for the analysis of our dataset and then discuss how they can be addressed.

21.3.1 Collinearity

A simple first step for identifying collinearity is to look at the pairwise correlations between explanatory variables. We can generate a matrix of pairwise correlations between the explanatory variables in our dataset using the following code:

```
> cor(Koalas[, 6:22], method = "spearman")
```

This outputs a matrix of the Spearman rank correlations (results are not given here as it is too large). We have used the Spearman rank correlation coefficient, rather than the Pearson correlation coefficient because the Spearman rank correlation makes no assumptions about linearity in the relationship between the two variables (Zar, 1996). One could also use the `pairs` command to view pairwise plots of the variables. Booth et al. (1994) suggest that correlations between pairs of variables with magnitudes greater than ± 0.5 indicate high collinearity, and we use this rough rule-of-thumb here.

The first thing you notice from the correlation matrix is that the landscape variables measuring the same characteristic at different landscape extents tend to be highly positively correlated. For example, `phss_5km`, `phss_2.5km`, and

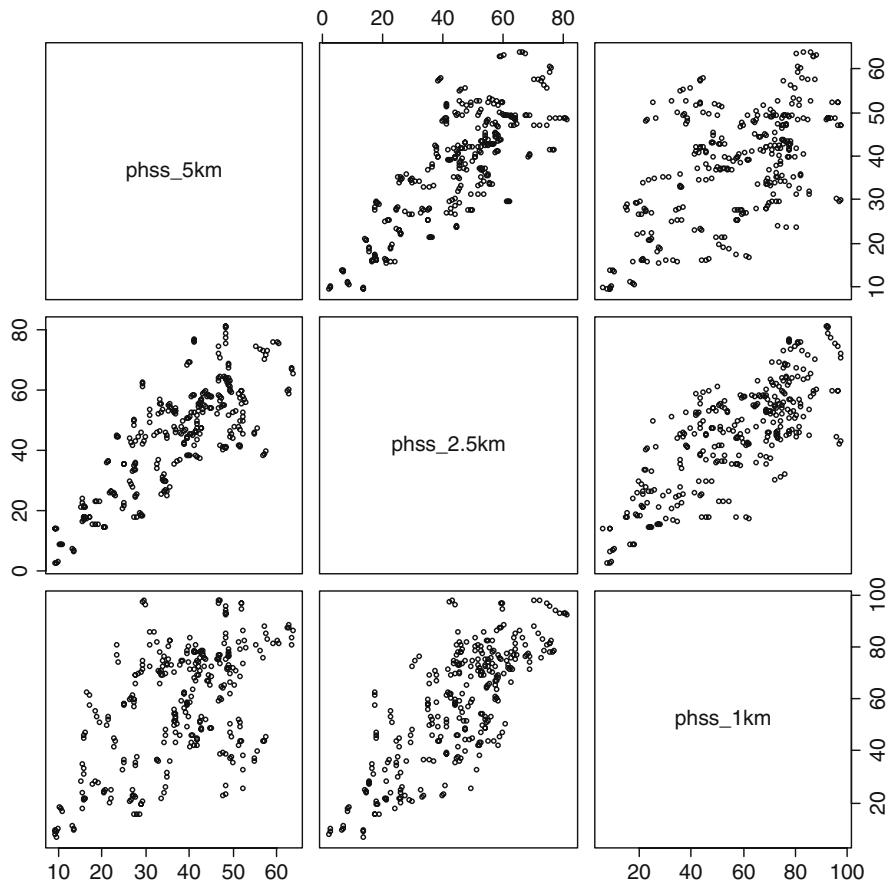
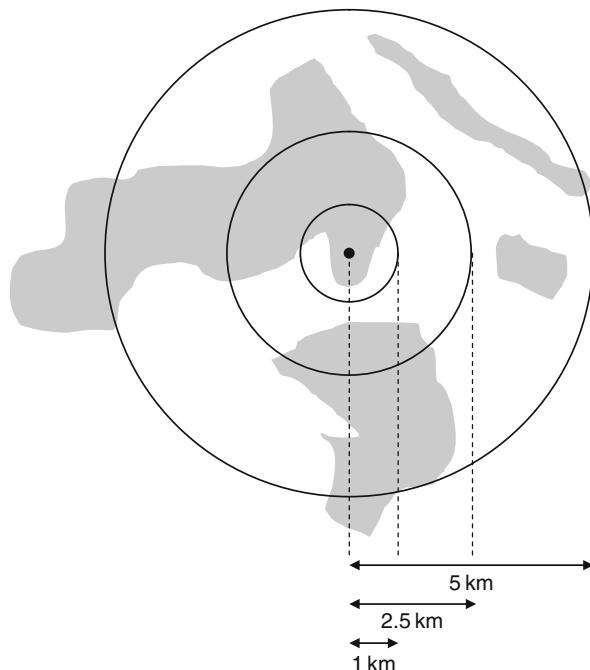


Fig. 21.3 Pairplot of the phss_5km, phss_2.5km, and phss_1km explanatory variables

phss_1km show high correlations with each other (Fig. 21.3). These variables measure the amount of highly suitable plus suitable habitat within distances of 5, 2.5, and 1 km of each subsite, respectively, and so they are spatially nested within each other (Fig. 21.4). The collinearity therefore arises because the variables calculated at the smaller landscape extents partly measure the same landscape characteristics as the variables calculated at the larger landscape extents.

You will also notice that the two landscape variables measuring habitat fragmentation (*pdens* and *edens*) are also highly positively correlated with each other. Areas with high patch densities tend to contain habitat patches that are smaller than those found in areas with low patch densities. Since small patches have more edge than large patches, this means that areas with high patch densities also tend to have high edge densities and vice versa, hence the high positive correlation. Finally, some of the patch density (*pdens*) and edge density (*edens*) variables tend to be somewhat negatively correlated with some of the habitat amount variables (*phss*

Fig. 21.4 Illustration of the nested landscape extents within which the landscape variables were calculated. The point in the centre represents a hypothetical subsite and the shaded areas represent hypothetical koala habitat



and pm). This occurs because the same processes that lead to habitat loss also tend to lead to a breaking apart of that habitat (i.e. fragmentation), resulting in greater numbers of patches with more edges. Therefore, landscape variables that measure fragmentation are often found to be correlated with those that measure habitat amount (Fahrig, 2003). However, in our data set, these correlations are only marginally more negative than -0.5 and are not considered a major concern at this stage.

There are several strategies that we could use to deal with the high collinearity found between the explanatory variables. These include (i) simply removing one or more variables so that the remaining variables are not highly correlated (Neter et al., 1990; Booth et al., 1994), (ii) using linear combinations of the variables rather than the variables directly in the model (Chatterjee and Price, 1991; Trzcinski et al., 1999; Villard et al., 1999), or (iii) using biased estimation procedures such as principal components regression or ridge regression (Neter et al., 1990; Chatterjee and Price, 1991). Here, we use the first two of these approaches to deal with collinearity because they are relatively straightforward to implement and appear adequate for our purposes.

We calculated the landscape variables at different landscape extents, because we were interested in the impact of landscape characteristics measured at different scales on koala presence at a site. We, therefore, ideally want to retain the nested structure, but reduce collinearity between the variables so that the coefficients in the model can be estimated precisely. To do this we recast each variable as linear

combination of the other variables. Suppose \mathbf{X}_5 , $\mathbf{X}_{2.5}$, and \mathbf{X}_1 are landscape variables measured at the 5, 2.5, and 1 km landscape extents respectively. We can then create a new set of variables \mathbf{Z}_5 , $\mathbf{Z}_{2.5}$, and \mathbf{Z}_1 such that:

$$\begin{aligned}\mathbf{Z}_5 &= \mathbf{X}_5 \\ \mathbf{Z}_{2.5} &= \mathbf{X}_{2.5} - \mathbf{X}_5 \\ \mathbf{Z}_1 &= \mathbf{X}_1 - \mathbf{X}_{2.5}\end{aligned}\quad (21.1)$$

Here the variable measured at the 5 km extent has remained the same, while the variables measured at the 2.5 and 1 km extents have been recalculated as the difference between the original variable and the one that it is nested within. We would expect the variables \mathbf{Z}_5 , $\mathbf{Z}_{2.5}$, and \mathbf{Z}_1 to be less correlated with each other than \mathbf{X}_5 , $\mathbf{X}_{2.5}$, and \mathbf{X}_1 . This is because the new variables represent the value of the original variables relative to those they are nested within, rather than their absolute values. Now, if we use the variables \mathbf{Z}_5 , $\mathbf{Z}_{2.5}$, and \mathbf{Z}_1 , instead of \mathbf{X}_5 , $\mathbf{X}_{2.5}$, and \mathbf{X}_1 , in our regression model, the collinearity problem should be reduced and our coefficient estimates will be more precise.

To demonstrate the reduction in collinearity, consider the percentage of highly suitable plus suitable habitat variable (`phss`). First we need to create the new variables:

```
> Koalas$phss_2.5km_new <- Koalas[, "phss_2.5km"] -
  Koalas[, "phss_5km"]
> Koalas$phss_1km_new <- Koalas[, "phss_1km"] -
  Koalas[, "phss_2.5km"]
```

Note that we do not need to create a new variable for `phss_5km`; this variable always remains the same. We will also need to create new variables, called `pm_2.5km_new`, `pm_1km_new`, `pdens_2.5km_new`, `pdens_1km_new`, `edens_2.5km_new`, `edens_1km_new`, `rdens_2.5km_new`, and `rdens_1km_new` for each of the other landscape variables in a similar way (the code is on the book website). The reduction in collinearity for the percentage of highly suitable plus suitable habitat variables can be seen by looking at the correlation matrix for the new variables using the code:

```
> cor(Koalas[, c("phss_5km", "phss_2.5km_new",
  "phss_1km_new")], method = "spearman")
```

which shows substantially lower correlation between the variables (results are not given here). This reduced collinearity can also be seen by looking at pair plots for the new variables (Fig. 21.5) compared to the pair plots for the original variables (Fig. 21.3). The same reduction in collinearity is also seen in the other landscape variables.

In using this approach, it is important to note that the regression coefficients for the new variables will have different interpretations to those for the original

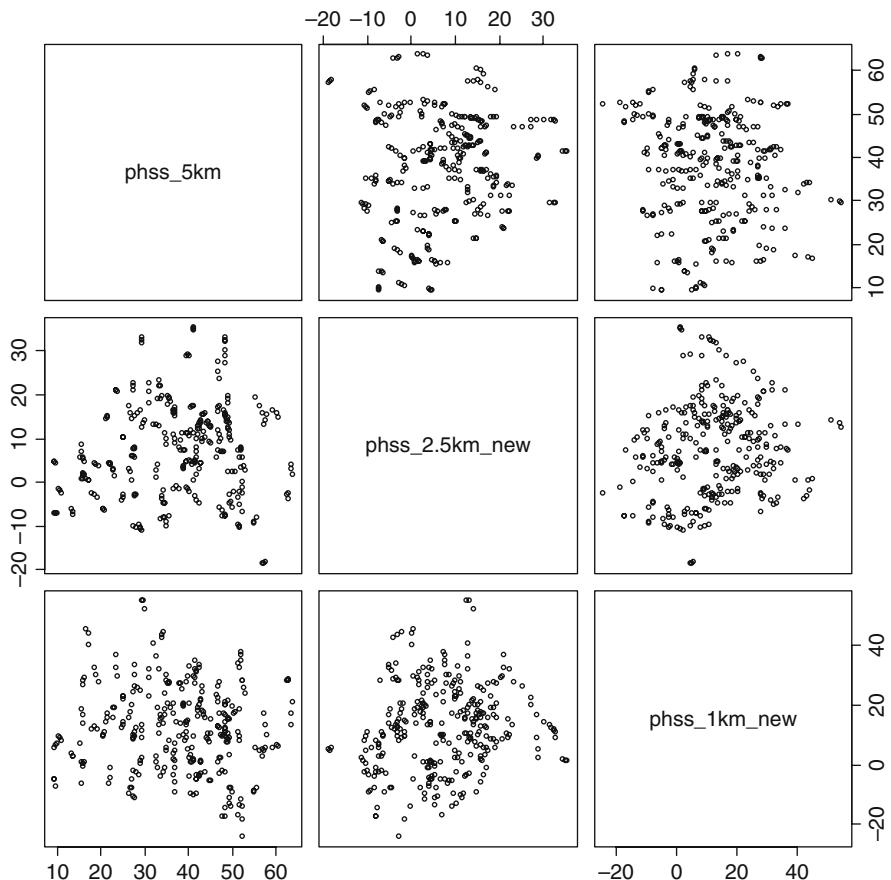


Fig. 21.5 Pairplot of the variables phss_5km, phss_2.5km_new, and phss_1km_new

variables. Fortunately, the coefficients for our new variables have a useful interpretation in terms of understanding the impact of landscape characteristics on koala presence. The interpretation of the coefficients for variables measured at the largest landscape extents remains the same. These coefficients quantify the broad-scale landscape effects on koala presence. However, the coefficients for variables measured at smaller landscape extents now represent landscape effects relative to the broader scale landscape context. This is a useful interpretation because it incorporates the dependence between fine-scale and broad-scale landscape effects on species distributions (O’Neil, 1989). Here, careful choice of the linear combinations of variables has resulted in new variables that are not highly correlated and have a useful interpretation. However new variables constructed from linear combinations of variables are not always so easily interpreted. Chatterjee and Price (1991) provide a good discussion on how to choose appropriate combinations of variables.

To deal with the collinearity between patch density (`pdens`) and edge density (`edens`) we could construct new variables based on linear combinations of the original variables. However, in this case, there are no obvious linear combinations that would result in easily interpreted coefficients. Many applications of species' distribution models require explanation to planners and the general public. Therefore, the ease of interpretation of the model is an important model building consideration, and rather than developing composite measures of patch density and edge density, we will simply retain only one of the variables as a measure of habitat fragmentation. The variable we retain is patch density because this is a straightforward and easily interpreted measure of fragmentation.

Having taken the steps described above, we now look at the variance inflation factors (VIFs) of the variables to assess the extent of any remaining collinearity. To do this, we first fit a generalised linear model with binomial response and logit link function (i.e. a logistic regression model), containing all explanatory variables, to the presence/absence data (McCullagh and Nelder, 1989; Hosmer and Lemeshow, 2000) and then calculate the VIFs for each variable from the resulting model. We use the `vif` function in the package `Design` to calculate the VIFs. The code to do this is as follows:

```
> Glm_5km <- glm(presence ~ pprim_sssite + psec_sssite +
+ phss_5km + phss_2.5km_new + phss_1km_new +
+ pm_5km + pm_2.5km_new + pm_1km_new + pdens_5km +
+ pdens_2.5km_new + pdens_1km_new + rdens_5km +
+ rdens_2.5km_new + rdens_1km_new,
+ data = Koalas, family = binomial)
> library(Design)
> vif(Glm_5km)
```

and the output is:

Variable	VIF	Variable	VIF
<code>pprim_sssite</code>	1.121	<code>psec_sssite</code>	1.099
<code>phss_5km</code>	3.196	<code>phss_2.5km_new</code>	1.584
<code>phss_1km_new</code>	1.495	<code>pm_5km</code>	1.931
<code>pm_2.5km_new</code>	1.575	<code>pm_1km_new</code>	1.973
<code>pdens_5km</code>	2.474	<code>pdens_2.5km_new</code>	1.600
<code>pdens_1km_new</code>	1.273	<code>rdens_5km</code>	2.130
<code>rdens_2.5km_new</code>	1.368	<code>rdens_1km_new</code>	1.095

You can see that all the VIFs are well below 10, suggesting that collinearity is no longer a major issue (Neter et al., 1990; Chatterjee and Price, 1991). However, some authors do suggest a more stringent cut-off than this. For example, Booth et al. (1994) suggest that VIFs should ideally be less than 1.5. Later in this chapter, we consider alternative regression models where the largest landscape extent is only

2.5 or 1 km, rather than 5 km. In these cases, the variables measured at the largest landscape extent remain as the original variables, and new variables are only constructed for those variables nested within the largest landscape extent. Therefore, we also need to check the VIFs for the variables included in these models because the variable set is slightly different. This can done using the code

```
> Glm_2.5km <- glm(presence ~ pprim_sssite +
+ psec_sssite + phss_2.5km + phss_1km_new +
+ pm_2.5km + pm_1km_new + pdens_2.5km +
+ pdens_1km_new + rdens_2.5km + rdens_1km_new,
+ data = Koalas, family = binomial)
> vif(Glm_2.5km)
```

for the 2.5 km maximum extent and the code

```
> Glm_1km <- glm(presence ~ pprim_sssite + psec_sssite +
+ phss_1km + pm_1km + pdens_1km + rdens_1km,
+ data = Koalas, family = binomial)
> vif(Glm_1km)
```

for the 1 km maximum extent. Note that for the 1 km maximum landscape extent, there are no new variables because there is no nesting within the 1 km extent. The VIFs for all variables are considerably less than 10 in both these cases. Therefore, the measures we have taken seem to have successfully reduced collinearity to acceptable levels.

21.3.2 Spatial Auto-correlation

There are two reasons for expecting spatial auto-correlation in the presence/absence data. First, spatial auto-correlation at the site-scale may occur because the distances between the subsites within individual sites are small relative to the size of koala home ranges. Average koala home range sizes in similar east coast habitats have been estimated at between 10–25 ha for females and 20–90 ha for males (AKF unpublished data, J. R. Rhodes unpublished data). Therefore, the occurrences of koalas at subsites within an individual site will tend to be correlated because they would often have been located within the same koala's home range. Second, spatial auto-correlation at broader scales may occur due to spatially constrained dispersal of koalas from their natal home ranges. Koala dispersal distances in nearby regions have been recorded to be around 3–4 km, but can be as high as 10 km (Dique et al., 2003). So, dispersal distances are substantially smaller than the spatial extent of the study area, and this could also lead to spatial auto-correlation between sites. We could also see spatial auto-correlation in the presence/absence data if the underlying spatial pattern of habitat is spatially auto-correlated. However, we would expect our explanatory variables to account for most of the spatial auto-correlation from this

source once the regression model is fitted to the data and is therefore considered to be of less concern.

One way to assess the extent of spatial auto-correlation is to look at correlograms of the data (Cliff and Ord, 1981; Bjørnstad and Falck, 2001). Correlograms are graphical representations of the spatial correlation between locations at a range of lag distances. Positive spatial correlation indicates that spatial auto-correlation between data points may be a problem. Negative spatial correlation may also indicate a problem, but this is fairly unusual in this kind of data; so we are mainly concerned with positive correlations. We use a spline correlogram to investigate auto-correlation in the presence/absence data. The spline correlogram that we use is essentially a correlogram that is smoothed using a spline function (Bjørnstad and Falck, 2001). To produce the correlograms, we need the *ncf* package (<http://asi23.ent.psu.edu/onb1/software.html>). A spline correlogram of the presence/absence data can be plotted using the code

```
> library(ncf)
> Correlog <- spline.correlog(x = Koalas[, "easting"],
+                                 y = Koalas[, "northing"],
+                                 z = Koalas[, "presence"], xmax = 10000)
> plot.spline.correlog(Correlog)
```

which produces Fig. 21.6A; a spline correlogram with 95% pointwise bootstrap confidence intervals and maximum lag distance of 10 km (note that it may take several minutes for this to run). You can see from the correlogram that significant positive spatial auto-correlation is present, but only at short lag distances of less than around 1 km. This suggests that spatial auto-correlation may be an issue for subsites located close to each other.

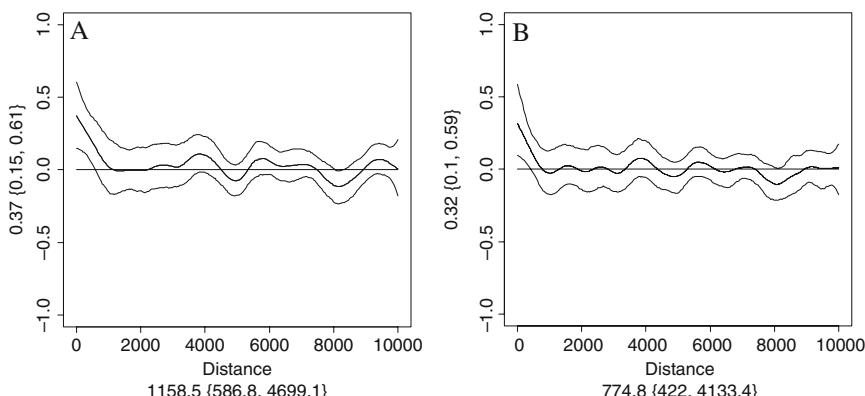


Fig. 21.6 Spline correlograms, with 95% pointwise bootstrap confidence intervals, of (A) the raw presence/absence data and (B) the Pearson residuals from a logistic regression model, including all the explanatory variables, fitted to the data

However, although spatial auto-correlation in the raw data is of interest, we are predominantly interested in whether there is any spatial auto-correlation in model residuals once any spatial auto-correlation explained by the explanatory variables has been accounted for. Therefore, we also look at the spatial auto-correlation in the Pearson residuals of the logistic regression model, containing all explanatory variables, that we fitted to the presence/absence data earlier in this chapter (Glm_5km). The following code will plot a spline correlogram of the Pearson residuals of this model:

```
> Correlog.Glm_5km <-  
  spline.correlog(x = Koalas[, "easting"],  
  y = Koalas[, "northing"], xmax = 10000,  
  z = residuals(Glm_5km, type = "pearson"))  
> plot.spline.correlog(Correlog.Glm_5km)
```

and it produces Fig. 21.6B. Although there seems to be some overall reduction in spatial auto-correlation, compared to the raw data, significant positive spatial auto-correlation at short lag distances still remains. As significant positive auto-correlation only exists at short lag distances, it is probably the result of correlation between subsites within sites, rather than correlation between sites. Since the data are nested and the spatial scale of nesting coincides with the spatial scale of auto-correlation, one reasonably straightforward way to deal with this problem is to use GLMM (McCulloch and Searle, 2001). This approach would take account of dependencies within sites and we discuss the approach in more detail in the next section. However, if the data were not nested or the spatial scale of auto-correlation and the spatial scale of nesting did not coincide (e.g. if the dependencies occurred between sites, rather than within sites), then mixed effects models are likely to be less useful and alternative approaches are likely to be required. Alternatives include a broad range of autoregressive and auto-correlation models that explicitly incorporate the spatial dependence between locations (Keitt et al., 2002; Lichstein et al., 2002; Miller et al., 2007). A full discussion of these methods is beyond the scope of the chapter, but they are worth being aware of as alternatives for dealing with spatial auto-correlation.

21.4 Generalised Linear Mixed Effects Modelling

GLMMs are useful when data are hierarchically structured in some way. They account for dependencies within hierarchical groups through the introduction of random-effects (Pinheiro and Bates, 2000; McCulloch and Searle, 2001). In this study, the data are hierarchically structured in the sense that subsites are nested within sites, and we want to use mixed effects models to account for the spatial dependencies within sites. A suitable mixed effects model for these purposes can be constructed by introducing a random-effect for site into the standard logistic regression model. The resulting mixed effects model looks like this:

$$\ln \left(\frac{p_{ij}}{1 - p_{ij}} \right) = \beta' \times \mathbf{X}_{ij} + b_i, \quad (21.2)$$

where p_{ij} is the probability of koala presence at subsite j in site i ; β is a vector of model coefficients; \mathbf{X}_{ij} is a vector of explanatory variables for subsite j in site i ; and b_i is the random-effect for site i . Here, the b_i are drawn from a random variable B , that we will assume is normally distributed with a mean of zero and variance of σ^2 , i.e., $B \sim \text{Normal}(0, \sigma^2)$. However, other random distributions can be assumed.

This provides an appropriate framework for modelling the distribution of koalas in our study area, but before progressing, we should first check that it will adequately account for the spatial auto-correlation that is present. To do this, we fit a logistic GLMM, including all the explanatory variables, to the data and once again look at a spline correlogram of the Pearson residuals. To fit the model, we will use the `glmmML` function in the package `glmmML`. Later in this chapter we compare alternative models using Akaike's information criteria (AIC) that require the calculation of the maximum log-likelihood of each model (Akaike, 1973; Burnham and Anderson, 2002). We use the `glmmML` function here because it estimates the model parameters by maximum likelihood and allows AICs to be calculated. An alternative would be to use the `lmer` function in the package `lme4` with the Lapacian or adaptive Gauss-Hermite methods. However, reliable AIC values cannot be calculated using some other mixed effects model functions such as `glmmPQL` in the package `MASS` because it maximises a penalised quasi-likelihood, rather than the full likelihood. The code to fit the mixed effects model is as follows:

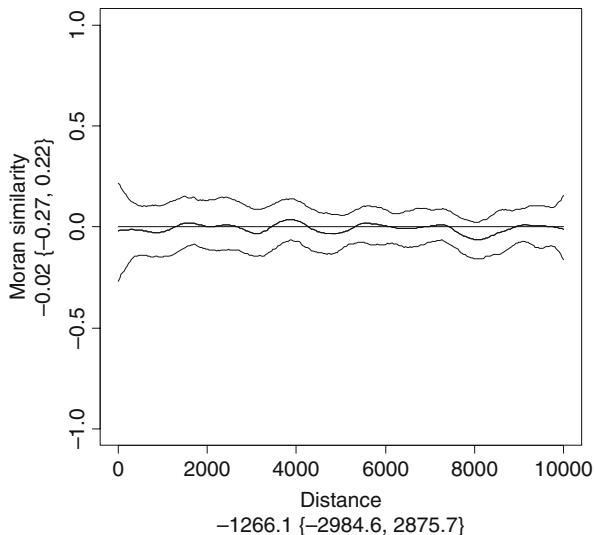
```
> library(glmmML)
> Glmm_5km <- glmmML(presence ~ pprim_sssite +
+                         psec_sssite + phss_5km + phss_2.5km_new +
+                         phss_1km_new + pm_5km + pm_2.5km_new +
+                         pm_1km_new + pdens_5km + pdens_2.5km_new +
+                         pdens_1km_new + rdens_5km + rdens_2.5km_new +
+                         rdens_1km_new, cluster = site, data = Koalas,
+                         family = binomial)
```

The `cluster` argument indicates the grouping level for the random-effect. A spline correlogram of the Pearson residuals can then be generated using the code:

```
> Correlog.Glmm_5km <- spline.correlog(
+   x = Koalas[, "easting"],
+   y = Koalas[, "northing"],
+   z = pres.glmmML(model = Glmm_5km,
+                     data = Koalas), xmax = 10000)
> plot.spline.correlog(Correlog.Glmm_5km)
```

which produces Fig. 21.7. Here the call to the function `pres.glmmML` (which can be found at the book website) calculates the Pearson residuals for the model. You

Fig. 21.7 Spline correlogram, with 95% pointwise bootstrap confidence intervals, of the Pearson residuals from a mixed effects logistic regression model, including all the explanatory variables, fitted to the data



can now see that there is no longer any obvious increase in spatial correlation at short lag distances. This suggests that the mixed effects model successfully accommodates the spatial auto-correlation within sites. This also helps to confirm that the main source of spatial auto-correlation at short lag distances is indeed the dependency between subsites within sites. In the following sections, we therefore use mixed effects logistic regression to model koala distributions in Noosa.

21.4.1 Model Selection

We have now identified a suitable set of explanatory variables and an appropriate modelling framework. The next step is to identify which of the variables are important determinants of koala distributions and to identify a suitable and parsimonious approximating model that we can use to make predictions. Rather than using traditional null-hypothesis testing procedures for variable selection to achieve these aims, we will use an information-theoretic approach (Burnham and Anderson, 2002). Information-theoretic approaches provide a framework that allows multiple model comparisons to be made and the most parsimonious of these models to be identified. The process of identifying a parsimonious model involves trading off model bias against model precision and information-theoretic approaches achieve this by using appropriately constructed criteria to compare models (Burnham and Anderson, 2002). The criteria we use here is AIC, which is defined as

$$\text{AIC} = -2L + 2K, \quad (21.3)$$

where L is the maximum log-likelihood of the model and K is the number of parameters in the model (Akaike, 1973). A model with a low AIC is more parsimonious

than a model with a high AIC. Note, however, that it is only the relative differences in AIC values between models that are important and that the absolute value of a model's AIC is meaningless (Burnham and Anderson, 2002). Information-theoretic approaches have certain advantages over traditional null-hypothesis testing approaches (Johnson, 1999; Anderson et al., 2000; Burnham and Anderson, 2001; Lukacs et al., 2007). These advantages include the ability to (i) evaluate multiple non-nested models relative to each other, (ii) quantify the relative support for multiple models simultaneously, and (iii) derive predictions that account for model uncertainty using model averaging; but see critiques by Guthery et al. (2005) and Stephens et al. (2005).

To implement this approach, we first develop a series of alternative mixed effects models that include different combinations of the explanatory variables. These alternative models can be thought of as different 'hypotheses' about the relationships between koala presence/absence and the explanatory variables. We then examine the support from the data for each of these models using AIC (*sensu* Hilborn and Mangel, 1997). This will be achieved by fitting each model to the data and ranking them by their AIC values. We will also calculate the relative probability of each model being the best model by calculating their Akaike weights, w_i . The Akaike weight for model i is defined as

$$w_i = \frac{\exp\left(-\frac{1}{2}\Delta_i\right)}{\sum_{j=1}^R \exp\left(-\frac{1}{2}\Delta_j\right)}, \quad (21.4)$$

where Δ_i is the difference between the AIC for model i and the model with the lowest AIC and the sum is over all the alternative models in the set $j = 1, \dots, R$. Akaike weights are useful because they can be used to identify a 95% confidence set of models, and ratios of Akaike weights (evidence ratios) provide quantitative information about the support for one model relative to another (Burnham and Anderson, 2002). A 95% confidence set of models can be constructed by starting with the model with the highest Akaike weight and repeatedly adding the model with the next highest weight to the set until the cumulative Akaike weight exceeds 0.95. Akaike weights can also be used to calculate the relative importance of a variable by summing the Akaike weights of all the models that include that variable (Burnham and Anderson, 2002). We will therefore also calculate the 95% confidence set of models and the relative importance of the landscape-scale habitat amount, fragmentation, and road density variables.

In constructing the alternative models, we group the explanatory variables into four functional groups (1) site-scale habitat (`pprim_sssite` and `psec_sssite`); (2) landscape-scale habitat amount (`phss` and `pm`); (3) landscape-scale habitat fragmentation (`pdens`); and (4) landscape-scale road density (`rdens`). There is good evidence from other studies that site-scale habitat characteristics are a key determinant of the use of a site by koalas (Phillips and Callaghan, 2000; Phillips et al., 2000). Therefore, we include site-scale habitat in all the models and for

each landscape extent (1, 2.5, and 5 km), construct a model for all combinations of the landscape-scale habitat amount, landscape-scale habitat fragmentation, and landscape-scale road density variables. This leads to a total of 22 alternative models. However, we also construct a ‘null’ model that includes no explanatory variables as a check of our assumption of the importance of the site-scale variables. Note that for each landscape extent, the variables spatially nested within that spatial extent are also included in the model.

Before fitting each of these models to the data, the explanatory variables should be standardised so that they each have a mean of zero and standard deviation of one. This helps to improve convergence of the fitting algorithm and puts the estimated coefficients on the same scale, allowing effect sizes to be more easily compared. We can standardise the explanatory variables using the code

```
> Koalas_St <- cbind(Koalas[, 1:5],
  apply(X = Koalas[, 6:ncol(Koalas)], MARGIN = 2,
  FUN = function(x) {(x - mean(x)) / sd(x)}))
```

which creates a new data frame, called Koala_St, of the standardised variables. We use these standardised variables as the explanatory variables in fitting the alternative models.

Rather than showing the code to fit each of the alternative models, we show the code to fit one of the models as an example. The code to fit the model including the site-scale habitat variables and the landscape-scale habitat amount variables at the 1 km extent is

```
> glmmML(presence ~ pprim_sssite + phss_1km + pm_1km, cluster = site,
  data = Koalas_St, family = binomial)
```

which gives the following output:

	coef	se(coef)	z	Pr(> z)
(Intercept)	-0.7427	0.2314	-3.210	0.001330
pprim_sssite	0.8576	0.2244	3.822	0.000132
psec_sssite	0.2319	0.1938	1.196	0.232000
phss_1km	0.2765	0.2479	1.115	0.265000
pm_1km	0.5573	0.2524	2.208	0.027200

```
Standard deviation in mixing distribution: 1.561
Std. Error: 0.3005
Residual deviance: 354.5 on 294 degrees of freedom
AIC: 366.5
```

This shows that the probability of koala presence increases with the percentage of preferred tree species at a subsite and the percentage of habitat in the surrounding landscape. The standard deviation of the random-effect is 1.56 and the model’s AIC is 366.5.

The AICs, Akaike weights, and model rankings for all the models in the 95% confidence set are shown in Table 21.2. This table also shows the relative importance of landscape-scale habitat amount, fragmentation, and road density variables. The first thing to note is the large number of models in the 95% confidence set of models (14), indicating there is considerable model uncertainty. The Akaike weights confirm this with no models much more likely to be the best model than the other models. The best model includes the site-scale habitat and landscape-scale habitat amount variables at the 1 km extent. However, this model is only 1.7 times more likely to be the best model than the next best model, which also includes landscape-scale road density (evidence ratio = 0.174/0.101). In general the models at the 1 km landscape extent performed better than the models at the 2.5 and 5 km landscape extents. This suggests there is little gain in predictive performance from adding additional variables representing the landscape at extents broader than 1 km. The relative variable importances suggests that landscape-scale habitat amount and landscape-scale road density are more important determinants of koala distributions than landscape-scale fragmentation. However, due to the high model uncertainty, the differences in relative importance are not particularly large. Finally, the null

Table 21.2 The 95% confidence set of models

Rank	Site-scale habitat	Landscape-scale habitat amount	Landscape-scale habitat fragmentation	Landscape-scale road density	Landscape extent (km)	AIC	w
1	✓	✓			1	366.5	0.174
2	✓	✓		✓	1	367.6	0.101
3	✓				—	367.7	0.097
4	✓			✓	5	367.8	0.092
5	✓	✓			5	367.9	0.087
6	✓	✓	✓		1	368.1	0.082
7	✓			✓	1	368.2	0.075
8	✓	✓	✓	✓	1	369.1	0.048
9	✓	✓			2.5	369.3	0.043
10	✓		✓		1	369.7	0.036
11	✓			✓	2.5	369.9	0.032
12	✓		✓	✓	1	370.2	0.028
13	✓		✓		5	370.7	0.021
14	✓	✓		✓	5	370.8	0.021
Relative importance	—	0.590	0.261	0.431			

AIC = Akaike's information criteria; w = Akaike weights; site-scale habitat = pprim_sssite + psec_sssite; landscape-scale habitat amount = phss_1km + pm_1km (1km extent), phss_2.5km + phss_1km_new + pm_2.5km + pm_1km_new (2.5km extent), phss_5km + phss_2.5km_new + phss_1km_new + pm_5km + pm_2.5km_new + pm_1km_new (5km extent); landscape-scale habitat fragmentation = pdens_1km (1km extent), pdens_2.5km + pdens_1km_new (2.5km extent), pdens_5km + pdens_2.5km_new + pdens_1km_new (5km extent); landscape-scale road density = rdens_1km (1km extent), rdens_2.5km + rdens_1km_new (2.5km extent), rdens_5km + rdens_2.5km_new + rdens_1km_new (5km extent).

model has an AIC of 382.2 and relative to the model only containing the site-scale habitat variables (AIC = 367.7), has an evidence ratio of almost zero. This indicates very strong support for our assumption that site-scale habitat variables are important determinants of koala presence or absence at a site.

Given there is no single model that is clearly the best, a sensible approach is to acknowledge this model uncertainty and make inferences based on model averaging (Burnham and Anderson, 2002). Model averaging allows coefficients to be estimated and model predictions to be made that account for the inherent model uncertainty in addition to parameter uncertainty. In essence, these approaches derive weighted average predictions, where the weights are the relative model probabilities. When model uncertainty is present, this has considerable advantages over more traditional step-wise and null-hypothesis approaches to model selection, where you only end up with a single best model. Model averaged predictions are likely to be more robust than those derived from a single best model. Burnham and Anderson (2002) provide useful guidelines for conducting model averaging using AIC, and see McAlpine et al. (2006) and Rhodes et al. (2006) for examples of model averaging applied to predicting koala distributions.

21.4.2 Model Adequacy

So far, we have examined the relative support from the data for each model. However, this tells us little about how well the models fit the data or whether there are any departures from model assumptions. Traditionally, the fit of logistic regression models have been assessed using global goodness-of-fit tests based on the deviance or Pearson χ^2 statistics. However, the distributional properties of these statistics are not well understood, making the tests somewhat difficult to apply in practice (Hosmer and Lemeshow, 2000). Further, despite the convenience of global goodness-of-fit tests, it is unclear to what extent it is sensible to condense model fit into a single number or test (Landwehr et al., 1984). An alternative to global goodness-of-fit tests is to use a range of graphical methods to assess how well a model fits the data. Here, we concentrate on quantile-quantile plots and partial residual plots (Landwehr et al., 1984). Logistic regression quantile-quantile plots are useful for assessing whether the error distribution of the data is modelled correctly and to detect more general departures from model assumptions. Partial residual plots are useful for assessing systematic departures from model assumptions, such as linearity. We will apply these diagnostic procedures to the most parsimonious model, although they can equally be applied to model averages if model averaged predictions are to be made.

A quantile-quantile plot consists of a graph of quantiles of residuals assuming the fitted model is the true model, against the actual quantiles of the residuals from the fitted model. If there are no major deviations from the model assumptions, then these points should lie close to the 1:1 line. Since the distribution of the residuals in logistic regression is not well understood, Landwehr et al. (1984) propose a simulation approach for constructing a logistic regression quantile-quantile plot. Their basic approach is as follows:

1. From the fitted model, calculate the residuals r_i .
2. Order the r_i , giving $r_{(i)}$.
3. Simulate M data sets from the fitted model.
4. Fit the model to the M data sets.
5. Compute the residuals r_i^* for the models fitted to the M data sets and order them to get $r_{(i)}^*$.
6. Calculate the medians of the ordered residuals from the M replicates. (Landwehr et al. (1984) use a slight modification here where they interpolate within the distribution of the simulated residuals to avoid plotting negative against positive residuals.)
7. Plot the median simulated ordered (interpolated) residuals against the ordered residuals from the original model fit.
8. Calculate confidence intervals for the simulated ordered (interpolated) residuals from the M replicates.
9. Plot the median simulated ordered (interpolated) residuals against the upper and lower confidence intervals.

If we apply this approach to the most parsimonious model, with $M = 1000$, we get the plot shown in Fig. 21.8. The code for creating this plot and the required functions `res.glmmML` and `fitted.glmmML` can be found at the book website. You will see that the points lie quite close to the 1:1 line and within the simulated 95% point-wise confidence interval. This suggests there are no major departures from the model assumptions.

The partial residual plot for a particular covariate consists of a graph of the values of the covariate against its partial residuals. Partial residuals (r_{par}) are defined as

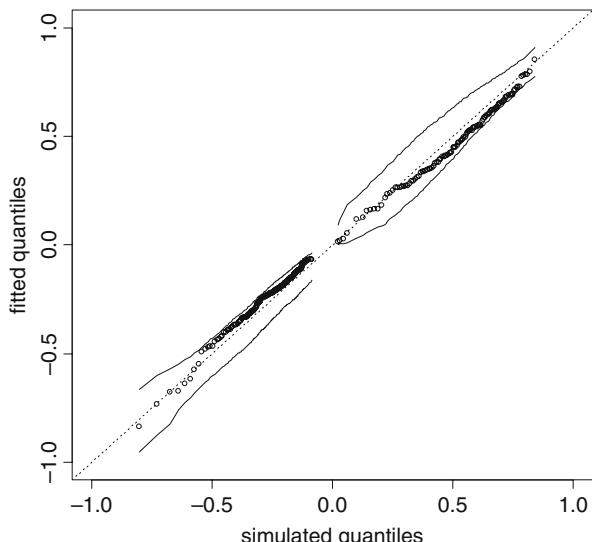


Fig. 21.8 Quantile-quantile plot with 95% pointwise confidence bounds

$$r_{\text{par}} = \frac{y - \hat{p}}{\hat{p} \times (1 - \hat{p})} + X \times \hat{\beta}_X \quad (21.5)$$

where y is the observed data (1 or 0), \hat{p} is the estimated probability for the fitted model, X is the covariate value, and $\hat{\beta}_X$ is the estimated coefficient for the covariate X for the fitted model (Landwehr et al., 1984). If a partial residual plot is linear, then a linear assumption for this covariate is appropriate. However, if a partial residual plot is non-linear, this indicates that a linear assumption may not be appropriate, and in that case, the shape of the curve can suggest an appropriate functional form for the covariate. Due to the dichotomous nature of binomial data, partial residual plots for logistic regression show two groups of points; one for the 1 observations and one for the 0 observations. Therefore, it is necessary to fit a smoothed curve to the points to assess whether it is linear or non-linear. The partial residual plots for the four covariates in the most parsimonious model with smoothed curves fitted using the `loess` function are shown in Fig. 21.9. The code for creating these plots and the required functions `res.glmmML` and `fitted.glmmML` can be found at the book website. All of the curves are moderately non-linear, but especially so for

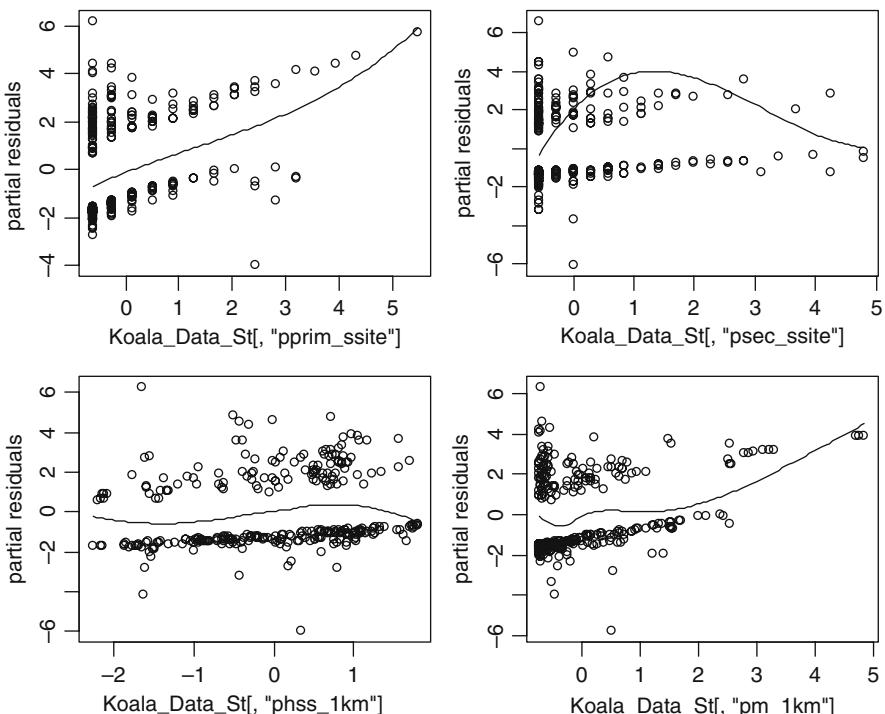


Fig. 21.9 Partial residual plots for `pprim_sssite`, `psec_sssite`, `phss_1km`, and `pm_1km` for the highest ranked model

the psec_sssite curve. The shape of the psec_sssite curve suggests that the inclusion of a quadratic term for this covariate might be appropriate.

Re-fitting the most parsimonious linear model with a quadratic term for psec_sssite gives the model

	coef	se(coef)	z	Pr(> z)
(Intercept)	-0.4809	0.2576	-1.866	0.062000
pprim_sssite	0.8908	0.2292	3.887	0.000101
psec_sssite	0.9161	0.3718	2.464	0.013700
I(psec_sssite^2)	-0.2820	0.1360	-2.074	0.038100
phss_1km	0.3095	0.2522	1.227	0.220000
pm_1km	0.5972	0.2581	2.314	0.020700

Standard deviation in mixing distribution: 1.581
 Std. Error: 0.3065
 Residual deviance: 349.3 on 293 degrees of freedom
 AIC: 363.3

which confirms the improvement in the model with a reduction in AIC of 3.2 units. Since this is a more parsimonious model than the linear model, the preference would be to use this to make predictions, rather than the linear model, or alternatively to include models with a quadratic term for psec_sssite in the model set for making model-averaged predictions.

In considering the adequacy of our models, we have only compared model predictions against the data that they were fitted to. However, we often want to use species' distribution models to make predictions for a new area or for a new site. In this case, simply comparing predictions to the data used to fit the models will tend to overestimate the predictive performance of the models. One way to overcome this is to fit the models to one data set and then compare model predictions to an independent data set (Pearce and Ferrier, 2000). This is known as cross-validation. However, we rarely have the luxury of a completely independent data set; so simulation-based cross-validation using random samples from the data used to fit the models is often used instead (Stone, 1974; Efron and Tibshirani, 1997). We do not consider these approaches in detail here, but they are important aspects of model validation and it is important to be aware of them. For specific discussion on the validation of wildlife distribution models, see Pearce and Ferrier (2000) and Vaughan and Ormerod (2005).

21.5 Discussion

In this chapter, we have demonstrated the use of GLMM for modelling species distributions. The use of GLMM was an effective way of dealing with spatial autocorrelation in the data, but this may not always be the case, such as if spatial autocorrelation existed between sites. However, other approaches, such as autoregres-

sive models, do exist that could be used to deal with between-site auto-correlation (e.g., Miller et al., 2007). We also found that constructing simple linear combinations of nested landscape variables was useful for reducing collinearity, while still maintaining an easily interpreted model. This approach is particularly useful for landscape-scale studies such as this, where landscape effects are often conceptualised as occurring at a range of nested spatial extents. We also demonstrated an information-theoretic approach (using AIC) to model selection and the identification of the most parsimonious models. The information-theoretic approach allowed us to quantify the level of model uncertainty and provided the potential to calculate model-averaged predictions. Model-averaged predictions are useful in contexts such as the one presented here, where there is reasonably high model uncertainty, because predictions are not conditional on a single model (Burnham and Anderson, 2002). The information-theoretic framework was also found to be useful for ranking the landscape-scale covariates in terms of their importance. Identifying the importance of each covariate in this way has an important practical application for prioritising management actions for the conservation of koalas.

One of the primary aims of this chapter was to model koala distributions to help understand the key landscape- and site-scale factors determining the presence of koalas. We found strong evidence that the percentage of preferred tree species at the site-scale was positively related to koala occupancy. This is consistent with other studies indicating that koalas often select certain preferred tree species (Phillips and Callaghan, 2000, Phillips et al., 2000) or select habitats containing high proportions of preferred tree species (Rhodes et al., 2005). We also found that koala occupancy was positively related to the amount of habitat at the landscape-scale, which was more important than the density of roads, which in turn was more important than habitat fragmentation. It is generally accepted that the amount of habitat tends to be more important than habitat fragmentation for the viability of wildlife populations (Fahrig, 2003). Our analyses suggest this is the case for the koala in Noosa and that the conservation priority should be habitat protection, rather than just seeking particular landscape configurations that minimise fragmentation. However, fragmentation effects may become more important as habitat is lost (Flather and Bevers, 2002). It is interesting to note that road density was almost as important as habitat amount. Increasing road density decreases the chance of finding koalas and this may simply reflect the general effects of urbanisation and associated threatening processes. It is generally accepted that areas around habitat patches, known as the habitat matrix, can have important implications for the viability of species (Ricketts, 2001). This may be what is happening here with factors associated with urban development, such as vehicle collision mortality and dog attacks, negatively impacting koala populations. Mitigation of these factors would therefore also seem to be an important conservation priority for koalas in Noosa.

It is interesting to note that the landscape-scale variables measured at the 1 km scale tended to be the best descriptors of koala presence (Table 21.2). We would expect the scale at which the landscape affects the presence of koalas to be related to the scale of koala movements such as natal dispersal and movements within individual home ranges. Koalas have average dispersal distances of several kilometres

(Dique et al., 2003), and so the scale of the landscape effects is at the shorter end of the distribution of koala dispersal distances. This suggests that the spatial dynamics of koala populations in Noosa are influenced predominantly by koalas dispersing over short distances and by movements of individuals within their home ranges, rather than by less common long distance dispersal movements.

21.6 What to Write in a Paper

When writing a scientific paper you need to be selective about what you include, while still ensuring that the methods are sufficiently detailed to allow readers to repeat your study and that the research findings are clearly explained. We have presented a great deal more information in this chapter than would be required for a scientific paper. Although there is no single recipe for what to include and what not to include in a paper, based on the analysis presented in this chapter, we give a broad outline of what we think should be included.

In the introduction section, we would aim to give a clear statement of the biological and wildlife management issues addressed by the research. The last paragraph of this section should explicitly state the specific questions that the research addresses, and very briefly, outline what was done. In the methods section, we would have a description of the study site and the data collection methods. Then we would briefly describe the exploratory analysis we conducted in relation to collinearity and spatial auto-correlation. Although the description of these steps should be brief, it would be important to describe the transformations of the explanatory variables and perhaps include the graphs showing the reduction in collinearity (e.g. Figs. 21.3 and 21.5). The remainder of the methods section should then describe the alternative models we fitted to the data, the use of AIC in comparing the models, and the methods used to assess model adequacy. The results section should include a description of the key findings of the statistical analyses and the assessment of model adequacy. It is not necessary to describe every single aspect of these results, but sufficient details should be included to give the reader a clear picture of the key findings. Other things to include here would be a table showing the model rankings with AICs, coefficient estimates, and standard errors for at least the best model(s) and graphical demonstration of model adequacy (e.g. Fig. 21.8). A useful additional figure that we do not show here would be a map of predictions and their associated standard errors based on the best, or model-averaged, model (see, e.g. Rhodes et al. (2006)). Finally, the discussion section should indicate the implications of the results in terms of the issues raised in the introduction and highlight the applied or theoretical advances the study has made. A key component of the discussion should be identifying any limitations of the work and suggesting future research directions.

Acknowledgments This work was funded by the Australian Research Council, the Australian Koala Foundation, and The University of Queensland.

Chapter 22

A Comparison of GLM, GEE, and GLMM Applied to Badger Activity Data

N.J. Walker, A.F. Zuur, A. Ward, A.A. Saveliev, E.N. Ieno, and G.M. Smith

22.1 Introduction

In this chapter, we analyse a data set consisting of signs of badger (*Meles meles*; see Fig. 22.1) activity around farms. The data are longitudinal and from multiple farms; so it is likely a temporal correlation structure is required. The response variable is binary; the presence or absence of badger activity. The dataset comes from a survey carried out on 36 farms over 8 consecutive seasons running from autumn 2003 to summer 2005. For analytical convenience, we consider these intervals to be exactly equal, which is a close enough approximation to the reality. All farms in the survey were in South-West England, which is a high-density badger country.



Fig. 22.1 Photograph of two badgers on the nightly hunt for food. The photo was taken by Dr Richard Yarnell, School of Animal, Rural and Environment Sciences, Nottingham Trent University, UK

N.J. Walker (✉)

Woodchester Park CSL, Tinkley Lane, Nympsfield, Gloucester GL10 3UJ, United Kingdom

This work was carried out in the wider context of badgers and their possible role in transmitting bovine tuberculosis to cattle. One avenue for tackling this problem might be to reduce the rates of badger visits to farms in particular areas where they may come into contact with resident cattle. The aim of this study was to predict the occurrence of signs of badger activity on farms.

There are many different ways of measuring badger activity, but for the purposes of this chapter, we just consider one of these: ‘signs of activity’. This was used as a binary variable that took the value 1 when signs of badger activity were recorded and 0 if no signs were recorded. Signs of activity included badger faeces, indications of digging, feeding evidence, etc. Several potential explanatory variables were recorded – these are detailed in Table 22.1.

Consecutive observations on badger activity at a given farm may be temporally auto-correlated. Because of this and because the data are in binary form, we

Table 22.1 List of variables with a short description. The response variable is Signs_in_yard

Variable	Description
Year	Calendar year
Season	Spring (Mar–May), Summer (Jun–Aug), autumn (Sept–Nov) and winter (Dec–Feb)
Farm_code_numeric	Blinded farm identifier
Survey	Which of the 8 survey occasions (i.e. the time indicator)
Signs_in_yard	Binary indicator of signs of badger activity
Latrines_with_farm_feed	Binary indicator – do (any) observed badger latrines contain farm feed? (This is a proxy for the fact that badgers must have been on farm).
No_latrines_with_farm_feed	The number of the above
No_scats_with_farm_feed	Number of badger faeces identified as containing farm feed
No_latrines	Number of badger latrines observed
No_setts_in_fields	Number of badger setts (i.e. homes) observed
No_active_setts_in_fields	Number of actively used setts observed
No_buildings	Number of buildings on farm
No_cattle_in_buildings_yard	Number of cattle housed in the building yard
Mode_feed_store_accessibility	Quantitative index of how easy it would be for badgers to access the farm’s feed store
Accessible_feed_store_present	Binary indicator – is such a feed store present?
Mode_cattle_house_accessibility	Quantitative index of how easy it would be for badgers to access the cattle house
Accessible_cattle_house_present	Binary indicator – is such a feed store present?
Accessible_feed_present	Binary indicator – is accessible feed present
Grass_silage	Binary indicator of presence of grass silage
Cereal_silage	Binary indicator of presence cereal silage
HayStraw	Binary indicator of presence of Hay/Straw
Cereal_grains	Binary indicator of presence of cereal grains
Concentrates	Binary indicator of presence of concentrates
Proteinblocks	Binary indicator of presence of protein blocks
Sugarbeet	Binary indicator of presence of sugar beet
Vegetables	Binary indicator of presence of vegetables
Molasses	Binary indicator of presence of molasses

used generalised estimating equations (GEE) and generalised linear mixed models (GLMM). If there would be no temporal auto-correlation, then generalised linear modelling (GLM) can be applied. The underlying GLM, GEE, and GLMM theory was discussed in Chapters 9, 12, and 13.

The aim of this chapter is not to find the best possible model for the data, but merely to contrast GLM, GEE, and GLMM. When writing this chapter, we considered two ways to do this, namely,

1. Apply a model selection in each of the three models (GLM, GEE, and GLMM). It is likely that the optimal GLM consists of a different set of explanatory variables than the GEE and GLMM. The reason for this is the omission of the dependence structure in the data. We have seen this behaviour already in various other examples in this book with the Gaussian distribution. Also, recall the California data set that was used to illustrate GLM and GEE in Chapter 12; the p -values of the GLM were considerably smaller than those of the GEE! Therefore, in a model selection, one ends up with different models. Using this approach, the story of the chapter is then that (erroneously) ignoring a dependence structure gives you a different set of significant explanatory variables.
2. Apply the GLM, GEE, and GLMM on the same set of explanatory variables and compare the estimated parameters and p -values. If they are different (especially if the GLM p -values are much smaller), then the message of the chapter is that ignoring the dependence structure in a GLM gives inflated p -values.

Both approaches are worthwhile presenting, but due to limited space, we decided to go for option 2 and leave the first approach as an exercise to the reader. The question is then: Which GLM model should we select? We decided to adopt the role of an ignorant scientist and apply the model selection using the GLM and contrast this with the GEE and GLMM applied on the *same* selection of covariates. Note that the resulting GEE and GLMM models are not the optimal models as we are not following our protocol from Chapters 4 and 5, which stated that we should first look for the optimal random structure using a model that contained as many covariates as possible.

22.2 Data Exploration

The first problem we encountered was the spreadsheet (containing data on 282 observations), which was characterised by a lot of missing values. Most R functions used so far have options to remove missing values automatically. In this section, we will use the `geepack` package, and its `geeglm` function requires the removal of all missing values.

Rows with missing values in the response variable were first removed. Some of the explanatory variables had no missing values at all and other explanatory variables had 71 missing values! Removing every row (observation) that contains a

Table 22.2 Number of missing values per variable. The data set contains 288 rows (observations). The notation ‘# NAs’ stands for the number of missing values. The response variable is Signs.in.yard and contains 6 missing values

Variable	# NAs	Variable	# NAs
Year	0	Accessible.feed.store.present	6
Season	0	Mode.cattle.house.accessibility	71
Farm_code_numeric	0	Accessible.cattle.house.present	6
Survey	0	Accessible.feed.present	6
Signs.in.yard	6	Grass.silage	6
Latrines.with.farm.feed	33	Cereal.silage	6
No.latrines.with.farm.feed	34	HayStraw	6
No.scats.with.farm.feed	59	Cereal.grains	6
No.latrines	30	Concentrates	6
No.setts.in.fields	10	Proteinblocks	6
No.active.setts.in.fields	15	Sugarbeet	6
No.buildings	6	Vegetables	6
No.cattle.in.buidlings.yard	6	Molasses	6
Mode.feed.store.accessibility	38		

missing value reduces the sample size. Therefore, it is perhaps better to remove entirely explanatory variables with several missing values. This is an arbitrary process; where do you draw the line when you stop removing explanatory variables? The answer should be based on biological knowledge and common sense (drop the variables with lots of missing values and that you also think are the least important). Table 22.2 shows the number of missing values per variable. The explanatory variable Mode.cattle.house.accessibility has 71 missing values. If we insist on using it, we end up removing 71 observations or 24% of the data! To avoid such a situation, we decided to omit all explanatory variables with more than 15 missing values from the analysis. From the remaining data, we removed all rows where there was at least one observation missing, ending up with 273 observations for analysis.

Table 22.2 was obtained with the following R code.

```
> library(AED); data(BadgersFarmSurveys.WithNA)
> Badgers.NA <- BadgersFarmSurveys.WithNA #Saves space
> colSums(sapply(Badgers.NA, FUN = is.na))
```

The sapply function creates a matrix of length 288 by 27 with the elements FALSE (corresponding element in Badger.NA is not a missing value) and TRUE (corresponding element is a missing value). The function colSums converts each FALSE into a 0 and TRUE into a 1 and takes the sum per column: the number of missing values per variable.

The number of explanatory variables is very large, and using a data exploration, we tried to find collinear explanatory variables. Pairplots (for the continuous

variables), Pearson correlation coefficients and variance inflation factors indicated that `No.setts.in.fields` and `No.active.setts.in.fields` are collinear; they have a correlation of 0.86.

We decided to drop the variable `No.active.setts.in.fields`. The variables `No.buildings` and `No.cattle.in.buildings.yard` have a correlation of 0.53. We decided to drop the second one. The explanatory variables `Proteinblocks` and `Vegetables` had only a few values of 1; the majority of observations had a 0 value. Including them caused numerical problems and we decided to drop them.

22.3 GLM Results Assuming Independence

The following code accesses the data (we removed the missing values in Excel and created a new data file), renames some of the longer variable names, and applies a GLM assuming independence. We could have renamed the variables in the data file, but the code below shows you the coding misery due to having long variable names (let it be a warning!). Always try to choose the names as short as possible when you create the data file. Most of the nominal variables are binary with values 0 (representing no) and 1 (representing yes), and for these, the `factor` command can be avoided because this is exactly what it does: making columns with zeros and ones. However, we decided to use it as it is too easy to make a mistake. The `drop1` function applies an analysis of deviance (Chapter 9).

```
> library(AED); data(BadgersFarmSurveysNoNA)
> Badgers <- BadgersFarmSurveysNoNA
> Badgers$fSeason <- factor(Badgers$Season)
> Badgers$fFeed.store <-
  factor(Badgers$Accessible.feed.store.present)
> Badgers$fCattle.house <-
  factor(Badgers$Accessible.cattle.house.present)
> Badgers$fFeed.present <-
  factor(Badgers$Accessible.feed.present)
> Badgers$fGrass.silage <- factor(Badgers$Grass.silage)
> Badgers$fCereal.silage <- factor(Badgers$Cereal.silage)
> Badgers$fHayStraw <- factor(Badgers$HayStraw)
> Badgers$fCereal.grains <- factor(Badgers$Cereal.grains)
> Badgers$fConcentrates <- factor(Badgers$Concentrates)
> Badgers$fSugarbeet <- factor(Badgers$Sugarbeet)
> Badgers$fMolasses <- factor(Badgers$Molasses)
> B.glm <- glm(Signs.in.yard ~ fSeason+
  No.setts.in.fields + No.buildings + fFeed.store +
  fCattle.house + fFeed.present + fGrass.silage +
  fCereal.silage + fHayStraw + fCereal.grains +
```

```
fConcentrates + fSugarbeet + fMolasses,
family = binomial, data = Badgers)
> drop1(B.glm, test = "Chi")
```

The results are not presented here, but most explanatory variables are not significant at the 5% level. We decided to drop the least significant explanatory variable, refit the model, reapply the `drop1` command, and continue to drop explanatory variables until all remaining variables in the model are significant. The final model contains `No.setts.in.fields` and `fFeed.store`. Applying this model in R and an analysis of deviance with the `drop1` function gave

```
> B2.glm <- glm(Signs.in.yard ~ No.setts.in.fields +
fFeed.store, family = binomial, data = Badgers)
> drop1(B2.glm, test = "Chi")
```

Single term deletions. Model:

```
Signs.in.yard ~ No.setts.in.fields + fFeed.store
```

	Df	Deviance	AIC	LRT	Pr(Chi)
<none>		182.509	188.509		
No.setts.in.fields	1	234.107	238.107	51.597	6.813e-13
fFeed.store	1	187.307	191.307	4.798	0.02849

The number of setts in the field is highly significant, and the presence of accessible feed store is weakly significant. The parameter estimates are obtained with the `summary(B2.glm)` command and are given below.

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-2.5891	0.4602	-5.626	1.85e-08
No.setts.in.fields	0.2862	0.0457	6.262	3.79e-10
fFeed.store1	-1.0341	0.4587	-2.254	0.0242

Dispersion parameter for binomial family taken to be 1

Null deviance: 237.79 on 272 degrees of freedom

Residual deviance: 182.51 on 270 degrees of freedom

AIC: 188.51

Note the number of setts in the field has a positive effect on the probability of finding badger activity. The nominal variable `fFeed.store` has values 0 and 1; hence, the summary output shows that on the linear predictor scale, for observations that have accessible feed storage, the intercept is lower by -1.03. Using the definition of the logit link function, this can be translated into probabilities (Chapter 10). The final GLM model is given by the following three steps.

1. Let Y_{is} be the value of the variable Signs.in.yard for farm i at time s . We assume that Y_{is} follows a binomial distribution with probability p_{is} ; hence, $E(Y_{is}) = p_{is}$ and $\text{var}(Y_{is}) = p_{is} \times (1 - p_{is})$.
2. The systematic component is given by: $\eta_{is} = -2.58 + 0.28 \times \text{No.setts.in.field}_{is} - 1.03 \times \text{fFeed.store}_{is}$.
3. The link between the expected value of Y_{is} and η_{is} is the logistic link: $\text{logit}(p_{is}) = \eta_{is}$.
4. All observations are independent.

22.4 GEE Results

This time around, we call GEE and fit, in addition to the previous model, an auto-regressive structure to the within-farm observations. As discussed in the introduction, we deliberately choose the same set of explanatory variables for optimal comparison between the statistical methods. An alternative option is to start from scratch with all explanatory variables and apply a new model selection. The following R code was used.

```
> B.gee <- geeglm(Signs.in.yard ~ No.setts.in.fields +
  fFeed.store, family = binomial,
  id = farm.code.numeric, corstr = "ar1",
  waves = Survey, data = Badgers)
> summary(B.gee)

Mean Model:
Mean Link:           logit
Variance to Mean Relation: binomial

Coefficients:
              estimate     san.se      wald      p
(Intercept) -2.97581231  0.53278887 31.196134543 2.332300e-08
No.setts.in.fields   0.21951360  0.06936777 10.013994983 1.553552e-03
fFeed.store1       0.01389024  0.40863960  0.001155416 9.728840e-01

Scale is fixed.
Correlation Model:
Correlation Structure:    ar1
Correlation Link:         identity

Estimated Correlation Parameters:
            estimate     san.se      wald      p
alpha 0.4901059  0.1137123 18.57656 1.632153e-05

Returned Error Value:      0
Number of clusters:      36  Maximum cluster size: 8
```

Note that the package `geepack` is not part of the base installation of R, and you need to download and install it. The `summary` command shows that the number of setts in the field is significant.

For optimal comparison with the GLM, we set the scale parameter ϕ equal to 1 (for binary data it does not make sense to correct for overdispersion). Note that the presence of the accessible feed store is no longer significant. Also the number of setts in the field is less significant ($p = 0.0015$ for the GEE and $p = 6.81 \times 10^{-13}$ for the GLM). The auto-correlation is moderate with a value of 0.49. Its standard error is small, indicating that the correlation is significant. However, the literature is not clear on the use of this standard error, and some packages will not print it. References were given in Chapter 12 that can be used to compare GEE models with and without a correlation structure.

Summarising, we can see that in comparison with the GLM approach, the GEE gives a more conservative result. Both models find the variable ‘no. setts in fields’ to be significant (although the p -value is lower, i.e. stronger association in the GLM). However, the GEE finds this to be the only significant variable, the GLM also gives ‘accessible feed store present’ as a significant predictor. This highlights the general effect to be expected by adjusting for inherent auto-correlation, i.e. more conservative results. This is particularly important in this example because the stepwise regression has an inherently high risk of including spurious explanatory variables in the final model. This result is not surprising given the multiple testing involved. (We could get round this by making some kind of adjustment in terms of significance thresholds, e.g. Bonferroni correction.)

The estimated correlation parameter indicates the presence of auto-correlation between within-farm observations, justifying our decision to use GEE. This is further evidence that the association between ‘accessible feed store present’ and ‘signs of badger activity’ as indicated by the original GLM model was statistically spurious. The final GEE model is given by the following three steps.

1. Let Y_{is} be the value of the variable Signs.in.yard for farm i at time s . We assume that $E(Y_{is}) = p_{is}$ and $\text{var}(Y_{is}) = p_{is} \times (1 - p_{is})$.
2. The systematic component is given by $\eta_{is} = -2.97 + 0.21 \times \text{No.setts.in.field}_{is} + 0.01 \times \text{fFeed.store.present}_{is}$. The link between the expected value of Y_{is} and η_{is} is the logistic link: $\text{logit}(p_{is}) = \eta_{is}$.
3. The correlation between Y_{is} and Y_{ik} is given by $\text{cor}(Y_{is}, Y_{ik}) = 0.49^{|s - k|}$.

22.5 GLMM Results

To compare results obtained with GEE and GLMM, which we also applied, the following R code was used. Again, for optimal comparison of the statistical methods, we used the same set of explanatory variables.

```
> library(lme4)
> B.glmm <- lmer(Signs.in.yard ~ No.setts.in.fields +
+ fFeed.store + (1 | farm.code.numeric),
+ family = binomial, data = Badgers)
```

The results obtained by the `summary(B.glmm)` command are given below.

Random effects:

Groups	Name	Variance	Std.Dev.
farm_code_numeric	(Intercept)	5.32	2.30

Estimated scale (compare to 1) 0.77

Fixed effects:

	Estimate	SE	z-val	p-val
(Intercept)	-5.34	0.98	-5.40	<0.001
No_setts_in_fields	0.37	0.10	3.38	0.0007
fFeed.store1	0.28	0.70	0.39	0.69

If we are happy to accept the random effect structure used in this model, then we again arrive at the same conclusion that number of setts in the fields is an important predictor of signs of badger activity. The *p*-value is slightly lower here, suggesting that in this instance at least, the GEE was the most conservative approach.

In both these models (GEE and GLMM), the coefficient for the relationship between number of setts and probability of observing signs of badger activity is positive, but note that the GLMM result was stronger (+0.21 for the GEE and +0.37 for the GLMM).

Note that the GLMM does estimate an overdispersion parameter ϕ , and the software does not allow you to set it to 1 (as would be normal for binary data). The final GLMM model is given by the following three steps.

1. Let Y_{is} be the value of the variable Signs_in_yard for farm i at time s . We assume that Y_{is} is binomial distributed with $E(Y_{is}) = p_{is}$ and $\text{var}(Y_{is}) = p_{is} \times (1 - p_{is})$.
2. The systematic component is given by: $\eta_{is} = -5.34 + 0.37 \times \text{No_setts_in_field}_{is} + 0.28 \times \text{fFeed.store}_{is} + \varsigma_i$, where ς_i is a random intercept with mean 0 and variance σ_ς^2 , which is estimated as 5.32.
3. The link between the expected value of Y_{is} and η_{is} is the logistic link: $\text{logit}(p_{is}) = \eta_{is}$.

22.6 Discussion

This simple example highlights how three different approaches can give three similar, but different results – and different in important respects. As stated earlier, by ignoring the inherent within-farm auto-correlation, we increase the risk of type I error. This is probably why ‘accessible feed store present’ was significant only in the first under-specified model.

In terms of inference, if we are happy to choose the GEE from the approaches tried here, we can say first of all that there is auto-correlation between within-farm observations with respect to observing signs of badger activity. This is not

surprising. On average, we are talking about a 3-month separation in time. So, if signs of badger activity are observed at one visit – it is easy to imagine that there will be a good chance of making the same observation 3 months later (and vice versa for non-observations). But the probability of making the same finding diminishes with time; so if we go back to the same farm, maybe 18 months later, then the chance of observing the same result is less compelling. Hence, the choice of the 1st-order autoregressive structure.

Having chosen what we hope is a suitable auto-correlation structure, we find that ‘number of sets in fields’ is a significant predictor and in a positive direction ($\beta = 0.21$, s.e. = 0.06). This is of course an intuitive result, i.e. the more badger sets observed close to the farm, the more likely that badger signs will be observed on the farm. This may seem at first glance an obvious conclusion. However, it offers support to our choice of model, and of equal importance, it gives insight into the variables not important in predicting badger activity on farms.

We should not forget that the correlation structure may be due to a missing covariate or interaction. The problem is that it is rather difficult to decide which interaction term to include as there are so many options. Good biological knowledge is required when considering which interactions to fit.

As stated in the introduction, the GEE and GLMM were applied on a selection of covariates that was determined by the GLM model selection. This is against our protocol presented in Chapters 4 and 5. Our motivation for this approach was explained in Section 22.1: to show that GLM gives a model with inflated p -values. If you want to find the optimal GEE (or GLMM) model, you should apply the model selection using these models! Because we were curious ourselves, we applied a model selection using GEE and GLMM. With the GEE, we ended up with a model that only contains the covariates ‘number of sets in a field’ and ‘presence/absence of sugar beets’. The GLMM picked only ‘number of sets in a field’. Hence, adding a dependence structure on the data gives a different set of covariates in the model selection, and the type of dependency (auto-regressive correlation from the GEE versus the symmetrical compound correlation from GLMM) also plays a role. Thus, it is important to give careful consideration to choice of correlation structure in advance of any analysis.

22.7 What to Write in a Paper

This depends of course on the journal and the audience. In general, most readers of ecological journals will not be interested in the more technical details of procedures such as GEE. A line or two on the reason for using a GEE (auto-correlation, non-standard data, e.g. binary or count) needs to be included, even when submitting to the most non-technical of journals.

All relevant parameters are given, by convention, along with the standard errors. This includes the auto-correlation parameter.

Chapter 23

Incorporating Temporal Correlation in Seal Abundance Data with MCMC

A.A. Saveliev, M. Cronin, A.F. Zuur, E.N. Ieno, N.J. Walker, and G.M. Smith

23.1 Introduction

Common or harbour seals (*Phoca vitulina* L.) are semi-aquatic mammals that spend time onshore at terrestrial sites where they haul-out to rest, breed, moult, engage in social activity, and escape predation (Fig. 23.1).

Over one-third of harbour seals in Ireland use haul-out sites in the southwest region (Cronin et al., 2007). Most of the haul-out sites in this region are located within Bantry Bay and the Kenmare River. Special Areas of Conservation (SACs) have been designated at each of these sites in accordance with the EU Habitats Directive. Assessing the year round changes in harbour seal abundance within SACs contributes to the monitoring obligations under the Habitats Directive and to the understanding of national population trends.

Between April 2003 and November 2005, regular standardised haul-out count surveys of both bays were carried out by boat. Counts of seals at each haul-out site were carried out independently and simultaneously by two observers, initially from a distance of approximately 200 m from the haul-out site and at progressively closer ranges whilst minimising disturbance to the seals. Surveys were carried out at least monthly all year-round and weekly during the summer and autumn, weather permitting. Surveys were scheduled to occur within two hours either side of low tide and during daylight hours.

In the original analysis of these data, Cronin (2007) used additive mixed modelling techniques. Residual auto-correlation structures and residual heterogeneity structures were included, albeit in the context of a Gaussian distribution. In this chapter, we do a similar statistical analysis, but replace the Gaussian distribution with a Poisson distribution because the response variable is a count (the number of seals). However, current software for generalised linear mixed modelling do not easily allow incorporating temporal correlation, and therefore, we use Markov

A.A. Saveliev (✉)
Kazan State University, Kazan, 420008, Russia

Fig. 23.1 Mother and pup hauling out. The photo was taken by Michelle Cronin



Chain Monte Carlo (MCMC) techniques to fit a model that contains a temporal correlation structure.

MCMC is based on Bayesian statistics: Something not every reader will be familiar with. Providing an introduction to Bayesian statistics is challenging and including one in a 20-page case study is near impossible. We could have simply dropped this chapter, but considered it important to include a final chapter showing there is life beyond the mixed modelling techniques discussed in earlier chapters. So, although we cannot give a detailed introduction to Bayesian statistics or MCMC, we can discuss salient points and recommend you spend some time digging a bit deeper into this approach. A good starting point for ecologists is McCarthy (2007). More technical references are given later in this chapter.

So, the aim of this chapter is to show how you can include an auto-regressive correlation structure in a generalised linear model (GLM) for count data (e.g. using the Poisson or negative binomial distribution). Our choice of a GLM instead of a generalised additive model (GAM) is that we already have various GAM case study chapters and we wanted a better balance between parametric and non-parametric case studies. Also, using the R functions from Chapter 3, a smoothing spline can easily be programmed inside a GLM; so you could program the GAM equivalent of our MCMC approach (relatively) easily yourself. Further details and references on MCMC and GAM can be found in Keele (2008). Furthermore, we only focus on the correlation aspects of the model, not on the selection of the optimal model in terms of the explanatory variables. We therefore take the explanatory variables of the optimal model presented in Cronin (2007), and use these in the GLM.

23.2 Preliminary Results

As indicated in the introduction of this chapter, the same data were analysed in Cronin (2007) using an additive mixed modelling with a Gaussian distribution. The optimal model was of the form

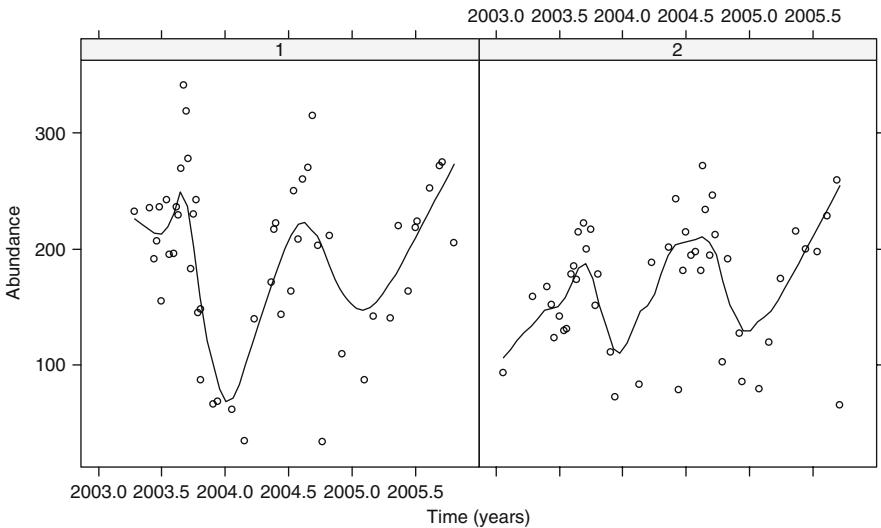


Fig. 23.2 Graph showing the observed abundances versus time (expressed as year + (week - 1)/52, for both sites). To aid visual interpretation, a LOESS smoother was added. The haul-out sites are Bantry Bay (site 1) and Kenmare River (site 2). Note the seasonal pattern at both sites

$$A_i = f(\text{Month}_i, \text{TDay}_i) + \text{WindDir}_i + \text{Site}_i + \varepsilon_i \quad \text{where } \varepsilon_i \sim N(0, \sigma_{\text{season}}^2) \quad (23.1)$$

A_i is the abundance of seals for observation number i . The explanatory variables month and time of day (expressed as TDay in the formula above) were fitted using a two-dimensional smoother $f(\text{Month}_i, \text{TDay}_i)$, allowing the day effect to change over the year. The model also contains two nominal variables: wind direction and site. Wind direction was categorised into 1 = North/Northeast, 2 = East/Southeast, 3 = South/Southwest, 4 = West/Northwest, and the haul-out sites are Bantry Bay (site 1) and Kenmare River (site 2). The residuals showed heterogeneity per season; so different residual variances per season (using the varIdent function from Chapter 4) were used.

Cronin (2007) also applied models with residual auto-regressive correlations of order 1 (using corAR1, see Chapter 6), but these were less optimal as judged by the AIC. The following R code reads the data and draws the time series plot shown in Fig. 23.2.

```
> library(AED); data(Seals)
> Seals$fSite <- factor(Seals$Site)
> Seals$Time <- Seals$Year + (Seals$Week - 1) / 52
> library(lattice)
> xyplot(Abun ~ Time | fSite, data = Seals,
  ylab = "Abundance", xlab = "Time (years)",
```

```
panel = function(x, y) {
  panel.loess(x, y, span = 0.3, col = 1)
  panel.xyplot(x, y, col = 1)}
```

We defined the variable Time as the year plus the week number (minus 1) divided by 52. Minus 1 ensures that an observation made in week 52 is still in the same year as an observation from week 1 or 51. The `xyplot` from the lattice package was used to make the figure, and it contains specific functions to add a LOESS smoother with a span of 0.3; higher values for the span resulted in smoothers that did not capture the seasonal pattern in the data.

Following Cronin (2007), we applied the GAM formulated in Equation (23.1), and the R code is given next. Instead of the long name `Timeofday`, we used a shorter notation, namely `TDay`. The dataset does not contain a variable `Season`; we created this variable using the following code:

```
> Seals$fSeason<-Seals$Month
> I1<-Seals$Month==1 | Seals$Month==2 | Seals$Month==12
> I2<-Seals$Month==3 | Seals$Month==4 | Seals$Month==5
> I3<-Seals$Month==6 | Seals$Month==7 | Seals$Month==8
> I4<-Seals$Month==9 | Seals$Month==10 | Seals$Month==11
> Seals$fSeason[I1] <- "a"
> Seals$fSeason[I2] <- "b"
> Seals$fSeason[I3] <- "c"
> Seals$fSeason[I4] <- "d"
> Seals$fSeason <- as.factor(fSeason)
```

The same was done for the wind direction:

```
> Seals$fWind2 <- Seals$Winddir
> Seals$fWind2[Seals$fWinddir==1|Seals$fWinddir==2]<-1
> Seals$fWind2[Seals$fWinddir==3|Seals$fWinddir==4]<-2
> Seals$fWind2[Seals$fWinddir==5|Seals$fWinddir==6]<-3
> Seals$fWind2[Seals$fWinddir==7|Seals$fWinddir==8]<-4
> Seals$fWind2 <- factor(Seals$fWind2)
```

Once we have this variable, the additive mixed model is run with the code below and also produces Fig. 23.3 and the numerical output.

```
> library(mgcv)
> Seals$TDay <- Seals$Timeofday
> fSeason2 <- Seals$fSeason #Avoids an error message
#from varIdent
> M1 <- gamm(Abun ~ s(Month,TDay) + fWind2 + fSite,
weights = varIdent(form =~ 1 | fSeason2),
data = Seals)
> plot(M1$gam, pers = TRUE)
> anova(M1$gam)
```

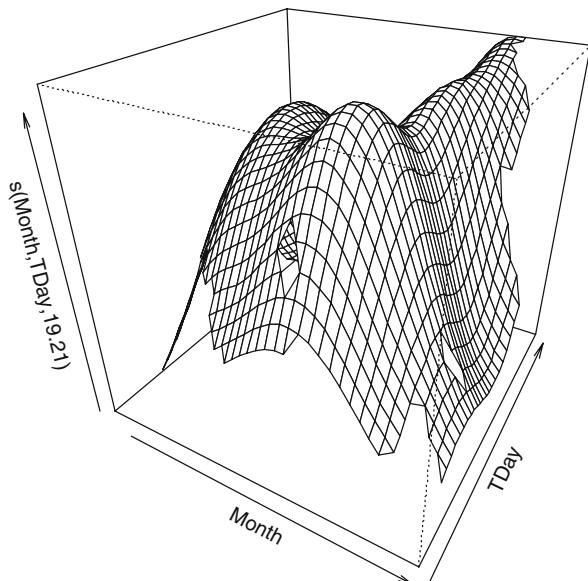


Fig. 23.3 Two-dimensional smoother for month and time of the day. The shape of the two-dimensional smoother indicates that an interaction is needed. A tunnel-type shape indicates that two additive main terms suffice

The output from the `anova` command is give below. There was a significant effect of month and time of day ($p < 0.001$), site effect ($p = 0.001$), and wind direction ($p = 0.005$) on the abundance of seals at haul-out sites in Bantry Bay and Kenmare River.

The summary command can be used to obtain the estimated parameters for each wind direction and site.

Parametric Terms:

	df	F	p-value
Wind2	3	4.526	0.00573
fSite	1	10.537	0.00176

Approximate significance of smooth terms:

	edf	Est.rank	F	p-value
s(Month, TDay)	19.21	29.00	17.08	<2e-16

23.3 GLM

There are a few things that we would like to improve in Equation (23.1). First, we want to work with a distribution for count data, and the most obvious ones are the Poisson and negative binomial distributions. Recall from Chapter 9 that a GAM with a Poisson distribution is specified by the following three steps.

1. A_i is Poisson distributed with mean μ_i . In mathematical notation: $A_i \sim P(\mu_i)$. As a result, we have $E(Y_i) = \mu_i = \text{var}(Y_i)$.
2. The systematic component is given by

$$\eta_i = f(\text{Month}_i, \text{TDay}_i) + \text{WindDir}_i + \text{Site}_i$$

3. The relationship between the mean μ_i and systematic component is specified by the logarithmic link function: $\log(\mu_i) = \eta_i$, which can also be written as $\mu_i = \exp(\eta_i)$.

In order to switch from a GAM to a GLM, we need to replace the $f(\text{Month}_i, \text{TDay}_i)$ by something parametric. One option is to use

$$\eta_i = \text{Month}_i + \text{Month}_i^2 + \text{TDay}_i + \text{TDay}_i^2 + \text{Month}_i \times \text{TDay}_i + \text{WindDir}_i + \text{Site}_i$$

This predictor function uses quadratic functions for month and time of the day and an interaction between these main terms. Later, in the discussion for this chapter, alternatives are mentioned. The following code applies the GLM. The `scale` function applies a standardisation (subtract the means and divide by the standard deviation) and avoids collinearity between Month and Month² and also between TDay and TDay².

```
> Seals$Month1 <- as.double(scale(Seals$Month))
> Seals$Month2 <- Seals$Month1^2
> Seals$TDay1 <- as.double(scale(Seals$TDay))
> Seals$TDay2 <- Seals$TDay1^2
> M2 <- glm(Abun ~ Month1 + Month2 + TDay1 + TDay2 +
  Month1:TDay1 + fWind2 + fSite,
  data = Seals, family = poisson)
> summary(M2)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 5.567644   0.021865 254.640 < 2e-16
Month1      -0.016750   0.009541  -1.756  0.07916
Month2      -0.208159   0.008039 -25.894 < 2e-16
TDay1       0.012404   0.007811   1.588  0.11229
TDay2      -0.054142   0.007000  -7.734 1.04e-14
fWind22     -0.228104   0.027422  -8.318 < 2e-16
fWind23     -0.053383   0.023327  -2.288  0.02211
fWind24     -0.059348   0.021829  -2.719  0.00655
fSite2      -0.133858   0.015293  -8.753 < 2e-16
Month1:TDay1 0.108726   0.011211   9.698 < 2e-16

Dispersion parameter for poisson family taken to be 1
Null deviance: 2502.7 on 97 degrees of freedom
Residual deviance: 1198.1 on 88 degrees of freedom
AIC: 1900.3
```

Note that there is overdispersion because the ratio of the residual deviance (1198.1) and residual degrees of freedom (88) is larger than 1! Instead of presenting a quasi-Poisson model, we will add an auto-correlation structure later in this chapter.

23.3.1 Validation

The hypothesis testing approach for the GLM also assumes independence of the residuals. Figure 23.4 shows a graph of the Pearson residuals plotted against time for both sites. Independence implies that we should not see any patterns in these panels, but we can, especially at site 1. Panel 2 also shows a general increasing trend: more negative residuals in the first year and more positive residuals in 2005. Instead of a subjective LOESS smoother, we can also use semi-variograms (or an auto-correlation function for regular spaced data) to test whether there is a significant temporal correlation, but we leave this as an exercise for the reader (see Chapters 6 and 7 for the R code). The following R code was used to create Fig. 23.4.

```
> Seals$E2 <- resid(M2, type = "pearson")
> xyplot(E2 ~ Time | fSite, data = Seals,
  ylab = "Pearson residuals", xlab = "Time (years)",
  panel = function(x, y) {
    panel.loess(x, y, span = 0.5, col = 1)
    panel.xyplot(x, y, col = 1)})
```

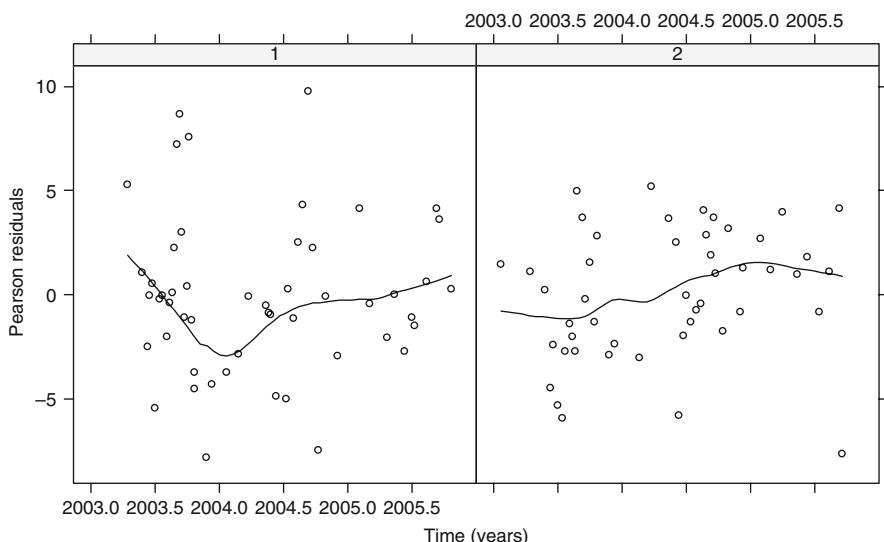


Fig. 23.4 Pearson residuals of the Poisson GLM plotted versus time for each site. The LOESS smoother indicates violation of independence over time

The first line extracts the Pearson residuals from the GLM object, and the rest is the familiar `xypplot` command with its options.

23.4 What Is Bayesian Statistics?

The methods discussed in previous chapters cannot easily be used to include a temporal correlation in a Poisson GLM, except perhaps for the `gamm` function, but it can cope with models that contain correlation structures and no random effects (e.g. random intercepts or slopes). However, as we said before, we want to do the analysis in a parametric context, and therefore, we now introduce techniques based on the Bayesian approach.

Before introducing Bayesian statistics, we will give a brief summary of the main characteristics of frequentist statistics, which is the approach used so far for analysing data. By this we mean that we have adopted a philosophy where we formulate a hypothesis for the regression parameters, apply the model in Equation (23.1) or its parametric equivalent, and estimate parameters, standard errors, 95% confidence intervals, and p -values. We can then say that if we were to repeat this experiment a large number of times, in 95% of cases, the real population regression parameters would lie inside the estimated confidence intervals. Furthermore, the p -values tell us how *often* (hence: frequency) we find an identical or larger test statistic. Key elements of frequentist statistics are as follows:

- The parameters (such as mean, variance, and regression parameters) that determine the behaviour of the population are fixed, but unknown.
- Based on observed data, these unknown parameters are then estimated in such a way that the observed data agree well with our statistical model; in other words, the parameter estimates are chosen such that the likelihood of the data is optimised (this is maximum likelihood estimation).
- Frequentist approaches are objective in that only the information contained in the current data set is used to estimate the parameters.

Bayesian statistics is based on a different philosophy. The main difference is assuming that the parameters driving the behaviour of the population are no longer fixed. Instead, it is assumed the parameters themselves follow some statistical distribution. To better explain this, we need to look at some theory.

23.4.1 Theory Behind Bayesian Statistics

The main components of Bayesian statistics are as follows.

Data. Suppose we have a stochastic variable Y with density function $f(Y | \boldsymbol{\theta})$, and let $\mathbf{y} = (y_1, \dots, y_n)$ denote n observations of Y . Now $\boldsymbol{\theta}$ is a vector containing unknown parameters which are to be estimated from the observed data \mathbf{y} . In the case of the Poisson model described in Chapter 9, \mathbf{y} would be the abundance observations and $\boldsymbol{\theta}$ the regression coefficient and overdispersion parameter.

Likelihood: The density $f(\mathbf{y} | \boldsymbol{\theta}) = \prod_i f(y_i | \boldsymbol{\theta})$ is the likelihood. When maximum likelihood estimation is carried out, $\boldsymbol{\theta}$ is chosen to maximise $f(\mathbf{y} | \boldsymbol{\theta})$. For example, in Chapter 9, we showed that for data following a Poisson distribution, the maximum likelihood estimates of the parameters are obtained from calculating the derivatives, setting those to zero, and solving the resulting equations.

Prior distribution: The major difference in Bayesian statistics is that instead of assuming that $\boldsymbol{\theta}$ is an unknown, but fixed, parameter vector, we now assume that $\boldsymbol{\theta}$ is stochastic. The distribution of $\boldsymbol{\theta}$ before the data are obtained, is called the prior distribution, and we denote it by $\pi(\boldsymbol{\theta})$. It reflects knowledge about $\boldsymbol{\theta}$, perhaps obtained from previous experiments, but it is also possible to choose $\pi(\boldsymbol{\theta})$ such that it reflects very little knowledge about $\boldsymbol{\theta}$ so that the posterior distribution is mostly influenced by the data. If the latter is the case, we say that the prior distribution is vague. (The term non-informative is also commonly used in reference to vague prior distributions.)

Posterior distribution: This forms the final component of a Bayesian analysis setting. Using some simple statistical theory (namely, Bayes' Theorem), the prior information is combined with information from the data to give us the posterior distribution $\pi(\boldsymbol{\theta} | \mathbf{y})$. It represents the information about $\boldsymbol{\theta}$ after observing the data \mathbf{y} :

$$\pi(\boldsymbol{\theta} | \mathbf{y}) = f(\mathbf{y} | \boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}) / \pi(\mathbf{y}) \propto f(\mathbf{y} | \boldsymbol{\theta}) \pi(\boldsymbol{\theta})$$

The latter follows because $\pi(\mathbf{y})$, which is the marginal density of the data, is constant. In contrast to maximum likelihood, where a point estimate for $\boldsymbol{\theta}$ is obtained, with Bayesian statistics a density of $\boldsymbol{\theta}$ is the final result. This density averages the prior information with information from the data. Gelman et al. (2003) provide an accessible book covering the basics of Bayesian analysis.

We have now discussed the three main components of Bayesian statistics: The prior distribution of $\boldsymbol{\theta}$, our observed data \mathbf{y} , and finally, how these two pieces of information are combined to obtain the posterior distribution of $\boldsymbol{\theta}$. In only a limited number of cases is the posterior distribution of a known form. More often than not, this distribution is complex, making it difficult to obtain summaries easily. For many years, this was the main reason why Bayesian statistics was not widely used. However, with the advent of computers, simulation tools such as Markov Chain Monte Carlo (MCMC) have become widely available and Bayesian analysis is more accessible. The development of freeware software implementing such simulation tools has also helped greatly to popularise Bayesian approaches.

23.4.2 Markov Chain Monte Carlo Techniques

The aim is to generate a sample from the posterior distribution. This is the Monte Carlo bit. Often, the exact form of the posterior distribution is unknown, but

fortunately another stochastic device, the Markov chain, can be used to deal with this. Assume we start with an initial value for θ , denoted by θ_0 . Then the next state of the chain θ_1 is generated from $P(\theta_1 | \theta_0)$, where $P(\cdot | \cdot)$ is the so-called transition kernel of the chain. Then, θ_2 is generated from $P(\theta_2 | \theta_1)$, ... and θ_t is generated from $P(\theta_t | \theta_{t-1})$. Under certain regularity conditions, the distribution of $P(\theta_t | \theta_0)$ will converge to a unique stationary distribution $\pi(\cdot)$. One important property of a Markov chain is that once it has reached its stationary distribution, it would have ‘forgotten’ about its initial starting value; so it no longer matters how inappropriate our initial value θ_0 was.

Assuming we can define an appropriate Markov chain (that is, an appropriate distribution $P(\theta_t | \theta_{t-1})$ can be constructed), we can then generate *dependent* draws (or realisations) from the posterior distribution. The samples are not independent as the distribution of θ_t depends on the value of θ_{t-1} . In turn, the distribution of θ_{t-1} depends on the value of θ_{t-2} and so on. This has the following consequences:

- The initial part of the chain should be discarded (this initial part is commonly referred to as ‘burn-in’) so that the influence of an arbitrary initial value θ_0 is eliminated.
- The MCMC samples are less variable compared to independent samples, and therefore, the variance of estimated summary statistics, such as the sample mean, is larger than would be the case if the samples had been independent.
- When stationarity has been reached (that is, the realisations no longer depend on the initial value θ_0), a large number of samples is needed to cover the entire region of the posterior distribution as small portions of consecutive samples tend to be concentrated in small regions of the posterior distribution.

When we generate many samples after the burn-in samples have been discarded, these will then be distributed appropriately from the entire posterior distribution. This distribution can be summarised by summary statistics such as the sample mean and sample quantiles. A useful property of MCMC is that statistics calculated from the MCMC sample will converge to the corresponding posterior distribution quantities; for example, the sample mean converges to the posterior mean and the sample quantiles converge to the posterior quantiles.

It may seem complicated to generate samples from the posterior distribution, but fortunately there are algorithms available that make this task easy. The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) and the Gibbs sampler (Geman and Geman, 1984; Gelfand and Smith, 1990), which is a special case of the former, are two commonly used algorithms for creating appropriate Markov chains. Gilks et al. (1996) provide an accessible introduction to the various MCMC techniques illustrated with many examples. Although these algorithms are easily programmed in R, there are many technical complexities, and it is better to use specialised software, such as the freeware package WinBUGS (Lunn et al., 2000; www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml). The R package BRugs is an interface to WinBUGS, and this is what we used for our seal abundance example.

23.5 Fitting the Poisson Model in BRugs

First, we have to recast the GLM model specification, given in Section 23.3, into a Bayesian framework. The abundance data are modelled as

$$\begin{aligned} A_i &\sim \text{Poisson}(\mu_i) \\ \eta_i &= \alpha + \beta_1 \times \text{Month}_i + \beta_2 \times \text{Month}_i^2 + \beta_3 \times \text{TDay}_i \\ &\quad + \beta_4 \times \text{TDay}_i^2 + \beta_5 \times \text{Month}_i \times \text{TDay}_i + \text{Winddir}_i + \text{Site}_i \\ \log(\mu_i) &= \eta_i \end{aligned} \tag{23.2}$$

This is the same as before. To make the model Bayesian, the unknown parameters are assigned prior distributions, as follows:

$$\begin{aligned} \alpha &\sim N(0, 10^6) \\ \beta_1, \dots, \beta_5 &\sim N(0, 10^6) \\ \text{Winddir}, \text{Site} &\sim N(0, 10^6) \end{aligned} \tag{23.3}$$

The notation $\text{Windir}, \text{Site} \sim N(0, 10^6)$ means that we assume all the regression parameters for the individual levels of these nominal variables are normally distributed. *A priori*, we assume that the unknown parameters are zero on average, but with a large variance so that the parameters can take on values anywhere between -2000 and $+2000$. This reflects that we are rather ‘vague’ about what we believe about the parameters before seeing the data and is often referred to as a vague or non-informative prior distribution. To complete the Bayesian approach, we now have to obtain the posterior distribution and we will use MCMC to do this.

23.5.1 Code in R

Applying MCMC in R is relatively easy, although it takes more programming effort than the frequentist approach via the `glm` function, and it is also more time consuming in terms of computing.

First you have to install the packages `coda` and `BRugs` from the R website. To implement the Bayesian Poisson model, you need to create a couple of ASCII text files containing the model, data, and initial parameter estimates. Then the following code is run in R:

```
> library(coda)
> library(BRugs)
> modelCheck("Modelglm1.txt")
> modelData("Sealmatrix.txt")
> modelCompile(numChains = 3)
> modelInits("InitializeParam1.txt")
```

```
> modelInits("InitializeParam2.txt")
> modelInits("InitializeParam3.txt")
> #Burn in
> samplesStats("alpha")
> modelUpdate(200, thin = 50)
> plotHistory("alpha", colour = c(1, 1, 1))
> #Monitor model parameters
> dicSet()
> samplesSet("alpha")
> samplesSet("b")
> samplesSet("W")
> samplesSet("S")
> modelUpdate(10000, thin = 10)
> dicSet()
> samplesStats("alpha")
> samplesStats("b")
> samplesStats("W")
> samplesStats("S")
```

As you can see, this requires more code than the `glm` command in Chapter 9! And it also takes longer to run. Let us go over these commands in more detail. First of all, the file `Sealmatrix.txt` contains the data and is given on our website. The website also contains a small macro to prepare the data in the required format. The remaining components of the code are described in detail in the following sections.

23.5.2 Model Code

The file `Modelglm1.txt` forms the heart of the MCMC code, and contains the following lines.

```
model{
  for(i in 1:98) {
    Abun[i] ~ dpois(mu[i])
    log(mu[i]) <- alpha +
      Month1[i] * b[1] + Month2[i] * b[2] +
      TDay1[i] * b[3] +
      TDay2[i] * b[4] +
      Month1[i] * TDay1[i] * b[5] +
      W[Wind2[i]] + S[Site[i]]
  }
  alpha ~ dnorm(0, 1.0E-6)
  for(j in 1:5) {
    b[j] ~ dnorm(0.0, 1.0E-6)
  }
```

```

for (i in 2:4) {
  W[i] ~ dnorm(0.0, 1.0E-6)
}
W[1] <- 0
S[1] <- 0
S[2] ~ dnorm(0.0, 1.0E-6)
}

```

Assuming you have read all previous 22 chapters, this code should not appear too alien. The first block of code specifies that the abundance at site i is Poisson distributed with mean μ_i . The log link is used and the right hand side of the equation, following ‘alpha +’, is simply the model in terms of the explanatory variables. The rest of the code, from $\alpha \sim \text{dnorm}(0, 1.0E-6)$ onwards specifies uninformative prior distributions for all parameters. The code including $W[1]$ and $S[1]$ ensures that the baseline levels of W (wind direction) and S (site) are nominal variables. It is important to note that the `dnorm` notation used in BUGS is different from the one used in R! The most important difference is that the variance is handed down to the `dnorm` function in terms of 1/variance. This is a Bayesian convention with 1/variance being the so-called precision of the distribution and is usually denoted by τ . So a variance of 10^6 is entered in the model framework as a precision of 10^{-6} . The reason for working with precision is that the posterior distribution of our parameters of interest will be a weighted combination of the prior distribution and the distribution of the data, with weights given by their respective precisions (i.e. 1/prior variance and 1/data variance). So, a large prior variance means that its precision is close to zero, and as a consequence, the prior distribution will receive almost no weight in deriving the posterior distribution of the parameters. This again reflects that our prior distribution is non-informative as it barely contributes to the posterior outcome. As a consequence, our posterior distribution should be similar to the one obtained from maximum likelihood estimation.

23.5.3 Initialising the Chains

Having formulated the model, we can now generate a sample from the posterior distribution using MCMC techniques. As described earlier, from a given starting value θ_0 , the MCMC routine will generate values θ_1, θ_2 , etc. When the chain is run for a sufficiently long period of time, it will have forgotten its initial starting value and from that point onward, the values drawn will represent samples from our posterior distribution of interest.

We used three chains, each of which is initialised with the `modelInits` command. The file `InitializeParam1.txt` contains the following text:

```

list(
  alpha=5.567643,

```

```

b=c(-0.016750,-0.208159,0.012404,-0.054141,0.108725),
S=c(NA, -0.133857),
W=c(NA, -0.228104,-0.053383,-0.059347)
)

```

These are the estimated values from the ordinary GLM. The content of the file InitializeParam2.txt is not presented here, but it contains similar information; we took estimated values plus twice their standard errors from the ordinary GLM. Finally, the file InitializeParam3.txt contains the estimated values minus twice their standard errors (also from the GLM). It is also possible to add some random variation around these estimated values. Hence, the initial values are based on the maximum likelihood estimates $\pm 2 \times$ standard errors. The NAs are needed for the baseline level of the factor.

The first `modelUpdate` statement in the main R code starts up the chains, and it states that every 50th realisation should be stored until 200 realisations have been kept in total per chain. The reason for not keeping all iterations is that the iterations will be auto-correlated, and therefore, do not provide much extra information on the posterior distribution. The command `plotHistory` provides a trace plot of these samples and is shown in Fig. 23.5 for one of the parameters: The intercept α . Note how the three chains start from different values and then gradually converge. The initial part, where the chains do not overlap, is the so-called burn-in period and reflects that the chains have not converged to the same stationary distribution yet. These samples will have to be discarded. Looking at Fig. 23.5 by eye, it seems that

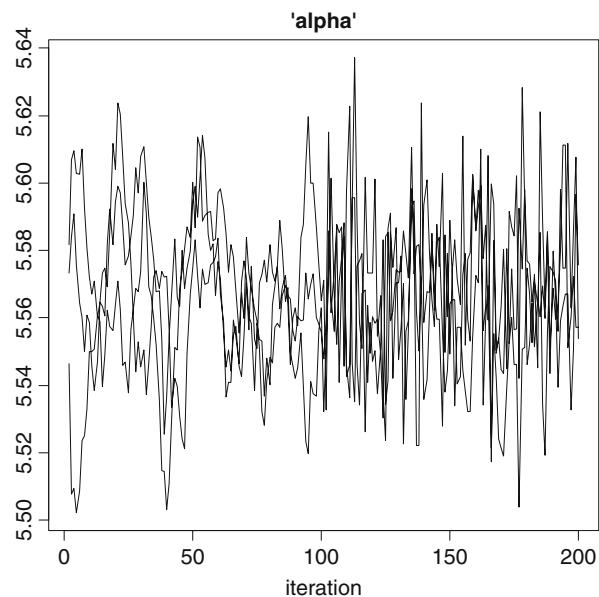


Fig. 23.5 Generated draws from the intercept α . If you run `plotHistory` with the default colours, you will see the colours mixed from different chains

a burn-in period of 150 samples (that is, $150 \times 50 = 7,500$ iterations) is enough, but we ran it for 10,000 samples to be sure.

More formal tests to assess the chain convergence are also available and are provided in the R package CODA (Plummer et al., 2008). One such test is the Gelman-Rubin statistic, which compares the variation between chains to the variation within a chain. Initially, the value of the Gelman-Rubin statistic will be large, but when convergence has been reached, it will have decreased to a value close to 1. Gelman (1996) suggests a value less than 1.1 or 1.2 is acceptable. This test should be applied to each of the model parameters. For our data, the Gelman-Rubin statistic was less than 1.2 for all parameters after 10,000 iterations, implying that from iteration 10,000 onwards, the draws come from the posterior distribution. Other tests (not shown here; full codes can be found at the book web site) reached similar conclusions.

23.5.4 Summarising the Posterior Distributions

Having decided on a burn-in of 10,000 iterations, we now want to formally keep the realised samples from iteration 10,000 onwards. This is achieved with the command `samplesSet`. The command `dicSet` allows for monitoring of the so-called Deviance Information Criterion, which can be used for model selection. Using the `modelUpdate` command, the chains are run for another 100,000 iterations. The R command `samplesStats` provides the following output.

```
> samplesStats("alpha")
      mean      sd MC_error val2.5pc median val97.5pc start sample
alpha 5.568 0.02176 0.0001381    5.526 5.568      5.61    201 30000
```

The last two columns indicate when monitoring started (from the 201st sample onwards – recall that the first 200 stored samples were discarded as burn-in) and how many samples have been stored since monitoring started (10,000 samples for each of three chains). As a consequence, the summary statistics are based on 30,000 samples. To save the space, the last two columns have not been printed (start and number of samples) in the following text as they are the same as above

```
> samplesStats("b")
      mean      sd MC_error val2.5pc median val97.5pc
b[1] -0.01670 0.009474 5.184e-05 -0.035220 -0.01668 0.001873
b[2] -0.20830 0.007988 4.446e-05 -0.224100 -0.20830 -0.192700
b[3]  0.01235 0.007782 4.579e-05 -0.002963 0.01238 0.027670
b[4] -0.05423 0.006975 3.980e-05 -0.067870 -0.05421 -0.040630
b[5]  0.10880 0.011200 6.361e-05  0.086900 0.10880 0.130800

> samplesStats("W")
```

```

          mean        sd MC_error val2.5pc   median val97.5pc
W[2] -0.22850 0.02725 0.0001631 -0.28210 -0.22860 -0.175000
W[3] -0.05374 0.02324 0.0001417 -0.09905 -0.05379 -0.008206
W[4] -0.05954 0.02170 0.0001434 -0.10190 -0.05969 -0.016920

> samplesStats("S")

          mean        sd MC_error val2.5pc   median val97.5pc
S[2] -0.1339 0.01535 9.53e-05   -0.164 -0.1339   -0.1036

```

We also list the DIC statistics as an overall model fit indicator (this is similar to AIC and discussed in details at the end of chapter):

```
> dicStats()
      Dbar Dhat   DIC     pD
Abun  1890 1880 1900 9.948
total 1890 1880 1900 9.948
```

The samples from the posterior distribution are summarised by the mean, median and 2.5 and 97.5 percentiles. Note that the mean values are similar to those obtained by the `glm` command. The *standard deviation* of the posterior distribution, given under the header `sdl`, is the Bayesian equivalent of the standard error of the mean (recall that the standard error of the mean is defined as the standard deviation of the mean values if the study were to repeated many times). Again, in this case, the values are similar to those obtained from `glm`.

23.5.5 Inference

The MCMC output contains thousands of realisations of all the model parameters, and these can be used to calculate various quantities of interest. For example, the correlation between the parameters can be obtained, and is shown in Table 23.1. High correlation is commonly expected for the constant and factor effects only. If

Table 23.1 Correlation between model parameters for the Poisson model

regression coefficients associated with continuous variables (such as $b[1], b[2], \dots, b[5]$) show a high correlation, it is best to standardise these variables. This will reduce the correlation and will improve mixing of the MCMC chains so that consecutive realisations will be less dependent, shortening the burn-in period and the total number of iterations to be run (as instead of storing only every 20th iteration, for example, we can now keep every 5th iteration, for example).

We can also obtain Pearson residuals. The simplest way is to take the mean values from the MCMC samples and use these to calculate the Pearson residuals, but it is more informative to calculate the Pearson residuals for each MCMC realisation individually. In addition, we can also generate ‘predicted’ residuals for each MCMC realisation obtained from simulating abundance data from a Poisson distribution (Congdon, 2005). The latter will be properly Poisson distributed so will not display any overdispersion.

The BRugs model code (this is added to the code in the `modelglm1.txt` file presented earlier) is given below:

```
for(i in 1:N) {
  Aprd[i] ~ dpois(mu[i])
  e.obs[i] <- (Abun[i] - mu[i]) / sqrt(mu[i])
  p2.obs[i] <- e.obs[i] * e.obs[i]
  e.prd[i] <- (Aprd[i] - mu[i]) / sqrt(mu[i])
  p2.prd[i] <- e.prd[i] * e.prd[i]
}
SS <- sum(p2.obs[1:N])
SS.prd <- sum(p2.prd[1:N])
```

In the absence of overdispersion, the sum of squares will follow a $\chi^2(N)$ distribution. The summary statistics of `SS` and `SS.prd` are

	mean	sd	MC_error	val2.5pc	median	val97.5pc
<code>SS</code>	1177	12.02	0.06561	1157	1176	1203
<code>SS.prd</code>	97.97	14.08	0.08001	72.47	97.23	127.7

For comparison, the true 2.5 and 97.5% percentiles of the $\chi^2(98)$ distribution are

```
> qchisq(c(0.025, 0.975), 98)
[1] 72.50094 127.28207
```

Note that the percentiles of the simulated `SS.prd` values correspond well with the theoretical percentiles. There are two ways of assessing overdispersion. The first one is to compare the distribution of `SS` to the distribution of the predicted `SS` in the absence of overdispersion (`SS.prd`). Clearly, the two distributions do not match, indicating substantial overdispersion. The second approach is to compare the `SS` distribution to the $\chi^2(98)$ distribution, which gives exactly the same information. The average `SS`, 1177, is similar to what was observed from the `glm` fit.

Another quantity of interest is the auto-correlation in the abundance data. For each realisation of the MCMC chain, we can estimate this in R as follows (Box-Pierce auto-correlation test with Ljung-Box modification, Congdon, 2005):

```
> for(k in 1:3) {
  for (t in k+1 : N) {
    p1[k,t] <- e.obs[t] * e.obs[t - k]
  }
  auto [1, k] <- sum(p1[k, (N11 + k) : N12]) /
    sum(p2.obs[N11: N12])
  auto2[1, k] <- auto[1,k] * auto[1,k]
  auto [2, k] <- sum(p1[k, (N21 + k) : N22]) /
    sum(p2.obs[N21 : N22])
  auto2[2, k] <- auto[2, k] * auto[2, k]
}
```

This code calculates the auto-correlation up to lag 3 based on the observed residuals for each of the two sites. The calculation is somewhat simplistic in that it assumes that the time series are regular, where in practice the data were collected at irregular one to two week intervals, but at this stage we are only exploring the possibility of auto-correlation. If the auto-correlation is zero, the value `BP.s1` and `BP.s2` below are distributed approximately as $\chi^2(k)$, where $k = 3$ is number of terms in sum

```
> BP.s1 <- (N12 - N11 - 1) * sum(auto2[1, ])
> BP.s2 <- (N22 - N21 - 1) * sum(auto2[2, ])
```

The summary statistics are as follows:

	mean	sd	MC_error	val2.5pc	median	val97.5pc
<code>BP.s1</code>	9.574	0.6612	0.004345	8.377	9.54	10.96
<code>BP.s2</code>	3.687	0.5958	0.003407	2.616	3.653	4.94

```
> qchisq(c(0.025, 0.975), 3)
[1] 0.2157953 9.3484036
```

In the absence of auto-correlation, the `BP` statistic follows a $\chi^2(3)$ distribution. Clearly, the mean and 2.5 and 97.5 percentiles of `BP.s1` are well in excess of those of the $\chi^2(3)$ distribution, providing strong evidence of auto-correlation at this site. For site 2, the auto-correlation is not significant, as the 2.5–97.5 percentile range of `BP.s2` is covered by the corresponding range of the $\chi^2(3)$ statistic. It should be mentioned that this criterion is approximate.

23.6 Poisson Model with Random Effects

The Poisson model can be extended into a GLMM by adding a random term ε_i to the linear predictor:

$$\begin{aligned} A_i &\sim \text{Poisson}(\mu_i) \\ \log(\mu_i) &= \eta_i + \varepsilon_i \end{aligned} \tag{23.4}$$

where η_i is defined as before, see Equation (23.2), and it is assumed that

$$\varepsilon_i \sim N(0, 1/\tau_\varepsilon)$$

The BRugs model code for the abundance data now becomes

```
for(i in 1:98) {
  Abun[i] ~ dpois(mu[i])
  log(mu[i]) <- alpha +
    Month1[i] * b[1] + Month2[i] * b[2] +
    TDDay1[i] * b[3] + TDay2[i] * b[4] +
    Month1[i] * TDay1[i] * b[5] +
    W[Wind2[i]] + S[Site[i]] + eps[i]
  eps[i] ~ dnorm(0.0, eps.tau)
}
```

Furthermore, in addition to the prior distributions on α , b , wind effect W , and site effect S , we now also need a prior distribution on τ_ε . A common choice is to assume that $\tau_\varepsilon \sim \text{Gamma}(0.001, 0.001)$, which reflects vague prior information on τ_ε :

```
eps.tau ~ dgamma(0.001, 0.001)
```

The complete BRugs code can be downloaded from the book website. To investigate overdispersion, the sum of squares SS was monitored:

	mean	sd	MC_error	val2.5pc	median	val97.5pc
SS	100.5	14.2	0.07761	74.68	99.85	130.4

The observed range of SS values corresponds to the 2.5 and 97.5 percentiles (72.5 and 128.3, respectively) of the $\chi^2(98)$ statistic, indicating that overdispersion is no longer present. The DIC statistics have also improved:

	Dbar	Dhat	DIC	pD
Abun	783	691	875	92.02
total	783	691	875	92.02

The MCMC output contains realisations for ε_i , $i = 1 \dots 98$, so that we can look at summary statistics of these for each observation i . To illustrate, for each i the mean and standard deviation of the realisations for ε_i was calculated, and then summarised in a histogram (Fig. 23.6).

Furthermore, the pooled variance of the ε_i was also calculated; it was 0.085. This agrees well with the sampled values for $\text{eps}.sigma = 1 / \text{eps}.tau$:

	mean	sd	MC_error	val2.5pc	median	val97.5pc
eps.sigma	0.08707	0.01491	9.457e-05	0.06213	0.08555	0.1205

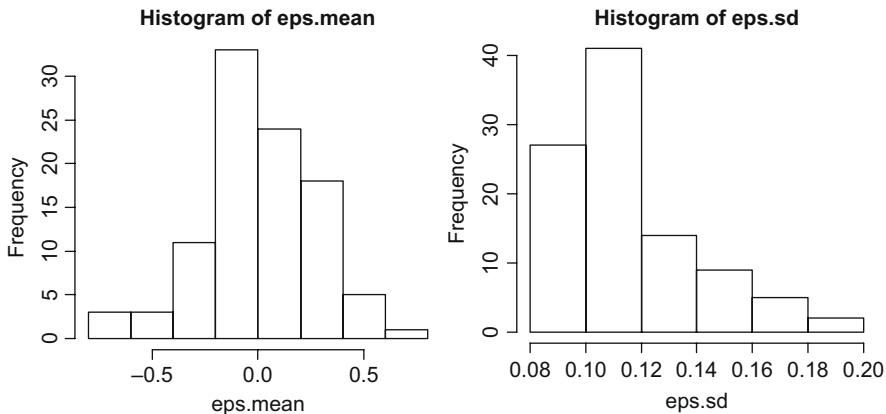


Fig. 23.6 Poisson model with random effects. For each observation i , the MCMC realisations of the residuals at the linear predictor level (ε_i , $i = 1 \dots 98$) are summarised by their mean and standard deviation. The 98 mean values and standard deviations are then plotted as a histogram

The auto-correlation summary statistics (at abundance level) are

	mean	sd	MC_error	val12.5pc	median	val97.5pc
BP.s1	2.724	2.28	0.01276	0.2018	2.147	8.616
BP.s2	2.635	2.185	0.01328	0.1985	2.073	8.233

and correspond to the $\chi^2(3)$ distribution (2.5 and 97.5 percentiles given by 0.22 and 9.35, respectively). This suggests there is no evidence of significant auto-correlation

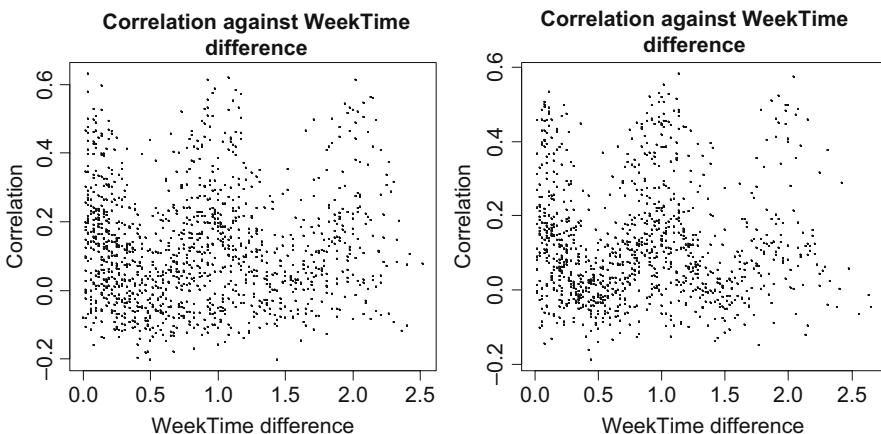


Fig. 23.7 Correlation in the random effects ε_i at the linear predictor level, calculated from MCMC results of the Poisson model with random effect, versus the lag (in weeks) between time points. The *left graph* shows the results for site 1 and the *right graph* for site 2

in the abundance data. Despite this, the correlation between the random effects, shown in Fig. 23.7, does exhibit strong pattern in time and therefore, in the next section, we investigate this further by extending the random effects model by including auto-correlation in the random effect.

23.7 Poisson Model with Random Effects and Auto-correlation

We now introduce auto-correlation into the model that allows the auto-correlation to differ between the two sites. First, we introduce a slightly different notation using subscript s for site s :

$$\begin{aligned} A_{si} &\sim \text{Poisson}(\mu_{si}) \\ \eta_{si} &= \alpha + \beta_1 \times \text{Month}_{si} + \beta_2 \times \text{Month}_{si}^2 + \beta_3 \times \text{TDay}_{si} \\ &\quad + \beta_4 \times \text{TDay}_{si}^2 + \beta_5 \times \text{Month}_{si} \times \text{TDay}_{si} + \text{Winddir}_{si} + \text{Site}_{si} \\ \log(\mu_{si}) &= \eta_{si} + \varepsilon_{si} \end{aligned} \tag{23.5}$$

and ε_{si} is a random effect. So far, this is the same model formulation as in Section 23.6. In addition, it is now assumed that ε_{si} follows an auto-regressive structure:

$$\varepsilon_{si} = \rho_s^{|W_{\text{dist}}|} \times \varepsilon_{s,i-1} + u_{si}$$

where W_{dist} is the time lag in weeks between data points $i-1$ and i . It is assumed that the u_{si} are independently and normally distributed:

$$u_{si} \sim N(0, 1/\tau_u)$$

The last part of Equation (23.5) can be rewritten into

$$\begin{aligned} \log(\mu_{si}) &= \eta_{si} + \rho_s^{|W_{\text{dist}}|} \times \varepsilon_{s,i-1} + u_{si} \\ &= \eta_{si} + \rho_s^{|W_{\text{dist}}|} \times \log(\mu_{s,i-1}) - \rho_s^{|W_{\text{dist}}|} \times \eta_{s,i-1} + u_{si} \end{aligned}$$

The full model then becomes

$$\begin{aligned} A_{si} &\sim \text{Poisson}(\mu_{si}) \\ \log(\mu_{si}) &= \eta_{si} + \rho_s^{|W_{\text{dist}}|} \times \log(\mu_{s,i-1}) - \rho_s^{|W_{\text{dist}}|} \times \eta_{s,i-1} + u_{si} \\ u_{si} &\sim N(0, 1/\tau_u) \end{aligned} \tag{23.6}$$

where η_{si} is given by Equation (23.5). Assigning prior distributions to the unknown parameters completes the Bayesian model formulation:

- $\alpha \sim N(0, 10^6)$
- $\beta_j \sim N(0, 10^6)$, for $j = 1, \dots, 5$
- $\tau_u \sim \text{Gamma}(0.001, 0.001)$

- $\rho_1 \sim \text{Uniform}[-1, 1]$
- $\rho_2 \sim \text{Uniform}[-1, 1]$

The full model code can be downloaded from our website. The summary statistics for the posterior distributions of the model parameters are as follows.

Parameter	mean	sd	2.5%	97.5%
Alpha	5.554	0.1145	5.338	5.786
b[1]	-0.07878	0.04421	-0.1661	0.006503
b[2]	-0.2238	0.0347	-0.2933	-0.1561
b[3]	0.01604	0.02676	-0.03674	0.06891
b[4]	-0.07792	0.02602	-0.1287	-0.02693
b[5]	0.1102	0.03435	0.04245	0.1776
S[2]	-0.0702	0.1018	-0.2838	0.1186
W[2]	-0.3426	0.0914	-0.5226	-0.163
W[3]	-0.0949	0.088	-0.2682	0.08002
W[4]	-0.05306	0.07948	-0.2089	0.1023
rho1	0.6395	0.1746	0.2318	0.9199
rho2	0.3545	0.3181	-0.3075	0.8311
u.sigma	0.07798	0.01372	0.05509	0.1087

The DIC statistic is similar to the previous

	Dbar	Dhat	DIC	pD
Abun	784.5	693.3	875.6	91.13
total	784.5	693.3	875.6	91.13

There is no indication of auto-correlation at the abundance level:

	mean	sd	MC_error	val2.5pc	median	val97.5pc
BP.s1	2.732	2.24	0.01306	0.2005	2.16	8.526
BP.s2	2.662	2.198	0.01146	0.1916	2.096	8.332

The random effects auto-correlation at site 1 is significant (zero is not contained in the 2.5–97.5 percentile interval for rho1), but is not significant for site 2:

	mean	sd	MC_error	val2.5pc	median	val97.5pc
rho1	0.6395	0.1746	0.002414	0.2318	0.6616	0.9199
rho2	0.3545	0.3181	0.003398	-0.3075	0.4126	0.8311

Furthermore, there is no sign of overdispersion as the SS and SSpred intervals are similar.

	mean	sd	MC_error	val2.5pc	median	val97.5pc
SS	101.8	14.37	0.08356	75.45	101.3	131.9
SS.prd	97.96	14.07	0.07975	72.43	97.34	127.6

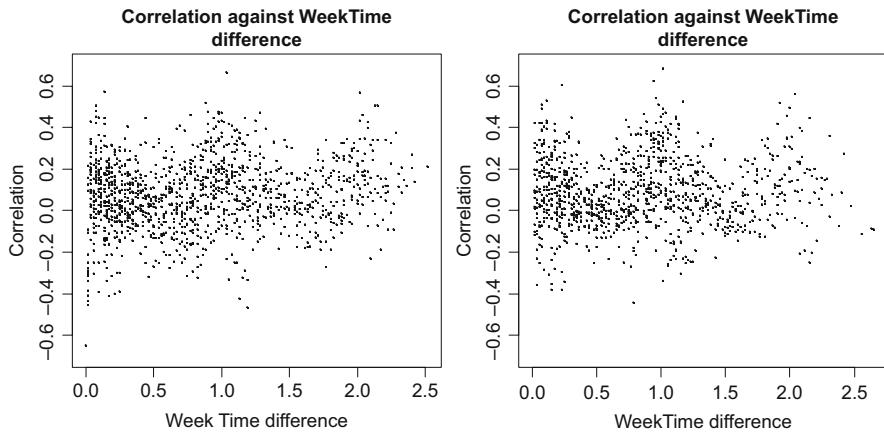


Fig. 23.8 Auto-correlation in the residuals u_{si} based on Poisson model with auto-correlated random effects (all auto-correlations over all time lags combined). The *left graph* shows the results for site 1, and the *right graph* for site 2

Finally, the auto-correlation in the residuals u_{si} was calculated and is shown in Fig. 23.8. Although the model assumes that the u_{si} are independent, the figure reveals that some auto-correlation pattern is still present (peaks at zero and multiples of year), but less severe compared to the previous model.

23.8 Negative Binomial Distribution with Auto-correlated Random Effects

Coming back to the overdispersion issue, we now use the negative binomial distribution (Chapter 9) instead of the Poisson distribution to model the overdispersion with the size parameter *size*. The variance of the abundances is given by $\mu + \mu^2/\text{size}$. To incorporate this into the Bayesian model structure, all we need to do is to change the Poisson distribution in Equation (23.6) to the negative binomial distribution:

$$A_i \sim NB(\mu_i, \text{size}) \quad (23.7)$$

As we have now introduced a new parameter, *size*, a distribution needs to be defined:

$$\text{size} \sim \text{Gamma}(0.001, 0.001)$$

This is a common choice reflecting vague prior information. The full model code can be found on our website. The posterior distributions are summarised as follows:

Parameter	mean	Sd	2.5%	97.5%
alpha	5.589	0.1055	5.39	5.81
b[1]	-0.05093	0.04148	-0.1373	0.02789
b[2]	-0.2188	0.03242	-0.2814	-0.1551
b[3]	0.02245	0.03231	-0.03867	0.08694
b[4]	-0.06466	0.03021	-0.1247	-0.005743
b[5]	0.1105	0.03911	0.03573	0.1878
S[2]	-0.1162	0.09203	-0.3104	0.04696
W[2]	-0.2827	0.1084	-0.4936	-0.06992
W[3]	-0.06669	0.0963	-0.2521	0.1244
W[4]	-0.05785	0.09231	-0.2347	0.1284
rho1	0.4544	0.4433	-0.5042	0.9796
rho2	0.2152	0.539	-0.8189	0.9669
u.sigma	0.01062	0.01133	0.0006066	0.04199
size	13.77	5.533	9.405	28.53

and the DIC statistic for the model is much larger:

	Dbar	Dhat	DIC	pD
Abun	1032	1009	1055	22.78
total	1032	1009	1055	22.78

The variance parameter $1/\text{size}$, which allows for modelling of overdispersion, is significantly different from zero (average $\text{size} = 13.77$ against the μ of order

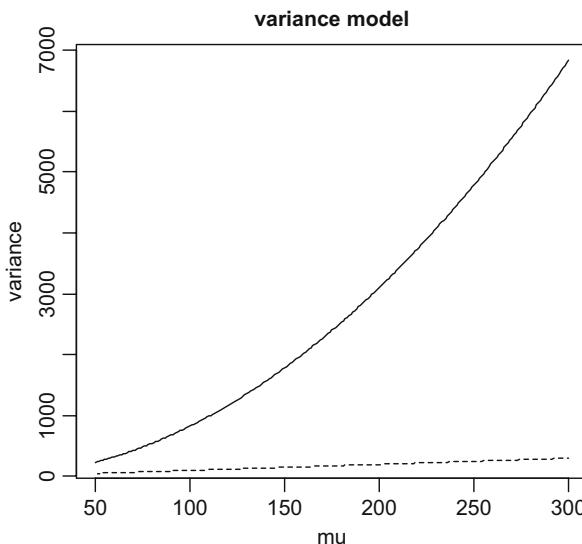


Fig. 23.9 Comparison of variance in abundance for Poisson model (dashed line) and negative binomial model (solid line)

100). The variance of the abundances is given by $\mu + \mu^2/\text{size}$, compared to μ for the Poisson model. Figure 23.9 compares the two variances and clearly shows the overdispersion for high abundance values.

As before, some evidence of auto-correlation in abundance levels is still expressed at both sites, even though the mean values almost certainly lie in the posterior confidence interval.

	mean	sd	MC_error	val12.5pc	median	val97.5pc
BP.s1	5.941	3.414	0.1355	0.6958	5.599	13.90
BP.s2	3.822	3.185	0.1080	0.2517	2.989	12.16

Residual test shows rather some signs of underdispersion:

	mean	sd	MC_error	val12.5pc	median	val97.5pc
SS	84.73	13.37	0.5135	64	82.77	116.2
SS.prd	97.96	14.07	0.07975	72.43	97.34	127.6

The auto-correlation in the residuals u_{si} is shown in Fig. 23.10. The increased auto-correlation at lag 0 and at lag multiples of year, as observed previously, has now disappeared. It is worth mentioning the significant decrease in the u_{si} variance against the previous model.

For both sites, the error auto-correlation parameters are now no longer significant:

	mean	sd	MC_error	val12.5pc	median	val97.5pc
rhol1	0.4544	0.443	0.02364	-0.5042	0.5832	0.9796
rho2	0.2152	0.539	0.01594	-0.8189	0.2956	0.9669

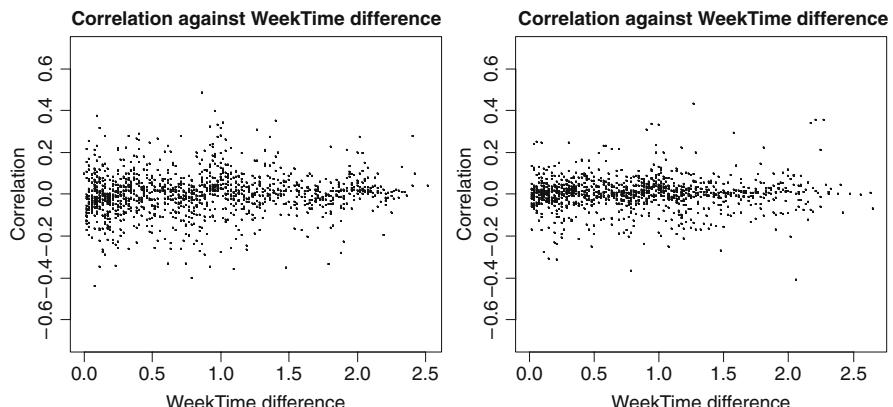


Fig. 23.10 Auto-correlation in u_{si} for negative binomial model with auto-correlated random effects. The left graph shows the results for site 1 and the right graph for site 2

and the negative binomial model without auto-correlation can also be tried. We leave it for the reader as an exercise.

23.8.1 Comparison of Models

With frequentist statistics, we have tools such as comparison of deviances (for nested models) or AIC at our disposal for comparing models. With Bayesian statistics, one such tool is the Deviance Information Criterion (DIC, Spiegelhalter et al., 2002). It is an extension of AIC and works as follows. Let $D(\theta)$ be defined as $-2 \log(f(y|\theta))$, where θ contains the model parameters, and $f(y|\theta)$ is the likelihood. $D(\theta)$ is calculated for each realisation of the MCMC chain. Let \bar{D} be the average of $D(\theta)$. The effective number of parameters, p_D , is calculated as the posterior mean deviance minus the deviance evaluated at the posterior mean:

$$p_D = \bar{D} - D(\bar{\theta})$$

where $\bar{\theta}$ is the average of all realisations of θ . Then, analogous to AIC, the DIC is defined as

$$\text{DIC} = D(\bar{\theta}) + 2 \times p_D = \bar{D} + p_D$$

As with AIC, models with a small DIC are preferred. In R, the command `dicSet` allows for monitoring of the DIC, and the results for the various models fitted in this chapter are given below:

Model	Dbar	Dhat	DIC	pD
Poisson	1890.0	1880.0	1900.0	10.0
Poisson + random effects	783.0	691.0	875.0	92.0
Poisson + auto-correlated random effects	784.5	693.3	875.6	91.1
Negative binomial + auto-correlated random effects	1032.0	1009.0	1055.0	22.8

Note that in the output above, Dhat refers to $D(\bar{\theta})$. Comparing the DIC criteria for all four models, it can be concluded that both Poisson models with random effects are similar in fit and are much better than the Poisson and Negative Binomial models despite them showing some random effect correlation pattern. Formally, the Poisson model with the random effects will be selected as the final model.

23.9 Conclusions

Fitting the Poisson model to the seal abundance data results in overdispersion and auto-correlation. To some extent, these can be addressed by fitting a quasi-Poisson model or generalised linear mixed model. In this chapter, however, we wanted

to introduce the alternative approach of Bayesian models. These form a flexible framework allowing for random effects, auto-correlation, various types of distribution, various (non-canonical) link functions, incorporation of missing values etc. This flexibility comes at a cost though, in that time-consuming simulation methods such as MCMC need to be employed to obtain summaries of the model parameters (although all the models used take no more than half an hour to get 300,000 iterations on a 1.5 GHz PC). Fortunately, freeware software such as WinBugs and its port to R, BRugs, is available to make the MCMC implementation comparatively simple.

It is important to keep in mind that Bayesian statistics is based on a different approach to statistics. It assumes that prior information is available for the parameters, which is then combined with information contained in the data to yield the posterior distribution. This is intrinsically different from frequentist statistics, where the data are analysed in a stand-alone manner, independent from any other sources of information. Although the use of prior information may seem a drawback (as we no longer perform an independent analysis), it can also be used to our advantage. For example, if we are collecting data on an annual basis, then the models can be updated annually where the results from previous years can form the prior distribution, which is then combined with the data from the current year. Furthermore, if an ‘independent’ Bayesian analysis is required, then this can be achieved by choosing non-informative prior distributions, which should give results similar to those obtained from maximum likelihood estimation. There are several introductory books on Bayesian Statistics, see, for example, Gelman et al. (2003) for a general introduction.

When working with small data sets, the prior distribution can become influential in the posterior result, especially with respect to the spread of the posterior distributions, even if non-informative settings are chosen. This can especially be an issue with prior distributions on the variance components. It may be worthwhile performing a sensitivity analysis where the parameters on the prior distribution are changed and the resulting output compared to the original output.

The aim of this chapter was to give a flavour of Bayesian statistics and by no means covers all aspects of Bayesian analysis. Therefore, no section ‘What to write in a paper’ is given. However, we hope that it has shown that there is life beyond ‘ordinary’ generalised linear mixed modelling.

Acknowledgments We would like to thank Grietje Holtrop from Biomathematics & Statistics Scotland for her help with writing this chapter.

Index

A

Additive modelling, 36
GAM in *gam*
and GAM in *mgcv*, 37
with LOESS, 38–42
and GAM in *mgcv*
cubic regression splines, 42–44
LOESS smoother and observed data, 37
with multiple explanatory variables, 53
Adelie penguin time series, R code, 356–357
AED package, 10
Agarwal, D. K., 262
Agresti, A., 200, 204, 209, 234, 246
Akaike, H., 2, 41, 61, 120, 274, 482–484, 486, 543, 553
Akaike information criteria (AIC), 61, 170
American Foulbrood (AFB), 447–448
Amphibian roadkills, 383
data exploration, 385–389
R code, for sampling positions, 385–386
VIF values, calculation of, 386–387
explanatory variables, identification and list of, 384–385
GAM, use of, 389
forward selection approach, 391
negative binomial, 390
R code for, 390
residuals *vs.* explanatory variables and spatial coordinates, plotting of, 392–393
shrinkage smoothers, use of, 390
Variogram function, use of, 396–397
Analysing Ecological Data, 11
ANCOVA model, 25
Anderson, D. R., 482–484, 487, 491, 550, 552, 553
Annual rainfall and bird abundances heterogeneity and, 157

linear regression model and AR-1, 152
numerical output for smoothing model, 154
smoother for, 156
time series for, 153
Antarctic birds and impact of climatic changes on, 343–344
data exploration, 345–350
explanatory variables, 344–345
sea ice extent as, 352–354
SOI and arrival and laying dates difference, 354–359
results obtained, 360–361
trends and auto-correlation, 350–352
Apis mellifera, *see* Honeybees, and AFB disease
Aptenodytes forsteri, *see* Penguin
AR-1 Correlation structure, 150–152
Austin, M. P., 270
Auto-regressive moving average (ARMA) model
error structure, 351–352
R code, 355–356
for residuals, 150
error structure, 151
parameters of, 152
structure, 351, 355
Azzalini, A., 36

B
Badger activity, data on, 495–497
data exploration, 495–497
number of missing values per variable, 496
explanatory variables, 496
GEE approach, 499–500
GLM, application of, 497–499
GLMM results, 500–501
Bagley, P., 30, 420, 421
Bagley, P. M., 401, 421

- Balguerías, E., 73
 Barbraud, C., 343, 344
 Barralb, M., 246, 254, 300, 324
 Barry, S. C., 262, 264
 Bartlett, M., 25, 400, 421
 Bartlett test for homogeneity, 20
 Bates, D., 1, 7, 8, 71, 80–82, 104, 107, 125, 145, 148, 151, 171, 308, 355, 384, 402, 431, 481
 Bathyphotometers, 400
 Bayesian statistics, 510–511
 Benthic biodiversity experiment data
 GLS applied on, 89–90
 linear regression for, 86–89
 protocol, 90–91
 application of, 92–100
 Bernoulli and binomial distributions, 202, 204
 density curves, 203
 Bernstein, S. A., 400
 Bersier, L. F., 129–131, 139, 333
 Bett, B. J., 401
 Bevers, M., 491
 Bird data analysis, 531, 551–552
 additive modelling, 548–552
 anova command, output of, 550
 cross-validation, use of, 549–550
 GAM with Gaussian distribution, use of, 548
 model validation process, 551–552
 R code, for GAM, 548
 smoothing function of, 551
 data exploration, 532
 collinearity, of explanatory variables, 533–536
 outliers, in response and explanatory variables, 532–533
 relationships, between response variable and explanatory variables, 536
 linear regression, 536–540
 drop1 function in, 540–541
 F-statistic and p-value, calculation of, 541–542
 model interpretation, 545–548
 model selection, 542–544
 model validation, 544–545
 summary command, for numerical output, 540
 variables, description of, 531
 Bissonette, D., 383
 Bivariate linear regression model, 17–19
 Bjørnstad, O. N., 480
 Blackinton, G., 400, 421
 Bodie, J. R., 348, 384
 Book outline
 case studies, 4
 GLM and GAM, 3–4
 for instructor guidelines, 6
 R
 and associated packages, citation of, 7–8
 getting data into, 9–10
 programming style, 8–9
 software packages, 5–6
 Booth, G. D., 473, 475, 478
 Bosker, R., 72, 101, 114, 324
 Boveng, P. L., 292
 Bowman, A., 36
 Boxplot, 15
 Bradner, H., 400, 421
 Braun, J., 11
 Brødsgaard, C. J., 447
 Broström, G., 8
 BRugs Packages, 8, 513–520
 Bryk, A. S., 72, 101, 324
 Burnham, K. P., 482–484, 487, 491, 552, 553
- C**
 California bird data
 GEE for, 314–316
 GLM, 295, 297–298
 R code, 297
 xyplot of, 296
 Callaghan, J., 471, 484, 491
 Cameron, A. C., 206, 263, 276, 277, 288
 Cape Petrel time series
 R code, 357–358
 car Package, 255
 Carrivick, P. J. W., 262
 Carroll, R. J., 11, 36, 71, 209
 Case, J. F., 400
 Cetaceans, age determination techniques for data analysis
 explanatory variables, as fixed part of model, 462
 intraclass correlations, calculation of, 466–467
 likelihood ratio test, use of, 464–465
 model with two random effects, 463
 multiple variance structure, use of, 464
 summary command, for numerical output of model, 465–466
 data exploration
 age conditional of species/animals, plot of, 460–461
 R code used, 461–462

- nested structure, of data, 459–460
staining methods, use of, 459, 465–466
step-down approach, use of, 460
- Chambers, J. M., 11, 36, 219
Chatfield, C., 145
Chatterjee, S., 475, 477, 478
Chi-square distribution, 222
Clarke, K. R., 423
Clem, J., 400, 421
Cleveland dotplot for Nereis concentration, 12–13
Cleveland, W. S., 39
Clevenger, J., 383
Cliff, A. D., 480
Climate change and phenology, relationships between, 343–344
Cloern, J. E., 425
Clog–log link, 248, 251
Collet, D., 209
Collins, M. A., 30, 401, 420, 421
Coluber hippocrepis, *see* Snakes, N days response variable
- Commands
 `abline`, 28
 `abline(0, 0)`, 131
 `AIC`, 61
 `attach`, 10
 `cbind`, 269
 `center = TRUE` and `scale = FALSE`, 139
 `coef`, 268
 `colnames`, 269
 `dotchart` function, 12–13
 `factor`, 139
 `gam.check`, 58, 60
 `header = TRUE`, 10
 `library(VGAM)`, 268
 `na.action` option, 145, 150
 `negative.binomial(1)`, 390
 `par`, 58
 `1-pchisq`, 222
 `plot`, 57
 `predict`, 39, 219
 `predict.gls` function, 99
 `print.trellis`, 369
 `rowSums`, 297
 `split` option in `print`, 370
 `stats::resid`, 268
 `step` or `stepAIC`, 235
 `strip` and `strip.default` options, 389
 `summary` and `anova`, 57
- upper.panel and lower.panel in pairs Command, 348
`varFixed`, 75
`varIdent`, 77
`varwidth = TRUE`, 15
WinBUGS, 512
- Constant plus power of variance covariate function, 80
- Cook, A. J. C., 159
Cook distance, 27
Coplot of wedge clam data, 22
`corAR1` Correlation argument, 150
Coronella girondica, *see* Snakes
Correlograms, 482
`corvif` Function, 255
Crawley, M. J., 11
Cronin, M., 9, 503–506
Cruikshanks, R., 177
Cunningham, R. B., 262, 264
Cutshall, A., 383
- D**
- Dale, C. V., 383
Dalgaard, P., 7, 8, 11, 77, 243, 253, 540
Daption capense, *see* Penguin
- Data exploration
 `boxplot`, 15
 Cleveland dotplots, 12–14
 `pairplots`, 14–15
 `xypot` from lattice package, 15–17
- Davis, I. M., 198, 239, 242, 243
Davison, A. C., 67, 177
Deep-sea pelagic bioluminescent organisms, 400
additive mixed modelling and smoothing curve
data collection, procedure of
 ISIT free-fall profiler, use of, 401
 station, location of, 401
model selection, 419–420
multi-panel graphs for grouped data,
 construction of, 401
clustering on correlation matrix, use of, 408–409
Euclidean distances between 16 stations, calculation of, 407–408
use of, one smoother, 406
`varPower` method, 405–406
`xypot`, for multi-panel figure, 404
- Deer data, 300
 binary data, 313
 GEE for, 319–320
GLMM predicted probabilities of parasitic infection along, 329

- Deer data (*cont.*)
 GLM on, 327–328
 probabilities of parasitic infection, 326
- Demetrio, C. G. B., 262
- Design Package, 8
- Deviance information criterion (DIC), 528
- Diggle, P. J., 8, 71, 121, 145, 147, 171, 307, 430
- Dique, D. S., 479, 492
- Dobson, A. J., 204, 209, 209
- Donnelly, C. F., 262, 264
- Draper, N., 11, 49, 66
- drop1 Command, 29, 221, 253, 256
- The Dumont d'Urville research station, 343
- E**
- Effective degrees of freedom (edf), 52–53
- Efron, B., 177, 490
- Eilers, P. H. C., 48
- Elaphe scalaris*, *see* Snakes
- Elaphostrongylus cervi* parasite, 254
 in deer, 255
 count data into presence and absence, 301
- R code, 257
- Elith, J., 292
- Ellis-Iversen, J., 159
- Elphick, C. S., 143, 152, 295, 297
- Emmerson, M. C., 86
- F**
- Fahrig, L., 383, 469, 475, 491
- Falck, W., 480
- family Commands
 family = binomial, 251
 family = poissonff, 268
 family = posnegbinomial argument, 268
 family = quasibinomial, 256
- Faraway, J. J., 11, 21, 25, 36, 201
- Fernández-de-Mera, I. G., 246, 254, 300, 324
- Fernández-Núñez, M. M., 73
- Ferrier, S., 490
- Field, S. A., 270, 271
- Fine, J. P., 8, 270, 271
- Fisher's iris data, 4
- Fitzmaurice, G. N., 19, 102, 113, 246, 295, 303, 312, 313, 324, 341, 430
- Flather, C. H., 491
- Ford, R., 86
- Forman, R. T., 383
- Fox, J., 2, 11, 27, 36, 39, 209, 351, 536
- France, R., 383
- F-Ratio test, 25
- Frequentist statistics, 510
- G**
- Gamma distribution, density curves for μ and v values, 201
- gam Package, 8, 37
- Garrido, J. M., 246, 254, 300, 324
- Gaussian linear regression
 Gaussian quadrature, 341
 as GLM, 210–211
 artificial data with, 212
 model formulation for, 211–212
 probability curves, 213
- geepack Package, 8
- Gelfand, A. E., 262, 512
- Gelman, A., 233, 511, 517, 529
- Gelman-Rubin statistic, 517
- Geman, D., 512
- Geman, S., 512
- Generalised additive mixed models (GAMMs), 2
- Generalised additive modelling (GAM), 1, 238, 258
 larval sea lice around Scottish fish farms, distribution of
 backward selection, 242
 Cleveland dotplot of, 240
 likelihood ratio test, 242–243
 R code, 241
 smoothing curves for, 243
- in mgcv package, 44–46
 additive models with multiple explanatory variables, 53–55
 backwards selection methods, 49
 bioluminescent data for two stations, 53–55
 collinearity and, 63–66
 cross-validation (CV), 51–53
 interaction between continuous and nominal variable, 59–62
 knots, values of, 48–49
 penalised splines and smoothing, 50–51
 regression splines and cubic regression spline, 46–47
- for presence-absence data
 gam commands, 259
 smoothing function of Length, 258
- Generalised cross-validation (GCV), 51–52
See also Generalised additive modelling (GAM)
- Generalised estimation equations (GEEs), 2, 302

- AR-1 correlation, 306–307
association structure, 304–305
exchangeable correlation, 307–308
and link function, 303–304
stationary correlation, 308–309
unstructured correlation, 305–306
variance structure, 304
- Generalised least squares (GLS), 1, 75
Generalised linear mixed models (GLMMs), 2
scene for binomial, 324–325
- Generalised linear modelling (GLM), 1
for presence-absence data
parasites in cod, 252–254
R code, 249–251
tuberculosis in wild boar, 246
for proportional data, 254
- Genersch, E., 447
- Gibbs sampler, 512
- Giller, P. S., 177
- Gillibrand, E. J. V., 30, 401, 420, 421
- Ginsberg, J. R., 469
- glmmML Packages, 8
- gls Function, 75
- Godwits
data
coplot of intake rate and time, 183
description of, 182
linear regression, 184–186
xyplot from lattice package, 184
- Goh, T. N., 262
- Goldman, C., 383
- Goldstein, H., 72, 324
- Gortazar, C., 246, 254, 300, 324
- Graham, M. H., 473
- gstat Package, 8
- Guthery, F. S., 484
- H**
- Hansen, H., 447
- Hanski, I., 469
- Harbour seals, 503–504
See also Seal abundance data
- Hardin, J. W., 194, 204, 234, 251, 263, 265, 295, 315, 320, 321
- Harrell, F. E. Jr., 8
- Harrison, A., 177
- Hartl, M. G. H., 177
- Harvey, A.C., 176, 177
- Hastie, T., 36, 37, 39, 42, 62
- Hastie, T. J., 11, 36, 219
- Hastings, W. K., 512
- Hawaiian bird data set, 171–172
- Heanue, K., 383
- He, B., 262
- Hediste diversicolor* and wedge clam data sets, 12, 72, 86
- Heger, A., 421
- Herring, P. J., 30, 401, 420, 421
- Heterogeneity
graphical model validation, 84–86
linear regression applied on squid, 72–74
variance structure
fixed, 74–75
varcomb, 81–82
varconstpower, 80–81
varexp, 80
varident, 75–78
varpower, 78–80
- Hilbe, J. M., 194, 204, 234, 251, 263, 265, 295, 315, 320, 321
- Hilborn, R., 484
- Hinde, J., 262
- Hindell, M. A., 469
- Hinkley, D. V., 67, 177
- Höfle, U., 246, 254, 300, 324
- Honeybees, and AFB disease
data analysis
explanatory variables, in fixed part of model, 451
intervals command, use of, 457–458
linear mixed effects models, selection approach for, 451–458
linear regression model, use of, 450–451
optimal fixed structure, 454–455
REML estimation, use of, 455–458
selected random structure, 454–456
- data exploration
Cleveland dotplot, for untransformed spores, 448, 449
logarithmic transformation, use of, 448–449
model used, interpretation of, 458
- Paenibacillus larvae*, as causative agent, 447
spores, detection of, 447–448
- Hornitzky, M. A. Z., 447, 448
- Hosmer, D. W., 209, 478, 487
- Hothorn, T., 8
- I**
- Ieno, E. N., 1–12, 22, 33, 35–69, 86, 143, 152, 182, 421, 459, 469, 493, 503
- Independence violation, tools for detection, 161

- Independence violation (*cont.*)
 linear regression model, 162
 R code
 `gstat` package, 162
 `summary` command, 162
 standardised residuals, 163
 variogram, 164–165
- Induced correlations, 112–113
 intraclass correlation coefficient, 114
- Inference, 66–67
 information theory and multi-model, 552–553
- Intensified silicon intensified target (ISIT), 400
 free-fall profiler, 401
- International Polar year 1957–58, 343
- Inverse Gaussian distribution, 204–205
- Iteratively reweighted least squares (IRWLS) algorithm, 214
- J**
 Jackman, S., 8, 262, 278
 Jamieson, A., 30, 420, 421
 Jamieson, A. J., 401
 Jansen, J. K., 293
 Jiang, J., 72, 324
Johanssonia arctica, *see* Leech
 Johnsen, S., 400
 Johnson, D. H., 484
 Jones, J., 383
 Juste, R., 246, 254, 300, 324
- K**
 Karl, D., 400, 421
 Karlovskis, S., 447, 448
 Keele, L. J., 36, 39, 40, 49, 67, 177, 209, 504
 Keitt, T. H., 481
 Kelly-Quinn, M., 177
 Keough, M. J., 2, 11, 21, 36, 531
 King, N. J., 421
 Kiriakoulakis, K., 401
 Kleiber, C., 8, 262, 278
 Koalas distribution, impact of landscape pattern on, 469–471
 collinearity, between explanatory variables
 landscape variables, 473–475
 linear combinations of variables, 475–479
 reduction in collinearity, 475–476
 Spearman rank correlations matrix, 473
 strategies for, high collinearity between explanatory variables, 475
 variance inflation factors (VIFs), calculation of, 478–479
 data, 471
- explanatory variables, 472–473
 exploration and preliminary analysis, 473
- generalised linear mixed-effects models (GLMM), use of, 471, 481–483
 AIC, use of, 483–484
 Akaike weight calculation, 484
 alternative models, construction of, 484–485
 95% confidence set of models, 486
`glmmML` function use, 482
 information-theoretic approach, to model selection, 483–484
 adequacy, methods for assess, 487–490
 averaged predictions, use of, 487
 simulation approach, for quantile-quantile plot, 487–488
 spline correlogram of Pearson residuals, code for, 482–483
 standardised variables, use of, 485
 uncertainty, presence of, 486–487
 koala conservation, implications of results for, 491–492
 Koala habitat, 471
 Noosa Local Government Area (LGA), as study area, 470–471
 spatial auto-correlation, 479–481
 Pearson residuals of, logistic regression model, 483
 spline correlogram, use of, 480–481
- Krill abundance and sea ice, 344
 Kuhnert, P. M., 262, 270, 271
- L**
 Laird and Ware model formulation, *see* Linear mixed effects model
 Laird, N.M., 19, 102, 113, 246, 295, 303, 312, 313, 324, 341
 Lambert, D., 262
 Lampitt, R. S., 401
 Land-use changes, impacts in Ythan catchment, 363
 birds
 and explanatory variables, 378–380
 time series, trends in, 365–366
 Common Agriculture Policy, 363
 data
 exploration, 364–367
 source, 363–364
 independence, dealing with, 374–377
 model validation, 368–372
 Landwehr, J. M., 487–489
 Langton, A. E. S., 384

- `lattice` Package, 8
Laursiden, R., 177
Learned, J., 400, 421
Lee, A. H., 262
Lee, A. K., 469
Leech, 252
Legendre, L., 176, 423
Legendre, P., 176, 423, 473
Lemeshow, S., 209, 478, 487
Lewitus, A., 400, 421
Liang, K. Y., 8, 71, 121, 145, 147, 171, 307
`library` Command and `mgcv` package, 57
Lichstein, J. W., 481
Ligges, U., 8
Likelihood criterion, 213–215
Limosa haemastica, *see* Godwits
Lindenmayer, D. B., 262, 264
Lindstrom, A., 447
Linear mixed effects model
 random effects model, 111–112
 random intercept and slope model
 within-group fitted curves, 111
 random intercept model
 fitted command, 109
 population fitted curve, 108
 in R, 107–109
 summary command, 107
Linear regression model, 17–19
 and multivariate time series, 152–157
`l1lines` Function, 30
`lmeControl` Settings, 169
LOESS smoother, 345–347
 F-statistics and p-values, 42
 and R code, 38–39
 smooth = FALSE, 169
 for span values, 41
Logit link, 248, 251
Log likelihood ratio test, 83
Log–log link, 248, 251
Log odds, 248–249
Loligo forbesi and dorsal mantle length
 (DML), 72–73
Longhurst, A. R., 400
Loyn, R. H., 531
Lukacs, P. M., 484
Lunn, D. J., 512
Luque, P. L., 459
- M**
Mächler, M., 8
Macroprotodon cucullatus, *see* Snakes, N days
 response variable
Maindonald, J., 11
- Mallow’s C_p pop up, 51
 See also Generalised additive modelling
 (GAM)
Mangel, M., 484
Marginal model
 compound symmetric structure, 115
 `corCompSymm`, 116
 general correlation matrix, 115
 R code for, 116
Marine biodiversity, and eutrophication, 424
Marine biological monitoring programme, by
 Rijkswaterstaat, 424, 424
Markov Chain Monte Carlo (MCMC),
 511–512
Martin, T. G., 262, 270
Marx, B. D., 48
MASS Packages, 8
Matsuno, S., 400, 421
Matthews, A., 469
Maui time series, 157
Maximum likelihood estimation and REML
 estimation, 116–119
 difference between
 R code for, 119–120
 in linear regression, 555
McAlpine, C. A., 471, 487
McCarthy, M. A., 504
McCullagh, P., 194, 204, 209, 215, 218, 230,
 246, 253, 478
McCulloch, C. E., 324, 341, 481
McKay, M., 471
Melosira nummuloides, 424
Melzer, A., 469
Mendes, S., 16, 176
Mengersen, K., 262, 270
Metropolis, N., 512
Metropolis-Hastings algorithm, 512
`mgcv` Package, 37
 nlme packages, 8
Millar, C.P., 198, 239, 242, 243
Miller, J., 481, 491
Milne, R., 30, 401, 420, 421
Mixed effects model, 107, 120–121
Moby’s data, 26
Model selection approach, 92–93
 in GLM
 anova command, 223
 drop1 command, 222
 optimal model, 221
 R code and output, 220–221
Model validation, 128–129
 heterogeneity, 20–21
 independence, 21–22

- Model validation (*cont.*)
 normality, 19–20
 in Poisson GLM, 228
 deviance residuals, 229–230
 Pearson residuals, 229
 in quasi-Poisson GLM
 R code, 231
 response residuals, 232
 wedge clam data, 22
 model validation graphs, 23–25
- Molenberghs, G., 71, 107, 121, 123, 125
- Montgomery, D. C., 2, 11, 49, 117
- Moore, B. D., 469
- Morris, K. J., 421
- Multinomial distribution, 204–205
- N**
- National Institute for Coastal and Marine Management (RIKZ), 427
- Naves, J., 469
- Negative binomial distribution, 199
 density curves, 200
 geometric distribution, 200
 mathematics for negative binomial
 truncated model, 265
- Negative binomial GLM, 233
 backward/forward selection based on AIC, 235–236
 explanatory variables, 234–235
 log likelihood test, 238
 probability function, 234
 R code, 236–237
 validation tools for, 237
- Neilson, D. J., 400
- Nelder, J., 194, 204, 209, 215, 218, 230, 246, 253, 478
- Nereis data set, 16, 28–30
- Nested models, 93, 221
- Nestling barn owls, begging behaviour, 128
 anova command, 133
 boxplot and plot commands, 134–135
 optimal model, 136–137
 R code, 130, 138–142
 axes = FALSE and text
 commands, 131
 gls function, 132
 REML and, 137
 variance structure, 132–133
- Neter, J., 473, 475, 478
- Newton, J., 16, 176
- Nitrogen
 concentration in teeth and age for whales stranded in Scotland, 16
- nlme Packages, 8
- Nordström, S., 447, 448
- Normal distribution
 histogram of weight, 194
 probability function, 195
 R code for, 195–196
- Null hypothesis, 25
See also Bartlett test for homogeneity
- O**
- Oahu time series, 155
- Oceans, distribution of living organisms in, 399–400
- O'Connor, D., 400, 421
- Odds, concept, 248
- odTest from pscl package, 238
- offset command, 241
- O'Halloran, J., 177
- O'Hara, B., 8
- O'Neil, R. V., 477
- Ordinary crossvalidation (OCV), 51–52
See also Generalised additive modelling (GAM)
- Ormerod, S. J., 490
- Outer iteration, 53
See also Generalised additive modelling (GAM)
- Overdispersion
 causes and solutions, 224
 model selection in quasi-Poisson, 227–228
 in Poisson GLM, 225–226
 R code and, 226–227
- Owl data
 GEE for, 316
 R code, 317
 Wald test, 318
- Poisson GLMM for, 333
 R code, 333–334
- R code, 299
- sibling negotiation data
 corAR1 structure, 159
 correlation structure with R code, 158–159
- P**
- Paenibacillus larvae*, 447–448
- Paiba, G. A., 159
- Pairplots, 14–15
 penguins, arrival and laying dates of, 348, 349
- panel Commands
 panel.grid, 30
 panel.smooth and panel.cor, 348
 panel = superpose, 404

- Paralithodes camtschaticus*, see Red king crab
- Parmesan, C., 343
- Partridge, J. C., 30, 401, 420, 421
- Pearce, J., 490
- Pearson residuals, 229
- Peatman, W., 400, 421
- Pebesma, E. J., 8, 162, 307
- Peck, E. A., 2, 11, 49, 117
- Penalised quasi-likelihood (PQL) methods, 341
- Penguin
- Adelie Penguin
 - auto-correlation function of laying dates of, 346
 - Emperor and Cape Petrel, 344
 - library and data commands, 347
 - pairplot, for arrival and laying dates, 349
 - R code
 - for differences between arrival and laying dates against time, 350
 - for laying dates, 347
 - for linear regression model, 353
 - time series of arrival and laying dates, 346
 - xypot function, 348
- See also* Antarctic birds and impact of climatic changes on
- Penston, M. J., 198, 239, 242, 243
- Phascolarctos cinereus*, see Koalas
- distribution, impact of landscape pattern on
- Phillips, S. S., 469, 471, 484, 491
- Phoca vitulina* L., see Harbour seals
- Phytoplankton time series data, 423–424, 445–446
 - environmental variables in, 426
 - marine biodiversity and
 - eutrophication, 424–425
 - monitoring programme, 424
 - temperature data analysis, 440–442
 - long-term trends, by area, 440
 - seasonal components, by area, 440
 - spatial trend, 441
 - temperature per month for each area, 439
 - water samples, collection of, 425
- Pierce, D. A., 71, 147, 151, 167, 230, 324
- Pierce, F. J., 147, 151, 167
- Pierce, G., 16, 176
- Pierce, G. J., 12, 22, 73, 182
- Pinheiro, J., 1, 7, 8, 71, 80–82, 104, 107, 125, 145, 148, 151, 171, 308, 355, 402, 431, 481
- Plummer, M., 8, 517
- Poisson distribution
 - function for, 196
 - in GLM, 198
 - probabilities for, 197
 - R code, 197–198
 - GLM with real example
 - deviance, 217–218
 - predicted values of, 218–219
 - R code and, 216–217
- Popova, E. E., 401
- Potts, J. M., 292
- Presence-absence data, 193
- Priede, I. G., 30, 401, 420, 421
- Probit link, 248, 251
- pscl Packages, 8
- Pygoscelis adeliae*, see Penguin
- Q**
- Quasi-Poisson distribution, 226
- GLM
 - R code, 299–300
- Quinn, G. P., 2, 11, 21, 36, 531
- R**
- Raffaelli, D. G., 86
- Raguenau, O., 401
- Random effects model, 111–112
- Random intercept model
 - fitted command, 109
 - population fitted curve, 108
 - in R, 107–109
 - and slope model
 - within-group fitted curves, 111
 - summary command, 107
- Raudenbush, S. W., 72, 101, 324
- Raya, C. P., 73
- R commands, 12
 - code, 62
 - auto-correlation function (ACF) for, 146–147
 - exponential variance structure, 80
 - and homogeneity, 73
 - for NB GLM, 267
 - random intercept model in, 107, 109
 - standardised residuals, 85–86
 - variogram, 167–169
 - dotchart, 13
 - function loess, 40
 - gam function, 42–43
 - interaction in mgcv package, 60
 - panel function, 30
- Red king crab, 252
- Reed, J. M., 143, 152

- Regression splines and technical information, 46–49
- Reichle, M., 400, 421
- Reid, R., 16, 176
- Rhodes, J. R., 270, 271, 469–492
- Ribeiro, P. J., 8, 307
- Ricketts, T. H., 491
- Ridout, M., 262
- RIKZ data, 102
- and model selection, 122
 - protocol for, 123–128
- Ripley, B. D., 8, 11, 13, 36, 233, 328, 332
- Rohlf, F. J., 18–20, 25
- Roos, C., 400, 421
- Roulin, A., 129–130, 139, 333
- Royal Research Ship Discovery*, 401
- Ruppert, D., 11, 36, 71, 209
- S**
- Sarkar, D., 1, 7, 8, 16, 71, 104, 107, 125, 145, 148, 151, 171, 308, 355, 369, 370, 376, 402
- Saveliev, A. A., 503–529
- `scale` Function, 139
- `scatterplot3d` Package, 8
- Schabenberger, O., 147, 151, 167
- Schafer, D. W., 71, 147, 151, 167, 230, 324
- Scheffé-Box test, 25
- Schimek, M. G., 36
- Seabrook, L., 471
- Seal abundance data, 503–504
- additive mixed modelling, with Gaussian distribution, 504–507
 - auto-correlation, calculation of, 520
 - Bayesian statistics, components of, 510–511
 - correlation between model parameters, for Poisson model, 518
 - DIC for model selection, 518, 528
 - frequentist statistics, characteristics of, 510
 - GAM, application of
 - season, as variable, 506
 - two-dimensional smoother, for month and time of day, 507 - GLM application of, 507
 - Pearson residuals plotted against time, graph of, 509–511
 - scale function, use of, 508 - Markov Chain Monte Carlo (MCMC) techniques, 511–512
 - negative binomial distribution, with auto-correlated random effects, 525–528
- overdispersion, assessment of, 519
- Poisson model
- with auto-correlated random effects, 525–528
 - with random effects, 520–523
- Poisson model in BRugs, fitting of, 513
- burn-in period, 516
 - code in R, 513–514
 - Gelman-Rubin statistic test, 517
 - inference, 518–520
 - initialising chains, 515–517
 - `InitializeParam3.txt`, 515–516
 - `Modelglm1.txt` file, as model code, 514–515
 - posterior distributions, summarising of, 517–518
- Searle, S. R., 324, 341, 481
- Season explanatory variables, 266–267
- Semlitsch, R. D., 384
- Shimanuki, H., 447
- Shorebirds, 364–365
- birds and explanatory variables, 378–380
 - independence over time, 374–377
 - LOESS smoother, 365
 - model on, shape of trends for birds, 367
- R code
- for additive mixed model, 367–368
 - `levels` option and `xypot` function, 366
 - `panel.text` function, 366
 - residuals *vs.* fitted values, 368–369
 - residuals *vs.* time, plot of, 370, 371
 - square root transformation, use of, 378
 - variance structure, and heterogeneity, 371
 - `xypot` function, graph with, 365
 - See also* Land-use changes, impacts in Ythan catchment
- Sikkink, P. G., 63
- Smith, A. F. M., 512
- Smith, G. M., 1–33, 35–69, 447–458, 469, 493, 503
- Smith, H., 11, 49, 66
- Smith, R.P., 159
- Smoothing models, 145
- Snakes, 266
- Snijders, T., 72, 101, 114, 324
- Snow, L. C., 159
- Sodium dominance index (SDI) for acid sensitivity of rivers, 177
- Ireland
- geographical position of sites in, 178
 - R code for, 179
 - variogram for pH data, 179–180

- normalised residuals of linear regression model, 181
- R code for, 179
- bubble plot, 182
 - `corRatio` and `corExp` structures, 180
 - experimental variogram, 181
 - `xypplot` from lattice package, 178
- Sokal, R. R., 18–20, 25
- Solan, M., 12, 22, 86, 182
- Southern oscillation index (SOI), 345, 348, 354–359
- Special areas of conservation (SACs), 503
- Species explanatory variables, 266–267
- Sperling, T., 383
- 2-Stage analysis method
- `anova` command, 104–105
 - `lmList` command from `nlme` package, 104
- `stats` Packages, 8
- `step` Function, 253
- Stephens, P. A., 484
- Stone, M., 490
- Sturtz, S., 8
- `summary` Command, 8
- `summary`, 28
 - `summary`, 141
- Sus scrofa*, *see* Wild boar
- Swanson, F., 383
- T**
- `tapply` Option, 373–374
- Temporal correlation and linear regression, 143–149
- auto-regressive moving average (ARMA) model for residuals, 150–152
- Tibshirani, R., 42, 62, 177, 490
- Tobler, W., 161
- Trzcinski, M. K., 475
- Turrentine, T., 383
- U**
- Unbiased risk estimator (UBRE), 51
- See also* Generalised additive modelling (GAM)
- V**
- Vangriesheim, A., 401
- Van Winden, S., 159
- `varComb` Function, 81
- `varConstPower` Function, 80–81
- `varExp` Function, 80
- `varFixed` Model, 83
- and `varIdent` Variance structures, 78–79
- Variance inflation factors (VIF), 386
- `Variogram` function from `nlme` package, 167
- `varPower` Function, 79
- Vaughan, I.P., 490
- Vector generalized additive models (VGAM), package with code, 8, 268
- Venables, W. N., 8, 11, 13, 36, 233, 328, 332
- Verbeke, G., 71, 107, 121, 123, 125
- Ver Hoef, J. M., 292, 293
- Verzani, J., 11
- Vicente, J., 246, 254, 300, 324
- Villard, M. A., 475
- `vis.gam` Function, 58
- W**
- Walker, N. J., 493–502
- Wand, M. P., 11, 36, 71, 209
- Waters, J., 400, 421
- Watson, E., 159
- Webster, M., 400, 421
- Wedge Clam Data, 22–23
- Welsh, A. H., 262, 264
- West, B., 5, 116, 121, 125, 459, 460, 462, 463, 466
- White book on S language, 219
- Widder, E. A., 400
- Wild boar
- tuberculosis-like lesions in, 246
- Wildlife conservation, management strategies for, 469
- See also* Koalas distribution, impact of landscape pattern on
- Winter, T., 383
- Wintle, B. A., 270, 271
- Wolff, G. A., 401
- Woodroffe, R., 469
- Wood, S. N., 1, 7, 8, 11, 36, 37, 45, 47–55, 62, 67, 71, 120, 175, 209, 242, 324, 336, 337, 339, 350, 431, 542, 549
- Wooldridge, J. M., 72
- X**
- `xypplot` from lattice package, 15–17, 345–348, 365
- Y**
- Yan, J., 8, 270, 271
- Yarbrough, M., 400, 421
- Yau, K. K. W., 262
- Yee, T. W., 8, 268
- Yellowstone National Park data, 63
- Ythan catchment, 363–364

Z

Zar, J. H., 19, 473

Zeger, S. L., 8, 71, 121, 145, 147, 171, 307

Zeileis, A, 8, 262, 278

Zero number

models comparisons, 291–292

sources of, 270

for cod parasite data, 271

two-part models and mixture models, and
hippos, 270–274

zero-altered negative binomial (ZANB)

mathematics of, 287–288

models, 262

R code, 268, 289–290

zero-altered Poisson (ZAP), 262

mathematics of, 287–288

R code, 288

zero-inflated count data, 261

zero-inflated GLM, 3

zero-inflated negative binomial (ZINB)

explanatory variables, 278–284

interpretation models, 286

mathematics of, 274–276

mean and variance, 277

validation models, 284–286

zero-inflated Poisson (ZIP), 262

GLMs and GAMs, 3–4

zero-truncated data, 261

mathematics for, 263

maximum likelihood criterion, 264

zero truncated distributions for count

data, 206

and Poisson distribution and, 207–208

probability of sampling 0 count, 207

Zuur, A. F., 1–33, 35–69, 459–468, 469,

493, 503



Springer

the language of science

springer.com

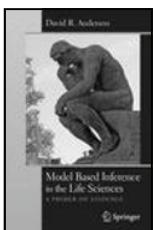


Analysing Ecological Data

Alain F. Zuur, Elena N. Ieno and Graham M. Smith

This book provides a practical introduction to analysing ecological data using real data sets collected as part of postgraduate ecological studies or research projects. The first part of the book gives a largely non-mathematical introduction to data exploration, univariate methods (including GAM and mixed modelling techniques), multivariate analysis, time series analysis (e.g. common trends) and spatial statistics. The second part provides 17 case studies, mainly written together with biologists who attended courses given by the first authors. The case studies include topics ranging from terrestrial ecology to marine biology. Data from all case studies are available from www.highstat.com. Guidance on software

2007. Approx. 672 pp. (Statistics for Biology and Health) Hardcover
ISBN 978-0-387-45967-7

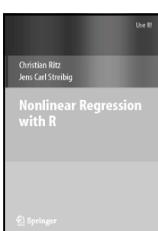


Model Based Inference in the Life Sciences A Primer on Evidence

David R. Anderson

This text focuses on a science philosophy based on "multiple working hypotheses" and statistical models to represent them. The fundamental science question relates to the empirical evidence for hypotheses in this set—a formal strength of evidence. Kullback-Leibler information is the information lost when a model is used to approximate full reality. Hirotugu Akaike found a link between K-L information (a cornerstone of information theory) and the maximized log-likelihood (a cornerstone of mathematical statistics). This combination has become the basis for a new paradigm in model based inference. The text advocates formal inference from all the hypotheses/models in the a priori set—multimodel inference.

2008. 184 pp. Softcover
ISBN 978-0-387-74073-7



Nonlinear Regression with R

Christian Ritz and Jens Carl Streibig

The book starts out giving a basic introduction to fitting nonlinear regression models in R. Subsequent chapters explain the salient features of the main fitting function `nls()`, the use of model diagnostics, how to deal with various model departures, and carry out hypothesis testing. In the final chapter grouped-data structures, including an example of a nonlinear mixed-effects regression model, are considered.

2009. 148pp. (Use R!) Softcover
ISBN 978-0-387-09615-5

Easy Ways to Order ►

Call: Toll-Free 1-800-SPRINGER • E-mail: orders-ny@springer.com • Write: Springer, Dept. S8113, PO Box 2485, Secaucus, NJ 07096-2485 • Visit: Your local scientific bookstore or urge your librarian to order.