

A flexible regression approach using GAMLSS in R.

Bob Rigby and Mikis Stasinopoulos

June 27, 2008

Preface

This book is designed for a short course in GAMLSS given at the University of Utrecht at the beginning of the International Workshop of Statistical Modelling 2008.

©: The copyright remains with the authors and any reproduction of the material in this booklet must have the authors permission.

Contents

1	Introduction	15
1.1	Notation used in this book	16
2	The GAMLSS framework	19
2.1	GAMLSS: the statistical framework	19
2.1.1	The model	19
2.1.2	Available distributions in GAMLSS	21
2.1.3	The additive terms in GAMLSS	25
2.1.4	The GAMLSS algorithms	29
2.1.5	Normalized (randomized) quantile residuals	29
2.2	The gamlss packages	30
2.2.1	How to input the GAMLSS framework packages	30
2.2.2	The different functions of the gamlss package	31
2.3	An introduction to the gamlss packages	34
2.4	Bibliographic notes for Chapter 2	42
3	Continuous response: fitting distributions	49
3.1	Introduction	49
3.1.1	Types of distribution in GAMLSS	49
3.1.2	Types of continuous distributions	50
3.2	Summary of methods generating distributions	52
3.2.1	Distributions generated by univariate transformation	52
3.2.2	Azzalini type methods	52
3.2.3	Splicing distributions	55
3.3	Comparison of properties of continuous distributions	55
3.4	Theoretical considerations for fitting parametric distribution	56
3.5	How to fit distributions in R	57
3.5.1	The function <code>distHist()</code>	57
3.6	Examples of fitting continuous distributions to data	58
3.6.1	Parzen79	58
3.6.2	The strengths of glass fibres data	66
3.6.3	The tensile strength data: response on $(0, 1)$	70
3.7	Bibliography	72
3.8	Appendix of Chapter 3: Methods of generating distributions	72
3.8.1	Distributions generated by univariate transformation	74
3.8.2	Distributions generated by transformation from two or more random variables	77

3.8.3	Distributions generated by truncation	78
3.8.4	Distributions generated by a mixture of distributions	78
3.8.5	Distributions generated by Azzalini's method	79
3.8.6	Distributions generated by splicing	80
3.8.7	Distributions generated by stopped sums	83
3.8.8	Systems of distributions	83
3.9	Exercises for Chapter 3	83
4	Continuous response: regression analysis	85
4.1	Algorithms for fitting parametric regression models	85
4.2	Examples of fitting regression models to continuous distributions	87
4.2.1	The CD4 count data	87
4.2.2	The abdominal circumference data	98
4.3	Bibliography	105
5	Discrete response: count data	109
5.1	Introduction	109
5.2	Explicit continuous mixture distributions	111
5.2.1	Negative binomial distribution	112
5.2.2	Poisson-inverse Gaussian	115
5.2.3	Sichel distribution	115
5.2.4	Delaporte distribution	115
5.2.5	Zero inflated Poisson	116
5.2.6	Comparison of the marginal distributions	117
5.2.7	Families modelling the variance-mean relationship	117
5.3	Examples: fitting a distribution	119
5.3.1	The computer failure data	119
5.3.2	The lice data	122
5.3.3	A stylometric application	125
5.4	Examples: regression analysis	127
5.4.1	The fish species data	127
5.5	Bibliography	136
6	Discrete response: binary and binomial data	141
6.1	Available distributions	141
6.2	Examples of fitting binomial data	141
6.2.1	The alveolar-bronchiolar adenomas data	141
6.2.2	The first year student examination results data	144
6.2.3	The hospital stay data	145
6.3	Bibliography	150
6.4	Appendix of Chapter 6	150
6.5	Exercises for Chapter 6	150
7	Fitting Finite Mixture Distributions	155
7.1	Introduction to finite mixtures	155
7.2	Finite mixtures with no parameters in common	156
7.2.1	The likelihood function	156
7.2.2	Maximizing the likelihood function using the EM algorithm	156
7.2.3	Modelling the mixing probabilities	158

7.2.4	Zero components	159
7.3	The <code>gamlssMX()</code> function	160
7.4	Examples using the <code>gamlssMX()</code> function	161
7.4.1	The enzyme data	161
7.4.2	The Old Faithful geyser data	165
7.5	Finite mixtures with parameters in common	177
7.5.1	Maximizing the likelihood using the EM algorithm	178
7.6	The <code>gamlssNP()</code> function	179
7.7	Examples using the <code>gamlssNP()</code> function	180
7.7.1	The animal brain data	180
7.8	Bibliographic notes	187
8	Model selection	189
8.1	Introduction on model selection	189
8.2	Data sets	191
8.2.1	The US pollution data	191
8.2.2	The AIDS data	191
8.3	Selecting explanatory variables using <code>addterm</code> , <code>dropterm</code> , and <code>stepGAIC</code>	191
8.3.1	The <code>addterm</code> and <code>dropterm</code> functions	192
8.3.2	The <code>stepGAIC</code> function	194
8.3.3	The <code>stepGAIC.VR</code> function	195
8.3.4	The <code>stepGAIC.CH</code> function	199
8.3.5	Selecting model for σ using the <code>stepGAIC.VR</code> function	202
8.4	Data example	204
8.4.1	The third party claims	204
8.5	Selecting hyperparameters	212
8.5.1	Selecting hyperparameters using <code>find.hyper</code>	212
8.6	Exercises for Chapter 8	220
9	Smoothing	223
9.1	Introduction	223
9.2	Data sets	225
9.2.1	The crash helmet data	225
9.2.2	The rent data	226
9.3	Cubic splines, the <code>cs()</code> function	228
9.4	Varying coefficient, the <code>vc()</code> function	229
9.5	Penalized splines, the <code>ps()</code> function	237
9.6	The <code>loess</code> function <code>lo()</code>	241
10	Nonlinear functions in GAMLSS	247
10.1	Introduction	247
10.2	The function <code>nlgamss</code>	247
10.2.1	The data	248
10.2.2	Fitting the model	249
10.3	The function <code>nl()</code>	257
10.4	Exercises	262

11 Centile estimation	265
11.1 Head circumference data	265
11.2 Exercises	279
12 Random effect models at the observational level	281
12.1 Introduction	281
12.2 Fitting an explicit continuous mixture distributions	282
12.3 Fitting non-explicit continuous mixture distributions using Gaussian quadrature	282
12.3.1 Introduction	282
12.3.2 Fitting a normal random effect model in the predictor for μ using Gaussian quadrature	283
12.3.3 Fitting a random effect model in the predictor for other distribution parameters.	285
12.4 Non parametric random effects models	286
12.4.1 Non parametric random intercept in the predictor for μ	286
12.4.2 Non parametric random coefficients in the predictor for μ	287
12.4.3 Non parametric random coefficients in the predictor for all distributional parameters	287
12.5 Examples	289
12.5.1 Schools example	289
13 Random effect models at the factor level	295
13.1 Non-parametric random intercept in the predictor for μ	295
13.1.1 Summary of the $(r + 1)^{th}$ iteration of the EM algorithm	297
13.2 Examples	299
13.2.1 Binomial response with random intercept and slope: A meta analysis	299
13.3 Normal random effect at the factor level in the predictor for μ	304
13.3.1 Summary of the $(r + 1)^{th}$ iteration of the EM algorithm	304
13.4 Examples	305
13.4.1 Binary response with random intercept: Respiratory infection in children	305
14 General random effect models	311
14.1 Random effects model in gamlss	311
14.2 Posterior mode estimation of the parameters β and random effects γ given λ	312
14.3 Estimation of the random effects hyper-parameters λ	313
14.4 Examples	316
14.4.1 Hodges HMO data example	316
A Distributions in the gamlss packages	325
A.1 Continuous two parameter distributions on $\Re$	325
A.1.1 Normal (or Gaussian) distribution (NO, NO2, NOF)	325
A.1.2 Logistic distribution (LO)	326
A.1.3 Gumbel distribution (GU)	326
A.1.4 Reverse Gumbel distribution (RG)	327
A.2 Continuous three parameter distributions on $\Re$	327
A.2.1 Exponential Gaussian distribution (exGAUS)	327
A.2.2 Power Exponential distribution (PE, PE2)	327
A.2.3 t family distribution (TF)	328
A.3 Continuous four parameter distributions on $\Re$	328

A.3.1	Exponential Generalized Beta type 2 distribution (EGB2)	328
A.3.2	Generalized t distribution (GT)	328
A.3.3	Johnson SU distribution (JSUo, JSU)	328
A.3.4	Normal-Exponential- t distribution (NET)	329
A.3.5	Sinh-Arcsinh (SHASH)	330
A.3.6	Skew Exponential Power type 1 distribution (SEP1)	330
A.3.7	Skew Exponential Power type 2 distribution (SEP2)	331
A.3.8	Skew Exponential Power type 3 distribution (SEP3)	331
A.3.9	Skew Exponential Power type 4 distribution (SEP4)	332
A.3.10	Skew t type 1 distribution (ST1)	332
A.3.11	Skew t type 2 distribution (ST2)	332
A.3.12	Skew t type 3 distribution (ST3)	332
A.3.13	Skew t type 4 distribution (ST4)	333
A.3.14	Skew t type 5 distribution (ST5)	333
A.4	Continuous one parameter distribution in $\mathbb{R}^+$	333
A.4.1	Exponential distribution (EXP)	333
A.5	Continuous two parameter distribution in $\mathbb{R}^+$	334
A.5.1	Gamma distribution (GA)	334
A.5.2	Log Normal distribution (LOGNO, LNO)	334
A.5.3	Inverse Gaussian distribution (IG)	334
A.5.4	Weibull distribution (WEI, WEI2, WEI3)	335
A.6	Continuous three parameter distribution in $\mathbb{R}^+$	335
A.6.1	Box-Cox Cole and Green distribution (BCCG)	335
A.6.2	Generalized gamma distribution (GG, GG2)	336
A.6.3	Generalized inverse Gaussian distribution (GIG)	337
A.6.4	Zero adjusted Inverse Gaussian distribution (ZAIG)	337
A.7	Continuous four parameter distribution in $\mathbb{R}^+$	337
A.7.1	Box-Cox t distribution (BCT)	337
A.7.2	Box-Cox power exponential distribution (BCPE)	337
A.7.3	Generalized Beta type 2 distribution (GB2)	338
A.8	Continuous two parameter distribution in $\mathbb{R}[0, 1]$	338
A.8.1	Beta distribution (BE, BEo)	338
A.8.2	Beta inflated distribution (BEINF)	339
A.8.3	Generalized Beta type 1 distribution (GB1)	339
A.9	Binomial type data	339
A.9.1	The Binomial distribution (BI)	339
A.9.2	Beta Binomial distribution (BB)	339
A.10	Count data	340
A.10.1	Poisson distribution (PO)	340
A.10.2	Negative Binomial distribution (NBI, NBII)	340
A.10.3	Poisson-inverse Gaussian distribution (PIG)	341
A.10.4	Delaporte distribution (DEL)	341
A.10.5	Sichel distribution (SI, SICHEL)	341
A.10.6	Zero inflated poisson (ZIP, ZIP2)	342

List of Figures

2.1	A plot of the abdominal circumference data	35
2.2	Residual plot from the Normal fitted model abd2 with $\mu = cs(x, 3)$ and $\sigma = cs(x, 3)$	39
2.3	Worm plot from the normal fitted model abd2 with $\mu = cs(x, 3)$ and $\log(\sigma) = cs(x, 3)$, (a) with default deviation range, (b) with deviation range $(-1.5, 1.5)$	40
3.1	Showing different types of continuous distributions	51
3.2	Showing different types of continuous distributions	53
3.3	Showing different types of continuous distributions	54
3.4	Parzen's snowfall data with a kernel density estimate (blue) and fitted (a) normal, (b) gamma, (c) power exponential, and (d) BCPE distributions respectively (red).	60
3.5	The residual plot from the fitted normal model to the snowfall data	61
3.6	The worm plot from the fitted normal model to the snowfall data	62
3.7	The profile deviance for the σ parameter for model modNO	64
3.8	The bootstrap for $\log \sigma$ from model modNO	65
3.9	The strengths of glass fibres data and the fitted SEP4 distribution model	69
3.10	A worm plot of the residuals for the model using the SEP4 distribution fitted to the strengths of glass fibres data.	71
3.11	Tensile strength data with fitted (a) truncated lognormal, (b) beta, (c) truncated gamma and (d) truncated t (e) generalized beta type 2) distributions.	73
4.1	The plot of the CD4 data	88
4.2	The CD4 data with various transformations for CD4 and age	89
4.3	The CD4 data and the fitted values using polynomial of degree 7 in age	90
4.4	The CD4 data and the best fitting fractional polynomials in age with one (—), two (— — —) and three (...) terms respectively	93
4.5	The CD4 data and the best fitting piecewise polynomials in age with 5 (—), 6, (— — —) and 7 (...), degrees of freedom respectively	94
4.6	The CD4 data and two different cubic spline fits in age, with four, (—), and twelve, (— — —), total effective degrees of freedom in the fit	95
4.7	A plot of the fitted μ and σ values against age for models m42 (continuous green) and m31 (dashed blue).	96
4.8	A worm plot of the residuals from models m42	97
4.9	The plot of the abdominal data data	99
4.10	The abdominal circumference data with the fitted median curve, (50%) centile, and 2.5% and 97.5% reference centile curves from the chosen m2LO model	103
4.11	The residual plot from model m2LO of the abdominal circumference data	104

5.1	NBI distribution for $\mu = 5$ and $\sigma = (0.01, 0.5, 1, 2)$	113
5.2	PIG distribution for $\mu = 5$ and $\sigma = (0.01, 0.5, 1, 2)$	114
5.3	ZIP distribution for $\mu = 5$ and $\sigma = (0.01, 0.1, 0.4, 0.7)$	116
5.4	Skewness-kurtosis combinations for different distributions for Y (for fixed mean 1 and variance 2 for Y)	118
5.5	The computer failure data fit with (a) Poisson , (b) negative binomial (c) Poisson inverse gaussian and (d) Sichel distributions respectively	120
5.6	The lice data with the fitted (a) negative binomial and (b) Sichel distributions respectively	124
5.7	The stylometric data: number of time a word appear in a text against the frequencies	126
5.8	The stylometric data fits with (a) Poisson (b) negative binomial type II (c) Delaporte and (d) Sichel distributions respectively	128
5.9	The fish species data	129
5.10	Fitted μ (the mean number of fish species) against log lake area	133
5.11	Fitted Sichel distributions for observations (a) 40 and (b) 67	133
6.1	The proportion of alveolar-bronchiolar adenomas	142
6.2	The proportion and fitted distribution to the artificial alveolar-bronchiolar adenomas data	143
6.3	The first year student examination results data: Fitted probabilities (a) binomial (b) beta binomial distributions	146
6.4	The rate of appropriateness against age, sex, ward and year	148
6.5	The fitted terms for μ in model IV	151
6.6	The fitted terms for σ in model IV	152
6.7	Six instances of the normalized randomised quantile residuals for model	153
7.1	A histogram of the enzyme activity data together with a non-parametric density estimator (— — —) and the fitted two component RG model m3 (—).	164
7.2	The residual plot from the enzyme activity data final two component RG model m3.	165
7.3	A histogram of variable waiting time (to next eruption from the Old Faithful geyser data), together with a non-parametric density estimator (dashed) and the fitted two component IG model (solid).	168
7.4	The residual plot from the fitted two component IG model for waiting time from the Old Faithful geyser data.	169
7.5	(a) A scatter plot of the waiting time against the previous eruption duration from the Old Faithful geyser data together with the fitted values from the two componets, (dotted and dashed for component 1 and 2 respectively) (b) a plot of the probability of belonging to componemt 2 as a function of duration estimated from model mIG4	173
7.6	Fitted conditional probability density function (f1) for waiting time given previous eruption duration for model mIG4	174
7.7	Comparison of the fitted values for μ for models mIG4 (dashed and dotted lines) and mIG6 (solid line).	176
7.8	Levelplot of the fitted conditional probability density function of the waiting time given the previous eruption time for models (a) mIG4 and model (b) mIG6	177
7.9	A plot of the brain size data.	181

7.10	A plot of the brain size data together with a plot of the three component fitted means of log brain size (<code>lbrain</code>) against log body size (<code>lbody</code>), (solid, dashed and dotted for component 1,2 and 3 respectively).	184
7.11	The residual plot of model <code>br.3</code> for the animal brain size data.	185
8.1	The plot of the third party insurance data	205
8.2	The additive parameter plots for the μ model	210
8.3	AIDS data: data and fitted values from the <code>aids.1</code> model	214
8.4	Profile global deviance and GAIC for different smoothing degrees of freedom fitted in the NBI model to the AIDS data. (a) global deviance (b) GAIC with penalty $\sharp = 2$, (c) $\sharp = 2.8$ (d) $\sharp = 3.8$, plotted against the smoothing degrees of freedom in the model for μ	217
8.5	AIDS data: fitted NBI models with $df = 10$ (red solid line) and $df = 30$ (blue dashed, line)	219
8.6	Abdominal circumference data: fitted μ , σ , ν and τ against x	221
9.1	Showing a scatter plot of the reported cases by media against the age of the of victims.	224
9.2	Plots of net rent (R) against each of the explanatory variables.	227
9.3	Rent data: Additive plots for the cubic splines model	230
9.4	Rent data: contour plot for the fitted additive cubic splines model model	231
9.5	Rent data: surface plot for the fitted additive cubic splines model model	232
9.6	Rent data: (a) the $\beta(A)$ function (b) a contour plot for the fitted additive cubic splines model	234
9.7	Rent data: surface plot for the fitted additive cubic splines model	235
9.8	B-basis functions for the crash helmet data for times with 9 inner knots.	239
9.9	The helmet data fit with different degrees 1,2,3 corresponding to red, green, and blue respectively	240
9.10	The helmet data fitted coefficients with different degrees 0,1,2,3 corresponding to red, green, blue and purple respectively.	242
9.11	Rent data: contour plot for the fitted <code>loess</code> surface model	245
9.12	Rent data: wireframe plot for the fitted <code>loess</code> surface model	246
10.1	Plotting the data together with the fitted values for <code>mu</code> for the fitted model <code>mod1</code> .	250
10.2	The plot of the residuals from model <code>mod1</code>	252
10.3	The plot of the residuals from model <code>mod3</code> using the command <code>plot(mod3, xvar=la\$bflow)</code> .	255
10.4	Model <code>mod3</code> fitted values: (a) the data together with the fitted values for μ , (b) the fitted vales for σ	256
10.5	Fitted values, Standard errors and partial residuals for the <code>sigma</code> parameter model: (a) model <code>mod4</code> fitted quadratic polynomials, (b) model <code>mod5</code> fitted smoothing cubic spline	261
11.1	Observed Head Circumference of Males (From the Fourth Dutch Growth Study) Against Age.	266
11.2	The fitted parameters against age for the 'best' BCT models using AIC (---), GAIC(3) (—), and SBC (···), criteria: (a) μ (b) σ (c) ν (d) τ .	271
11.3	The Residuals From Model <code>BCT(12.3, 5.7, 2, 2, 0.33)</code> . (a) against fitted values of μ (b) against age (c) kernel density estimate (d) normal QQ plot.	273

11.4	Worm Plot of the Residuals From Model $BCT(12.3, 5.7, 2, 2, 0.33)$. The 16 detrended QQ plots, from bottom left to top right plot, correspond to 16 age ranges (displayed in steps above the worm plot from 0 to 22 years).	276
11.5	Observed Head Circumference With Nine Fitted Model Centile Curves (0.4, 2, 10, 25, 50, 75, 90, 98, 99.6), From Model $BCT(12.3, 5.7, 2, 2, 0.33)$, against Age: (a) 0-2.5 years (b) 2.5-22 years	278
12.1	Plot showing how the continuous distribution $NO(0, 1)$ is approximated by Gaussian quadrature with $K = 10$	284
12.2	Plot showing a non parametric mixture distribution with $K = 5$	286
12.3	Plot showing a non parametric mixture distribution in two dimensions with $K = 10$	288
12.4	schools : plots of the fitted means for different degrees of freedom (on the left) and different standard deviations of the random effect (on the right)	293
13.1	Plot of the sample log odds ratios for the different 27 smoking cessation studies .	300
13.2	Plot of the joint (a) and (c) and marginal (b) and (d) non-parametric distribution for γ_0, γ_1	303
14.1	HMO data: box plots of prind against stage	317
14.2	HMO data: the sample (o) and fitted medians (+) of prind against state	321
14.3	HMO data: The sample and approximate fitted coefficients of variation of prind against state	322

List of Tables

1.1	Notation for the random and systematic part of a model used	17
1.2	Notation for likelihood and residuals used	18
2.1	Continuous distributions implemented within the gamlss packages (with default link functions)	22
2.2	Discrete distributions implemented within the gamlss packages (with default link functions)	23
2.3	Additive terms implemented within the gamlss packages	25
3.1	Continuous GAMLSS family distributions defined on $(-\infty, +\infty)$	50
3.2	Continuous GAMLSS family distributions defined on $(0, +\infty)$	51
3.3	Continuous GAMLSS family distributions defined on $(0, 1)$	52
3.4	Showing distributions generated by univariate transformation	74
3.5	Showing distributions generated by continuous mixtures	79
3.6	Showing distributions generated by Azzalini type methods using equation (3.20)	80
3.7	Showing distributions generated by splicing	82
4.1	Models for the abdominal circumference data	101
4.2	Abdominal circumference data: SBC for the logistic distribution	102
5.1	Discrete gamlss family distributions for count data (derived from Poisson mixtures)	111
5.2	Discrete gamlss family distributions for count data	112
5.3	Comparison of models for the fish species data	135
6.1	The first year student examination results data where the binomial denominator is constant at 8.	144
6.2	Models for the AEP data	149
7.1	Table showing the expansion of data for fitting the multinomial model at M step (2) of the EM algorithm	160
7.2	Table showing the expansion of data use in M-step (1) of the EM algorithm for fitting the common parameter mixture model	179
7.3	Possible alternative models for the animal brain data	186
10.1	Models for the blood flow data with their fitted global deviance GDEV and generalized information criterion with penalty $\# = 3$ i.e. GAIC(3)	263
11.1	Selected hyperparameters for different penalties $\#$ in the BCT model	269

11.2	Choosing the distribution using $GAIC(\sharp = 3)$	271
11.3	Comparison of model and sample percents lying below each centile curve for model $BCT(12.3, 5.7, 2, 2, 0.33)$	279
12.1	Table showing the expansion of data for fitting the Gauss quadrature model at M step (2) of the EM algorithm	285
12.2	The educational testing experiments data	289
13.1	Table showing the expansion of data for fitting the Gauss quadrature model at M step (2) of the EM algorithm. (The column headed by k indicates the factor MASS).	298
13.2	The meta analysis data	309
14.1	Comparison of random effects models in BCT model for Hodges data	324
14.2	Comparison of distribution models with random effect for μ only for Hodges data	324

Chapter 1

Introduction

This book is designed as an introduction to the GAMLSS framework. It describes how generalized additive models for location, scale and shape (GAMLSS) can be used for statistical modelling.

Regression analysis is one of the most popular statistical techniques for modelling the relationship between a response variable and explanatory variables. Practitioners who use standard regression methods soon find that the classical assumptions about normality of the errors terms and linearity of the relationship between the response variable (Y) and the explanatory variables (the X 's) very seldom hold. Generalized linear models (GLM) and generalized additive models (GAM), widely used by practitioners today, were designed to cope with some of the above problems but increasingly, especially with larger data sets, are found to be inadequate. This booklet is an introduction to generalized additive models for location, scale and shape (GAMLSS), a framework where some of the limitations of GLM and GAM can be overcome.

This book is designed for the practical statistician. The examples used here are real data examples. Each chapter and the examples within chapters are self contained and can be read separately.

Most of the chapters in this book follow the following format:

- a review
- theoretical considerations for the problem
- software availability
- practical examples
- bibliography
- appendices for more demanding theoretical material
- exercises

Chapter 2 is a general introduction to the GAMLSS framework. Section 2.1 defines the GAMLSS framework and contains a description of the type of regression models that the GAMLSS framework is capable of modelling. Section 2.2 also contains an introduction to the R implementation of the **gamlss** packages and a description of the available functions within the basic **gamlss** package. A brief demonstration of the use of the basic **gamlss** package is also given here.

Chapter 3 is about fitting parametric distributions to data involving a single continuous response variable (with no covariates). This chapter serves also as an introduction to all continuous distributions implemented into the **gamlss** packages. The extensive Appendix of Chapter 3 shows methods of generating continuous distributions using available known distributions.

Regression type of models for continuous response variables are introduced in Chapter 4. Here we introduce modelling the relationship between the continuous response Y and explanatory variables X 's.

Chapter 5 analyzes a count response variable. It introduces the count data distributions available in GAMLSS. Here the response variable takes values $0, 1, 2, \dots, \infty$. It gives examples of how to fit a parametric discrete distribution to a count data sample (with no explanatory variables) and also simple regression situations where the response is a count variable.

Chapter 6 analyzes binomial response variables, that is when the response variable takes values $0, 1, 2, \dots, n$ for a finite n .

Chapter 7 expands the available distributions by the use of finite mixture distributions for modelling the response variable.

Some model selection techniques are discussed in Chapter 8. Different types of smoothers that can be used for explanatory variables within GAMLSS are discussed in Chapter 9. Non-linear models within GAMLSS is discussed in Chapter 10. The important topic of centile estimation is given in Chapter 11.

Random effect models are the subject of Chapters 12, 13 and 14.

Appendix A summarizes all distributions available in the **gamlss** packages.

1.1 Notation used in this book

Vectors in general will be represented in a lower case bold letters, e.g. $\mathbf{x} = (x_1, x_2, \dots, x_n)$ while matrices in an upper case bold letter, for example $\mathbf{X}$. Scalar random variables are represented by upper case, for example Y . The observed value of a random variable is represented by lower case, for example y .

Tables 1.1 and 1.2 show notation that will be used throughout this book.

<i>Systematic part</i>	
Y :	univariate response variable
$\mathbf{y}$:	vector of observed values of the response variable, i.e. $(y_1, y_2, \dots, y_n)^\top$
n :	total number of observations
$\mathbf{x}$:	explanatory variables
$\mathbf{X}$:	fixed effects design matrix
$\boldsymbol{\beta}$:	vector of fixed effect parameters
$\boldsymbol{\gamma}$:	vector of random effects
q :	dimension of the random effect vector $\boldsymbol{\gamma}$
$\mathbf{Z}$:	random effect design matrix
$\boldsymbol{\eta}$:	predictor for a distribution parameter
δ :	$(0, 1)$ indicator variable
ξ :	power parameter for x , i.e. x^ξ
J :	total number of factor levels
$\mathbf{H}$:	hat matrix
$\mathbf{z}$:	adjusted dependent variable
$g()$:	link function applied to model a distribution parameter
$h()$:	non-parametric or non-linear function (in the predictor $\boldsymbol{\eta}$)
$\mathbf{W}$:	matrix of weights
$\mathbf{w}$:	vector of weights
$\mathbf{S}$:	smoothing matrix
<i>Distributions and parameters</i>	
$f_Y()$:	theoretical probability density function of the random variable Y
$f_P()$:	the population probability function
$f_E()$:	the empirical probability density function
$\phi()$:	probability density function of a standard normal distribution
$F_Y()$:	cumulative distribution function of the random variable Y
$\Phi()$:	cumulative distribution function of a standard normal distribution
$Q_Y()$:	inverse cumulative distribution function of the random variable Y , i.e. $F_Y^{-1}()$
$E_Y()$:	Expectation of random variable Y
$V_Y()$:	Variance of random variable Y
$\pi()$:	prior probability density function
$\boldsymbol{\pi}$:	vector of prior (or mixing) probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_k)^\top$
θ :	a parameter of the model distribution, e.g. μ
$\boldsymbol{\theta}$:	vector of the parameters of the distribution, e.g. $\boldsymbol{\theta} = (\mu, \sigma, \nu, \tau)^\top$
K :	total number of distribution parameters
K :	total number of mixture components
μ :	location parameter of the distribution
σ :	scale parameter of the distribution
ν :	shape (eg. skewness) parameter of the distribution
τ :	shape (eg. kurtosis) parameter of the distribution
$\boldsymbol{\lambda}$:	vector of hyperparameters
σ_k :	standard deviation of the normal random effect for parameter θ_k
Z :	normal random variable, $NO(\mu, \sigma)$
z :	standard normal (Gaussian) quadrature mass point

Table 1.1: Notation for the random and systematic part of a model used

Likelihood and information criteria	
L :	likelihood function
ℓ :	log likelihood function
Λ :	generalized likelihood ratio test statistic
$\mathbf{i}()$:	Fisher's expected information matrix
$I()$:	observed information matrix
GD :	global deviance, i.e. minus twice the fitted log-likelihood
$GAIC$:	generalized Akaike information criterion (i.e. $GD + \sharp df$)
df :	total (effective) degrees of freedom used in the model
$\sharp$:	penalty for each degree of freedom in the model
Residuals	
$\mathbf{u}$:	vector of (randomised) quantile residuals
$\mathbf{r}$:	vector of normalised (randomised) quantile residuals
$\boldsymbol{\varepsilon}$:	vector of (partial) residuals
Q :	Q statistic calculated from the residuals
Z :	z-statistic calculated from the residuals

Table 1.2: Notation for likelihood and residuals used

Chapter 2

The GAMLSS framework

This Chapter serves as an introduction to generalized additive models for location, scale and shape (GAMLSS). Section 2.1 describes the GAMLSS statistical framework. That is, the different statistical models, the different distributions, the different additive terms and the different algorithms used within GAMLSS. It also contains an introduction to the different **gamlss** packages and a simple demonstration of how to use the main package **gamlss**.

2.1 GAMLSS: the statistical framework

Generalized additive models for location, scale and shape (GAMLSS) are semi-parametric regression type models. They are parametric, in that they require a parametric distribution assumption for the response variable, and “semi” in the sense that the modelling of the parameters of the distribution, as functions of explanatory variables, may involve using non-parametric smoothing functions. GAMLSS were introduced by Rigby and Stasinopoulos (2001, 2005), Stasinopoulos and Rigby (2007) and Akantziliotou *et al.* (2002) as a way of overcoming some of the limitations associated with the popular generalized linear models (GLM) and generalized additive models (GAM) (Nelder and Wedderburn, 1972 and Hastie and Tibshirani, 1990, respectively).

In GAMLSS the exponential family distribution assumption for the response variable (Y) is relaxed and replaced by a general distribution family, including highly skew and/or kurtotic continuous and discrete distributions. The systematic part of the model is expanded to allow modelling not only of the mean (or location) but other parameters of the distribution of Y as, linear and/or non-linear, parametric and/or smooth non-parametric functions of explanatory variables and/or random effects. Hence GAMLSS is especially suited to modelling a response variable which does not follow an exponential family distribution, (eg. leptokurtic or platykurtic and/or positively or negatively skew response variable, or overdispersed counts response variable) or which exhibit heterogeneity, (eg. where the scale or shape of the distribution of the response variable changes with explanatory variables(s)).

2.1.1 The model

A GAMLSS model assumes that, for $i = 1, 2, \dots, n$, independent observations Y_i have probability (density) function $f_Y(y_i|\boldsymbol{\theta}^i)$ conditional on $\boldsymbol{\theta}^i = (\theta_{1i}, \theta_{2i}, \theta_{3i}, \theta_{4i}) = (\mu_i, \sigma_i, \nu_i, \tau_i)$ a vector of four distribution parameters, each of which can be a function to the explanatory variables.

This is denoted by $Y_i|\boldsymbol{\theta}^i \sim D(\boldsymbol{\theta}^i)$, i.e. $Y_i|(\mu_i, \sigma_i, \nu_i, \tau_i) \sim D(\mu_i, \sigma_i, \nu_i, \tau_i)$ independently for $i = 1, 2, \dots, n$, where D represent the distribution of Y . We shall refer to $(\mu_i, \sigma_i, \nu_i, \tau_i)$ as the *distribution parameters*. The first two population distribution parameters μ_i and σ_i are usually characterized as location and scale parameters, while the remaining parameter(s), if any, are characterized as shape parameters, e.g., skewness and kurtosis parameters, although the model may be applied more generally to the parameters of any population distribution, and can be generalized to more than four distribution parameters.

Let $\mathbf{Y}^\top = (Y_1, Y_2, \dots, Y_n)$ be the n length vector of the response variable. Rigby and Stasinopoulos (2005) define the original formulation of a GAMLSS model as follows. For $k = 1, 2, 3, 4$, let $g_k(\cdot)$ be known monotonic link functions relating the distribution parameters to explanatory variables by

$$g_k(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = \mathbf{X}_k \boldsymbol{\beta}_k + \sum_{j=1}^{J_k} \mathbf{Z}_{jk} \boldsymbol{\gamma}_{jk}, \quad (2.1)$$

i.e.

$$\begin{aligned} g_1(\boldsymbol{\mu}) &= \boldsymbol{\eta}_1 = \mathbf{X}_1 \boldsymbol{\beta}_1 + \sum_{j=1}^{J_1} \mathbf{Z}_{j1} \boldsymbol{\gamma}_{j1} \\ g_2(\boldsymbol{\sigma}) &= \boldsymbol{\eta}_2 = \mathbf{X}_2 \boldsymbol{\beta}_2 + \sum_{j=1}^{J_2} \mathbf{Z}_{j2} \boldsymbol{\gamma}_{j2} \\ g_3(\boldsymbol{\nu}) &= \boldsymbol{\eta}_3 = \mathbf{X}_3 \boldsymbol{\beta}_3 + \sum_{j=1}^{J_3} \mathbf{Z}_{j3} \boldsymbol{\gamma}_{j3} \\ g_4(\boldsymbol{\tau}) &= \boldsymbol{\eta}_4 = \mathbf{X}_4 \boldsymbol{\beta}_4 + \sum_{j=1}^{J_4} \mathbf{Z}_{j4} \boldsymbol{\gamma}_{j4}. \end{aligned}$$

where $\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\nu}, \boldsymbol{\tau}$ and $\boldsymbol{\eta}_k$ are vectors of length n , $\boldsymbol{\beta}_k^\top = (\beta_{1k}, \beta_{2k}, \dots, \beta_{J'_k k})$ is a parameter vector of length J'_k , $\mathbf{X}_k$ is a fixed known design matrix of order $n \times J'_k$, $\mathbf{Z}_{jk}$ is a fixed known $n \times q_{jk}$ design matrix and $\boldsymbol{\gamma}_{jk}$ is a q_{jk} dimensional random variable which is assumed to be distributed as $\boldsymbol{\gamma}_{jk} \sim N_{q_{jk}}(\mathbf{0}, \mathbf{G}_{jk}^{-1})$, where $\mathbf{G}_{jk}^{-1}$ is the (generalized) inverse of a $q_{jk} \times q_{jk}$ symmetric matrix $\mathbf{G}_{jk} = \mathbf{G}_{jk}(\boldsymbol{\lambda}_{jk})$ which may depend on a vector of hyperparameters $\boldsymbol{\lambda}_{jk}$, and where if $\mathbf{G}_{jk}$ is singular then $\boldsymbol{\gamma}_{jk}$ is understood to have an improper prior density function proportional to $\exp\left(-\frac{1}{2} \boldsymbol{\gamma}_{jk}^\top \mathbf{G}_{jk} \boldsymbol{\gamma}_{jk}\right)$, while if $\mathbf{G}_{jk}$ is nonsingular then $\boldsymbol{\gamma}_{jk}$ has a q_{jk} dimensional multivariate normal distribution with mean $\mathbf{0}$ and variance-covariance matrix $\mathbf{G}_{jk}^{-1}$.

The model in (2.1) allows the user to model each distribution parameter as a linear function of explanatory variables and/or as linear functions of stochastic variables (random effects). Note that seldom will all distribution parameters need to be modelled using explanatory variables.

There are several important sub-models of GAMLSS. For example for readers familiar with smoothing, the following GAMLSS sub-model formulation may be more familiar. Let $\mathbf{Z}_{jk} = \mathbf{I}_n$, where $\mathbf{I}_n$ is an $n \times n$ identity matrix, and $\boldsymbol{\gamma}_{jk} = \mathbf{h}_{jk} = h_{jk}(\mathbf{x}_{jk})$ for all combinations of j and k in (2.1), then we have the *semi-parametric additive* formulation of GAMLSS given by

$$g_k(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = \mathbf{X}_k \boldsymbol{\beta}_k + \sum_{j=1}^{J_k} h_{jk}(\mathbf{x}_{jk}) \quad (2.2)$$

where to abbreviate the notation use $\boldsymbol{\theta}_k$ for $k = 1, 2, 3, 4$ to represent the distribution parameter vectors $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, $\boldsymbol{\nu}$ and $\boldsymbol{\tau}$, and where $\mathbf{x}_{jk}$ for $j = 1, 2, \dots, J_k$ are also vectors of length n . Note that design vector $\mathbf{x}_{jk}$ may be the same or different from a design column of matrix $\mathbf{X}_k$. The function h_{jk} is an unknown function of the explanatory variable X_{jk} and $\mathbf{h}_{jk} = h_{jk}(\mathbf{x}_{jk})$ is the vector which evaluates the function h_{jk} at $\mathbf{x}_{jk}$. If there are no additive terms in any of the distribution parameters we have the simple *parametric linear* GAMLSS model,

$$g_1(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = \mathbf{X}_k \boldsymbol{\beta}_k \quad (2.3)$$

Model (2.2) can be extended to allow non-linear parametric terms to be included in the model for μ , σ , ν and τ , as follows, see Rigby and Stasinopoulos (2006):

$$g_k(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = h_k(\mathbf{X}_k, \boldsymbol{\beta}_k) + \sum_{j=1}^{J_k} h_{jk}(\mathbf{x}_{jk}) \quad (2.4)$$

where h_k for $k = 1, 2, 3, 4$ are non-linear functions and $\mathbf{X}_k$ is a known design matrix of order $n \times J_k''$. We shall refer to the model in (2.4) as the *non-linear semi-parametric additive* GAMLSS model. If, for $k = 1, 2, 3, 4$, $J_k = 0$, that is, if, for all distribution parameters, we do not have additive terms, then model (2.4) is reduced to a *non-linear parametric* GAMLSS model:

$$g_k(\boldsymbol{\theta}_k) = \boldsymbol{\eta}_k = h_k(\mathbf{X}_k, \boldsymbol{\beta}_k). \quad (2.5)$$

If, in addition, $h_k(\mathbf{X}_k, \boldsymbol{\beta}_k) = \mathbf{X}_k^\top \boldsymbol{\beta}_k$ for $i = 1, 2, \dots, n$ and $k = 1, 2, 3, 4$ then (2.5) reduces to the linear parametric model (2.3). Note that some of the terms in each $h_k(\mathbf{X}_k, \boldsymbol{\beta}_k)$ may be linear, in which case the GAMLSS model is a combination of linear and non-linear parametric terms. We shall refer to any combination of models (2.3) or (2.5) as a *parametric* GAMLSS model.

The parametric vectors $\boldsymbol{\beta}_k$ and the random effects parameters $\boldsymbol{\gamma}_{jk}$, for $j = 1, 2, \dots, J_k$ and $k = 1, 2, 3, 4$ are estimated within the GAMLSS framework (for fixed values of the smoothing hyper-parameters λ_{jk} 's) by maximising a penalized likelihood function $\ell_p(\boldsymbol{\beta}, \boldsymbol{\gamma})$ given by

$$\ell_p(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \ell(\boldsymbol{\beta}, \boldsymbol{\gamma}) - \frac{1}{2} \sum_{k=1}^p \sum_{j=1}^{J_k} \lambda_{jk} \boldsymbol{\gamma}_{jk}^\top \mathbf{G}_{jk} \boldsymbol{\gamma}_{jk} \quad (2.6)$$

where $\ell(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \sum_{i=1}^n \log f_Y(y_i | \boldsymbol{\theta}^i) = \sum_{i=1}^n \log f_Y(y_i | \mu_i, \sigma_i, \nu_i, \tau_i)$ is the log likelihood function of the distribution parameters given the data. Note we used $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ as argument in the penalized log-likelihood to emphasize what is maximized here; $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ represent all the $\boldsymbol{\beta}_k$'s and the $\boldsymbol{\gamma}_{jk}$'s, for $j = 1, 2, \dots, J_k$ and $k = 1, 2, 3, 4$. More details about the algorithms used to maximize the penalized log likelihood ℓ_p are given in Section 2.1.4. For parametric GAMLSS model (2.3) or (2.5), $\ell_p(\boldsymbol{\beta}, \boldsymbol{\gamma})$ reduces to $\ell(\boldsymbol{\beta})$, and the $\boldsymbol{\beta}_k$ for $k = 1, 2, 3, 4$ are estimated by maximizing the likelihood function $\ell(\boldsymbol{\beta})$. The available distributions and the different additive terms in the current GAMLSS implementation in R are given in Sections 2.1.2 and 2.1.3 respectively. The R function to fit a GAMLSS model is `gamlss()` in the package `gamlss`.

2.1.2 Available distributions in GAMLSS

The form of the distribution assumed for the response variable Y , $f_Y(y | \mu, \sigma, \nu, \tau)$, can be very general. The only restriction that the R implementation of GAMLSS has is that the function $\log f_Y(y | \mu, \sigma, \nu, \tau)$ and its first (and optionally expected second and cross) derivatives with respect to each of the parameters of $\boldsymbol{\theta} = (\mu, \sigma, \nu, \tau)$ must be computable. Explicit derivatives are preferable but numerical derivatives can be used.

Distributions	R Name	μ	σ	ν	τ
beta	BE()	logit	logit	-	-
beta inflated (at 0)	BE0I()	logit	log	logit	-
beta inflated (at 1)	BEZI()	logit	log	logit	-
beta inflated (at 0 and 1)	BEINF()	logit	logit	log	log
Box-Cox Cole and Green	BCCG()	identity	log	identity	-
Box-Cox power exponential	BCPE()	identity	log	identity	log
Box-Cox t	BCT()	identity	log	identity	log
exponential	EXP()	log	-	-	-
exponential Gaussian	exGAUS()	identity	log	log	-
exponential gen. beta type 2	EGB2()	identity	identity	log	log
gamma	GA()	log	log	-	-
generalized beta type 1	GB1()	logit	logit	log	log
generalized beta type 2	GB2()	log	identity	log	log
generalized gamma	GG()	log	log	identity	-
generalized inverse Gaussian	GIG()	log	log	identity	-
generalized t	GT()	identity	log	log	log
Gumbel	GU()	identity	log	-	-
inverse Gaussian	IG()	log	log	-	-
Johnson's SU (μ the mean)	JSU()	identity	log	identity	log
Johnson's original SU	JSUo()	identity	log	identity	log
logistic	LO()	identity	log	-	-
log normal	LOGNO()	log	log	-	-
log normal (Box-Cox)	LN0()	log	log	fixed	-
NET	NET()	identity	log	fixed	fixed
normal	NO()	identity	log	-	-
normal family	NOF()	identity	log	identity	-
power exponential	PE()	identity	log	log	-
reverse Gumbel	RG()	identity	log	-	-
skew power exponential type 1	SEP1()	identity	log	identity	log
skew power exponential type 2	SEP2()	identity	log	identity	log
skew power exponential type 3	SEP3()	identity	log	log	log
skew power exponential type 4	SEP4()	identity	log	log	log
sinh-arcsinh	SHASH()	identity	log	log	log
skew t type 1	ST1()	identity	log	identity	log
skew t type 2	ST2()	identity	log	identity	log
skew t type 3	ST3()	identity	log	log	log
skew t type 4	ST4()	identity	log	log	log
skew t type 5	ST5()	identity	log	identity	log
t Family	TF()	identity	log	log	-
Weibull	WEI()	log	log	-	-
Weibull (PH)	WEI2()	log	log	-	-
Weibull (μ the mean)	WEI3()	log	log	-	-
zero adjusted IG	ZAIG()	log	log	logit	-

Table 2.1: Continuous distributions implemented within the **gamlss** packages (with default link functions)

Distributions	R Name	μ	σ	ν
beta binomial	BB()	logit	log	-
binomial	BI()	logit	-	-
Delaporte	DEL()	log	log	logit
Negative Binomial type I	NBI()	log	log	-
Negative Binomial type II	NBII()	log	log	-
Poisson	PO()	log	-	-
Poisson inverse Gaussian	PIG()	log	log	-
Sichel	SI()	log	log	identity
Sichel (μ the mean)	SICHEL()	log	log	identity
zero inflated poisson	ZIP()	log	logit	-
zero inflated poisson (μ the mean)	ZIP2()	log	logit	-

Table 2.2: Discrete distributions implemented within the **gamlss** packages (with default link functions)

Table 2.1 shows a variety of one, two, three and four parameter families of continuous distributions implemented in our current **gamlss** software version. Table 2.2 shows the discrete distributions. We shall refer to the distributions in Tables 2.1 and 2.2 as the **gamlss.family** distributions, a name to coincide with the R object created by the package **gamlss**. Johnson *et al.* (1994, 1995) are the classic reference books for continuous distributions, while Johnson *et al.* (2005) is the classic reference book for discrete distributions, and cover most of the distributions in Tables 2.1 and 2.2 respectively. The appendix of Chapter 3 describes methods of generating most of the continuous distributions in Table 2.1. The probability (density) functions of all of the distributions in Tables 2.1 and 2.2 are shown in Appendix A. The BCCG distribution in Table 2.1 is the Box-Cox transformation model used by Cole and Green (1992) (also known as the LMS method of centile estimation). The BCPE and BCT distributions, described in Rigby and Stasinopoulos (2004) and Rigby and Stasinopoulos (2006) respectively, generalize the BCCG distribution to allow modelling of both skewness and kurtosis.

For some of the distributions shown in Tables 2.1 and 2.2 more than one parameterization has been implemented in the **gamlss** packages. For example, the two parameter Weibull distribution can be parameterized as $f_Y(y|\mu, \sigma) = (\sigma y^{\sigma-1}/\mu^\sigma) \exp\{-(y/\mu)^\sigma\}$, denoted as WEI, or as $f_Y(y|\mu, \sigma) = \sigma \mu y^{\sigma-1} e^{-\mu y^\sigma}$, denoted as WEI2, or as $f_Y(y|\mu, \sigma) = (\sigma/\beta) (y/\beta)^{\sigma-1} \exp\{-(y/\beta)^\sigma\}$ denoted as WEI3, for $\beta = \mu/[\Gamma(1/\sigma) + 1]$. Note that the second parameterization WEI2 is suited to proportional hazard (PH) models. In the WEI3 parameterization, parameter μ is equal to the mean of y . The choice of parameterization depends upon the particular problem, but some parameterizations are computationally preferable to others in the sense that maximization of the likelihood function is easier. This usually happens when the parameters μ , σ , ν and τ are information orthogonal or almost orthogonal. For interpretation purposes we favour parameterizations where the parameter μ is a location parameter (mean, median or mode). The specific parameterizations used in the **gamlss.family** distributions are given in Appendix A.

For the R implementation of GAMLSS all of the distributions in Tables 2.1 and 2.2 have **d**, **p**, **q** and **r** functions corresponding respectively to the probability density function (pdf), the cumulative distribution function (cdf), the quantiles (i.e. inverse cdf) and random value generating functions. For example, the gamma distribution has the functions **dGA**, **pGA**, **qGA** and **rGA**. In addition each distribution has a *fitting* function which helps the fitting procedure by providing link functions, first and (exact or approximate) expected second derivatives, starting

values etc. All fitting functions have as arguments the link functions for the distribution parameters. For example, the fitting function for the gamma distribution is called `GA` with arguments `mu.link` and `sigma.link`. The default link functions for all `gamlss.family` distributions are shown in columns 3-6 of Tables 1 and 2. The function `show.link()` can be used to identify which are the available links for the distribution parameter within each of the `gamlss.family`. Available link functions can be the usual `glm()` link functions plus `logshifted`, `logitshifted` and `own`. The `own` option allows the user to define his/her own link function, for an example see the help file on the function `make.link.gamlss()`.

There are several ways to extend the `gamlss.family` distributions. This can be achieved by

- creating a new `gamlss.family` distribution
- truncating an existing `gamlss.family`
- using a censored version of an existing `gamlss.family`
- mixing different `gamlss.family` distributions to create a new finite mixture distribution.

New `gamlss.family` distributions

To create a new `gamlss.family` distribution is relatively simple, if the pdf function of the distribution can be evaluated easily. To do that, find a file of a current `gamlss.family` distribution, (having the same number of distribution parameters) and amend accordingly. For more details, on how this can be done, see Stasinopoulos *et.al.*, (2008) Section 4.2.

Truncating `gamlss.family` distributions

Truncating existing `gamlss.family` distributions can be achieved by using the add-on package `gamlss.tr`. The function `gen.trun()`, within the `gamlss.tr` package, can take any `gamlss.family` distribution and generate the `d`, `p`, `q`, `r` and fitting `R` functions for the specified truncated distribution. The truncation can be left, right or in both tails of the range of the response y variable.

Censored `gamlss.family` distributions

The package `gamlss.cens` is designed for the situation where the response variable is censored or, more generally, it has been observed in an interval form, eg. $(3, 10]$ an interval from 3 to 10 (including only the right end point 10). The function `gen.cens()` will take any `gamlss.family` distribution and create a new function which can fit a response of “interval” type. Note that for “interval” response variables the usual likelihood function for independent response variables defined as

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n f(y_i|\boldsymbol{\theta}) \quad (2.7)$$

changes to

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n [F(y_{2i}|\boldsymbol{\theta}) - F(y_{1i}|\boldsymbol{\theta})] \quad (2.8)$$

where $F(y)$ is the cumulative distribution function and $(y_{1i}, y_{2i}]$ is the observed interval.

Additive terms	R Name
cubic splines	<code>cs()</code>
varying coefficient	<code>vc()</code>
penalized splines	<code>ps()</code>
<code>loess</code>	<code>lo()</code>
fractional polynomials	<code>fp()</code>
power polynomials	<code>pp()</code>
non-linear fit	<code>nl()</code>
random effects	<code>random()</code>
random effects	<code>ra()</code>
random coefficient	<code>rc()</code>

Table 2.3: Additive terms implemented within the **gamlss** packages

Finite mixtures of `gamlss.family` distributions

Finite mixtures of `gamlss.family` distributions can be fitted using the package **gamlss.mx**. A finite mixture of `gamlss.family` distributions will have the form

$$f_Y(y|\boldsymbol{\psi}) = \sum_{k=1}^K \pi_k f_k(y|\boldsymbol{\theta}_k) \quad (2.9)$$

where $f_k(y|\boldsymbol{\theta}_k)$ is the probability (density) function of y for component k , and $0 \leq \pi_k \leq 1$ is the prior (or mixing) probability of component k , for $k = 1, 2, \dots, K$. Also $\sum_{k=1}^K \pi_k = 1$ and $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\pi})$ where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_K)$ and $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_K)$. Any combination of (continuous or discrete) `gamlss.family` distributions can be used. The model in this case is fitted using the EM algorithm. The component probability (density) functions may have different parameters [fitted using the function `gamlssMX()`] or may have parameters in common [fitted using the function `gamlssNP()`]. In the former case, the mixing probabilities may also be modelled using explanatory variables and the finite mixture may have a zero component (eg. zero inflated negative binomial etc.). Both functions `gamlssMX()` and `gamlssNP()` are in the add on package **gamlss.mx**. Chapter 7 gives more details about modelling and fitting finite mixtures models using the package **gamlss.mx**.

2.1.3 Available additive terms in GAMLSS

Equation (2.1) allows to model all the distribution parameters μ , σ , ν and τ as linear parametric and/or non-linear parametric and/or non-parametric (smooth) function of the explanatory variables and/or random effects terms. In the GAMLSS implementation in R, the function `gamlss()` in **gamlss** allows formulae for all the distribution parameters. For modelling linear functions the Wilkinson and Rogers (1973) notation as applied for model formulae in the S language by Chambers and Hastie (1992) can be used. [It is the model formulae notation used in R the fit of linear models, `lm()`, and generalized lineal models, `glm()`, see for example Venables and Ripley (2002), Section 6.2.] For fitting non-linear and/or non-parametric (smooth) functions and/or random effects terms, appropriate additive term functions have to be included in the distribution parameters' formulae within the `gamlss()` function. Parametric non-linear models can be also fitted using the function `nlgamlss()` of the add-on package **gamlss.nl**.

Table 2.3 shows the additive term functions implemented in the current R implementation of GAMLSS. Note that all available additive terms names are stored in the list `.gamlss.sm.list`.

Cubic splines

The cubic spline function `cs()` is based on the `smooth.spline()` function of R and can be used for univariate smoothing. Cubic splines are covered extensively in the literature, see e.g. Reinsch (1967), Green and Silverman (1994) and Hastie and Tibshirani (1990). They assume in model (2.2) that the functions $h(t)$ are arbitrary twice continuously differentiable functions and we maximize a penalized log likelihood, given by ℓ subject to penalty terms of the form $\lambda \int_{-\infty}^{\infty} [h''(t)]^2 dt$. The solution for the maximizing functions $h(t)$ are all natural cubic splines, and hence can be expressed as linear combinations of their natural cubic spline basis functions de Boor (1978). In `cs()` each distinct x -value is a knot.

Varying coefficients

The varying coefficient terms were introduced by Hastie and Tibshirani (1993) to accommodate a special type of interaction between explanatory variables. This interaction takes the form of $\beta(r)x$, i.e. the linear coefficient of the explanatory variable x is changing smoothly according to another explanatory variable r . In some applications r will be time. In general r should be a continuous variable, while x can be either continuous or categorical. In the current GAMLSS implementation x has to be continuous or a two level factor with levels 0 and 1.

Penalized splines

Penalized splines were introduced by Eilers and Marx (1996). Penalized Splines (or P-splines) are piecewise polynomials defined by B-spline basis functions in the explanatory variable, where the coefficients of the basis functions are penalized to guarantee sufficient smoothness, see Eilers and Marx (1996). More precisely consider the model $\theta = \mathbf{Z}(\mathbf{x})\gamma$ where θ can be any distribution parameter in a GAMLSS model, $\mathbf{Z}(\mathbf{x})$ is $n \times q$ basis design matrix for the explanatory variable $\mathbf{x}$ defined at q -different knots mostly within the range of $\mathbf{x}$ and γ is a $q \times 1$ vector of coefficients which have some stochastic restrictions imposed by the fact that $\mathbf{D}\gamma \sim N_{q-r}(\mathbf{0}, \lambda^{-1}\mathbf{I})$ [or equivalently by $\gamma \sim N_q(\mathbf{0}, \lambda^{-1}\mathbf{K}^-)$ where $\mathbf{K}^-$ is a generalized inverse of $\mathbf{K} = \mathbf{D}^\top \mathbf{D}$]. The matrix $\mathbf{D}$ is a $(q-r) \times q$ matrix giving r th differences of the q -dimensional vector γ . So to define a penalized spline we need: i) q the number of knots in the x -axis defined by argument `ps.intervals` [and of course where to put them; `ps()` uses equal spaces in the x -axis], ii) the degree of the piecewise polynomial used in the B-spline basis so we can define $\mathbf{X}$, defined by argument `degree` iii) r the order of differences in the $\mathbf{D}$ matrix indicating the type of the penalty imposed on the the coefficients of the B-spline basis functions, defined by argument `order` and iv) the amount of smoothing required defined either by the desired equivalent degrees of freedom defined by argument `df` [or alternatively by the smoothing parameter defined by argument `lambda`]. The `ps()` function in `gamlss` is based on an S-PLUS function of Marx (2003).

Local polynomials, loess

The function `lo()` allows the user to use a `loess` fit in a `gamlss` formula. A `loess` fit is a polynomial (surface) curve determined by one or more explanatory (continuous) variables, which are fitted locally see Cleveland *et al.* (1993). The implementation of the `lo()` function is very similar to the function with the same name in the S-PLUS implementation of `gam`. However `gamlss` `lo()` function uses the R `loess()` function as its engine and this creates some minor differences between the two `lo()` even when the same model is fitted. `lo()` is the only function

currently available in **gamlss** which allows smoothing in more than one explanatory (continuous) variables.

Fractional polynomials

The **fp()** function is an implementation of the fractional polynomials introduced by Royston and Altman (1994). The functions involved in **fp()** and **bfp()** are loosely based on the fractional polynomial function **fracpoly()** for S-PLUS given by Ambler (1999). The function **bfp** generates the correct design matrix for fitting a power polynomial of the type $b_0 + b_1x^{p_1} + b_2x^{p_2} + \dots + b_kx^{p_k}$. For given powers $p_1, p_2, \dots, p_k$, given as the argument **powers** in **bfp()**, the function can be used to fit power polynomials in the same way as the functions **poly()** or **bs()** of the package **splines** are used to fit orthogonal or piecewise polynomials respectively. The function **fp()**, [which uses **bfp()**] works as an additive smoother term in **gamlss**. It is used to fit the best fractional polynomials among a specific set of power values. Its argument **npoly** determines whether one, two or three fractional polynomials should be used in the fitting. For a fixed number **npoly** the algorithm looks for the best fitting fractional polynomials in the list $c(-2, -1, -0.5, 0, 0.5, 1, 2, 3)$. Note that **npoly=3** is rather slow since it fits all possible 3-way combinations at each backfitting iteration.

Power polynomials

The power polynomial function **pp()** is an experimental function and is designed for the situation in which the model is in the form $b_0 + b_1x^{p_1} + b_2x^{p_2}$ with powers p_1, p_2 to be estimated non-linearly by the data. Initial values for the non-linear parameters p_1, p_2 have to be supplied.

Non-linear terms

The function **nl()** exists in the add-on package **gamlss.nl** designed for fitting non-linear parametric models within GAMLSS. It provides a way of fitting non-linear terms together with linear or smoothing terms in the same model. The function takes a non-linear object, (created by the function **nl.obs**), and uses the R **nlm()** function within the backfitting cycle of **gamlss()**. The success of this procedure depends on the starting values of the non-linear parameters (which must be provided by the user). No starting values are required for the other, e.g., linear terms, of the model. [An alternative method of fitting non-linear parametric models is using the function **nlgamlss()** of the package **gamlss.nl**.]

Random effects

The function **random()** allows the fitted values for a factor (categorical) predictor to be shrunk towards the overall mean, where the amount of shrinking depends either on the parameter λ , or on the equivalent degrees of freedom (df). This function is similar to the **random()** function in the **gam** package of Hastie (2006) documented in Chambers and Hastie (1992). The function **ra()** is similar to the function **random()** but its fitting procedure is based on augmented least squares, a fact that makes **ra()** more general, but also slower to fit, than **random()**. The random coefficient function **rc()** is experimental. Note that the “random effects” functions, **random()**, **ra()** and **rc()** are used to estimate the random effect γ ’s *given* the hyperparameters λ ’s. In order to obtain estimates for the hyperparameters, methods discussed in Rigby and Stasinopoulos (2005) Appendix A can be used. Alternatively, for models only requiring a single random effect in one distribution parameter only, the function **gamlssNP()** of the package

`gamlss.mx`, which uses Gaussian quadrature, can be used.

The `gamlss()` function uses the same type of additive backfitting algorithm implemented in the `gam()` function of the R package **gam** Hastie (2006). Note that the function `gam()` implementation in the R recommended package **mgcv** Wood (2001) does not use backfitting. The reason that we use backfitting here that it is easier to extend the algorithm so new additive terms can be included.

Each new additive term in the `gamlss()` requires two new functions. The first one, (the one that is seen by the user) is the one which defines the additive term and sets the additional required design matrices for the linear part of the model. The names of the existing additive functions are shown in the second column of Table 2.3. For example `cs(x)` defines a cubic smoothing spline function for the continuous explanatory variable `x`. It is used during the definition of the design matrix for the appropriate distribution parameter and it adds a linear term for `x` in the design matrix. The second function is the one that actually performs the additive backfitting algorithm. This function is called `gamlss.name()` where the `name` is one of the names in column two of Table 2.3. For example the function `gamlss.cs()` performs the backfitting for cubic splines. New additive terms can be implemented by defining those two functions and adding the new names in the `.gamlss.sm.list` list.

The general policy when backfitting is used in `gamlss()` is to include the linear part of an additive term in the appropriate linear term design matrix. For example, in the cubic spline function `cs()` the explanatory variable say `x` is put in the linear design matrix of the appropriate distribution parameter and the smoothing function is fitted as a deviation from this linear part. This is equivalent of fitting a modified backfitting algorithm, see Hastie and Tibshirani (1990). In other additive functions where the linear part is not needed (or defined) a column on zeros is put in the design matrix. For example, this is the case when the fractional polynomials additive term `fp()` is used.

If the user wishes to create a new additive term, care should be taken on how the degrees of freedom of the model are defined. The degrees of freedom for the (smoothing) additive terms are usually taken to be the extra degrees of freedom on top of the linear fit. For example to fit a single smoothing cubic spline term for say `x` with 5 total degrees of freedom, `cs(x,df=3)` should be used since already 2 degrees of freedom have been used for the fitting of the constant and the linear part of the explanatory variable `x`. [This is different from the `s()` function of the **gam** package which uses `s(x,df=4)`, assuming that only the constant term has been fitted separately]. After a GAMLSS model containing additive (smoothing) terms is used to fit a specific distribution parameter the following components are (usually) saved for further use. In the output below replace `mu` with `sigma`, `nu` or `tau` if a distribution parameter other than `mu` is involved.

mu.s: a matrix, each column containing the fitted values of the smoothers used to model the specific parameter. For example given a fitted model say `mod1`, then `mod1$mu.s` would access the additive terms fitted for `mu`.

mu.var: a matrix containing the estimated variances of the smoothers.

mu.df: a vector containing the extra degrees of freedom used to fit the smoothers.

mu.lambda: a vector containing the smoothing parameters (or random effects hyperparameters).

mu.coefSmo: a list containing coefficients or other components from the additive smooth fitting.

2.1.4 The GAMLSS algorithms

There are two basic algorithms used for maximizing the penalized likelihood given in (2.6). The first, the CG algorithm, is a generalization of the Cole and Green (2002) algorithm [and uses the first and (expected or approximated) second and cross derivatives of the likelihood function with respect to the distribution parameters $\theta = (\mu, \sigma, \nu, \tau)$ for a four parameter distribution]. Note that we have dropped the subscripts here to simplify the notation. However for many population probability (density) functions, $f_Y(y|\theta)$, the parameters θ are information orthogonal (since the expected values of the cross derivatives of the likelihood function are zero), e.g., location and scale models and dispersion family models, or approximately so. In this case the simpler RS algorithm, which is a generalization of the algorithm used by Rigby and Stasinopoulos (1996a) and Rigby and Stasinopoulos (1996b) for fitting mean and dispersion additive models, (MADAM), [and does not use the cross derivatives], is more suited. The parameters $\theta = (\mu, \sigma)$ are fully information orthogonal for distributions NBI, GA, IG, LO and NO only in Tables 2.1 and 2.2. Nevertheless, the RS algorithm has been successfully used for fitting all distributions in Tables 2.1 and 2.2, although occasionally it can be slow to converge. Note also that the RS algorithm is not a special case of the CG algorithm, see Appendix 2A.

The object of the algorithms is to maximize the penalized likelihood function ℓ_p , given by (2.6), for fixed hyperparameters λ . Appendix 2B show how this can be achieved. For fully parametric models, (2.3) or (2.5), the algorithms maximize the likelihood function ℓ . The algorithms, which are fully described in Appendix 2A, are implemented in the option `method` in the function `gamlss()` where a combination of both algorithms is also allowed (using the `mixed()` function). The major advantages of the algorithms are i) the modular fitting procedure (allowing different model diagnostics for each distribution parameter); ii) easy addition of extra distributions; iii) easy addition of extra additive terms; and iv) easily found starting values, requiring initial values for the $\theta = (\mu, \sigma, \nu, \tau)$ rather than for the β parameters. The algorithms have generally been found to be stable and fast using very simple starting values (e.g., constants) for the θ parameters. The function `nlgamlss()` in the package `gamlss.nl` provides a third algorithm for fitting parametric linear or non-linear GAMLSS models in equation (2.3) or (2.5) respectively. However the algorithm needs starting values for all the β parameters, rather than $\theta = (\mu, \sigma, \nu, \tau)$, which can be difficult for the user to choose. This method uses the `nlm()` R function for maximization of the likelihood, which uses numerical derivatives (if the actual derivatives are not provided). This function is used by the function `summary.gamlss()` to get more accurate standard errors for the beta parameters of the parameteric linear terms in the predictors after convergence of the GAMLSS algorithm.

Clearly, for a specific data set and model, the (penalized) likelihood can potentially have multiple local maxima. This is investigated using different starting values and has generally not been found to be a problem in the data sets analyzed, possibly due to the relatively large sample sizes used.

Singularities in the likelihood function similar to the ones reported by Crisp and Burridge (1994) can potentially occur in specific cases within the GAMLSS framework, especially when the sample size is small. The problem can be alleviated by appropriate restrictions on the scale parameter (penalizing it for going close to zero).

2.1.5 Normalized (randomized) quantile residuals

For each fitted GAMLSS model, say $\mathcal{M}$, the (normalized randomized quantile) residuals of Dunn and Smyth (1996) are used to check the adequacy of $\mathcal{M}$ and, in particular, its distribution component. The (normalized randomized quantile) residuals are given by $\hat{r}_i = \Phi^{-1}(u_i)$ where

Φ^{-1} is the inverse cumulative distribution function of a standard normal variate and $u_i = F(y_i|\hat{\theta}^i)$ if y_i is an observation from a continuous response, while u_i is a random value from the uniform distribution on the interval $[F(y_i - 1|\hat{\theta}^i), F(y_i|\hat{\theta}^i)]$ if y_i is an observation from a discrete integer response, where $F(y|\theta)$ is the cumulative distribution function. For a right censored continuous response u_i is defined as a random value from a uniform distribution on the interval $[F(y_i|\hat{\theta}^i), 1]$. Note that, when randomization is used, several randomized sets of residuals (or a median set from them) should be studied before a decision about the adequacy of model $\mathcal{M}$ is taken. The true residuals r_i have a standard normal distribution if the model is correct.

2.2 The gamlss packages

Section 2.2.1 provides some information to download the main **gamlss** packages. Section 2.2.2 shows the different functions available in the different packages available for modelling GAMLSS. Section 2.3 provides a basic introduction of the **gamlss** package.

2.2.1 How to input the GAMLSS framework packages

The GAMLSS framework comprise of seven different packages written in the free software R, i.e. the original **gamlss** package and six add-on packages

1. the original **gamlss** package for fitting a GAMLSS model
2. the **gamlss.boot** package for bootstrapping centiles.
3. the **gamlss.cens** package for fitting censored (left, right or interval) response variables.
4. the **gamlss.dist** package for additional distributions
5. the **gamlss.mx** package for fitting finite mixture distributions.
6. the **gamlss.nl** package for fitting non-linear models
7. the **gamlss.tr** package for fitting truncated distributions.

The GAMLSS framework packages can be downloaded and installed from CRAN, the R library at <http://www.r-project.org/>. Test versions may be found at the GAMLSS web site at <http://www.gamlss.com/>.

The following paragraph only applies to PC's with Microsoft Windows software. If your PC is connected to the internet you can install the package by going in the R-menu **Packages/install package(s)** and get the package from CRAN. If you are not connected but you have download the zip file earlier use **Packages/install package(s) from local drive** to install the package. The package **gamlss** will now be in the R library and you can load it using the menu **Packages/load package.../gamlss** or using the command `library(gamlss)`.

Help files are provided for all functions in the **gamlss** package in the usual way. For example using

```
?gamlss
```

will bring you to the HTML help menu for the **gamlss()** function and similarly for other functions within the package. The **gamlss** manual, *Instructions on how to use the gamlss package in R*, (2nd edition), Stasinopoulos *et al.* (2008), and the help files of the package can be found in a pdf form at the "browse directory" folder of the **Help/Html help/Packages/gamlss**.

2.2.2 The different functions of the `gamlss` package

The main function of the `gamlss` package is `gamlss()`. This function is used to fit a GAMLSS model and consequently to create a `gamlss` object in R. Section 2.3 shows the basic use of the function while Chapter 3 of Stasinopoulos *et al.* (2008) provides a more detailed examination of the function. Note that all commands in R are case sensitive.

The following functions are used for fitting or updating a model:

- `gamlss()` : for fitting and creating a `gamlss` object
- `refit()` : to refit a `gamlss` object (i.e. continue iterations) if it has not converged
- `update()` : to update a given `gamlss` model object
- `histDist()` : to fit a parametric distribution to a single (response) variable and plot simultaneously a histogram and the fitted distribution of this variable

Note that the `histDist()` is designed for fitting a parametric distribution to data where no explanatory variables exist. The functions which extract information from the fitted model (object) are:

- `AIC()` or `GAIC()` : to extract the generalized Akaike information criterion (GAIC) from a fitted `gamlss` model object
- `coef()` : to extract the linear coefficients from a fitted `gamlss` model object
- `deviance()` : to extract the global deviance of the `gamlss` model object
- `extractAIC()` : to extract the generalized Akaike information criterion from a fitted `gamlss` model object
- `fitted()` : to extract the fitted values from a fitted `gamlss` model object
- `formula()` : to extract a model formula
- `fv()` : to extract the fitted values for a distribution parameter (see also `fitted()` and `lpred()`)
- `logLik()` : to extract the log likelihood
- `lp()` : to extract the linear predictor for a distribution parameter (see also `lpred()`)
- `lpred()` : to extract the fitted values, linear predictor or specified terms (with standard errors) for a distribution parameter.
- `model.frame()` : to extract the model frame of a specified distribution parameter
- `model.matrix()` : to extract the design matrix of a specified distribution parameter
- `predict()` : to predict from new data individual distribution parameter values (see also `lpred` below)
- `predictAll()` : to predict from new data all the distribution parameter values
- `print()` : to print a `gamlss` object

- **residuals()** : to extract the normalized (randomized) quantile residuals from a fitted **gamlss** model object. See Dunn and Smyth (1996) for a definition of the normalized (randomized) quantile residuals given in Section 2.1.5.
- **summary()** : to summarize the fit in a **gamlss** object
- **terms()** : to extract terms from a **gamlss** object
- **vcov()** : to extract the variance-covariance matrix of the beta estimates (for all distribution parameter models).

Note that some of the functions above are distribution parameter dependent. That is, these functions have an extra argument **what**, which can be used to specify which of the distribution parameters values are required i.e. "**mu**", "**sigma**", "**nu**" or **tau**. For example **fitted(m1, what="sigma")** would give the fitted values for the σ parameter from model **m1**.

Functions which can be used for selecting a model are:

- **addterm()** : to add a single term, from those supplied, to a fitted **gamlss** model object (used by **stepGAIC()** below).
- **dropterm()** : to fit all models that differ from the current fitted **gamlss** model object by dropping a single term (used by **stepGAIC()** below).
- **find.hyper()** : to find the hyperparameters (e.g. degrees of freedom for smoothing terms and/or non-linear parameters) by minimizing the profile Generalized Akaike Information Criterion (GAIC) based on the global deviance, see Appendix A2.1 of Rigby and Stasinopoulos (2005) .
- **gamlss.scope()** : to define the scope for **stepGAIC()**
- **stepGAIC()** : to select explanatory terms using GAIC
- **stepGAIC.CH()** : to select (additive) using the method of Chambers and Hastie (1992).
- **stepGAIC.VR()** : to select (parametric) terms using the method of Venables and Ripley (2002).
- **VGD()** : to calculate the global deviance of model using the validation set data set, (where the training part of the data is used for fitting and the validation for calculating the global deviance).
- **VGD1()** : identical to **VGD()** but the output is a list rather than values as in **VGD()**.
- **VGD2()** : identical to **VGD1()** but it takes as argument the new data, (**newdata**), rather than a factor which splits the combined data in two as in functions **VGD()** or **VGD1()**.
- **TGD()** : to calculate the global deviance for a new (test) data set, given a fitted **gamlss** model.

Functions for plotting or diagnostics:

- **plot()** : a plot of four graphs for the normalized (randomized) quantile residuals of a **gamlss** object. The residual plots are: (i) against an x-variable (ii) against the fitted values, (iii) a density plot and (iv) a QQ-plot. Note that residuals are randomized only for discrete response variables, see Dunn and Smyth (1996) or Section 2.1.5.

- `par.plot()` : for plotting parallel profile plots for individual participants in repeated measurement analysis
- `pdf.plot()` : for plotting the pdf functions for a given fitted `gamlss` object or a given `gamlss.family` distribution
- `prof.dev()` : for plotting the profile global deviance of one of the distribution parameters μ , σ , ν or τ .
- `prof.term()` : for plotting the profile global deviance of one of the model (beta) parameters. It can be also used to study the GAIC($\#$) information profile of a hyperparameter for a given penalty $\#$ for the GAIC.
- `Q.stats()` : for printing the Q statistics of Royston and Wright (2000).
- `rqres.plot()` : for plotting QQ-plots of different realizations of normalized randomized quantile residuals for a model with a discrete `gamlss.family` distribution.
- `show.link()` : for showing available link functions for distribution parameters in any `gamlss.family` distribution
- `term.plot()` : for plotting additive (smoothing) terms in any distribution parameter model
- `wp()` : worm plot of the residuals from a fitted `gamlss` object. See van Buuren and Fredriks (2001) for the definition of a worm plot.

Functions created specially for centile estimation which can be applied if only one explanatory variable is involved are:

- `centiles()` : to plot centile curves against an x-variable.
- `centiles.com()`: to compare centiles curves for more than one `object`.
- `centiles.split()`: as for `centiles()`, but splits the plot at specified values of x.
- `centiles.pred()`: to predict and plot centile curves for new x-values.
- `fitted.plot()` : to plot fitted values for all the parameters against an x-variable

The following two functions are used in the definition of a new `gamlss.family` distribution so the casual user does not need them:

- `make.link.gamlss()` : defines the available link functions in `gamlss` package
- `checklink()`: used to define the link function for each of the distribution parameters.

Some functions like `gamlss()`, `print.gamlss()`, `summary.gamlss()`, `fitted.gamlss()`, `predict.gamlss()`, `plot.gamlss()`, `wp()`, AIC and GAIC are introduced in the next sections.

The function `gamlss()` is considered in more detail in Chapter 3 of Stasinopoulos *et al.* (2008).

Appendix A contains a summary of the distributions available in `gamlss` packages.

2.3 An introduction to the gamlss packages

The function `gamlss()` of the package **gamlss** is similar to the `gam()` function in the R package **gam**, Hastie (2006), but can fit more distributions (not only the ones belonging to the exponential family) and can model all the parameters of the distribution as functions of the explanatory variables.

This implementation of `gamlss()` allows modelling of up to four parameters in a distribution family, which are conventionally called `mu`, `sigma`, `nu` and `tau`. Here we will try to give a simple demonstration of the **gamlss** package.

Data summary:

R data file: `abdom` in package **gamlss** of dimensions 610×2

variables

`y` : abdominal circumference

`x` : gestational age

purpose: to demonstrate the fitting of a simple regression type models in GAMLSS

The following data `abdom`, kindly provided by Dr. Eileen M. Wright, are used here for demonstration purposes. Data `abdom` comprises 610 observations of Y = abdominal circumference in mm. and x = gestational age in weeks. Load **gamlss** from the R library and then load the `abdom` data set:

```
> library("gamlss")
> data("abdom")
> plot(y ~ x, data = abdom, col = "blue", xlab = "age", ylab = "circumference")
```

The data are plotted in Figure 2.1. To fit a normal distribution to the data with the mean of Y modelled as a cubic polynomial in x , i.e. `poly(x,3)`, use

```
> abd0 <- gamlss(y ~ poly(x, 3), data = abdom, family = NO)
```

```
GAMLSS-RS iteration 1: Global Deviance = 4939.735
```

```
GAMLSS-RS iteration 2: Global Deviance = 4939.735
```

Since the normal distribution `NO` is also the default value we could omit the `family` argument. To get a summary of the results use

```
> summary(abd0)
```

```
*****
```

```
Family:  c("NO", "Normal")
```

```
Call:  gamlss(formula = y ~ poly(x, 3), family = NO, data = abdom)
```

```
Fitting method: RS()
```

```
-----
Mu link function:  identity
```

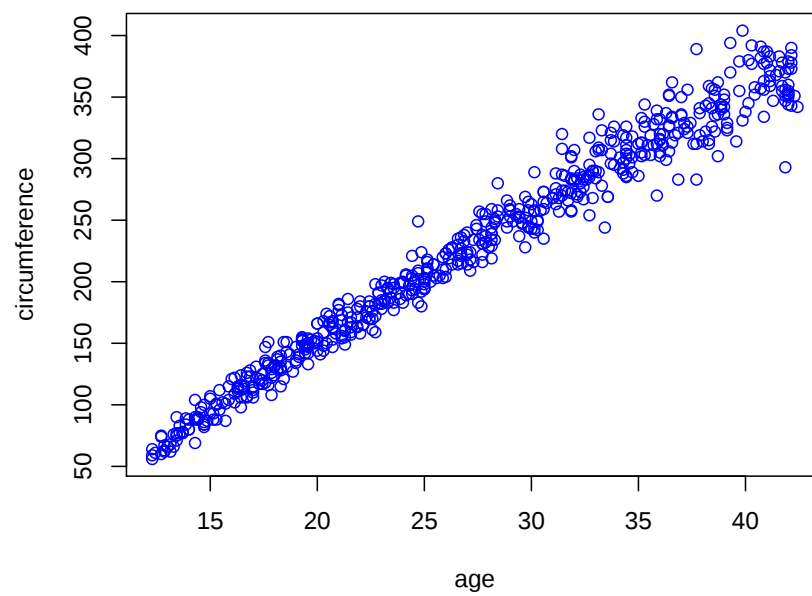


Figure 2.1: A plot of the abdominal circumference data

Mu Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	226.7	0.5618	403.581	0.000e+00
poly(x, 3)1	2157.7	13.8741	155.521	0.000e+00
poly(x, 3)2	-109.6	13.8748	-7.896	1.360e-14
poly(x, 3)3	-26.7	13.8748	-1.924	5.480e-02

Sigma link function: log

Sigma Coefficients:

Estimate	Std. Error	t value	Pr(> t)
2.63002	0.02867	91.74924	0.00000

No. of observations in the fit: 610
Degrees of Freedom for the fit: 5
Residual Deg. of Freedom: 605
at cycle: 2

Global Deviance: 4939.735
AIC: 4949.735
SBC: 4971.802

We used the R function `poly()` to fit orthogonal polynomials, but we could have fit the same model using the `I()` function i.e.

```
> abd00 <- gamlss(y ~ x + I(x^2) + I(x^3), data = abdom, family = NO)
```

GAMLSS-RS iteration 1: Global Deviance = 4939.735

GAMLSS-RS iteration 2: Global Deviance = 4939.735

```
> summary(abd00)
```

Family: c("NO", "Normal")

Call: `gamlss(formula = y ~ x + I(x^2) + I(x^3), family = NO, data = abdom)`

Fitting method: RS()

Mu link function: identity

Mu Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-65.340953	19.528047	-3.346	8.705e-04
x	9.577417	2.354505	4.068	5.375e-05
I(x^2)	0.104515	0.089438	1.169	2.430e-01
I(x^3)	-0.002075	0.001078	-1.924	5.479e-02

```

-----
Sigma link function:  log
Sigma Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      2.63    0.02863   91.86      0

-----

No. of observations in the fit:  610
Degrees of Freedom for the fit:   5
      Residual Deg. of Freedom: 605
                        at cycle:  2

Global Deviance:      4939.735
      AIC:             4949.735
      SBC:             4971.802
*****

```

Note that for large data sets it is more efficient (and may be essential) to calculate the polynomials terms in advance prior to using the `gamlss()` function, i.e.

```
x2<-x^2; x3<-x^3
```

since the evaluation is done then only once. The fitted model is given by $Y \sim NO(\hat{\mu}, \hat{\sigma})$ where $\hat{\mu} = \hat{\beta}_{01} + \hat{\beta}_{11}x + \hat{\beta}_{21}x^2 + \hat{\beta}_{31}x^3$ i.e. $\hat{\mu} = -65.34 + 9.577x + 0.1045x^2 - 0.002075x^3$ and $\log(\hat{\sigma}) = \hat{\beta}_{02} = 2.63$ so $\hat{\sigma} = \exp(2.63) = 13.87$ (since σ has a default log link function).

The `summary` function (used after convergence of the `gamlss()` function) has two ways of producing standard errors. The default value is `type="vcov"`. This uses the `vcov` method for `gamlss` objects which (starting from the fitted beta parameter values given by the `gamlss()` function) uses a non-linear fitting, with only one iteration to obtain the full Hessian matrix of all the beta parameters in the model (from all the distribution parameters), i.e. $\beta_{01}, \beta_{11}, \beta_{21}, \beta_{31}$ and β_{02} in the above model. Standard errors are obtained from the observed information matrix (the inverse of the Hessian). The standard errors obtained this way are reliable, since they take into account the information about the interrelationship between the distribution parameters, i.e. μ and σ in the above case. On occasions, when the above procedure fails, the standard errors are obtained from `type="qr"`, which uses the individual fits of the parameters (used in the `gamlss()` algorithms) and therefore should be used with caution. The standard errors produced this way do not take into the account the correlation between the estimates of the distribution parameters μ, σ, ν and τ , [although in the example above the estimates of the distribution parameters μ and σ of the normal distribution are asymptotically uncorrelated]. Note also that when smoothing additive terms are involved in the fitting, both methods, that is, "vcov" and "qr", produce incorrect standard errors, since they are effectively assume that the estimated smoothing terms were fixed at their estimated values. The functions `prof.dev()` and `prof.term()` can be used for obtaining more reliable individual parameter confidence intervals.

Model `abd0` is a linear parametric GAMLSS model, as defined in (2.3). In order to fit a semi-parametric model in age using a non-parametric smoothing cubic spline with 3 effective degrees of freedom on top of the constant and linear terms use

```
> abd1 <- gamlss(y ~ cs(x, df = 3), data = abdom, family = NO)
```

```
GAMLSS-RS iteration 1: Global Deviance = 4937.16
```

```
GAMLSS-RS iteration 2: Global Deviance = 4937.16
```

The effective degrees of freedom used in the fitting of the μ parameters in the above model are 5 (one for the constant, one for the linear and 3 for smoothing). Note that the `gamlss()` notation is different to the `gam()` notation in S-PLUS where the equivalent model is fitted using `s(x,4)`. [Note also that when you use `gam()` in S-PLUS (or R package **gam**) that their default convergence criteria may need to be reduced for proper convergence in S-PLUS and comparison with `gamlss()` results.]

The total degrees of freedom used for the above model `abd1` are six, i.e. 5 for μ the mean, and 1 for the constant scale parameter `sigma` the standard deviation.

Fitted values of the parameters of the object can be obtained using the `fitted()` function. For example `plot(x, fitted(abd1,"mu"))` will plot the fitted values of μ against x . The constant estimated scale parameter (the standard deviation of the normal in this case) can be obtained:

```
> fitted(abd1, "sigma")[1]

      1
13.84486
```

where `[1]` indicates the first value of the vector. The same values can be obtained using the more general function `predict()`:

```
> predict(abd1, what = "sigma", type = "response")[1]

      1
13.84486
```

The function `predict()` can also be used to predict the response variable parameters for both old and new data values of the explanatory variables.

To model both the mean, μ , and the scale parameter, σ , as non-parametric smoothing cubic spline functions of x (with a normal distribution for the response Y) use:

```
> abd2 <- gamlss(y ~ cs(x, 3), sigma.formula = ~cs(x, 3), data = abdom,
+   family = NO)
```

```
GAMLSS-RS iteration 1: Global Deviance = 4785.698
GAMLSS-RS iteration 2: Global Deviance = 4784.711
GAMLSS-RS iteration 3: Global Deviance = 4784.718
GAMLSS-RS iteration 4: Global Deviance = 4784.718
```

The function `resid(abd2)` (an abbreviation of `residuals()`) can be used to obtain the (normalized randomized quantile) residuals of the model, subsequently just called residuals throughout this manual. [The residuals only need to be randomized for discrete distributions, see Dunn and Smyth (1996) and Section 2.1.5.] Residuals plots can be obtained using `plot()`.

```
> plot(abd2)
```

```
*****
      Summary of the Quantile Residuals
      mean    = 0.0005115742
```

```

variance      = 1.001641
coef. of skewness = 0.2397172
coef. of kurtosis = 3.718456
Filliben correlation coefficient = 0.9962348
*****

```

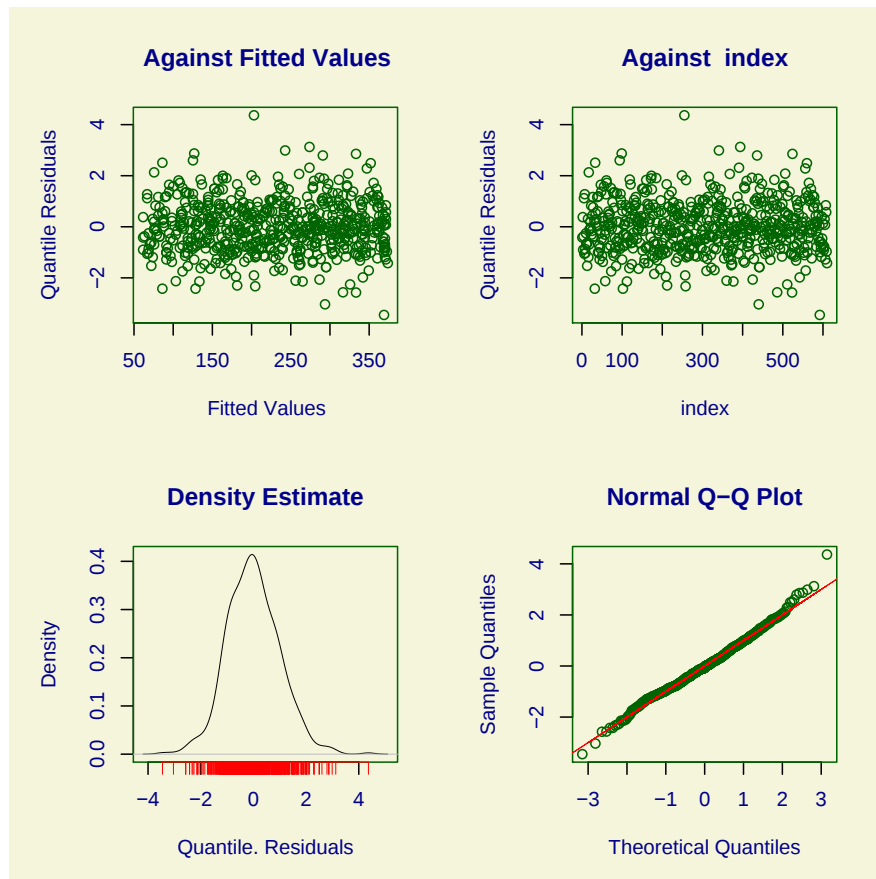


Figure 2.2: Residual plot from the Normal fitted model `abd2` with $\mu = cs(x, 3)$ and $\sigma = cs(x, 3)$

See Figure 2.2, for the plot.

Note that the `plot()` function does not produce additive term plots [as it does for example in the `gam()` function of the package `mgcv`] in R. The function which does this in the `gamlss` package is `term.plot()`

A worm plot of the residuals, see van Buuren and Fredriks (2001), can be obtained by using the `wp()` function:

```
> wp(abd2)
```

See Figure 2.3(a) for the plot.

To include all points in the worm plot change the Deviation axis range by increasing the value of `ylim.all`:

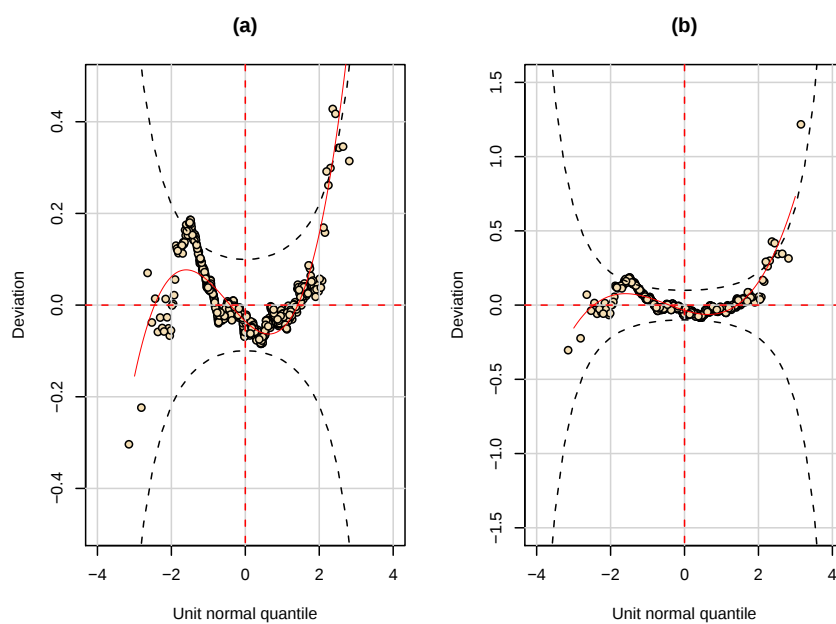


Figure 2.3: Worm plot from the normal fitted model `abd2` with $\mu = cs(x, 3)$ and $\log(\sigma) = cs(x, 3)$, (a) with default deviation range, (b) with deviation range $(-1.5, 1.5)$.

```
> wp(abd2, ylim.all = 1.5)
```

Since there is no warning message all points have been included in the worm plot. See Figure 2.3(b) for the plot. [Clearly one point was omitted from Figure 2.3(a).]

The default worm plot above is a detrended normal Q-Q plot of the residuals, and indicates a possible inadequacy in modelling the distribution, since some points plotted lie outside the (dotted) confidence bands.

If you wish to use loess curves instead of cubic splines use:

```
> abd3 <- gamlss(y ~ lo(x, span = 0.4), sigma.formula = ~lo(x,
+   span = 0.4), data = abdom, family = NO)
```

```
GAMLSS-RS iteration 1: Global Deviance = 4785.934
GAMLSS-RS iteration 2: Global Deviance = 4785.498
GAMLSS-RS iteration 3: Global Deviance = 4785.491
GAMLSS-RS iteration 4: Global Deviance = 4785.491
```

You can find all the implemented smoothers and additive terms in the **gamlss** package in Table 2.3.

If you wish to use a different distribution instead of the normal, use the option **family** of the function **gamlss**. For example to fit a *t*-distribution to the data use:

```
> abd3 <- gamlss(y ~ cs(x, 3), sigma.formula = ~cs(x, 3), data = abdom,
+   family = TF)
```

```
GAMLSS-RS iteration 1: Global Deviance = 4777.554
GAMLSS-RS iteration 2: Global Deviance = 4776.539
GAMLSS-RS iteration 3: Global Deviance = 4776.544
GAMLSS-RS iteration 4: Global Deviance = 4776.544
```

A list of the different continuous distributions implemented in the package **gamlss()** is given in Tables 2.1. The details of all the distributions currently available in **gamlss()** are given in Appendix A. Chapter 4 of the GAMLSS manual, Stasinopoulos *et.al.*, (2008), describes how the user can set up their own distribution in **gamlss()**.

Different models can be compared using their global deviances, $GD = -2\hat{\ell}$, (if they are nested) or using a generalized Akaike information criterion, $GAIC = -2\hat{\ell} + (\sharp.df)$, where $\hat{\ell} = \sum_{i=1}^n \log f(y_i | \hat{\mu}_i, \hat{\sigma}_i, \hat{\nu}_i, \hat{\tau}_i)$ is the log-likelihood function and $\sharp$ is a required penalty e.g. $\sharp = 2$ for the usual Akaike information criterion. The function **deviance()** provides the global deviance of the model. Notes that the GAMLSS global deviance is different from the deviance that it is provided by the functions **glm()** and **gam()** in R. The global deviance is **exactly** minus twice the fitted log likelihood function, *including* all constant terms in the log-likelihood. The **glm()** deviance is calculated as a deviation from the saturated model and it does not include 'constant' terms (which do not depend on the mean of distribution) in the fitted log likelihood and so cannot be used to compare different distributions. To obtain the generalized Akaike information criterion use the functions **AIC()** or **GAIC()**. The functions are identical. For example to compare the models **abd1**, **abd2** and **abd3** use

```
> AIC(abd1, abd2, abd3)

      df      AIC
abd3 11.001504 4798.547
abd2  9.999872 4804.718
abd1  6.000680 4949.162
```

The AIC function uses default penalty $\sharp = 2$, giving the usual Akaike information criterion (AIC). Hence the usual AIC [equivalent to $\text{GAIC}(\sharp = 2)$] selects model **abd3** as the best model (since it has the smallest value of AIC). If you wish to change the penalty $\sharp$ use the argument **k**.

```
> AIC(abd1, abd2, abd3, k = 3)
```

	df	AIC
abd3	11.001504	4809.548
abd2	9.999872	4814.718
abd1	6.000680	4955.162

Hence, $\text{GAIC}(\sharp = 3)$ also selects model **abd3** as the best model.

2.4 Bibliographic notes for Chapter 2

Appendices 2A and 2B

The Appendix of this Chapter describes the two main algorithms, RS and CG, of GAMLSS and show how the maximization of the penalized log-likelihood function ℓ_p given by equation (2.6) over the parameters β_k and terms γ_{jk} for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$ leads to the algorithms.

Appendix 2A: The Algorithms

Introduction

Let $\mathbf{u}_k = \frac{\partial \ell}{\partial \boldsymbol{\eta}_k}$ be the score functions, $\mathbf{z}_k = \boldsymbol{\eta}_k + [\mathbf{W}_{kk}]^{-1} \mathbf{u}_k$ be the adjusted dependent variables and $\mathbf{W}_{ks}$ be diagonal matrices of iterative weights, for $k = 1, 2, \dots, p$ and $s = 1, 2, \dots, p$, which can have one of the following forms $-\frac{\partial^2 \ell}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_s^T}$, $-E \left[\frac{\partial^2 \ell}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_s^T} \right]$ or $\text{diag} \left\{ \left[\frac{\partial \ell_i}{\partial \eta_{ik}} \frac{\partial \ell_i}{\partial \eta_{is}} \right] \right\}$, over $i = 1, 2, \dots, n$, i.e. the observed information, expected information or product score function, depending respectively on whether a Newton-Raphson, Fisher scoring or quasi Newton-Raphson algorithm is used, (see Lange, 1999, Chapter 11 for a definition of the techniques), in the RS and CG algorithms below.

Let r be the outer cycle iteration index, k the parameter index, i the inner cycle iteration index, m the backfitting index and j the random effects (or nonparametric) term index. Also, for example, let $\gamma_{jk}^{(r,i,m)}$ stand for the current value of the vector γ_{jk} in the r^{th} outer, i^{th} inner and m^{th} backfitting cycle iteration and let $\gamma_{jk}^{(r,i,\cdot)}$ stand for the value of γ_{jk} at the convergence of the backfitting cycle for the i^{th} inner cycle of the r^{th} outer cycle, which is also the starting value $\gamma_{jk}^{(r,i+1,1)}$ for the $(i+1)^{\text{th}}$ inner cycle of the r^{th} outer cycle, for $j = 1, 2, \dots, J_k$ and $k = 1, \dots, p$. Note also, for example, $\gamma_{jk}^{(r,i,c)}$ means the current (i.e. most recently) updated estimate of γ_{jk} while the algorithm operates in the backfitting cycle of the i^{th} inner cycle of the r^{th} outer cycle.

The RS Algorithm

Essentially the RS algorithm has an outer cycle which maximizes the penalized likelihood with respect to β_k and γ_{jk} , for $j = 1, \dots, J_k$ in the model successively for each θ_k in turn, for

$k = 1, \dots, p$. Note at each calculation in the algorithm the current updated values of all quantities are used.

The RS algorithm is not a special case of the CG algorithm because in RS the diagonal weight matrix $\mathbf{W}_{kk}$ is evaluated (i.e. updated) *within* the fitting of each parameter θ_k , while in CG all weight matrices $\mathbf{W}_{ks}$ for $k = 1, 2, \dots, p$ and $s = 1, 2, \dots, p$ are evaluated *after* fitting *all* θ_k for $k = 1, 2, \dots, p$.

The RS algorithm is as follows :

- **Start:** Initialize fitted values $\theta_k^{(1,1)}$ and random effects $\gamma_{jk}^{(1,1,1)}$, for $j = 1, \dots, J_k$ and $k = 1, 2, \dots, p$. Evaluate the initial linear predictors $\eta_k^{(1,1)} = g_k \left[\theta_k^{(1,1)} \right]$, for $k = 1, 2, \dots, p$.
- **START OUTER CYCLE** $r = 1, 2, \dots$ **UNTIL CONVERGENCE. FOR** $k = 1, 2, \dots, p$
 - **START INNER CYCLE** $i = 1, 2, \dots$ **UNTIL CONVERGENCE** .
 - * Evaluate the current $\mathbf{u}_k^{(r,i)}$, $\mathbf{W}_{kk}^{(r,i)}$ and $\mathbf{z}_k^{(r,i)}$
 - * **START BACKFITTING CYCLE** $m = 1, 2, \dots$ **UNTIL CONVERGENCE:**
 - * Regress the current partial residuals $\varepsilon_{0k}^{(r,i,m)} = \mathbf{z}_k^{(r,i)} - \sum_{j=1}^{J_k} \mathbf{Z}_{jk} \gamma_{jk}^{(r,i,m)}$ against design matrix $\mathbf{X}_k$ using the iterative weights $\mathbf{W}_{kk}^{(r,i)}$ to obtain the updated parameter estimates $\beta_k^{(r,i,m+1)}$.
 - * For $j = 1, 2, \dots, J_k$ smooth the partial residuals $\varepsilon_{jk}^{(r,i,m)} = \mathbf{z}_k^{(r,i)} - \mathbf{X}_k \beta_k^{(r,i,m+1)} - \sum_{t=1, t \neq j}^{J_k} \mathbf{Z}_{tk} \gamma_{tk}^{(r,i,c)}$ using the shrinking (smoothing) matrix S_{jk} given by (??) to obtain the updated (and current) additive predictor term $\mathbf{Z}_{jk} \gamma_{jk}^{(r,i,m+1)}$
 - * **END BACKFITTING CYCLE**, on convergence of $\beta_k^{(r,i,\cdot)}$ and $\mathbf{Z}_{jk} \gamma_{jk}^{(r,i,\cdot)}$ and set $\beta_k^{(r,i+1)} = \beta_k^{(r,i,\cdot)}$ and $\gamma_{jk}^{(r,i+1)} = \gamma_{jk}^{(r,i,\cdot)}$ for $j = 1, 2, \dots, J_k$ and otherwise update m and continue the backfitting cycle.
 - * **Calculate updated** $\eta_k^{(r,i+1)}$ and $\theta_k^{(r,i+1)}$.
 - **END INNER CYCLE** on convergence of $\beta_k^{(r,\cdot)}$ and the additive predictor terms $\mathbf{Z}_{jk} \gamma_{jk}^{(r,\cdot)}$ and set $\beta_k^{(r+1,1)} = \beta_k^{(r,\cdot)}$, $\gamma_{jk}^{(r+1,1)} = \gamma_{jk}^{(r,\cdot)}$, for $j = 1, 2, \dots, J_k$, $\eta_k^{(r+1,1)} = \eta_k^{(r,\cdot)}$ and $\theta_k^{(r+1,1)} = \theta_k^{(r,\cdot)}$, otherwise update i and continue inner cycle.

UPDATE value of k

- **END OUTER CYCLE:** if the change in the (penalized) likelihood is sufficiently small, otherwise update r and continue outer cycle.

The CG algorithm

Algorithm CG, based on Cole and Green (1992) is as follows :

- **Start** Initialize : $\theta_k^{(1,1)}$ and $\gamma_{jk}^{(1,1,1)}$ for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$. Evaluate $\eta_k^{(1)} = \eta_k^{(1,1)} = g_k \left[\theta_k^{(1,1)} \right]$ for $k = 1, 2, \dots, p$.

- **START OUTER CYCLE** $r = 1, 2, \dots$ **UNTIL CONVERGENCE**
- Evaluate and **fix** the current $\mathbf{u}_k^{(r)}$, $\mathbf{W}_{ks}^{(r)}$ and $\mathbf{z}_k^{(r)}$ for $k = 1, 2, \dots, p$ and $s = 1, 2, \dots, p$. Perform a single r^{th} step of the Newton-Raphson algorithm by:
 - **START INNER CYCLE** $i = 1, 2, \dots$ **UNTIL CONVERGENCE** :
 - * **FOR** $k = 1, 2, \dots, p$
 - * **START BACKFITTING CYCLE** $m = 1, 2, \dots$ **UNTIL CONVERGENCE**

$$X_k \beta_k^{(r,i,m+1)} = \mathbf{H}_k^{(r)} \boldsymbol{\varepsilon}_{ok}^{(r,i,m)}$$
 and for $j = 1, 2, \dots, J_k$

$$\mathbf{Z}_{jk} \gamma_{jk}^{(r,i,m+1)} = \mathbf{S}_{jk}^{(r)} \boldsymbol{\varepsilon}_{jk}^{(r,i,m)}$$
 - * **END BACKFITTING CYCLE**, on convergence of $\beta_k^{(r,i,\cdot)}$ and $z_{jk} \gamma_{jk}^{(r,i,\cdot)}$ and set $\beta_k^{(r,i+1)} = \beta_k^{(r,i,\cdot)}$ and $\gamma_{jk}^{(r,i+1)} = \gamma_{jk}^{(r,i,\cdot)}$ for $j = 1, 2, \dots, J_k$ and otherwise update m and the continue backfitting cycle.
 - * Calculate updated $\boldsymbol{\eta}_k^{(r,i+1)}$ and $\boldsymbol{\theta}_k^{(r,i+1)}$ and then **UPDATE** k .
 - **END INNER CYCLE** on convergence of $\beta_k^{(r,\cdot)}$ and the additive predictor terms $\mathbf{Z}_{jk} \gamma_{jk}^{(r,\cdot)}$ and set $\beta_k^{(r+1,1)} = \beta_k^{(r,\cdot)}$, $\gamma_{jk}^{(r+1,1)} = \gamma_{jk}^{(r,\cdot)}$, $\boldsymbol{\eta}_k^{(r+1)} = \boldsymbol{\eta}_k^{(r+1,1)} = \boldsymbol{\eta}_k^{(r,\cdot)}$ and $\boldsymbol{\theta}_k^{(r+1,1)} = \boldsymbol{\theta}_k^{(r,\cdot)}$, for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$, otherwise update i and continue inner cycle
- **END OUTER CYCLE**: if the change in the (penalized) likelihood is sufficiently small, otherwise update r and continue outer cycle.

The matrices $\mathbf{H}_k^{(r)}$ and $\mathbf{S}_{jk}^{(r)}$, defined in Appendix C, are the projection matrices and the shrinking matrices, for the parametric and additive components of the model respectively, at the r^{th} iteration, for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$

The partial residuals $\boldsymbol{\varepsilon}_{ok}^{(r,i,m)}$ and $\boldsymbol{\varepsilon}_{jk}^{(r,i,m)}$ are the current working variables for fitting the parametric and the additive (random effects or smoothing) components of the model respectively and are defined as

$$\boldsymbol{\varepsilon}_{ok}^{(r,i,m)} = \mathbf{z}_k^{(r)} - \sum_{t=1}^{J_k} \mathbf{z}_{tk} \gamma_{tk}^{(r,i,c)} - \mathbf{W}_{kk}^{(r)-1} \sum_{s=1, s \neq k}^p \mathbf{W}_{ks}^{(r)} [\boldsymbol{\eta}_s^{(r,c)} - \boldsymbol{\eta}_s^{(r)}]$$

$$\boldsymbol{\varepsilon}_{jk}^{(r,i,m)} = \mathbf{z}_k^{(r)} - \mathbf{X}_k \beta_k^{(r,i,m+1)} - \sum_{t=1, t \neq j}^{J_k} \mathbf{z}_{tk} \gamma_{tk}^{(r,i,c)} - \mathbf{W}_{kk}^{(r)-1} \sum_{s=1, s \neq k}^p \mathbf{W}_{ks}^{(r)} [\boldsymbol{\eta}_s^{(r,c)} - \boldsymbol{\eta}_s^{(r)}].$$

The full Newton-Raphson step length in the algorithm can be replaced by a step of size α , by updating the linear predictors as

$$\boldsymbol{\eta}_k^{(r+1)}(\alpha) = \alpha \boldsymbol{\eta}_k^{(r+1)} + (1 - \alpha) \boldsymbol{\eta}_k^{(r)}$$

rather than $\eta_k^{(r+1)}$ for $k = 1, 2, \dots, p$, at the end of the inner cycle for the r^{th} outer cycle and then evaluating $\mathbf{u}_k^{(r+1)}$, $\mathbf{W}_{ks}^{(r+1)}$ and $\mathbf{z}_k^{(r+1)}$, for $k = 1, 2, \dots, p$ and $s = 1, 2, \dots, p$, using the $\eta_k^{(r+1)}(\alpha)$ for $k = 1, 2, \dots, p$. The optimum step length for a particular iteration r can be obtained by maximizing $\ell_p(\alpha)$ over α .

The inner (backfitting) cycle of the algorithm can be shown to converge (for cubic smoothing splines and similar linear smoothers), Hastie and Tibshirani (1990, Ch 5). The outer cycle is simply a Newton-Raphson algorithm. Thus if step size optimization is performed, the outer loop will converge as well. Standard general results on Newton-Raphson algorithm ensure convergence (Ortega and Rheinboldt, 1970). Step optimization is rarely needed in practice in our experience.

Appendix 2B : Maximization of the Penalized Likelihood

In this Section it is shown that maximization of the penalized log-likelihood function ℓ_p given by equation (2.6) over the parameters β_k and terms γ_{jk} for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$ leads to the algorithm described in Appendix 2A.

This is achieved by the following two steps :

- (i) The first and second derivatives of the penalized likelihood (2.6) are obtained to give a Newton-Raphson step for maximizing (2.6) with respect to β_k and γ_{jk} for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$.
- (ii) Each step of the Newton-Raphson is achieved using a backfitting procedure cycling through the parameters and through the additive terms of the k linear predictors.

Step(i) The algorithm maximizes the penalized likelihood function ℓ_p , given by (2.6), using a Newton-Raphson algorithm. The first derivative (score function) and the second derivatives of ℓ_p with respect to β_k and γ_{jk} for all $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$ are evaluated at iteration r at the current predictors $\eta_k^{(r)}$ for $k = 1, 2, \dots, p$.

Let $\alpha_k^T = [\beta_k^T, \gamma_{1k}^T, \gamma_{2k}^T, \dots, \gamma_{J_k k}^T]$, $\mathbf{a}_k = \frac{\partial \ell_p}{\partial \alpha_k}$ and $\mathbf{A}_{ks} = -\frac{\partial^2 \ell_p}{\partial \alpha_k \partial \alpha_s^T}$ for $k = 1, 2, \dots, p$ and $s = 1, 2, \dots, p$, and let $\alpha^T = [\alpha_1^T, \alpha_2^T, \dots, \alpha_p^T]$, $\mathbf{a} = \frac{\partial \ell_p}{\partial \alpha}$ and $\mathbf{A} = -\frac{\partial^2 \ell_p}{\partial \alpha \partial \alpha^T}$.

The Newton-Raphson step is given by $\mathbf{A}^{(r)}[\alpha^{(r-1)} - \alpha^{(r)}] = \mathbf{a}^{(r)}$, i.e.

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1p} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2p} \\ \vdots & \cdots & \cdots & \vdots \\ \mathbf{A}_{p1} & \mathbf{A}_{p2} & \cdots & \mathbf{A}_{pp} \end{bmatrix}^{(r)} \begin{bmatrix} \alpha_1^{(r+1)} - \alpha_1^{(r)} \\ \alpha_2^{(r+1)} - \alpha_2^{(r)} \\ \vdots \\ \alpha_p^{(r+1)} - \alpha_p^{(r)} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_p \end{bmatrix}^{(r)}$$

where the matrix $\mathbf{A}_{ks}$ is given by

$$\begin{bmatrix} \mathbf{X}_k^T \mathbf{W}_{ks} \mathbf{X}_s & \mathbf{X}_k^T \mathbf{W}_{ks} \mathbf{Z}_{1s} & \cdots & \mathbf{X}_k^T \mathbf{W}_{ks} \mathbf{Z}_{J_s s} \\ \mathbf{Z}_{1k}^T \mathbf{W}_{ks} \mathbf{X}_s & \mathbf{Z}_{1k}^T \mathbf{W}_{ks} \mathbf{Z}_{1s} + \mathbf{G}_{1k} \text{ (if } s = k) & \cdots & \mathbf{Z}_{1k}^T \mathbf{W}_{ks} \mathbf{Z}_{J_s s} \\ \vdots & \cdots & \cdots & \vdots \\ \mathbf{Z}_{J_k k}^T \mathbf{W}_{ks} \mathbf{X}_s & \mathbf{Z}_{J_k k}^T \mathbf{W}_{ks} \mathbf{Z}_{1s} & \cdots & \mathbf{Z}_{J_k k}^T \mathbf{W}_{ks} \mathbf{Z}_{J_s s} + \mathbf{G}_{J_k k} \text{ (if } s = k) \end{bmatrix}$$

and the vector

$$\mathbf{a}_k^{(r)} = \begin{bmatrix} \mathbf{X}_k^T \mathbf{u}_k^{(r)} \\ \mathbf{Z}_{1k}^T \mathbf{u}_k^{(r)} - \mathbf{G}_{1k} \boldsymbol{\gamma}_{1k}^{(r)} \\ \vdots \\ \mathbf{Z}_{J_k k}^T \mathbf{u}_k^{(r)} - \mathbf{G}_{J_k k} \boldsymbol{\gamma}_{J_k k}^{(r)} \end{bmatrix}$$

where $\mathbf{u}_k = \frac{\partial \ell}{\partial \boldsymbol{\eta}_k}$ and $\mathbf{W}_{ks} = -\frac{\partial^2 \ell}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_s^T} = -\text{diag} \left\{ \frac{\partial^2 \ell_i}{\partial \eta_{ik} \partial \eta_{is}} \right\}$ over $i = 1, 2, \dots, n$, for $k = 1, 2, \dots, p$ and $s = 1, 2, \dots, p$ (see Appendix B for alternative weight matrices).

Step (ii) Now considering the row corresponding to updating $\boldsymbol{\gamma}_{jk}$ gives

$$\mathbf{G}_{jk} \left[\boldsymbol{\gamma}_{jk}^{(r+1)} - \boldsymbol{\gamma}_{jk}^{(r)} \right] + \mathbf{Z}_{jk}^T \sum_{s=1}^p \mathbf{W}_{ks}^{(r)} \left[\boldsymbol{\eta}_s^{(r+1)} - \boldsymbol{\eta}_s^{(r)} \right] = \mathbf{Z}_{jk}^T \mathbf{u}_k^{(r)} - \mathbf{G}_{jk} \boldsymbol{\gamma}_{jk}^{(r)}$$

Expanding and rearranging this gives

$$\mathbf{Z}_{jk} \boldsymbol{\gamma}_{jk}^{(r+1)} = \mathbf{S}_{jk}^{(r)} \boldsymbol{\varepsilon}_{jk}^{(r)} \quad (2.10)$$

where, for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$,

$$\mathbf{S}_{jk}^{(r)} = \mathbf{Z}_{jk} \left(\mathbf{Z}_{jk}^T \mathbf{W}_{kk}^{(r)} \mathbf{Z}_{jk} + \mathbf{G}_{jk} \right)^{-1} \mathbf{Z}_{jk}^T \mathbf{W}_{kk}^{(r)}$$

is a shrinking (smoothing) matrix and where

$$\boldsymbol{\varepsilon}_{jk}^{(r)} = \mathbf{z}_k^{(r)} - \mathbf{X}_k \boldsymbol{\beta}_k^{(r+1)} - \sum_{t=1, t \neq j}^{J_k} \mathbf{z}_{tk} \boldsymbol{\gamma}_{tk}^{(r+1)} - \mathbf{W}_{kk}^{(r)-1} \sum_{s=1, s \neq k}^p \mathbf{W}_{ks}^{(r)} \left[\boldsymbol{\eta}_s^{(r+1)} - \boldsymbol{\eta}_s^{(r)} \right]$$

are the partial residuals and $\mathbf{z}_k^{(r)} = \boldsymbol{\eta}_k^{(r)} + \mathbf{W}_{kk}^{(r)-1} \mathbf{u}_k^{(r)}$ is the adjusted dependent variable.

[Note a device for obtaining updated estimate $\boldsymbol{\gamma}_{jk}^{(r+1)}$ in (2.10) is to apply weighted least squares estimation to an augmented data model given by:

$$\begin{bmatrix} \boldsymbol{\varepsilon}_{jk}^{(r)} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{jk} \\ -\mathbf{D}_{jk} \end{bmatrix} \boldsymbol{\gamma}_{jk} + \begin{bmatrix} \mathbf{e}_{0k} \\ \mathbf{e}_{jk} \end{bmatrix} \quad (2.11)$$

where $\mathbf{0}$ is a vector of zeros of length q_{jk} , $\mathbf{D}_{jk}^T \mathbf{D}_{jk} = \mathbf{G}_{jk}$, $\mathbf{e}_{0k} \sim N(0, \mathbf{W}_{kk}^{(r)-1})$ and $\mathbf{e}_{jk} \sim N(0, \mathbf{I})$. This device can be generalized to estimate $\boldsymbol{\alpha}_k$ and even $\boldsymbol{\alpha}$.]

Similarly taking the row corresponding to $\boldsymbol{\beta}_k$ and rearranging gives

$$\mathbf{X}_k \boldsymbol{\beta}_k^{(r+1)} = \mathbf{H}_k^{(r)} \boldsymbol{\varepsilon}_{ok}^{(r)} \quad (2.12)$$

where, for $k = 1, 2, \dots, p$,

$$\mathbf{H}_k^{(r)} = \mathbf{X}_k \left(\mathbf{X}_k^T \mathbf{W}_{kk}^{(r)} \mathbf{X}_k \right)^{-1} \mathbf{X}_k^T \mathbf{W}_{kk}^{(r)}$$

and

$$\boldsymbol{\varepsilon}_{ok}^{(r)} = \mathbf{z}_k^{(r)} - \sum_{t=1}^{J_k} \mathbf{z}_{tk} \boldsymbol{\gamma}_{tk}^{(r+1)} - \mathbf{W}_{kk}^{(r)-1} \sum_{s=1, s \neq k}^p \mathbf{W}_{ks}^{(r)} \left[\boldsymbol{\eta}_s^{(r+1)} - \boldsymbol{\eta}_s^{(r)} \right]$$

for $k = 1, 2, \dots, p$.

A single r^{th} Newton-Raphson step is achieved using a backfitting procedure for each k , cycling through (2.12) and then (2.10) for $j = 1, 2, \dots, J_k$ and cycling over $k = 1, 2, \dots, p$ until convergence of the set of updated values $\alpha_k^{(r+1)}$ for $k = 1, 2, \dots, p$. The updated predictors $\eta_k^{(r+1)}$, first derivatives $\mathbf{u}_k^{(r+1)}$, diagonal weighted matrices $\mathbf{W}_{ks}^{(r+1)}$ and adjusted dependent variables $\mathbf{z}_k^{(r+1)}$, for $k = 1, 2, \dots, p$ and $s = 1, 2, \dots, p$, are then calculated and the $(r + 1)^{th}$ Newton-Raphson step is performed, until convergence of the Newton-Raphson algorithm.

Exercises for Chapter 2

Practical 1: Introduction to gamlss packages

- Q1 To familiarise yourself with the GAMLSS package repeat the commands given in Section 2.3.
- Q2 The **gamlss** and **gamlss.dist** packages contain several distributions. Typing

```
?gamlss.family
```

in R will show all the available distributions in the two **gamlss** packages.

You can explore the shape and other properties of the distributions. For example the following R script will produce the probability density function (pdf), cumulative distribution function (cdf), inverse c.d.f., and a histogram of a random sample obtained from a Gamma distribution:

```
PPP <- par(mfrow=c(2,2))
plot(function(y) dGA(y, mu=10 ,sigma=0.3),0.1, 25) # pdf
plot(function(y) pGA(y, mu=10 ,sigma=0.3), 0.1, 25) #cdf
plot(function(y) qGA(y, mu=10 ,sigma=0.3), 0, 1) # inverse cdf
hist(rGA(100,mu=10,sigma=.3)) # randomly generated values
par(PPP)
```

Note that the same type of plots produced say by

```
plot(function(y) dGA(y, mu=10 ,sigma=0.3), 0, 25) # pdf
```

can also be produced by using the function `curve()` as in

```
curve(dGA(y=x, mu=10, sigma=.3),0, 25)
```

To explore discrete distributions use:

```
PPP <- par(mfrow=c(2,2))
plot(function(y) dNBI(y, mu = 10, sigma =0.5 ), from=0, to=40, n=40+1, type="h",
      main="pdf", ylab="pdf(x)")
cdf <- stepfun(0:39, pNBI(0:40, mu=10, sigma=0.5 ), f = 0)
plot(cdf,main="cdf", ylab="cdf(x)", do.points=FALSE )
```

```

invcdf <- stepfun(seq(0.01,.99,length=39), qNBI(seq(0.01,.99,length=40),
      mu=10, sigma=0.5 ), f = 0)
plot(invcdf, main="inverse cdf",ylab="inv-cdf(x)", do.points=FALSE )
tN <- table(Ni <- rNBI(1000,mu=5, sigma=0.5))
r <- barplot(tN, col='lightblue')
par(PPP)

```

Note that to find moments or to check if a distribution integrates or sums to one, the functions `integrate()` or `sum` can be used. For example

```
integrate(function(y) dGA(y, mu=10, sigma=.1),0, Inf)
```

will check that the distribution integrates to one, and

```
integrate(function(y) y*dGA(y, mu=10, sigma=.1),0, Inf)
```

will give the mean of the specific gamma distribution.

The density function of a GAMLSS family distribution can be plotted also using the `pdf.plot()` of the GAMLSS package. Use for example

```
pdf.plot(family=GA, mu=10, sigma=c(.1,.5,1,2),
      min=0.01,max=20, step=.5)
```

Try plotting other continuous distributions, e.g. IG (inverse Gaussian), PE (power exponential) and BCT (Box-Cox t) and discrete distributions, e.g. NBI (negative binomial type I), and PIG (Poisson inverse Gaussian).

Note that you should first load the package **gamlss.dist** by: `library(gamlss.dist)`, so that all current GAMLSS family distributions are available.

Chapter 3

Continuous response: fitting distributions

3.1 Introduction

The great advantage of GAMLSS is its ability to fit a variety of different distributions to a response variable so that an appropriate distribution can be chosen among different alternatives. This chapter focuses on how the `gamlss` package can be used to fit continuous distributions to a simple random sample of observations of a response variable Y . We shall also take the opportunity to introduce the different continuous distributions available in the package. We are assuming that the “continuous” response variable Y comes from a population distribution which can be modelled by a theoretical probability density function $f_Y(y|\boldsymbol{\theta})$, where the parameter vector $\boldsymbol{\theta}$ can be up to four dimensions, i.e. $\boldsymbol{\theta}^\top = (\mu, \sigma, \nu, \tau)$, i.e. $Y|(\boldsymbol{\theta}) \sim f_Y(y|(\boldsymbol{\theta}))$. Our task is to find the appropriate distribution $f_Y(y|\boldsymbol{\theta})$ and estimate the parameters $\boldsymbol{\theta}$. The case where the distribution of Y depends on an explanatory variable X , i.e. $Y|(\boldsymbol{\theta}, X = x) \sim f_Y(y|(\boldsymbol{\theta}, x))$ is examined in the Chapter 4.

3.1.1 Types of distribution in GAMLSS

In the GAMLSS model in Section (2.1), the population probability (density) function $f_Y(y|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\mu, \sigma, \nu, \tau)$, is deliberately left general with no explicit conditional distribution specified for the response variable Y . The only restriction that the R implementation of GAMLSS, Stasinopoulos *et al.* (2008), has for specifying the distribution of Y is that the function $f(Y|\boldsymbol{\theta})$ and its first (and optionally expected second and cross) derivatives with respect to each of the parameters of $\boldsymbol{\theta}$ must be computable. Explicit derivatives are preferable, but numerical derivatives can be used (resulting in reduced computational speed).

Fitting a parametric distribution within the GAMLSS family can be achieved using the command `gamlss(y ~ 1, family = " ")` where the argument `family` can take any `gamlss.family` distribution. The type of distribution of course depends on the type of response variable. There are three distinct types of distribution in GAMLSS:

1. continuous distributions,
2. discrete distributions,

Table 3.1: Continuous GAMLSS family distributions defined on $(-\infty, +\infty)$

Distributions	family	no parameters	skewness	kurtosis
Exponential Gaussian	exGAUS	3	positive	-
Exponential Generalized beta type 2	EGB2	4	both	lepto
Generalized t	GT	4	(symmetric)	lepto
Gumbel	GU	2	(negative)	-
Johnson's SU	JSU	4	both	lepto
Johnson's original SU	JSUo	4	both	lepto
Logistic	LO	2	(symmetric)	(lepto)
Normal-Exponential- t	NET	2 + (2 fixed)	(symmetric)	lepto
Normal	NO-NO2	2	(symmetric)	
Normal Family	NOF	3	(symmetric)	(meso)
Power Exponential	PE-PE2	3	(symmetric)	both
Reverse Gumbel	RG	2	positive	-
Sinh Arcsinh	SHASH	4	both	both
Skew Exponential Power (type 1 to 4)	SEP1-SEP4	4	both	both
Skew t (type 1 to 5)	ST1-ST5	4	both	lepto
t Family	TF	3	(symmetric)	lepto

3. mixed distributions.

Continuous distributions are usually defined on $(-\infty, +\infty)$, $(0, +\infty)$ or $(0, 1)$, but can take other support ranges. Discrete distributions are usually defined on $y = 0, 1, 2, \dots, n$, where n is a known finite value or n is infinite, i.e. usually discrete (count) values.

Mixed distributions are a special case of finite mixture distributions described in Chapter 7 and are mixtures of continuous and discrete distributions, i.e. continuous distributions where the range of Y has been expanded to include some discrete values with non-zero probabilities.

In this Chapter we will look at the continuous distributions. In Chapter 5 we will discuss the discrete distributions. Chapter 7 will deal with the general case of finite mixtures.

3.1.2 Types of continuous distributions

Continuous distributions can be symmetric, negatively or positively skewed, and also mesokurtic, leptokurtic or platykurtic. Figure 3.1 shows i) a negatively skew, ii) a positively skew, iii) a platykurtic and iv) a leptokurtic distribution. Note that leptokurtic and platykurtic distributions are judged by comparison to the normal distribution which is mesokurtic. A leptokurtic distribution has thicker (fatter) tails than the normal distribution while a platykurtic distribution has thinner (or shorter) tails than the normal.

Table 3.1 provides a list of the continuous `gamlss.family` distributions defined on support range $(-\infty, +\infty)$ available in the current version of GAMLSS software, while Tables 3.2 and 3.3 provide a list of distributions defined on $(0, +\infty)$ and $(0, 1)$ respectively. Note that 'both' in the skewness column of Tables 3.1, 3.2 and 3.3 indicates that the distribution can be negative or positive skew, while 'both' in the kurtosis column indicates that the distribution can be platykurtic or leptokurtic. A brackets indicates that the skewness or kurtosis cannot be modelled independently of the location and scale parameters.

Many of the distributions of Tables 3.1, 3.2 and 3.3 can be generated by one (or more) of the methods described in Section 3.8.

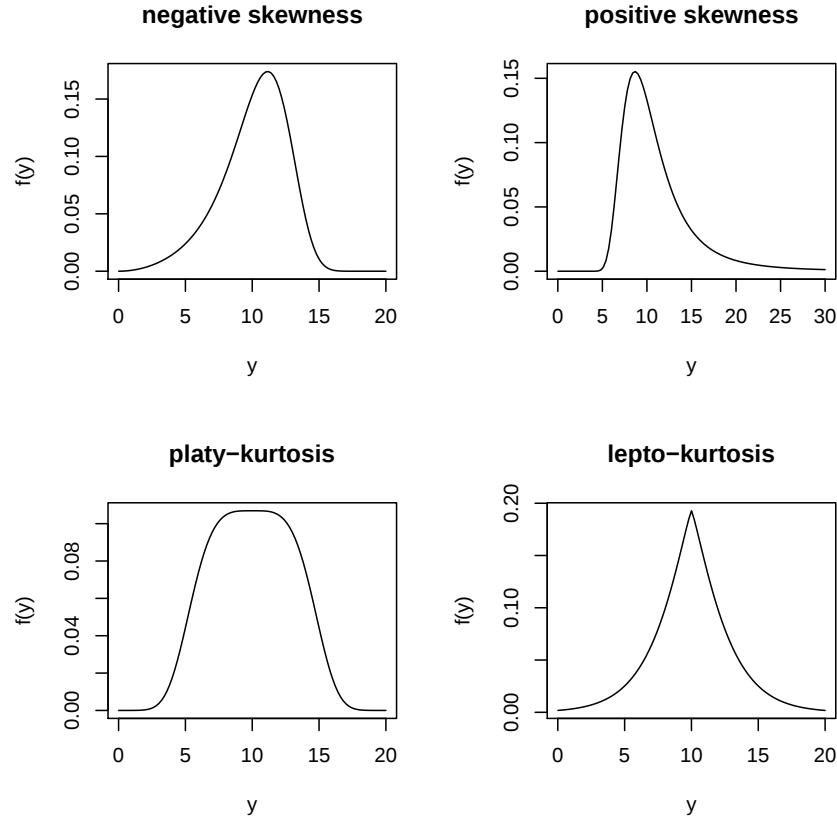


Figure 3.1: Showing different types of continuous distributions

Table 3.2: Continuous GAMLSS family distributions defined on $(0, +\infty)$

Distributions	family	no parameters	skewness	kurtosis
Box-Cox Cole and Green	BCCG	3	both	-
Box-Cox Power Exponential	BCPE	4	both	both
Box-Cox- t	BCT	4	both	lepto
Exponential	EXP	1	(positive)	-
Gamma	GA	2	(positive)	-
Generalized Beta type 2	GB2	4	both	both
Generalized Gamma	GG-GG2	3	positive	-
Generalized Inverse Gaussian	GIG	3	positive	-
Inverse Gaussian	IG	2	(positive)	-
Log Normal	LOGNO	2	(positive)	-
Log Normal family	LN0	2 + (1 fixed)	positive	
Reverse Generalized Extreme	RGE	3	positive	-
Weibull	WEI-WEI3	2	(positive)	-

Table 3.3: Continuous GAMLSS family distributions defined on $(0, 1)$

Distributions	family	no parameters	skewness	kurtosis
Beta	BE	2	(both)	-
Beta original	BEo	2	(both)	-
Generalized beta type 1	GB1	4	(both)	(both)

3.2 Summary of methods generating distributions

Here we give a summary of the methods on how many of the distributions in Tables 3.1, 3.2 and 3.3 for the random variable Y can be generated. Distribution families for Y can be generated by one (or more) of the following methods:

1. univariate transformation from a single random variable
2. transformation from two or more random variables
3. truncation distributions
4. a (continuous or finite) mixture of distributions
5. Azzalini type methods
6. splicing distributions
7. stopped sums
8. systems of distributions

These methods are discussed in detail in the Appendix of this chapter.

Here we look briefly at three of the methods and give examples.

3.2.1 Distributions generated by univariate transformation

Many three and four parameter families of continuous distribution for Y can be defined by assuming that a transformed variable Z , obtained from Y , has a simple well known distribution. The parameters of the distribution of Y may come from parameters of the univariate transformation or from parameters of the distribution of Z or both.

A simple example of a distribution obtained by transformation is a lognormal distribution, $\text{LOGNO}(\mu, \sigma)$. Let $Y = \exp(Z)$ where $Z \sim \text{NO}(\mu, \sigma)$, then $Y \sim \text{LOGNO}(\mu, \sigma)$.

An example where the univariate transform introduces an extra parameter is a generalized gamma distribution, $\text{GG}(\mu, \sigma, \nu)$. Let $Y = \mu Z^{1/\nu}$ where $Z \sim \text{GA}(1, \sigma\nu)$, then $Y \sim \text{GG}(\mu, \sigma, \nu)$.

Table 3.5 (in the Appendix) gives distributions generated by univariate transformation available in GAMLSS and shows how they can be generated.

3.2.2 Azzalini type methods

Azzalini type methods have been used to introduce skewness into symmetric distribution family. Let $f_{Z_1}(z)$ be a probability density function symmetric about z equals zero and let $F_{Z_2}(z)$ be an absolutely continuous cumulative distribution function such that $dF_{Z_2}(z)/dz$ is symmetric

about zero. Then if $w(z)$ is any odd function of z , $f_Y(y)$ is a proper probability density function given by: Hence

$$f_Y(y) = \frac{2}{\sigma} f_{Z_1}(z) F_{Z_2}[w(z)] \quad (3.1)$$

where $z = (y - \mu)/\sigma$, Azzalini and Capitanio (2003) Proposition 1.

Table 3.6 (in the Appendix) gives distributions generated by Azzalini type methods.

For example the *skew exponential power type 1* family for $-\infty < Y < \infty$, Azzalini (1986), denoted by $\text{SEP1}(\mu, \sigma, \nu, \tau)$, is defined by assuming Z_1 and Z_2 have power exponential type 2, $\text{PE2}(0, \tau^{1/\tau}, \tau)$, distributions in (3.18). Figure 3.2 shows symmetric SEP1 distributions (equivalent to power exponential distributions) obtained by setting $\nu = 0$. Figure 3.2 plots the $\text{SEP1}(\mu, \sigma, \nu, \tau)$ distribution for $\mu u = 0$, $\sigma = 1$, $\nu = 0$ and $\tau = .5, 1, 2, 5, 10, 1000$. Figure 3.3 shows skew SEP1(μ, σ, ν, τ) distributions where $\mu u = 0$ and $\sigma = 1$. Figure 3.3(a) and (b) plot the SEP1 distribution for $\nu = 0, 1, 3, 100$ and $\tau = .5$ and $\tau = 2$ respectively. Note the when $\tau = 2$ the SEP1 becomes the skew normal type 1 distribution. Figure 3.3(c) and (d) plot the SEP1 distribution for $\nu = 0, -.2, -.4, -.6, -.8, -1$ and $\tau = 10$ and $\tau = 1000$ respectively.

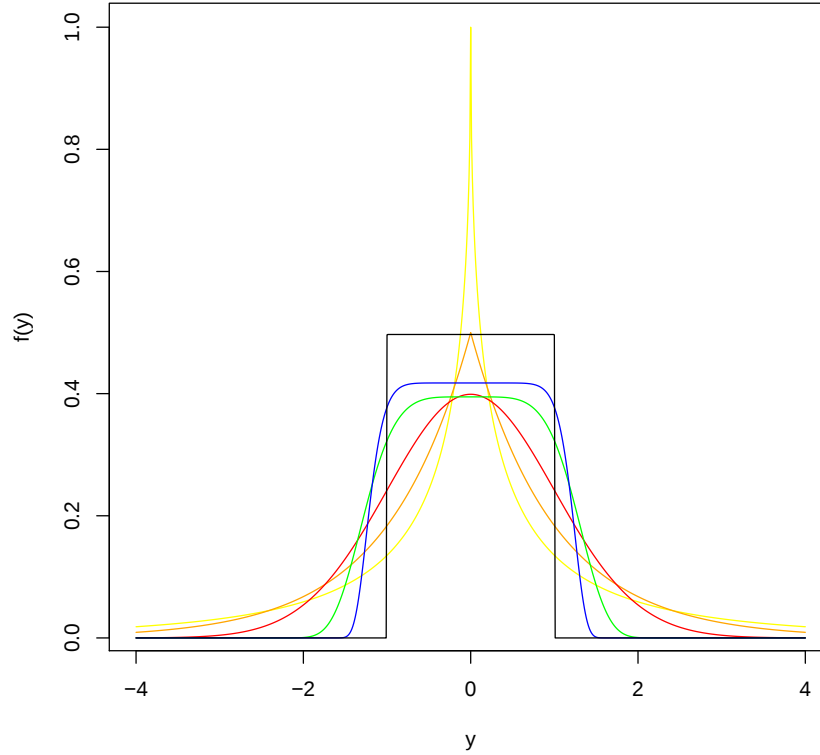


Figure 3.2: Showing different types of continuous distributions

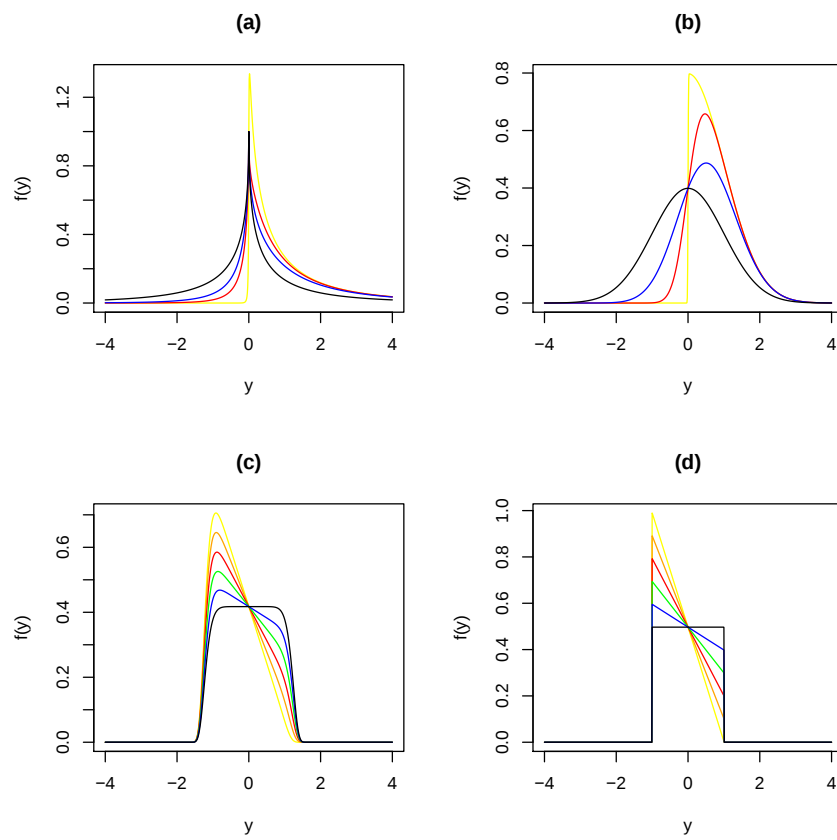


Figure 3.3: Showing different types of continuous distributions

3.2.3 Splicing distributions

Splicing has been used to introduce skewness into symmetric distribution family. Let Y_1 and Y_2 have probability density functions that are symmetric about μ . A spliced distribution for Y may be defined by

$$f_Y(y) = \frac{2}{(1+k)} \{f_{Y_1}(y)I(y < \mu) + kf_{Y_2}(y)I(y \geq \mu)\}. \quad (3.2)$$

where $k = f_{Y_1}(\mu)/f_{Y_2}(\mu)$ and $I()$ is an indicator variable taking value 1 if the condition is true and 0 otherwise.

Table 3.7 (in the Appendix) gives distributions generated by splicing two distributions.

3.3 Comparison of properties of continuous distributions

The choice of model distribution for a particular response variable Y is usually based on how well it fits the data as judged by the fitted global deviance $GDEV = -2 \log \hat{l}$, i.e. minus twice the fitted log likelihood function, and tests and information criteria (e.g. AIC or SBC) based on $GDEV$.

Where more than one distribution fits the data adequately the choice of distribution may be made on other criteria, e.g. properties of the particular distribution. For example an explicit formula for the mean, median or mode of Y may be desirable in a particular application.

The term explicit indicates that the particular function or measure can be obtained using standard functions (available in R), i.e. not requiring numerical integration or numerical solution.

The following are properties of the distribution that may be relevant in choosing the model distribution:

1. Explicit probability density function, cumulative distribution function and inverse cumulative distribution function
2. Explicit centiles and centile based measures of location, scale, skewness and kurtosis (e.g. median and semi-interquartile range)
3. Explicit moment based measures of location, scale, skewness and kurtosis (e.g. mean, standard deviation, skewness $\sqrt{\beta_1}$ and kurtosis β_2)
4. Explicit mode(s)
5. Continuity of the probability density function $f_Y(y|\mu, \sigma, \nu, \tau)$ and its derivatives with respect to y
6. Continuity of the derivatives of the probability density function $f_Y(y|\mu, \sigma, \nu, \tau)$ with respect to μ, σ, ν and τ
7. Flexibility in modelling skewness and kurtosis

Distributions generated by univariate transformation often satisfy all the desirable properties above, except perhaps the flexibility in modelling skewness and kurtosis.

An important disadvantage of distributions generated by Azzalini type methods are that their cumulative distribution function (cdf) is not explicitly available, but requires numerical integration. Their inverse cdf requires a numerical search and many integrations. Consequently

both functions can be slow, particularly for large data sets. Centiles and centile based measures (e.g. the median) are not explicitly available. Moment based measures are usually complicated, if available. However they can be flexible in modelling skewness and kurtosis.

An important disadvantage of distributions generated by splicing is often a lack of continuity of the first and/or second derivatives of the probability density function with respect to y and μ at the splicing point. However they can be flexible in modelling skewness and kurtosis.

3.4 Theoretical considerations for fitting parametric distribution

The standard method of fitting a parametric family to a random sample of values of a single response variable Y is the method of maximum likelihood. That is, maximizing the likelihood of observing the sample of independent observations $\mathbf{y} = (y_1, y_2, \dots, y_n)$,

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n f_Y(y_i|\boldsymbol{\theta}) \quad (3.3)$$

with respect to the parameter(s) $\boldsymbol{\theta}$. This is the exact probability of observing the data $\mathbf{y}$ given the parameters $\boldsymbol{\theta}$, provided the distribution of Y is discrete. If however the distribution of Y is continuous, then in practice value y_i is observed to a certain level of accuracy, say $y_i \pm \Delta_i$. [For example, if y_i is rounded to the nearest first decimal place then $\Delta_i = 0.05$ and, for example, an observed value $y_i = 5.7$ corresponds to $5.65 < y < 5.75$.] Hence the true likelihood can be defined as:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n [F_Y(y_i + \Delta_i|\boldsymbol{\theta}) - F_Y(y_i - \Delta_i|\boldsymbol{\theta})] \quad (3.4)$$

where $F_Y(\cdot)$ is the cumulative distribution function of Y . The likelihood in (3.4) is bounded above by one and so cannot go to infinity. Assume the Δ_i 's are sufficiently small then

$$L(\boldsymbol{\theta}) \approx \prod_{i=1}^n f_Y(y_i|\boldsymbol{\theta})\Delta_i = \left[\prod_{i=1}^n \Delta_i \right] \left[\prod_{i=1}^n f_Y(y_i|\boldsymbol{\theta}) \right] \quad (3.5)$$

Hence the log likelihood $\ell(\boldsymbol{\theta})$ is given approximately by:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^n \log f_Y(y_i|\boldsymbol{\theta}) + \sum_{i=1}^n \log \Delta_i \quad (3.6)$$

Clearly the second summation does not depend on $\boldsymbol{\theta}$ and hence when maximizing $\ell(\boldsymbol{\theta})$ over $\boldsymbol{\theta}$ only the first term needs to be maximized. Occasionally this creates problems, (especially in flexible models such as GAMLSS) where the fact that we ignored the accuracy with which the response variable is measured can occasionally lead to the likelihood shooting up to infinity. To demonstrate the point consider a single observation y from a normal distribution, i.e. $Y \sim \text{NO}(\mu, \sigma)$. The likelihood is maximized as $\mu \rightarrow y$ and $\sigma \rightarrow 0$ and the likelihood goes to ∞ . The problem is avoided by taking account of the measurement accuracy by using (3.4) instead of (3.5).

Within GAMLSS we have adopted the definition of the likelihood given in (3.3). Models can be maximized using (3.4) with the help of the package **gamlss.cens** which is designed for censored or interval response variables. In this case one can think of the response variable having the form

- $(-\infty, y_{i2})$ if the response is left censored
- $(y_{i1}, +\infty)$ if the response is right censored
- (y_{i1}, y_{i2}) if the response lies within an interval

In all three cases the likelihood takes the form

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n [F_Y(y_{i2}|\boldsymbol{\theta}) - F_Y(y_{i1}|\boldsymbol{\theta})] \quad (3.7)$$

In practice it is more convenient to work with the log likelihood $\log L(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta})$, or $-2\ell(\boldsymbol{\theta})$ a quantity we call the Global Deviance (or GD) for abbreviation. Assuming that the population distribution of Y is $f_Y(y|\boldsymbol{\theta})$ for some value of $\boldsymbol{\theta}$, then provided regularity conditions hold, using well known asymptotic results, we have that the expected value of the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ is $\boldsymbol{\theta}$ and its asymptotic variance is $\text{Var}(\hat{\boldsymbol{\theta}}) = I_E^{-1}(\hat{\boldsymbol{\theta}})$ where $I_E(\hat{\boldsymbol{\theta}})$ is the (Fisher's) expected information $I_E(\boldsymbol{\theta}) = E[d^2\ell/d\boldsymbol{\theta}\boldsymbol{\theta}^T]$, evaluated at the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$. The expected information can be substituted with the observed information matrix $I(\hat{\boldsymbol{\theta}}) = H^{-1}(\hat{\boldsymbol{\theta}})$ where H is the Hessian matrix, that is, the second derivative of the log likelihood, $H = d^2\ell/d\boldsymbol{\theta}\boldsymbol{\theta}^T$, evaluated at the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$.

In order to test whether a specific predictor parameter is different from zero, a (generalized likelihood ratio) Chi-squared test is used, comparing the deviance change Δ when the parameter is set to zero with a $\chi^2_{1,0.05}$ critical value. Confidence intervals for individual parameters θ can be constructed in several ways:

- the usual symmetric way using the standard errors obtained from the observed or expected information matrix.
- using the profile likelihood.
- using bootstrap

An alternative more flexible way of estimating the probability density function of a continuous response variable Y (particularly in the absence of explanatory variables) is using nonparametric density function estimation, as described for example in Silverman (1986) or Wand and Jones (1995). This is a well established technique and in certain cases could be a preferable method of density function estimation. Nonparametric density function estimation has its own problems, notably the need to estimate the smoothing parameter and the inability to cope with discrete data. Here we concentrate on fitting parametric distributions to the data. Our approach is to fit different parametric distributions and choose between them using a generalized Akaike information criterion, (GAIC). For continuous data, nonparametric density estimation could be a useful tool in helping us with this choice of the parametric distribution.

3.5 How to fit distributions in R

3.5.1 The function `distHist()`

The specifically designed function `histDist()` will fit a `gamlss.family` distribution and automatically produce a graphical representation of the data and the fitted distribution. Section 3.6

shows how to use `histDist()` to fit a distribution to a continuous variable. Section 5.3 shows the same for discrete variables.

To fit a particular distribution to a random sample `y` from a random variable Y , and plot the fitted distribution, use the function `histDist()`. The function `histDist()` has the following arguments:

- i) `y`: the vector containing the values of the Y variable
- ii) `family`: appropriate GAMLSS family distribution
- iii) `freq`: the observed frequencies corresponding to the values in `y` (usually appropriate for a discrete response variable)
- iv) `xmin` and `xmax`: for minimum and maximum values in the x-axis
- v) `g.control`: for passing `gamlss.control` parameters into the fitting of the distribution
- vi) `density`: whether to plot a nonparametric density plot, together with the parametric fitted distribution (only for continuous distributions).

In the next subsections we will use examples to demonstrate the use of the function `histDist()`.

3.6 Examples of fitting continuous distributions to data

3.6.1 The Parzen data

Data summary:

R data file: `parzen` in package `gamlss.dist` of dimensions 63×1

source: Hand *et al.* (1994)

variables

`snowfall` : the annual snowfall in Buffalo, NY (inches) for the 63 years, from 1910 to 1972 inclusive.

purpose: to demonstrate the fitting of continuous distribution to a single variable.

conclusion: normal assumption seems adequate

Fitting and display the model

The first data set is the snowfall data used by Parzen (1979) and also contained in Hand *et al.* (1994), data set 278. The data give the annual snowfall (inches) in Buffalo, NY for the 63 years, from 1910 to 1972 inclusive.

```
> library(gamlss.dist)
```

```
***** GAMLSS Version 1.9-0 *****
```

```
For more on GAMLSS look at http://www.gamlss.com/
```

```
Type gamlssNews() to see new features/changes/bug fixes.
```

```
> data(parzen)
> names(parzen)

[1] "snowfall"
```

Here we fit the data to a normal (NO), gamma (GA), power exponential (PE) and a Box-Cox power exponential (BCPE) distribution. A comparison of normal with gamma explores whether there is positive skewness in the data. A comparison of normal with power exponential explores the possibility of kurtosis, while the BCPE will show whether both skewness and kurtosis are exhibited in the data. The GAIC will help us with the choice between the different distributions.

```
> op <- par(mfrow = c(2, 2))
> mNO <- histDist(parzen$snowfall, "NO", density = TRUE, main = "(a)",
+   ymax = 0.017)
> mGA <- histDist(parzen$snowfall, "GA", density = TRUE, main = "(b)",
+   ymax = 0.017)
> mPE <- histDist(parzen$snowfall, "PE", density = TRUE, main = "(c)",
+   ymax = 0.017)
> mBCPE <- histDist(parzen$snowfall, "BCPE", density = TRUE, main = "(d)",
+   ymax = 0.017)
> par(op)
```

Note that the option `density=TRUE` requests a non-parametric kernel density estimate to be included in the plot.

```
> GAIC(mNO, mGA, mPE, mBCPE)
```

	df	AIC
mNO	2	580.7331
mPE	3	581.3780
mBCPE	4	583.2114
mGA	2	583.8153

The default penalty for the `GAIC()` function is `k=2` the Akaike information criterion. (Note that we could have used the equivalent function `AIC()`). The AIC criterion shows that the normal distribution fits the data adequately. Figure 3.4 shows the four different distributions fitted.

Checking the model

Testing the adequacy of the normal model (or any other model) using the Kolmogorov-Smirnov goodness of fit test, as provided by the R function `ks.test()`, is not advisable here since we have to estimate the distributional parameters μ and σ so the test is invalid. A check of the (normalized quantile) residuals would provide a way of investigating the adequacy of the fit. The true (normalized quantile) residuals are defined as $r_i = \Phi^{-1}(u_i)$ where Φ^{-1} is the inverse cumulative distribution function of a standard normal variate and $u_i = F_Y(y_i|\mu_i, \sigma_i, \nu_i, \tau_i)$. The true (normalized quantile) residuals are independent standard normal variables. We expect the fitted (normalized quantile) residuals $\hat{r}_i$ to behave approximately as normally distributed variables (even though the original observations Y_i are not necessarily normal), so a normal Q-Q plot of the residuals is appropriate here. The `gamlss` package provides the functions i)

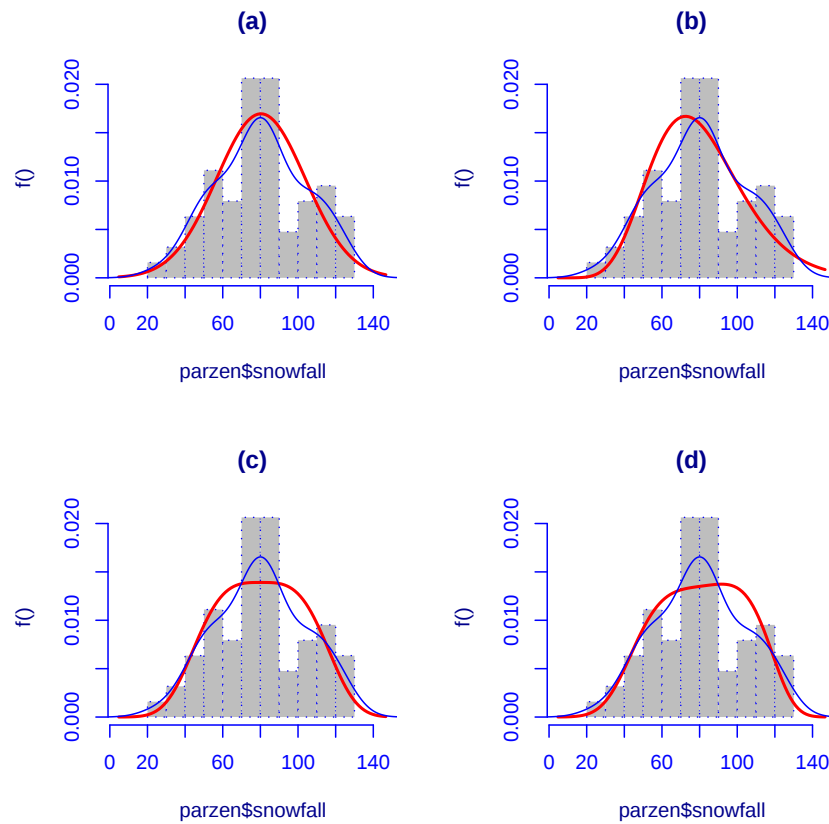


Figure 3.4: Parzen's snowfall data with a kernel density estimate (blue) and fitted (a) normal, (b) gamma, (c) power exponential, and (d) BCPE distributions respectively (red).

`plot()` and ii) `wp()` for plotting QQ-plots. Figure 3.5 shows the results of using `plot(mNO)` while Figure 3.6 shows the result of using `wp(mNO)`. Both the QQ-plot, (in the right bottom corner of Figure 3.5 and the worm plot in Figure 3.6 indicate that there is no reason to worry about the inadequacy of the fit. Note that a worm plot is a detrended Q-Q plot (i.e. where the line in a Q-Q plot has been transformed horizontally). Note also that not all the plots in Figure 3.5 are useful as they will be in a regression type situation.

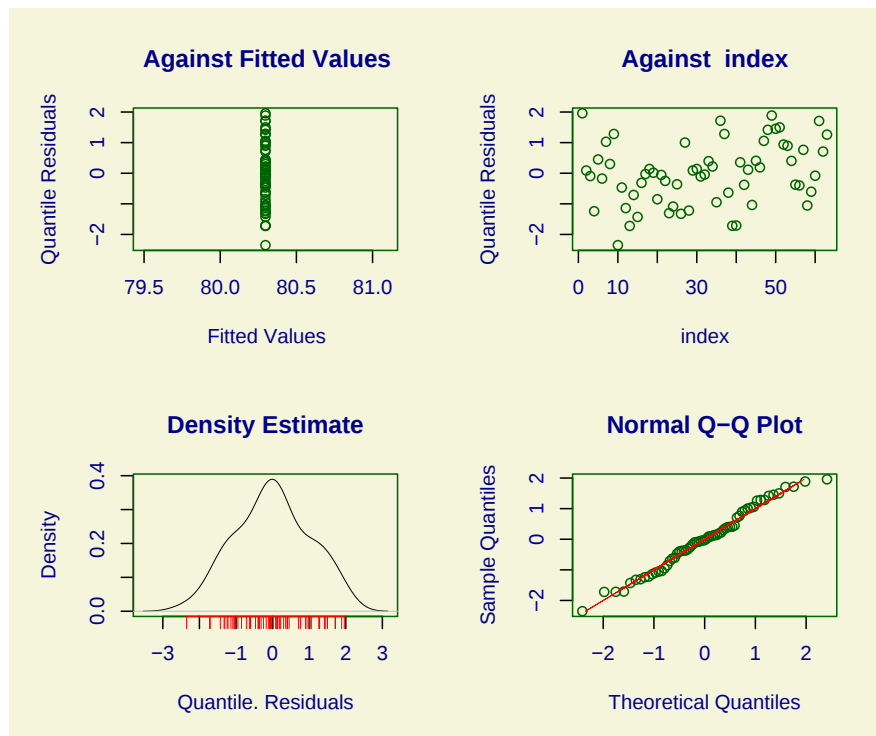


Figure 3.5: The residual plot from the fitted normal model to the snowfall data

The function `Q.stats()` calculates and prints Q statistics which are useful to test the normality of the (normalized quantile) residuals usually in a situation where an explanatory variable exists, Royston and Wright (2000).

Testing hypotheses from the fitted model

There are several methods to check the reliability of the fitted parameters of the distribution. Standard errors for the fitted parameters are provided by two functions: i) the `summary()` and ii) by the `vcov()` function. In general the two values should be identical, since by default `summary` is the standard errors obtained by `vcov`. The standard errors obtained by `vcov()` are the ones obtained by inverting the full observed information matrix and they do take into account the correlations between the distribution parameter estimates. Note that the `vcov()` function refits the final model one more time in order to obtain the Hessian matrix. Occasionally this could fail in which case `summary()` will use an alternative method called `qr`. This alternative method uses the QR decomposition of the individual distribution parameter estimation fits. The standard errors given by the `qr` method of `summary()` are not very reliable since they are the (conditional)

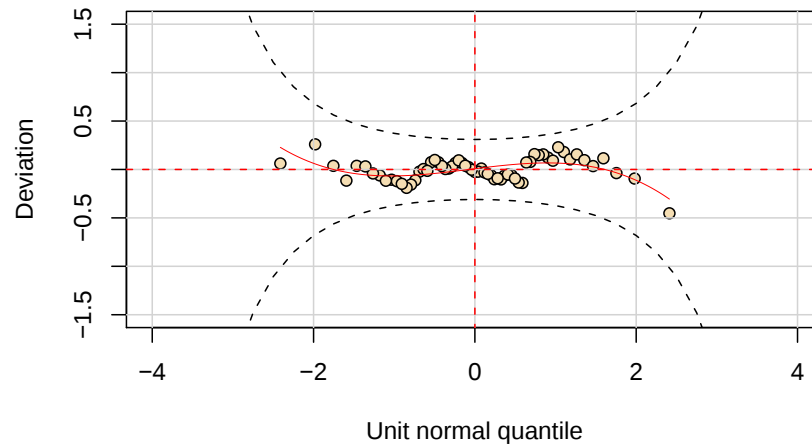


Figure 3.6: The worm plot from the fitted normal model to the snowfall data

standard errors obtained by assuming that the other distribution parameters are fixed at their maximum (penalized) likelihood estimates. We refit the chosen final model so we can use the `vcov()` function.

```
> modNO <- gamlss(snowfall ~ 1, data = parzen)

GAMLSS-RS iteration 1: Global Deviance = 576.7331
GAMLSS-RS iteration 2: Global Deviance = 576.7331

> summary(modNO)

*****
Family:  c("NO", "Normal")

Call:  gamlss(formula = snowfall ~ 1, data = parzen)

Fitting method: RS()

-----
Mu link function:  identity
Mu Coefficients:
  Estimate Std. Error   t value   Pr(>|t|)
 8.030e+01  2.965e+00   2.709e+01  1.052e-35

-----
Sigma link function:  log
Sigma Coefficients:
  Estimate Std. Error   t value   Pr(>|t|)
```

```
3.158e+00  8.922e-02  3.540e+01  2.245e-42
```

```
-----
No. of observations in the fit: 63
Degrees of Freedom for the fit: 2
      Residual Deg. of Freedom: 61
                        at cycle: 2
```

```
Global Deviance:      576.7331
      AIC:            580.7331
      SBC:            585.0194
```

```
*****
```

```
> vcov(modNO, type = "se")
```

```
(Intercept) (Intercept)
 2.9645994   0.0892178
```

The fitted model is given by $Y_i \sim \text{NO}(\hat{\mu}, \hat{\sigma})$ where $\hat{\mu} = \hat{\beta}_{01} = 80.3$ and $\log(\hat{\sigma}) = \hat{\beta}_{02} = 3.158$, so $\hat{\sigma} = 23.52$. Note that $\hat{\mu}$ and $\hat{\sigma}$ are maximum likelihood estimates of μ and σ .

The standard errors obtained are 2.965 for $\hat{\mu}$ and 0.08922 for $\log(\hat{\sigma}) = \hat{\beta}_{02}$ respectively, using either the `summary()` or `vcov()` functions. Note that since the normal fitting function `NO()` uses the identity link for μ and the log link for σ the standard errors given are those for $\hat{\mu}$ and for $\log(\hat{\sigma}) = \hat{\beta}_{02}$. For example, an approximate 95% confidence interval (CI) for $\log(\sigma) = \beta_{02}$, using the `vcov()` results, will be $[3.158 - (1.96 * 0.08922), 3.158 + (1.96 * 0.08922)] = (2.983, 3.333)$. Hence an approximate 95% CI confidence interval for σ is given by $[\exp(2.983), \exp(3.333)] = (19.75, 28.02)$. This can be compared with a profile deviance interval (19.96, 28.32) obtained using the `prof.dev(modNO, "sigma", min=19, max=28.5, step=.1, type="l")` function in Figure 3.8 or to a bootstrap CI given by $[\exp(3.021), \exp(3.33)] = (20.51, 27.93)$ obtained using the following R script.

Note that the function `boot` is given in Venables and Ripley (2000) page 173.

```
> library(boot)
> set.seed(1453)
> modNO <- gamlss(snowfall ~ 1, data = parzen, control = gamlss.control(trace = F))
> funB <- function(data, i) {
+   d <- data.frame(snowfall = data[i, ])
+   coef(update(modNO, data = d), "sigma")
+ }
> (modNO.boot <- boot(parzen, funB, R = 199))
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP
```

```
Call:
```

```
boot(data = parzen, statistic = funB, R = 199)
```

```
Bootstrap Statistics :
```

```
      original      bias    std. error
t1* 3.158309 -0.01715738  0.07906377
```

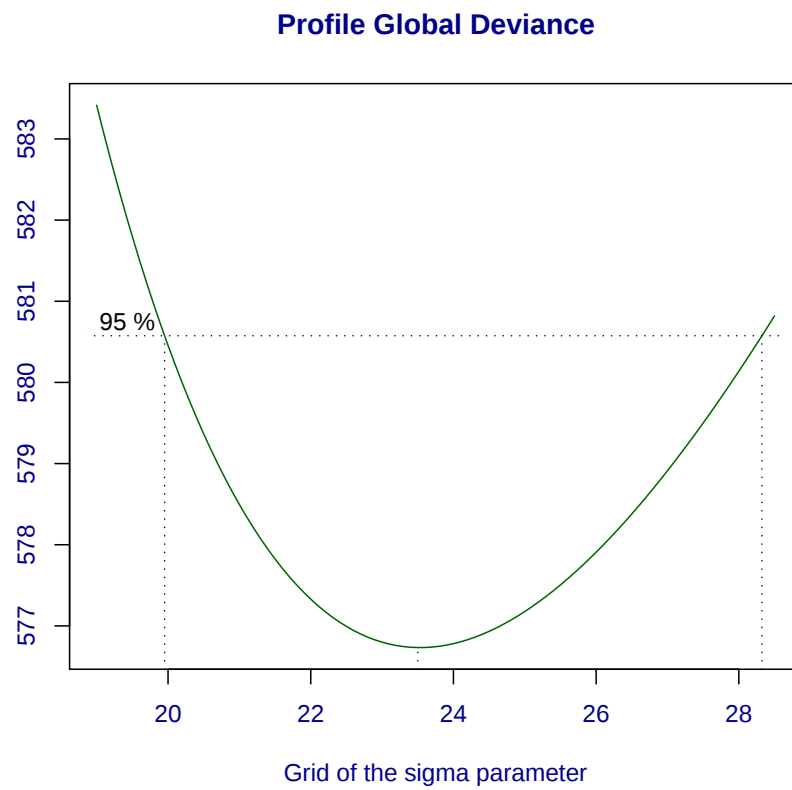


Figure 3.7: The profile deviance for the σ parameter for model modNO

```
> plot(modNO.boot)
> boot.ci(modNO.boot, type = c("norm", "basic"))
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 199 bootstrap replicates

CALL :

```
boot.ci(boot.out = modNO.boot, type = c("norm", "basic"))
```

Intervals :

Level	Normal	Basic
95%	(3.021, 3.330)	(3.011, 3.332)

Calculations and Intervals on Original Scale
Some basic intervals may be unstable

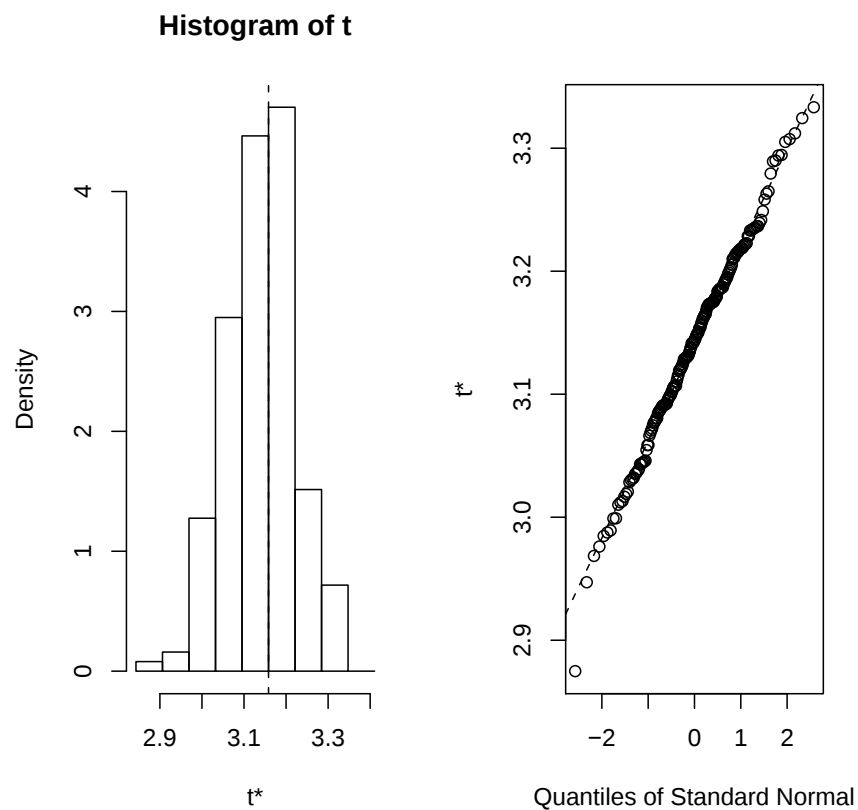


Figure 3.8: The bootstrap for $\log \sigma$ from model modNO

3.6.2 The strengths of glass fibres data

Data summary:

R data file: `glass` in package `gamlss.dist` of dimensions 63×1

source: Smith and Naylor (1987)

variables

strength : the strength of glass fibres (the unit of measurement is not given).

purpose: to demonstrate the fitting of a parametric distribution to the data.

conclusion a SEP4 distribution fits adequately

The following data show the strength of glass fibres, measured at the National Physical Laboratory, England, see Smith and Naylor (1987), (the unit of measurement was not given in the paper). Here we fit different distribution to the data and we select the "best" model using first the Akaike information criterion, then the Schwartz Bayesian criterion.

Note the use of the `gen.trun()` function in the package `gamlss.tr` designed to create a truncated distribution from an existing `gamlss.family` distribution. Here we generated a positive t distribution called `TFtr` and we fit it to the data together with the rest of the distributions. To obtain a positive t distribution we use the command `gen.trun()` with `par=0` to truncate it at zero with left truncation by default:

```
> data(glass)
> library(gamlss.dist)
> library(gamlss.tr)
> gen.trun(par = 0, family = TF)
```

A truncated family of distributions from TF has been generated and saved under the names:

```
dTFtr pTFtr qTFtr rTFtr TFtr
```

The type of truncation is left and the truncation parameter is 0

Next we fit the normal, truncated t , Box-Cox t (BCT), Box-Cox power exponential (BCPE), Box-Cox Cole and Green (BCCG), generalized gamma (GG), Skew exponential power type 1,2,3,4 (SEP1 to SEP4), skew t type 1,2,3,4,5 (ST1 to ST5) and Jonhson's SU (JSU) distributions to the data:

```
> mno <- gamlss(strength ~ 1, data = glass, trace = FALSE)
> mtftr <- gamlss(strength ~ 1, family = TFtr, data = glass, trace = FALSE)
> mbct <- gamlss(strength ~ 1, family = BCT, data = glass, trace = FALSE)
> mbcpe <- gamlss(strength ~ 1, family = BCPE, data = glass, trace = FALSE)
> mbccg <- gamlss(strength ~ 1, family = BCCG, data = glass, trace = FALSE)
> mGG <- gamlss(strength ~ 1, family = GG, data = glass, trace = FALSE)
> msep1 <- gamlss(strength ~ 1, family = SEP1, data = glass, trace = FALSE)
> msep2 <- gamlss(strength ~ 1, family = SEP2, data = glass, trace = FALSE)
> msep3 <- gamlss(strength ~ 1, family = SEP3, data = glass, trace = FALSE,
+   method = mixed(10, 40))
> msep4 <- gamlss(strength ~ 1, family = SEP4, data = glass, trace = FALSE,
```

```

+     method = mixed(20, 50))
> mst1 <- gamlss(strength ~ 1, family = ST1, data = glass, trace = FALSE)
> mst2 <- gamlss(strength ~ 1, family = ST2, data = glass, trace = FALSE,
+     method = mixed(10, 40))
> mst3 <- gamlss(strength ~ 1, family = ST3, data = glass, trace = FALSE,
+     method = mixed(10, 40))
> mst4 <- gamlss(strength ~ 1, family = ST4, data = glass, trace = FALSE)
> mst5 <- gamlss(strength ~ 1, family = ST5, data = glass, trace = FALSE)
> mjsu <- gamlss(strength ~ 1, family = JSU, data = glass, trace = FALSE)

```

Now we compare the distribution models using the Akaike information criterion, AIC, (given by the default penalty argument $k = 3$ in the generalized AIC function `GAIC()`).

```

> GAIC(mno, mtftr, mbct, mbcpe, mbccg, mGG, msep1, msep2, msep3,
+     msep4, mst1, mst2, mst3, mst4, mst5, mjsu)

```

	df	AIC
msep4	4	27.81106
msep3	4	28.05412
msep2	4	28.85255
msep1	4	28.87961
mjsu	4	30.41705
mbcpe	4	31.17762
mst2	4	31.40287
mbct	4	31.47897
mst3	4	31.58503
mst1	4	31.86362
mst5	4	31.86512
mbccg	3	33.07760
mst4	4	33.69735
mtftr	3	35.22183
mno	2	39.82364
mGG	3	50.60666

Next we compute the distribution models the Schwartz Bayesian criterion (given by penalty $k = \log(n)$ where $n = 63$ in the `GAIC()` function).

```

> GAIC(mno, mtftr, mbct, mbcpe, mbccg, mGG, msep1, msep2, msep3,
+     msep4, mst1, mst2, mst3, mst4, mst5, mjsu, k = log(63))

```

	df	AIC
msep4	4	36.38360
msep3	4	36.62665
msep2	4	37.42509
msep1	4	37.45215
mjsu	4	38.98959
mbccg	3	39.50701
mbcpe	4	39.75016
mst2	4	39.97541
mbct	4	40.05151

```

mst3  4 40.15757
mst1  4 40.43616
mst5  4 40.43766
mtftr 3 41.65123
mst4  4 42.26989
mno   2 44.10991
mGG   3 57.03607

```

The best model for the glass fibre strength according to both the Akaike and Schwartz Bayesian information criteria is the SEP4 distribution. Our truncated t distribution, TFtr, did not fit well to this particular data set. The fitted SEP4 distribution together with the data is shown in Figure 3.9 obtained using the command:

```

> histDist(glass$strength, SEP4, nbins = 13, main = "SEP4 distribution",
+   method = mixed(20, 50), trace = FALSE)

```

```

Family:  c("SEP4", "skew exponential power type 4")
Fitting method: mixed(20, 50)

```

```

Call:  gamlss(formula = y ~ 1, family = FA, method = ..1, trace = FALSE)

```

```

Mu Coefficients:
(Intercept)
      1.581
Sigma Coefficients:
(Intercept)
     -1.437
Nu Coefficients:
(Intercept)
     -0.1280
Tau Coefficients:
(Intercept)
      0.3183

```

```

Degrees of Freedom for the fit: 4 Residual Deg. of Freedom    59
Global Deviance:      19.8111
      AIC:      27.8111
      SBC:      36.3836

```

The fitted distribution has a spike at its mode. Distributions which involve the power exponential distribution, (eg. all the SEP's), with values of the kurtosis parameter(s) less or equal to 1 often have discontinuity in the gradient, leading to a spike at the mode. This often results in a multimodal likelihood function (with respect to μ , and leads to inferential problem. In the SEP4 distribution the parameters ν and τ adjust the kurtosis at the left and right side of the distribution respectively. The estimates of those two parameters are $\hat{\nu} = \exp(-0.1280) = 0.880$ and $\hat{\tau} = \exp(0.3183) = 1.375$ respectively, indicating possible problems with the inferential procedures since $\nu \leq 1$. [Note we can extract the fitted coefficients using either of the functions `coef()` and `fitted()`, e.g. `coef(msep5, "nu")` and `fitted(msep4, "nu")[1]`.]

```

> summary(msep4)

```

```
> histDist(glass$strength, SEP4, nbins = 13, main = "SEP4 distribution",
+          method = mixed(20, 50))
```

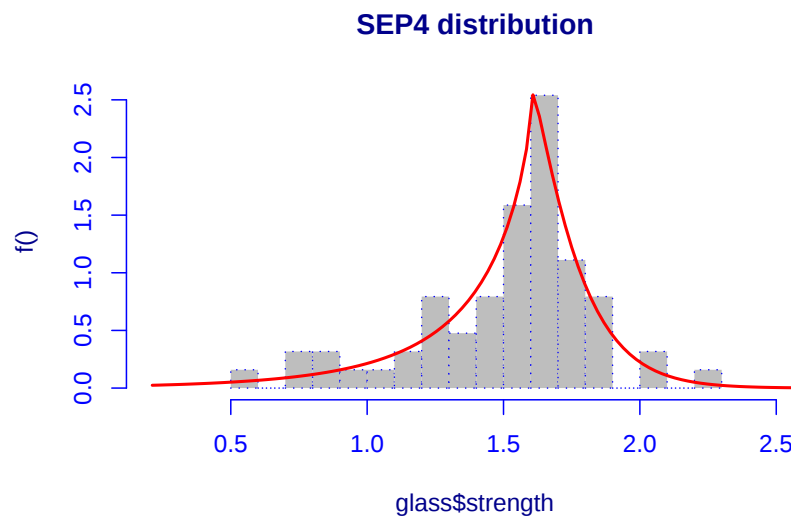


Figure 3.9: The strengths of glass fibres data and the fitted SEP4 distribution model

```
*****
Family:  c("SEP4", "skew exponential power type 4")

Call:
gamlss(formula = strength ~ 1, family = SEP4, data = glass, method = mixed(20,
  50), trace = FALSE)

Fitting method: mixed(20, 50)

-----
Mu link function:  identity
Mu Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.581     0.01818   86.97 1.656e-66

-----
Sigma link function:  log
Sigma Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.437     0.1371  -10.48 2.323e-15

-----
Nu link function:  log
Nu Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.1280	0.1078	-1.187	0.2396

Tau link function: log

Tau Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.3183	0.1422	2.239	0.02876

No. of observations in the fit: 63

Degrees of Freedom for the fit: 4

Residual Deg. of Freedom: 59

at cycle: 20

Global Deviance: 19.81106

AIC: 27.81106

SBC: 36.3836

Using the function `summary()` we are getting the warning “`vcov has failed, option qr is used instead`”. This is because the function `vcov` which is the default option for obtaining standard errors in `summary()` has failed, probably because of the peculiarity of the likelihood function at the point of the (possibly local) maximum, a consequence of a kurtotic parameters being less than one. The standard errors given, obtained from the individual fits by assuming that the rest of the parameters are fixed at their point of maximum, should be treated with caution.

A worm plot of the residuals of the fitted model in Figure 3.10 shows that all the points are close to the horizontal line indicating that the `SEP4` distribution provides an appropriate fit to the data.

3.6.3 The tensile strength data: response on $(0, 1)$

Data summary:

R data file: `tensile` in package `gamlss.dist` of dimensions 30×1

source: Hand *et al.* (1994)

variables

str : the strength of polyester fibres (the unit of measurement are not given).

purpose: to demonstrate the fitting of a parametric distribution to the data.

conclusion a truncated lognormal distribution fit best

These data come from Quesenberry and Hales (1980) and were also reproduced in Hand *et al.* (1994), data set 180, page 140. They contain measurements of tensile strength of polyester fibres and the authors were trying to check if they were consistent with the lognormal distribution. According to Hand *et al.* (1994) “these data follow from a preliminary transformation. If the

```
> wp(msep4)
```

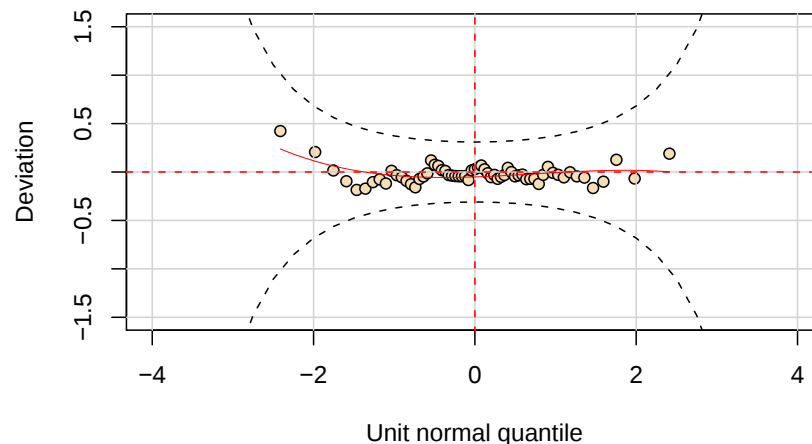


Figure 3.10: A worm plot of the residuals for the model using the SEP4 distribution fitted to the strengths of glass fibres data.

lognormal hypothesis is correct, these data should have been uniformly distributed". Here we use them as an example of data from a variable restricted to the range $(0, 1)$ and try to fit appropriate distributions. Note that apart from the beta (BE), a two parameter distribution, and the generalized beta type 1, (GB1), a four parameter distribution, there are no other distributions in the current version of GAMLSS software which are restricted to the range 0 to 1. So we create some using the `gen.trun()` function of the `gamlss.tr` package. The distributions we create are lognormal and Gamma distributions right truncated at one, and a t distribution truncated outside the range 0 to 1. First we fit the distributions and then we select the "best" using an Akaike information criterion.

```
> data(tensile)
> gen.trun(par = 1, family = "GA", type = "right")

A truncated family of distributions from GA has been generated
and saved under the names:
dGAttr pGAttr qGAttr rGAttr GAttr
The type of truncation is right and the truncation parameter is 1

> gen.trun(par = 1, "LOGNO", type = "right")

A truncated family of distributions from LOGNO has been generated
and saved under the names:
dLOGNOtr pLOGNOtr qLOGNOtr rLOGNOtr LOGNOtr
The type of truncation is right and the truncation parameter is 1

> gen.trun(par = c(0, 1), "TF", type = "both")
```

A truncated family of distributions from TF has been generated and saved under the names:

```
dTFtr pTFtr qTFtr rTFtr TFtr
```

The type of truncation is both and the truncation parameter is 0 1

```
> mbe <- gamlss(str ~ 1, data = tensile, family = BE, trace = FALSE,
+   n.cyc = 50)
> mgb1 <- gamlss(str ~ 1, data = tensile, family = GB1, trace = FALSE,
+   n.cyc = 50)
> mgatr <- gamlss(str ~ 1, data = tensile, family = GAtr, trace = FALSE,
+   n.cyc = 50)
> mlnotr <- gamlss(str ~ 1, data = tensile, family = LOGN0tr, trace = FALSE,
+   n.cyc = 50)
> mtftr <- gamlss(str ~ 1, data = tensile, family = TFtr, trace = FALSE,
+   n.cyc = 50)
> GAIC(mbe, mgb1, mgatr, mlnotr, mtftr)
```

	df	AIC
mlnotr	2	-3.6714336
mgatr	2	-2.9742870
mbe	2	-2.6101273
mtftr	3	-0.6404115
mgb1	4	0.8353512

The truncated lognormal distribution gives the best fit according to the Akaike information criterion although the beta and the truncate gamma fit almost as well. Figure 3.11 shows the fitted distributions. The plots in the figure were created using the `histDist()` function i.e. `histDist(tensile$str, "LOGN0tr" , nbins=10, xlim=c(0,1), main="(a) LOGN0tr")`.

3.7 Bibliography

3.8 Appendix of Chapter 3: Methods of generating distributions

Here we examine how many of the distributions in Tables 3.1, 3.2 and 3.3 for the random variable Y can be generated. Distribution families for Y can be generated by one (or more) of the following methods:

1. univariate transformation from a single random variable
2. transformation from two or more random variables
3. truncation distributions
4. a (continuous or finite) mixture of distributions
5. Azzalini type methods
6. splicing distributions

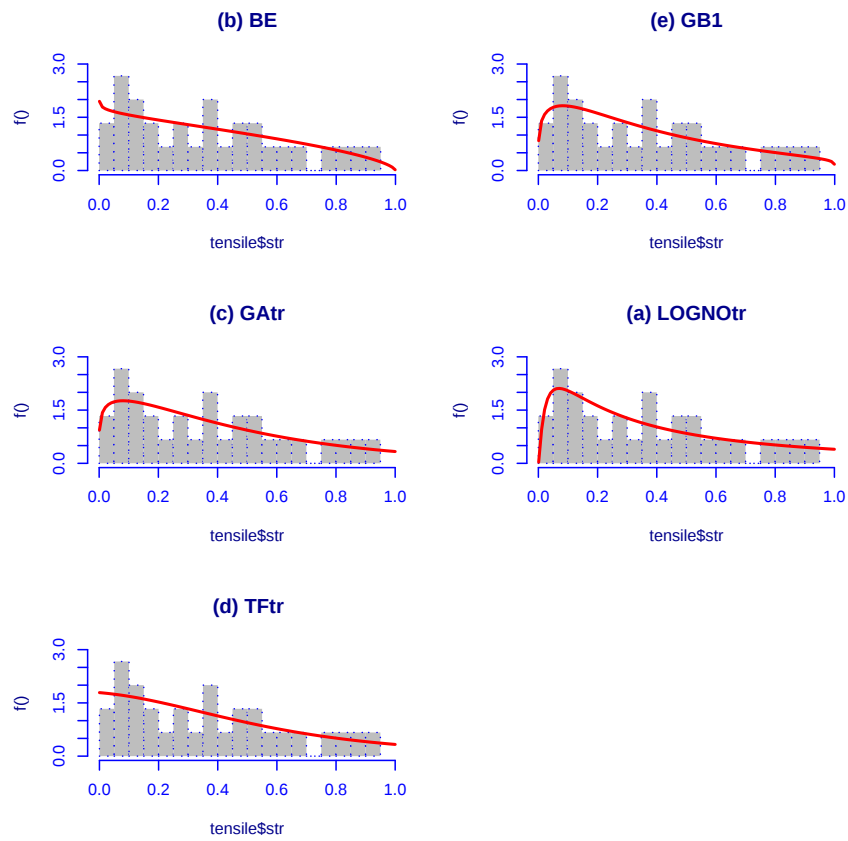


Figure 3.11: Tensile strength data with fitted (a) truncated lognormal , (b) beta, (c) truncated gamma and (d) truncated t (e) generalized beta type 2) distributions.

Distributions of Y	Random variable Z	Transformation to Z	References
BCCG	$\text{NO}(0, 1)$	(3.8)	Cole and Green (1992)
BCPE	$\text{PE}(0, 1, \tau)$	(3.8)	Rigby and Stasinopoulos (2004)
BCT	$\text{TF}(0, 1, \tau)$	(3.8)	Rigby and Stasinopoulos (2006)
EGB2	$\text{F}(2\nu, 2\tau)$	(3.9)	Johnson <i>et al.</i> (1995) p.142
GB1	$\text{BE}(\mu, \sigma)$	(3.10)	McDonald and Xu (1995)
GB2	$\text{F}(2\nu, 2\tau)$	$(\tau/\nu) (Y/\mu)^\sigma$	McDonald and Xu (1995)
GG	$\text{GA}(1, \sigma\nu)$	$(Y/\mu)^\nu$	Lopatatazidis and Green (2000)
JSU _o	$\text{NO}(0, 1)$	(A.15)	Johnson (1949)
JSU	$\text{NO}(0, 1)$	(A.15)	Rigby and Stasinopoulos (2006)
PE	$\text{GA}(1, \nu^{1/2})$	$\nu \left \frac{Y-\mu}{c\sigma} \right ^\nu$	Nelson (1991)
SHASH	$\text{NO}(0, 1)$	(3.13)	Jones (2005)
ST3	$\text{BEo}(\alpha, \beta)$	(3.14)	Jones and Faddy (2003)

Table 3.4: Showing distributions generated by univariate transformation

7. stopped sums

8. systems of distributions

There methods are discussed next in Sections 3.8.1 to 3.8.8 respectively.

3.8.1 Distributions generated by univariate transformation

Many three and four parameter families of continuous distribution for Y can be defined by assuming that a transformed variable Z , obtained from Y , has a simple well known distribution. The parameters of the distribution of Y may come from parameters of the univariate transformation or from parameters of the distribution of Z or both. Below we consider distributions available in GAMLSS which can be obtained by a univariate transformation.

Box-Cox, Cole and Green (BCCG)

The *Box-Cox Cole and Green* family for $Y > 0$ used by Cole and Green (1992), denoted by $\text{BCCG}(\mu, \sigma, \nu)$, assumes that Z has a standard normal distribution, $\text{NO}(0, 1)$, with mean 0 and standard deviation 1, where

$$Z = \begin{cases} \frac{1}{\sigma\nu} \left[\left(\frac{Y}{\mu} \right)^\nu - 1 \right], & \text{if } \nu \neq 0 \\ \frac{1}{\sigma} \log\left(\frac{Y}{\mu}\right), & \text{if } \nu = 0. \end{cases} \quad (3.8)$$

Cole and Green (1992) were the first to model all three parameters of a distribution as nonparametric smooth functions of a single explanatory variable. Note that the parameterization above is different from and more orthogonal than the one used originally by Box and Cox (1964). Rigby and Stasinopoulos (2000) and Stasinopoulos *et al.* (2000) used the original parameterization, $Z = (Y^\nu - 1)/\nu$ (if $\nu \neq 0$) + $\log(Y)$ (if $\nu = 0$) where $Z \sim \text{NO}(\mu, \sigma)$, to model the mean μ and the variance σ^2 of Z as functions of explanatory variables for a constant ν . They obtained the maximum likelihood estimate of the power parameter ν from its profile likelihood. This model for $Y > 0$ is denoted by $\text{LNO}\{\mu, \sigma, \nu\}$ where ν is fixed by the user in the GAMLSS software.

Box-Cox Power Exponential (BCPE)

The *Box-Cox power exponential* family for $Y > 0$, denoted by $\text{BCPE}(\mu, \sigma, \nu, \tau)$, is defined by assuming Z given by (3.8) has a (truncated) standard Power Exponential distribution, $\text{PE}(0, 1, \tau)$, see Rigby and Stasinopoulos (2004). This distribution is useful for modelling (positive or negative) skewness combined with (lepto or platy) kurtosis in continuous data.

Box-Cox t (BCT)

The *Box-Cox t* family for $Y > 0$, denoted by $\text{BCT}(\mu, \sigma, \nu, \tau)$, is defined by assuming Z given by (3.8) has a (truncated) standard t distribution with τ degrees of freedom, i.e. $\text{TF}(0, 1, \tau)$, see Rigby and Stasinopoulos (2006).

Exponential generalized beta type 2 (EGB2)

The *exponential generalized beta type 2* family for $-\infty < Y < \infty$, denoted by $\text{EGB2}(\mu, \sigma, \nu, \tau)$, assumes that $\exp(Y)$ has a generalized beta type 2 distribution. This distribution was called the exponential generalized beta of the second kind by McDonald (1991) and was investigated by McDonald and Xu (1995). The distribution may also be defined by assuming the Z has an F distribution with degrees of freedom 2ν and 2τ , i.e. $Z \sim F_{2\nu, 2\tau}$, where

$$Z = (\tau/\nu) \exp[(Y - \mu)/\sigma], \quad (3.9)$$

Johnson *et al.* (1995) p142. The distribution has also been called a generalized logistic distribution type IV, see Johnson *et al.* (1995) p 142, who report its long history from Perks (1932). Note also that $R = \exp[(Y - \mu)/\sigma]$ has a beta distribution of the second kind $\text{BE2}(\nu, \tau)$, Johnson *et al.* (1995) p248 and p325 and $B = R/(1 + R)$ has an original beta $\text{BEo}(\nu, \tau)$ distribution.

Generalized Beta type 1 (GB1)

The *generalized beta type 1* family for $0 < Y < 1$, denoted by $\text{GB1}(\mu, \sigma, \nu, \tau)$, is defined by assuming Z has a beta, $\text{BE}(\mu, \sigma)$, distribution where

$$Z = \frac{Y^\tau}{\nu + (1 - \nu) Y^\tau} \quad (3.10)$$

where $0 < \mu < 1$, $0 < \sigma < 1$, $\nu > 0$ and $\tau > 0$. Note that GB1 always has range $0 < y < 1$ and so is different from the generalized beta of the first kind, McDonald and Xu (1995), whose range depends on the parameters.

Note that for $0 < \nu \leq 1$ only, $\text{GB1}(\mu, \sigma, \nu, \tau)$ is a reparameterization of the submodel with range $0 < Y < 1$ of the five parameter generalized beta, $\text{GB}(a, b, c, p, q)$ distribution of McDonald and Xu (1995) given by

$$\text{GB1}(\mu, \sigma, \nu, \tau) \equiv \text{GB}\left(\tau, \nu^{1/\tau}, 1 - \nu, \mu(\sigma^{-2} - 1), (1 - \mu)(\sigma^{-2} - 1)\right).$$

Note also that $\tau = 1$ in $\text{GB1}(\mu, \sigma, \nu, \tau)$ gives a reparameterization of the *generalized 3 parameter beta* distribution, $\text{G3B}(\alpha_1, \alpha_2, \lambda)$, distribution, Pham-Gia and Duong (1989) and Johnson *et al.* (1995) p251, given by $\text{G3B}(\alpha_1, \alpha_2, \lambda) = \text{GB1}\left(\alpha_1/(\alpha_1 + \alpha_2), (\alpha_1 + \alpha_2 - 1)^{-1/2}, 1/\lambda, 1\right)$. Hence $\text{G3B}(\alpha_1, \alpha_2, \lambda)$ is a reparameterized submodel of $\text{GB1}(\mu, \sigma, \nu, \tau)$.

Generalized Beta type 2 (GB2)

The *generalized beta type 2* family for $Y > 0$, McDonald (1996), denoted by $\text{GB2}(\mu, \sigma, \nu, \tau)$, is defined by assuming Z has an F distribution with degrees of freedom 2ν and 2τ , i.e. $Z \sim F_{2\nu, 2\tau}$, where $Z = (\tau/\nu) (Y/\mu)^\sigma$.

The distribution is also called the generalized beta distribution of the second kind. Note also that $R = (Y/\mu)^\sigma$ has a beta distribution of the second kind, $\text{BE2}(\nu, \tau)$, Johnson *et al.* (1995) p248 and p325 and $B = R/(1 + R)$ has an original beta, $\text{BEo}(\nu, \tau)$, distribution.

Generalized Gamma (GG, GG2)

The *generalized gamma* family for $Y > 0$, parameterized by Lopatzidis and Green (2000), denoted by $\text{GG}(\mu, \sigma, \nu)$, assumes that Z has a gamma $\text{GA}(1, \sigma\nu)$ distribution with mean 1 and variance $\sigma^2\nu^2$, where $Z = (Y/\mu)^\nu$. A reparametrization of $\text{GG}(\mu, \sigma, \nu)$, Johnson *et al.* (1995) p401, given by setting $\mu = \alpha_2\alpha_3^{1/\alpha_1}$, $\sigma = (\alpha_1^2\alpha_3)^{-1/2}$ and $\nu = \alpha_1$, is denoted $\text{GG2}(\alpha_1, \alpha_2, \alpha_3)$.

Johnson Su (JSUo, JSU)

The original *Johnson Su* family for $-\infty < Y < \infty$, denoted by $\text{JSUo}(\mu, \sigma, \nu, \tau)$, Johnson (1949), is defined by assuming

$$Z = \nu + \tau \sinh^{-1}[(Y - \mu)/\sigma] \quad (3.11)$$

has a standard normal distribution. The *reparameterized Johnson Su* family, for $-\infty < Y < \infty$, denoted by $\text{JSU}(\mu, \sigma, \nu, \tau)$, has exact mean μ and standard deviation σ for all values of ν and τ , see Appendix A.3.3 for details.

Power Exponential (PE, PE2)

The *power exponential* family for $-\infty < Y < \infty$, denoted by $\text{PE}(\mu, \sigma, \nu)$ is defined by

$$f_Y(y) = \frac{\nu}{2c\sigma\Gamma(1/\nu)} \exp \left\{ - \left| \frac{y - \mu}{c\sigma} \right|^\nu \right\} \quad (3.12)$$

where $c = [\Gamma(1/\nu)/\Gamma(3/\nu)]^{1/2}$, and $-\infty < \mu < \infty$, $\sigma > 0$ and $\nu > 0$.

This parameterization, used by Nelson (1991), ensures that μ and σ are the mean and standard deviation of Y respectively for all $\nu > 0$. This distribution assumes that $Z = \nu \left| \frac{Y - \mu}{c\sigma} \right|^\nu$ has a gamma $\text{GA}(1, \nu^{1/2})$ distribution. A reparametrization of $\text{PE}(\mu, \sigma, \nu)$ used by Nandi and Mämpel (1995), denoted by $\text{PE2}(\alpha_1, \alpha_2, \alpha_3)$, is given by setting $\mu = \alpha_1$, $\sigma = \alpha_2/c$ and $\nu = \alpha_3$.

The Subbotin distribution, Subbotin (1923) and Johnson *et al.* (1995), p195, which uses as parameters (θ, ϕ, δ) is also a reparametrization of $\text{PE2}(\alpha_1, \alpha_2, \alpha_3)$ given by setting $\alpha_1 = \theta$, $\alpha_2 = \phi 2^{\delta/2}$ and $\alpha_3 = 2/\delta$. Box and Tiao (1973) p 157 equations (3.2.3) and (2.2.5) are respectively reparameterizations of the Subbotin parameterization and (3.12) in which $\delta = 1 + \beta$ and $\nu = 2/(1 + \beta)$. The distribution is also called the exponential power distribution or the Box-Tiao distribution.

Sinh-Arcsinh (SHASH)

The *sinh-arcsinh* family for $-\infty < Y < \infty$, Jones (2005), denoted by $\text{SHASH}(\mu, \sigma, \nu, \tau)$, is defined by assuming that Z has a standard normal distribution $\text{NO}(0, 1)$, where

$$Z = \frac{1}{2} \{ \exp [\tau \sinh^{-1}(R)] - \exp [-\nu \sinh^{-1}(R)] \} \quad (3.13)$$

where $R = (Y - \mu)/\sigma$.

Skew t type 5 (ST5)

The *skew t type 5* family for $-\infty < Y < \infty$, Jones and Faddy (2003), denoted by $\text{ST5}(\mu, \sigma, \nu, \tau)$, assumes that Z has a beta $\text{BEo}(\alpha, \beta)$ distribution with $f_Z(z) = z^{\alpha-1} (1-z)^{\beta-1} / B(\alpha, \beta)$ where

$$Z = \frac{1}{2} \left[1 + R/(\alpha + \beta + R^2)^{1/2} \right] \quad (3.14)$$

where $R = (Y - \mu)/\sigma$, $\alpha = \tau^{-1} [1 + \nu(2\tau + \nu^2)^{-1/2}]$ and $\beta = \tau^{-1} [1 - \nu(2\tau + \nu^2)^{-1/2}]$.

3.8.2 Distributions generated by transformation from two or more random variables

Distributions can be generated from a function of two (or more) random variables.

Student t family (TF)

The *Student t* family for $-\infty < Y < \infty$ (e.g. Lange *et al.*, 1989), denoted by $\text{TF}(\mu, \sigma, \nu)$, is defined by assuming that $Y = \mu + \sigma T$ where $T \sim t_\nu$ has a standard t distribution with ν degrees of freedom, defined itself by $T = Z(W/\nu)^{-1/2}$ where $Z \sim \text{NO}(0, 1)$ and $W \sim \chi_\nu^2 \equiv \text{GA}(\nu, [2/\nu]^{1/2})$, a Chi-square distribution with ν degrees of freedom treated as a continuous parameter, and where Z and W are independent random variables.

Skew t type 2 (ST2)

The *skew t type 2* family for $-\infty < Y < \infty$, Azzalini and Capitanio (2003), denoted $\text{ST2}(\mu, \sigma, \nu, \tau)$, is defined by assuming that $Y = \mu + \sigma T$ where $T = Z(W/\tau)^{-1/2}$ and $Z \sim \text{SN}(0, 1, \nu)$ has a skew normal type 1 distribution (see Section 3.8.5) and $W \sim \chi_\tau^2 \equiv \text{GA}(\tau, [2/\tau]^{1/2})$ has a Chi-square distribution with $\tau > 0$ degrees of freedom, and where Z and W are independent random variables. Note that $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$ and $\tau > 0$.

The distribution $\text{ST2}(\mu, \sigma, \nu, \tau)$ is the one dimensional special case of the multivariate skew t used in R package **Sn**, Azzalini (2006).

An important special case of a function of two independent random variables is their sum, i.e. $Y = Z_1 + Z_2$. The probability density function of Y is obtained by convolution, i.e.

$$f_Y(y) = \int_{-\infty}^y f_{Z_1}(z) f_{Z_2}(y-z) dz. \quad (3.15)$$

The following are two examples.

Exponential Gaussian (exGAUS)

If $Z_1 \sim \text{NO}(\mu, \sigma)$ and $Z_2 \sim \text{EXP}(\nu)$ in (3.15), then Y has an exponential Gaussian distribution, denoted by $\text{exGAUS}(\mu, \sigma, \nu)$, for $-\infty < Y < \infty$. The distribution has been also called a lagged normal distribution, Johnson *et al.* (1994), p 172.

Generalized Erlangian

As pointed out by Johnson *et al.* (1994), p 172, the convolution of two or more exponential probability density functions with different mean parameters gives the generalized Erlangian distribution, while the convolution of a normal, $\text{NO}(\mu, \sigma)$, with a generalized Erlangian probability density function gives a generalized lagged normal distribution, see Davis and Kutner (1976).

3.8.3 Distributions generated by truncation

3.8.4 Distributions generated by a mixture of distributions

A distribution for Y can be generated by assuming that a parameter γ of a distribution for Y itself comes from a distribution.

Assume that, given γ , Y has conditional probability (density) function $f(y|\gamma)$ and marginally γ has probability (density) function $f(\gamma)$. Then the marginal of Y is given by

$$f_Y(y) = \begin{cases} \int f(y|\gamma)f(\gamma)d\gamma, & \text{if } \gamma \text{ is continuous,} \\ \sum f(y|\gamma)p(\gamma = \gamma_i), & \text{if } \gamma \text{ is discrete.} \end{cases} \quad (3.16)$$

The marginal distribution of Y is called a continuous mixture distribution if γ is continuous and a discrete (or finite) mixture distribution if γ is discrete.

Discrete (or finite) mixture distributions are considered in detail in Chapter 7. Continuous mixture density functions may be explicitly defined if the integral in (3.16) is tractable. This is dealt with in this section. However the integral in (3.16) is often intractable (and so the density functions is not explicitly defined), but may be approximated, e.g. using Gaussian quadrature points. This is dealt with in Section ??, where the model is viewed as a random effect model at the observational level.

Explicitly defined continuous mixture distributions

The marginal distribution of Y will, in general, be continuous if the conditional distribution of Y is continuous.

Student t family (TF)

The *Student t* family for $-\infty < Y < \infty$, denoted $\text{TF}(\mu, \sigma, \nu)$, may be generated from a continuous mixture by assuming $Y|\gamma \sim \text{NO}(\mu, \gamma)$ and $\gamma \sim \sqrt{\nu}\sigma\chi_\nu^{-1} \equiv \text{GG}(\sigma, [2\nu]^{-1/2}, -2)$ has a scale inverted Chi distribution (which is a special case of the generalized gamma distribution), Box and Tiao (1973).

Distributions of Y	Distribution of $Y \gamma$	Distribution of γ	References
TF(μ, σ, ν)	NO(μ, γ)	GG($\sigma, [2\nu]^{-1/2}, -2$)	Box and Tiao (1973)
GT (μ, σ, ν, τ)	PE2(μ, γ, τ)	GG2($-\tau, \sigma\nu^{1/\tau}, \nu$)	McDonald(1991)
GB2(μ, σ, ν, τ)	GG2(σ, γ, ν)	GG2($-\sigma, \mu, \tau$)	McDonald (1996)
EGB2(μ, σ, ν, τ)	EGG2($1/\sigma, \gamma, \nu$)	GG2($-1/\sigma, e^\mu, \tau$)	McDonald (1996)

Table 3.5: Showing distributions generated by continuous mixtures

Generalized t (GT)

The *generalized t* family for $-\infty < Y < \infty$, denoted GT(μ, σ, ν, τ), may be generated by assuming $Y|\gamma \sim \text{PE2}(\mu, \gamma, \tau)$ has a power exponential type 2 distribution and $\gamma \sim \text{GG2}(-\tau, \sigma\nu^{1/\tau}, \nu)$ has a generalized gamma type 2 distribution, McDonald (1991).

Generalized Beta type 2 (GB2)

The *generalized beta type 2* family for $Y > 0$, denoted GB2(μ, σ, ν, τ), may be generated by assuming $Y|\gamma \sim \text{GG2}(\sigma, \gamma, \nu)$ and $\gamma \sim \text{GG2}(-\sigma, \mu, \tau)$, McDonald (1996).

Exponential Generalized Beta type 2 (EGB2)

The *exponential generalized beta type 2* family for $-\infty < Y < \infty$, denoted EGB2(μ, σ, ν, τ) may be generated by assuming $Y|\gamma \sim \text{EGG2}(1/\sigma, \gamma, \nu)$ has an exponential generalized gamma type 2 distribution and $\gamma \sim \text{GG2}(-1/\sigma, e^\mu, \tau)$, McDonald (1996). [Note that the exponential generalized gamma type 2 distribution is defined by: if $Z \sim \text{EGG2}(\mu, \sigma, \nu)$ then $e^Z \sim \text{GG2}(\mu, \sigma, \nu)$.]

3.8.5 Distributions generated by Azzalini's method

Lemma 1 of Azzalini (1985) proposed the following method of introducing skewness into a symmetric probability density function. Let $f_{Z_1}(z)$ be a probability density function symmetric about z equals zero and let $F_{Z_2}(z)$ be an absolutely continuous cumulative distribution function such that $dF_{Z_2}(z)/dz$ is symmetric about zero. Then, for any real ν , $f_Z(z)$ is a proper probability density function where

$$f_Z(z) = 2f_{Z_1}(z)F_{Z_2}(\nu z). \quad (3.17)$$

Let $Y = \mu + \sigma Z$ then

$$f_Y(y) = \frac{2}{\sigma} f_{Z_1}(z) F_{Z_2}(\nu z) \quad (3.18)$$

where $z = (y - \mu)/\sigma$.

Skew Normal type 1 (SN1)

The *skew normal type 1* family for $-\infty < Y < \infty$, Azzalini (1985), denoted by SN1(μ, σ, ν), is defined by assuming Z_1 and Z_2 have standard normal, NO(0, 1), distributions in (3.18).

Distributions of Y	Distribution of Z_1	Distribution of Z_2	$w(z)$
SN1(μ, σ, ν)	NO(0, 1)	NO(0, 1)	νz
SEP1(μ, σ, ν, τ)	PE2($0, \tau^{1/\tau}, \tau$)	PE2($0, \tau^{1/\tau}, \tau$)	νz
SEP2(μ, σ, ν, τ)	PE2($0, \tau^{1/\tau}, \tau$)	NO(0, 1)	$\nu (2/\tau)^{1/2} \text{sign}(z) z ^{\tau/2}$
ST1(μ, σ, ν, τ)	TF(0, 1, τ)	TF(0, 1, τ)	νz
ST2(μ, σ, ν, τ)	TF(0, 1, τ)	TF(0, 1, $\tau + 1$)	$\nu \lambda^{1/2} z$

Table 3.6: Showing distributions generated by Azzalini type methods using equation (3.20)

Skew exponential power type 1 (SEP1)

The *skew exponential power type 1* family for $-\infty < Y < \infty$, Azzalini (1986), denoted by SEP1(μ, σ, ν, τ), is defined by assuming Z_1 and Z_2 have power exponential type 2, PE2($0, \tau^{1/\tau}, \tau$), distributions in (3.18). Azzalini (1986) called this distribution type I. The skew normal type 1, SN1(μ, σ, ν), is a special case of SEP1(μ, σ, ν, τ) obtained by setting $\tau = 2$.

Skew t type 1 (ST1)

The *skew t type 1* family for $-\infty < Y < \infty$, Azzalini (1986), denoted by ST1(μ, σ, ν, τ), is defined by assuming Z_1 and Z_2 have Student t distributions with $\tau > 0$ degrees of freedom, i.e., TF(0, 1, τ), in (3.18).

Equation (3.17) was generalized, in Azzalini and Capitanio (2003) Proposition 1, to

$$f_Z(z) = 2f_{Z_1}(z)F_{Z_2}[w(z)] \quad (3.19)$$

where $w(z)$ is any odd function of z . Hence

$$f_Y(y) = \frac{2}{\sigma} f_{Z_1}(z) F_{Z_2}[w(z)] \quad (3.20)$$

where $z = (y - \mu)/\sigma$. This allows a wider generation of family of distributions.

Skew exponential power type 2 (SEP2)

The *skew exponential power type 2* family, denoted by SEP2(μ, σ, ν, τ), Azzalini (1986) and DiCiccio and Monti (2004) is expressed in the form (3.20) by letting $Z_1 \sim \text{PE2}(0, \tau^{1/\tau}, \tau)$, $Z_2 \sim \text{NO}(0, 1)$ and $w(z) = \nu(2/\tau)^{1/2} \text{sign}(z)|z|^{\tau/2}$. Azzalini (1986) developed a reparametrization of this distribution given by setting $\nu = \text{sign}(\lambda)|\lambda|^{\tau/2}$ and called it type II. The skew normal type 1, SN1(μ, σ, ν), distribution is a special case of SEP2(μ, σ, ν, τ) obtained by setting $\tau = 2$.

Skew t type 2 (ST2)

The *skew t type 2* family, denoted by ST2(μ, σ, ν, τ) and discussed in Section 3.8.2, is expressed in the form (3.20) by letting $Z_1 \sim \text{TF}(0, 1, \tau)$, $Z_2 \sim \text{TF}(0, 1, \tau + 1)$ and $w(z) = \nu \lambda^{1/2} z$ where $\lambda = (\tau + 1) / (\tau + z^2)$, Azzalini and Capitanio (2003).

3.8.6 Distributions generated by splicing**Splicing using two components**

Splicing has been used to introduce skewness into symmetric distribution family. Let Y_1 and Y_2 have probability density functions that are symmetric about μ . A spliced distribution for Y

may be defined by

$$f_Y(y) = \pi_1 f_{Y_1}(y) I(y < \mu) + \pi_2 f_{Y_2}(y) I(y \geq \mu). \quad (3.21)$$

where $I(\cdot)$ is an indicator variable taking value 1 if the condition is true and 0 otherwise. Ensuring that $f_Y(y)$ is a proper probability density function requires $(\pi_1 + \pi_2)/2 = 1$. Ensuring continuity at $y = \mu$ requires $\pi_1 f_{Y_1}(\mu) = \pi_2 f_{Y_2}(\mu)$. Hence $\pi_1 = 2/(1+k)$ and $\pi_2 = 2k/(1+k)$ where $k = f_{Y_1}(\mu)/f_{Y_2}(\mu)$ and

$$f_Y(y) = \frac{2}{(1+k)} \{f_{Y_1}(y) I(y < \mu) + k f_{Y_2}(y) I(y \geq \mu)\}. \quad (3.22)$$

Splicing using two components with different scale parameters

A “scale-spliced” distribution for Y may be defined by assuming that probability density function $f_Z(z)$ is symmetric about 0 and that $Y_1 = \mu + \sigma Z/\nu$ and $Y_2 = \mu + \sigma \nu Z$ in (3.22). Hence

$$f_Y(y) = \frac{2}{(1+k)} \left\{ \frac{\nu}{\sigma} f_Z(\nu z) I(y < \mu) + \frac{k}{\nu \sigma} f_Z(z/\nu) I(y \geq \mu) \right\}. \quad (3.23)$$

for $z = (y - \mu)/\sigma$ and where $k = f_{Y_1}(\mu)/f_{Y_2}(\mu) = \nu^2$. Hence

$$f_Y(y) = \frac{2\nu}{\sigma(1+\nu^2)} \{f_Z(\nu z) I(y < \mu) + f_Z(z/\nu) I(y \geq \mu)\}. \quad (3.24)$$

The formulation (3.24) was used by Fernandez, Osiewalski and Steel (1995) and Fernandez and Steel (1998).

Skew normal (SN2)

A *skew normal type 2* distribution (or two-piece normal distribution) for $-\infty < Y < \infty$, denoted by $\text{SN2}(\mu, \sigma, \nu)$, is defined by assuming $Z \sim \text{NO}(0, 1)$ in (3.24) or equivalently $Y_1 \sim \text{NO}(\mu, \sigma/\nu)$ and $Y_2 \sim \text{NO}(\mu, \sigma\nu)$ in (3.22), giving

$$f_Y(y) = \frac{2\nu}{\sqrt{2\pi}\sigma(1+\nu^2)} \left\{ \exp\left[-\frac{1}{2}(\nu z)^2\right] I(y < \mu) + \exp\left[-\frac{1}{2}\left(\frac{z}{\nu}\right)^2\right] I(y \geq \mu) \right\} \quad (3.25)$$

where $z = (y - \mu)/\sigma$. References to this distribution are given in Johnson *et al.* (1994) p 173 and Jones and Faddy (2003). The earliest reference appears to be Gibbons and Mylroie (1973).

Skew exponential power type 3 (SEP3)

A *skew exponential power type 3* distribution for $-\infty < Y < \infty$, Fernandez, Osiewalski and Steel (1995), denoted by $\text{SEP3}(\mu, \sigma, \nu, \tau)$, is defined by assuming $Z \sim \text{PE2}(0, 2^{1/\tau}, \tau)$ in (3.24) or equivalently, $Y_1 \sim \text{PE2}(\mu, \sigma 2^{1/\tau}/\nu, \tau)$ and $Y_2 \sim \text{PE2}(\mu, \sigma \nu 2^{1/\tau}, \tau)$ in (3.22). Note that the skew normal type 2 distribution, $\text{SN2}(\mu, \sigma, \nu)$, is a special case of $\text{SEP3}(\mu, \sigma, \nu, \tau)$ given by setting $\tau = 2$.

Skew t type 3 (ST3)

A *skew type 3* distribution for $-\infty < Y < \infty$, Fernandez and Steel (1998), denoted by $\text{ST3}(\mu, \sigma, \nu, \tau)$ is defined by assuming $Z \sim \text{TF}(0, 1, \tau) \equiv t_\tau$ in (3.24), or equivalently $Y_1 \sim \text{TF}(\mu, \sigma/\nu, \tau)$ and $Y_2 \sim \text{TF}(\mu, \sigma \nu, \tau)$ in (3.22). A reparametrization of ST3, in which μ and σ are the mean and the standard deviation of Y is given by Hansen (1994). Theodossiou (1998) extended the Hansen reparametrization to a five parameter skew generalized t distribution.

Distributions of Y	Distribution of Y_1	Distribution of Y_2	References
SN2(μ, σ, ν)	NO($\mu, \sigma/\nu$)	NO($\mu, \sigma\nu$)	Gibbons and Mylroie (1973)
SEP3(μ, σ, ν, τ)	PE2($\mu, \sigma 2^{1/\tau}/\nu, \tau$)	PE2($\mu, \sigma\nu 2^{1/\tau}, \tau$)	Fernandez, Osiewolski and Steel (1995)
SEP4(μ, σ, ν, τ)	PE2(μ, σ, ν)	PE2(μ, σ, τ)	Jones (2005)
ST3(μ, σ, ν, τ)	TF($\mu, \sigma/\nu, \tau$)	TF($\mu, \sigma\nu, \tau$)	Fernandez and Steel (1998)
ST4(μ, σ, ν, τ)	TF(μ, σ, ν)	TF(μ, σ, τ)	

Table 3.7: Showing distributions generated by splicing

Splicing using two components with different shape parameters

A “shape-spliced” distribution for Y may be defined by assuming Y_1 and Y_2 in (3.22) have different shape parameters.

Skew exponential power type 4 (SEP4)

A *skew exponential power type 4* family for $-\infty < Y < \infty$, Jones (2005), denoted by SEP4(μ, σ, ν, τ), is defined by assuming $Y_1 \sim \text{PE2}(\mu, \sigma, \nu)$ and $Y_2 \sim \text{PE2}(\mu, \sigma, \tau)$ in (3.22). Note that μ is the mode of Y .

A similar distribution was used by Nandi and Mämpel (1995) who set $Y_1 \sim \text{PE2}(\mu, \sigma, \nu)$ and $Y_2 \sim \text{PE2}(\mu, \sigma/q, \tau)$ in (3.22), where $q = \Gamma[1 + (1/\tau)] / \Gamma[1 + (1/\nu)]$. However this distribution constrains *both* the median and mode of Y to be μ , which is perhaps rather restrictive.

Skew t type 4 (ST4)

A *skew t type 4* family for $-\infty < Y < \infty$, denoted by ST4(μ, σ, ν, τ), is defined by assuming $Y_1 \sim \text{TF}(\mu, \sigma, \nu)$ and $Y_2 \sim \text{TF}(\mu, \sigma, \tau)$ in (3.22).

Splicing using three components

Splicing has also been used to introduce robustness into the normal distribution, as in the NET distribution below.

Normal-exponential- t (NET)

The *normal-exponential- t* family for $-\infty < Y < \infty$, denoted by NET(μ, σ, ν, τ), Rigby and Stasinopoulos (1994), is defined by $Y = \mu + \sigma Z$, where Z has a standard normal density function for $|Z| < \nu$, an exponential density function for $\nu \leq |Z| < \tau$, and a Student t density function for $|z| \geq \tau$, given by

$$f_Z(z) = \pi_1 f_{Z_1}(z) I(|z| < \nu) + \pi_2 f_{Z_2}(z) I(\nu \leq |z| < \tau) + \pi_3 f_{Z_3}(z) I(|z| \geq \tau) \quad (3.26)$$

where $Z_1 \sim \text{NO}(0, 1)$, $Z_2 \sim \text{EXP}(\nu)$, $Z_3 \sim \text{TF}(0, 1, \nu\tau - 1) \equiv t_{\nu\tau-1}$ and π_1, π_2 and π_3 are defined to ensure $f_Z(z)$ is a proper density function and to ensure continuity of $f_Z(z)$ at ν and τ . In `gamlss()` parameters ν and τ are constants which are either chosen by the user or estimated using the function `prof.dev()`. For fixed ν and τ , the NET distribution has bounded influence functions for both μ and σ , Rigby and Stasinopoulos (1994), and hence provides a robust method of estimating μ and σ for contaminated normal data.

3.8.7 Distributions generated by stopped sums

3.8.8 Systems of distributions

Pearson system

The Pearson system of probability density functions $f_Y(y|\boldsymbol{\theta})$, where $\boldsymbol{\theta}^\top = (\theta_1, \theta_2, \theta_3, \theta_4)$, is defined by solutions of the equation:

$$\frac{d}{dy}f_Y(y|\boldsymbol{\theta}) = -\frac{\theta_1 + y}{\theta_2 + \theta_3 y + \theta_4 y^2} \quad (3.27)$$

The solutions of (3.27) fall into one of seven families of distributions called Type I to Type VII. Type I, IV, and VI cover disjoint regions of the skewness-kurtosis $(\sqrt{\beta_1}, \beta_2)$ space, while the other four types are boundary types, see Johnson *et al.* (1994), Figure 12.2. Type I is a shifted and scaled beta $\text{BE}(\mu, \sigma)$ distribution, with the resulting arbitrary range defined by two extra parameters. Type II is a symmetrical form of type I. Type III is a shifted gamma distribution. Types IV and V are not well known distribution (probably because the constants of integration are intractable). Type VI is a generalization of the F distribution. Type VII is a scaled t distribution, i.e. $\text{TF}(0, \sigma, \nu)$.

Stable distribution system

Stable distributions are defined through their characteristic function, given by Johnson *et al.* (1994) p57. In general their probability density function cannot be obtained explicitly (except using complicated infinite summations). McDonald (1996) and Lambert and Lindsey (1999) discuss the application of stable distributions to modelling stock returns.

Exponential Family

The *Exponential Family* for Y with mean μ and variance $\phi\mu^\nu$ (where $\phi = \sigma^2$ and σ is a scale parameter), McCullagh and Nelder (1989), does not transform to a simple well known distribution. This is also called the Tweedie family.

The probability (density) function exists only for $\nu \leq 0$ or $\nu > 1$ and suffers from being intractable (except using complicated series approximations) except for specific values $\nu = 0, 2, 3$. Furthermore, in general for $1 < \nu < 2$, the distribution is a combination of a mass probability at $Y = 0$ together with a continuous distribution for $Y > 0$, (which cannot be modelled independently), which is inappropriate for a continuous dependent variable Y , see Gilchrist (2000). This distribution is not currently available in GAMLSS.

Generalized inverse Gaussian family

This family was developed by Jorgensen (1982).

3.9 Exercises for Chapter 3

- Q1 The Turkish stock exchange index, I_i was recorded daily from 1/1/1988 to 31/12/1998. The daily returns, $ret = \log_e(I_{i+1}/I_i)$, were obtained for $i = 1, 2, \dots, 2868$.

- (a) Input the data into data.frame `tse` from file `TSE.r`, e.g. by

```
tse<-read.table("D:/gamlss/data/utrecht/TSE.r",header=TRUE)
```

and plot the data sequentially using

```
with(tse, plot(ret,type="l"))
```

- (b) Fit each of the following distributions for `ret` using the command `histDist()` (and using different model names for later comparison):

- two parameter: normal $NO(\mu, \sigma)$, i.e. `mN0<-histDist(tse$ret,"N0")`
- three parameter: t family $TF(\mu, \sigma, \nu)$ and power exponential $PE(\mu, \sigma, \nu)$
- four parameter: Johnson Su $JSU(\mu, \sigma, \nu, \tau)$, skew exponential power type 1 to 4 i.e. $SEP1(\mu, \sigma, \nu, \tau)$, skew t type 1 to 5 i.e. $ST1(\mu, \sigma, \nu, \tau)$ and sinh-arc-sinh $SHASH(\mu, \sigma, \nu, \tau)$.

[Note that to ensure convergence you may need to increase the number of GAMLSS iterations using for example `n.cyc=100` or switch the algorithm from `RS()` to `CG()` after few iterations using the argument `method` i.e.

```
method=mixed(10,100),
```

for both `gamlss()` and `histDist()` functions.

Also if you are using `histDist()` increase the default value of the argument `nbins` to 30 i.e.

```
histDist(tse$ret, family=SEP4, nbins=30,n.cyc=100).]
```

- (c) Use the AIC command with each of the penalties $k = 2, 3.8$ and $7.96 = \log(2868)$, [corresponding to criteria AIC, $\chi^2_{1,0.05}$ and SBC respectively], in order to select a distribution model. Output the parameter estimates for your chosen model using the function `summary`.

Chapter 4

Continuous response: regression analysis

4.1 Algorithms for fitting parametric regression models

A typical parametric regression model within the GAMLSS framework assumes that the response variable $Y_i \sim D(y_i, |\mu_i, \sigma_i, \nu_i, \tau_i)$, independently for $i = 1, \dots, n$, where the n length vectors of the distributional parameters can be modelled as a function of explanatory variables as

$$\begin{aligned} g_1(\boldsymbol{\mu}) &= \boldsymbol{\eta}_1 = \mathbf{X}_1\boldsymbol{\beta}_1 \\ g_2(\boldsymbol{\sigma}) &= \boldsymbol{\eta}_2 = \mathbf{X}_2\boldsymbol{\beta}_2 \\ g_3(\boldsymbol{\nu}) &= \boldsymbol{\eta}_3 = \mathbf{X}_3\boldsymbol{\beta}_3 \\ g_4(\boldsymbol{\tau}) &= \boldsymbol{\eta}_4 = \mathbf{X}_4\boldsymbol{\beta}_4. \end{aligned} \tag{4.1}$$

where the $\mathbf{X}$ matrices contain the explanatory variable values, the $\boldsymbol{\eta}$'s are the (linear) predictors, the $g()$ are known link functions (usually there to guarantee that the distributional parameters have the correct range) and the $\boldsymbol{\beta}$'s are the parameters to be estimated.

The likelihood to be maximized in this case with respect to the $\boldsymbol{\beta}$ parameters will be

$$L(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4) = \prod_{i=1}^n f(y_i | \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4). \tag{4.2}$$

As it turns out the likelihood in (4.2) can be maximized using an iterative algorithm, (described below) which repeatedly uses simple weighted linear regressions. The quantities needed for the RS algorithm are:

- the score function: $\mathbf{u}_k = \frac{\partial \ell}{\partial \boldsymbol{\eta}_k}$, for $k = 1, 2, 3, 4$.
- the adjusted dependent variables: $\mathbf{z}_k = \boldsymbol{\eta}_k + [\mathbf{W}_{kk}]^{-1} \mathbf{u}_k$, for $k = 1, 2, 3, 4$ and
- the diagonal matrices of iterative weights: $\mathbf{W}_{kk}$ which can have one of the following forms $-\frac{\partial^2 \ell}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_k^T}$, $-E \left[\frac{\partial^2 \ell}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_k^T} \right]$ or $\text{diag} \left\{ \left[\frac{\partial \ell}{\partial \eta_k} \right]^2 \right\}$ i.e. the observed information, expected information or product score function, depending respectively on whether a Newton-Raphson,

Fisher scoring or quasi Newton-Raphson algorithm is used, (see Lange, 1999, Chapter 11 for a definition of the techniques),

Below we describe a simplified version of the RS algorithm (the default method in the `gamlss()` function) used for fitting model (4.1). Let r be the outer cycle iteration index, k the parameter index and i the inner cycle iteration index. Essentially the RS algorithm has an outer cycle which checks the maximization of the overall likelihood with respect to the β 's and an inner cycle for fitting a model for each distributional parameter in turn, for $k = 1, 2, 3, 4$, where the other distribution parameters are fixed at their current values. Note at each calculation in the algorithm the most current updated values of all quantities are used. Note that $\theta^\top = (\theta_1, \theta_2, \theta_3, \theta_4) = (\mu, \sigma, \nu, \tau)$. The RS algorithm can be described as follows:

Algorithm RS (simple)

A simple version of the RS algorithm

- **Start:** Initialize fitted values $\theta_k^{(1,1)}$ for $k = 1, 2, 3, 4$ for distributional parameter vectors of length n , μ , σ , ν and τ respectively. Evaluate the initial linear predictors $\eta_k^{(1,1)} = g_k \left[\theta_k^{(1,1)} \right]$, for $k = 1, 2, 3, 4$.
- **START OUTER CYCLE** $r = 1, 2, \dots$ **UNTIL CONVERGENCE.**
 - FOR** $k = 1, 2, 3, 4$
 - **START INNER CYCLE** $i = 1, 2, \dots$ **UNTIL CONVERGENCE .**
 - * Evaluate the current $\mathbf{u}_k^{(r,i)}$, $\mathbf{W}_{kk}^{(r,i)}$ and $\mathbf{z}_k^{(r,i)}$
 - * Regress the current $\mathbf{z}_k^{(r,i)}$ against design matrix $\mathbf{X}_k$ using the iterative weights $\mathbf{W}_{kk}^{(r,i)}$ to obtain the updated parameter estimates $\beta_k^{(r,i)}$.
 - **END INNER CYCLE** on convergence of $\beta_k^{(r,\cdot)}$ and set $\beta_k^{(r+1,1)} = \beta_k^{(r,\cdot)}$, $\eta_k^{(r+1,1)} = \eta_k^{(r,\cdot)}$ and $\theta_k^{(r+1,1)} = \theta_k^{(r,\cdot)}$, otherwise update i and continue inner cycle.
 - UPDATE** value of k
- **END OUTER CYCLE:** if the change in the (penalized) likelihood is sufficiently small, otherwise update r and continue outer cycle.

4.2 Examples of fitting regression models to continuous distributions

4.2.1 The CD4 count data

Data summary:

R data file: CD4 in package `gamlss.dist` of dimensions 609×2

source: Wade and Ades (1994)

variables

`cd4` : CD4 counts from uninfected children born to HIV-1 mothers

`age` : age in years of the child.

purpose: to demonstrate regression fitting of different functional forms for the explanatory variable `age`.

conclusion: models for both μ and σ are needed with a 4-parameter distribution for the response variable

The above data are given by Wade and Ades (1994) and they refer to `cd4` counts from uninfected children born to HIV-1 mothers and the `age` in years of the child. Here we input and plot the data in Figure 4.1. This is a simple regression example with only one explanatory variable, the `age`, which is a continuous variable. The response while, strictly speaking is a count, is sufficiently large for us to treat it at this stage as a continuous response variable.

```
> library(gamlss.dist)
```

```
***** GAMLSS Version 1.8-2 *****
```

```
For more on GAMLSS look at http://www.gamlss.com/
```

```
Type gamlssNews() to see new features/changes/bug fixes.
```

```
> data("CD4")
```

```
> plot(cd4 ~ age, data = CD4)
```

There are several striking features in this specific set of data in Figure 4.1. The first has to do with the relationship between the mean of `cd4` and `age`. It is hard to see from the plot whether this relationship is linear or not. The second has to do with the heterogeneity of variance in the response variable `cd4`. It appears that the variation in `cd4` is decreasing with `age`. The final problem has to do with the distribution of `cd4` given the `age`. Is this distribution normal? It is hard to tell from the figure but probably we will need a more flexible distribution. Traditionally, problems of this kind were dealt with by a transformation in the response variable or a transformation in both in the response and the explanatory variable(s). One could hope that this would possibly correct for some or all the above problems simultaneously. Figure 4.2 shows plots where several transformations for `cd4` and `age` were tried. It is hard to see how we can improve the situation.

```
> op <- par(mfrow = c(3, 4), mar = par("mar") + c(0, 1, 0, 0),
+           pch = "+", cex = 0.45, cex.lab = 1.8, cex.axis = 1.6)
```

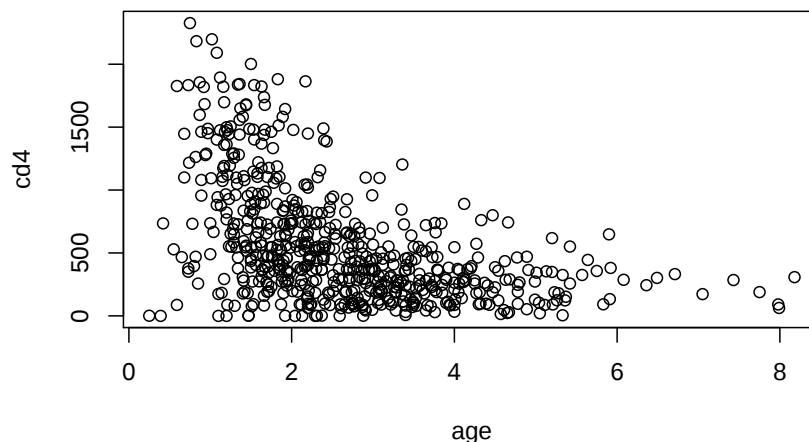


Figure 4.1: The plot of the CD4 data

```

> page <- c("age^-0.5", "log(age)", "age^.5", "age")
> pcd4 <- c("cd4^-0.5", "log(cd4+0.1)", "cd4^.5")
> for (i in 1:3) {
+   yy <- with(CD4, eval(parse(text = pcd4[i])))
+   for (j in 1:4) {
+     xx <- with(CD4, eval(parse(text = page[j])))
+     plot(yy ~ xx, xlab = page[j], ylab = pcd4[i])
+   }
+ }
> par(op)

```

Within the GAMLSS framework we can deal with these problems one at the time. First we start with the relationship between the mean of `cd4` and `age`. We will fit orthogonal polynomials of different orders to the data and choose the best using a GAIC criterion. For now we fit a constant variance and a default normal distribution.

```

> con <- gamlss.control(trace = FALSE)
> m1 <- gamlss(cd4 ~ age, sigma.fo = ~1, data = CD4, control = con)
> m2 <- gamlss(cd4 ~ poly(age, 2), sigma.fo = ~1, data = CD4, control = con)
> m3 <- gamlss(cd4 ~ poly(age, 3), sigma.fo = ~1, data = CD4, control = con)
> m4 <- gamlss(cd4 ~ poly(age, 4), sigma.fo = ~1, data = CD4, control = con)
> m5 <- gamlss(cd4 ~ poly(age, 5), sigma.fo = ~1, data = CD4, control = con)
> m6 <- gamlss(cd4 ~ poly(age, 6), sigma.fo = ~1, data = CD4, control = con)
> m7 <- gamlss(cd4 ~ poly(age, 7), sigma.fo = ~1, data = CD4, control = con)
> m8 <- gamlss(cd4 ~ poly(age, 8), sigma.fo = ~1, data = CD4, control = con)

```

First we compare the model using the Akaike Information criterion (AIC) which has penalty $k = 2$ for each parameter in the model, (the default value in the function `GAIC()`):

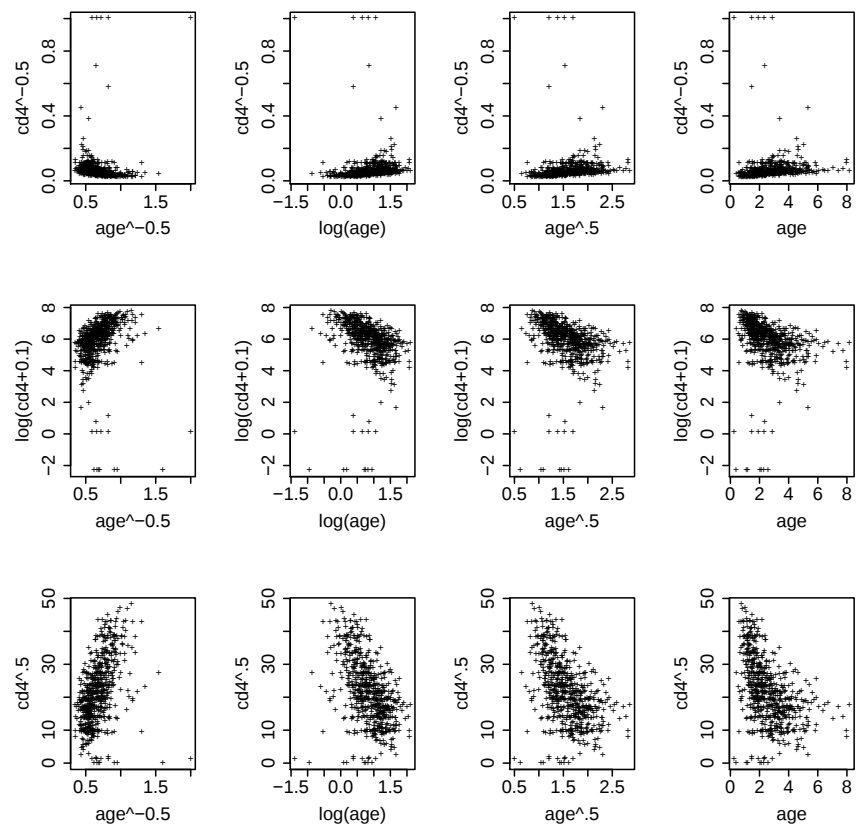


Figure 4.2: The CD4 data with various transformations for CD4 and age

```
> GAIC(m1, m2, m3, m4, m5, m7, m8)
```

	df	AIC
m7	9	8963.263
m8	10	8963.874
m5	7	8977.383
m4	6	8988.105
m3	5	8993.351
m2	4	8995.636
m1	3	9044.145

Next we compare the models using Schwartz Bayesian Criterion (SBC) which uses penalty $k = \log(n)$:

```
> GAIC(m1, m2, m3, m4, m5, m7, m8, k = log(length(CD4$age)))
```

	df	AIC
m7	9	9002.969
m8	10	9007.992
m5	7	9008.266
m2	4	9013.284
m4	6	9014.576
m3	5	9015.410
m1	3	9057.380

```
> plot(cd4 ~ age, data = CD4)
> lines(CD4$age[order(CD4$age)], fitted(m7)[order(CD4$age)], col = "red")
```

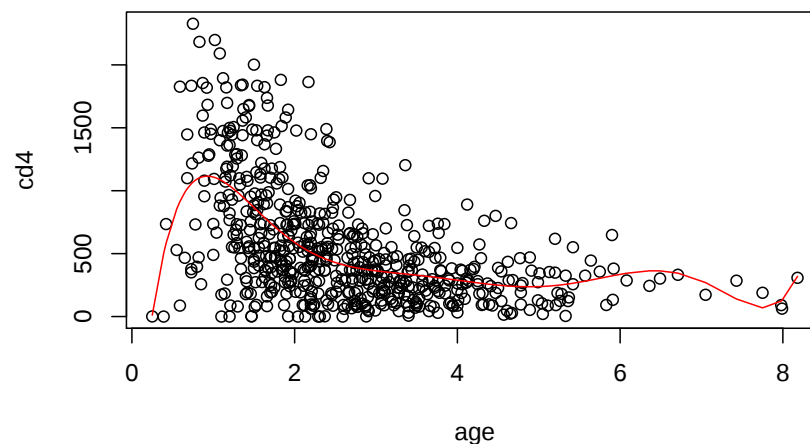


Figure 4.3: The CD4 data and the fitted values using polynomial of degree 7 in age

Remarkably with both AIC and SBC select model m7, with a polynomial of degree 7, as the best model. Unfortunately the fitted values for the mean of `cd4` shown together with the data in Figure 4.3 look rather unconvincing. The line is too wobbly at the ends of the range of `age`, trying to be very close to the data. This is a typical behavior of polynomial fitting.

Now we will try alternatives methods, two parametric, using *fractional polynomials* and piecewise polynomials, and one non-parametric smoothing method using *cubic smoothing splines*. We start first with the fractional polynomials. Fractional polynomials were introduced by Royston and Altman (1994). The function `fp()` which we are going to use to fit them works in `gamlss()` as an additive smoother. It can be used to fit the best (fractional) polynomial within a specific set of possible power values. Its argument `npoly` determines whether one, two or three terms in the fractional polynomial will be used in the fitting. For example with `npoly=3` the following polynomial functions are fitted to the data $\beta_0 + \beta_1 x^{p_1} + \beta_2 x^{p_2} + \beta_3 x^{p_3}$ where each p_j , for $j = 1, 2, 3$ can take any value within the set $(-2, -1, -0.5, 0, 0.5, 1, 2, 3)$. If two powers, p_j 's, happen to be identical then the two terms $\beta_{1j} x^{p_j}$ and $\beta_{2j} x^{p_j} \log(x)$ are fitted instead. Similarly if three powers p_j 's are identical the terms fitted are $\beta_{1j} x^{p_j}$, $\beta_{2j} x^{p_j} \log(x)$ and $\beta_{3j} x^{p_j} [\log(x)]^2$. Here we fit fractional polynomials with one, two and three terms respectively and we choose the best using GAIC:

```
> m1f <- gamlss(cd4 ~ fp(age, 1), sigma.fo = ~1, data = CD4, control = con)
> m2f <- gamlss(cd4 ~ fp(age, 2), sigma.fo = ~1, data = CD4, control = con)
> m3f <- gamlss(cd4 ~ fp(age, 3), sigma.fo = ~1, data = CD4, control = con)
> GAIC(m1f, m2f, m3f)
```

	df	AIC
m3f	8	8966.375
m2f	6	8978.469
m1f	4	9015.321

```
> GAIC(m1f, m2f, m3f, k = log(length(CD4$age)))
```

	df	AIC
m3f	8	9001.669
m2f	6	9004.940
m1f	4	9032.968

```
> m3f
```

```
Family: c("NO", "Normal")
Fitting method: RS()
```

```
Call: gamlss(formula = cd4 ~ fp(age, 3), sigma.formula = ~1, data = CD4,
  control = con)
```

```
Mu Coefficients:
```

```
(Intercept)    fp(age, 3)
      557.5             NA
```

```
Sigma Coefficients:
```

```
(Intercept)
      5.93
```

```

Degrees of Freedom for the fit: 8 Residual Deg. of Freedom    601
Global Deviance:      8950.37
                    AIC:      8966.37
                    SBC:      9001.67

> m3f$mu.coefSmo

[[1]]
[[1]]$coef
      one
-599.2970 1116.7924 1776.1937  698.6097

[[1]]$power
[1] -2 -2 -2

[[1]]$varcoeff
[1] 2016.238 4316.743 22835.146 7521.417

> plot(cd4 ~ age, data = CD4)
> lines(CD4$age[order(CD4$age)], fitted(m1f)[order(CD4$age)], col = "blue")
> lines(CD4$age[order(CD4$age)], fitted(m2f)[order(CD4$age)], col = "green")
> lines(CD4$age[order(CD4$age)], fitted(m3f)[order(CD4$age)], col = "red")

```

Both AIC and BSC favour the model `m3f` with a fractional polynomial with three terms. Note that by printing `m3f` the model for μ gives a value of 557.5 for the "Intercept" and NULL for the coefficient for `fp(age, 3)`. This is because within the backfitting the constant is fitted first and then the fractional polynomial is fitted to the partial residuals of the constant model. As a consequence the constant is fitted twice. The coefficients and the power transformations of the fractional polynomials can be obtained using the `mu.coefSmo` component of the `gamlss` fitted object. For the `CD4` data all powers happens to be -2 indicating that the following terms are fitted in the model, age^{-2} , $age^{-2} \log(age)$ and $age^{-2} [\log(age)]^2$. Hence the fitted model `m3f` is given by $cd4 \sim NO(\hat{\mu}, \hat{\sigma})$, where $\hat{\mu} = 557.5 - 599.3 + 1116.8 \, age^{-2} + 17776.2 \, age^{-2} \log(age) + 698.6 \, age^{-2} [\log(age)]^2$ and $\hat{\sigma} = \exp(5.93) = 376.2$. Figure 4.4 shows the best fitted models using one, two or three fractional polynomial terms. The situation remains unconvincing. None of the models seem to fit particular well.

Next we will fit piecewise polynomials using the R function `bs`. We try different degrees of freedom (effectively different number of knots) and we choose the best model using AIC and SBC:

```

> m2b <- gamlss(cd4 ~ bs(age), data = CD4, trace = FALSE)
> m3b <- gamlss(cd4 ~ bs(age, df = 3), data = CD4, trace = FALSE)
> m4b <- gamlss(cd4 ~ bs(age, df = 4), data = CD4, trace = FALSE)
> m5b <- gamlss(cd4 ~ bs(age, df = 5), data = CD4, trace = FALSE)
> m6b <- gamlss(cd4 ~ bs(age, df = 6), data = CD4, trace = FALSE)
> m7b <- gamlss(cd4 ~ bs(age, df = 7), data = CD4, trace = FALSE)
> m8b <- gamlss(cd4 ~ bs(age, df = 8), data = CD4, trace = FALSE)
> GAIC(m2b, m3b, m4b, m5b, m6b, m7b, m8b)

      df      AIC
m7b  9 8959.519

```

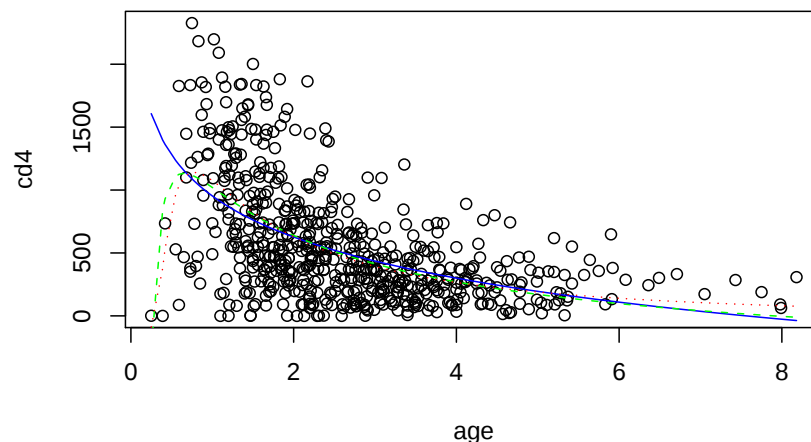


Figure 4.4: The CD4 data and the best fitting fractional polynomials in age with one (—), two (— —) and three (...) terms respectively

```

m6b  8 8960.353
m8b 10 8961.073
m5b  7 8964.022
m4b  6 8977.475
m2b  5 8993.351
m3b  5 8993.351

> GAIC(m2b, m3b, m4b, m5b, m6b, m7b, m8b, k = log(length(CD4$age)))

      df      AIC
m5b   7 8994.904
m6b   8 8995.648
m7b   9 8999.225
m4b   6 9003.946
m8b  10 9005.191
m2b   5 9015.410
m3b   5 9015.410

```

The best model with AIC uses 7 degrees of freedom while with SBC 5. Figure 4.5 shows the fitted models using 5, 6 and 7 degrees of freedom for the piecewise polynomial in age.

We will proceed by fitting cubic smoothing splines to the data. In smoothing splines the problem is how to choose a sensible value for the smoothing parameter λ . The smoothing parameter is a function of the (effective) degrees of freedom, so we will use the following procedure: we will use the `optim()` function in R to find the model with the optimal (effective) degrees of freedom according to a GAIC. Again we do not commit ourselves to what penalty we should use in the GAIC but we will try both AIC and SBC, with penalties $k = 2$ and $\log(n)$ respectively, in the `GAIC()` function:

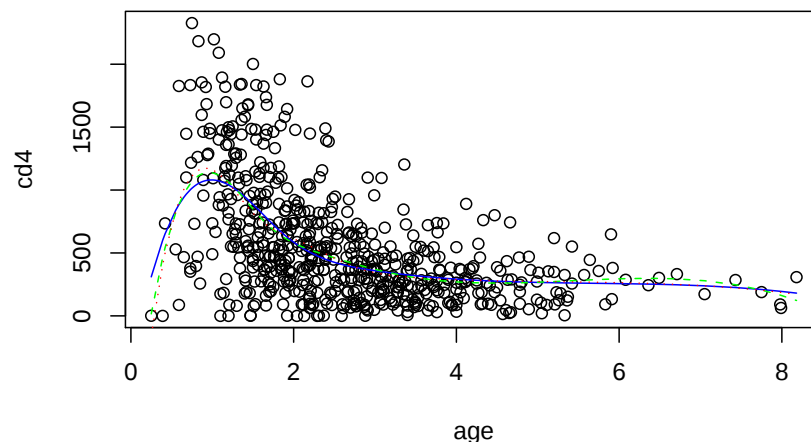


Figure 4.5: The CD4 data and the best fitting piecewise polynomials in age with 5 (—), 6, (— — —) and 7 (...), degrees of freedom respectively

```
> fn <- function(p) AIC(gamlss(cd4 ~ cs(age, df = p[1]), data = CD4,
+   trace = FALSE), k = 2)
> opAIC <- optim(par = c(3), fn, method = "L-BFGS-B", lower = c(1),
+   upper = c(15))
> fn <- function(p) AIC(gamlss(cd4 ~ cs(age, df = p[1]), data = CD4,
+   trace = FALSE), k = log(length(CD4$age)))
> opSBC <- optim(par = c(3), fn, method = "L-BFGS-B", lower = c(1),
+   upper = c(15))
> opAIC$par

[1] 10.85157

> opSBC$par

[1] 1.854689

> maic <- gamlss(cd4 ~ cs(age, df = 10.85), data = CD4, trace = FALSE)
> msbc <- gamlss(cd4 ~ cs(age, df = 1.85), data = CD4, trace = FALSE)
```

According to AIC the best model is the one with smoothing degrees of freedom $10.85 \approx 11$. This model seems to overfit the data as can be seen in Figure 4.6, (green continuous line). This is typical behaviour of AIC when it is used in this context. Note that 11 degrees of freedom in the fit refers to the extra degrees of freedom after the constant and the linear part is fitted to the model, so the overall degrees of freedom are 13. The best model using SBC has $1.854 \approx 2$ degrees of freedom for smoothing (i.e. 4 overall) and is shown in Figure 4.6 (blue dashed line). It fits well for most of the observations but not at small values of `age`. It appears that we need

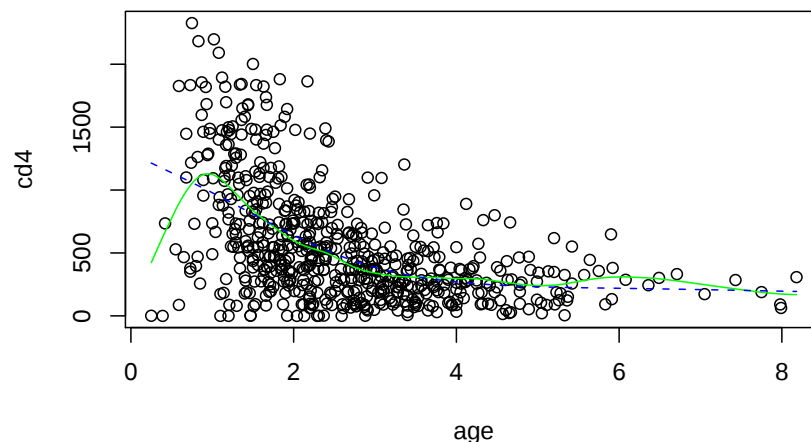


Figure 4.6: The CD4 data and two different cubic spline fits in age, with four, (—), and twelve, (---), total effective degrees of freedom in the fit

a model with smoothing degrees of freedom for the cubic spline with a value between 2 and 11 (i.e. 4 and 13 overall).

Given that the smooth cubic spline model for μ appears reasonable, we proceed by looking for a suitable model for $\log \sigma$ (since the default link function for σ for the normal NO distribution is a log link). We try a smooth cubic splines model for both terms. The following code will give us the best choice of degrees of freedom for both μ and $\log \sigma$ according to AIC. [Note the fit of model `m1` is used as the starting values.]

```
> m1 <- gamlss(cd4 ~ cs(age, df = 10), sigma.fo = ~cs(age, df = 2),
+   data = CD4, trace = FALSE)
> fn <- function(p) AIC(gamlss(cd4 ~ cs(age, df = p[1]), sigma.fo = ~cs(age,
+   p[2]), data = CD4, trace = FALSE, start.from = m1), k = 2)
> opAIC <- optim(par = c(8, 3), fn, method = "L-BFGS-B", lower = c(1,
+   1), upper = c(15, 15))
> opAIC$par
[1] 3.717336 1.808830
```

The resulting smoothing degrees of freedom for μ and σ are 3.72 and 1.81 respectively. The degrees of freedom for μ are lower than expected from the previous analysis. It appears that by picking a suitable model for σ the model for μ is less complicated. Rerunning the code for SBC results in estimated degrees of freedom (2.55, 0.93). Note that the `lower` argument in `optim` had to change to `lower=c(0.1, 0.1)` allowing the smoothing degrees of freedom to be lower than 1. We now fit the two chosen models:

```
> m42 <- gamlss(cd4 ~ cs(age, df = 3.72), sigma.fo = ~cs(age, df = 1.81),
+   data = CD4, trace = FALSE)
```

```
> m31 <- gamlss(cd4 ~ cs(age, df = 2.55), sigma.fo = ~cs(age, df = 0.93),
+   data = CD4, trace = FALSE)
```

Figure 4.7 shows the fitted values for both the models. The plot is obtained using the command `fitted.plot(m42,m31, x=CD4$age, line.type=TRUE)`. The function `fitted.plot()` is appropriate when only one explanatory variable is fitted to the data. The models are almost identical apart from early `age` where the SBC model has a slightly lower mean and standard deviation of `cd4` than the AIC.

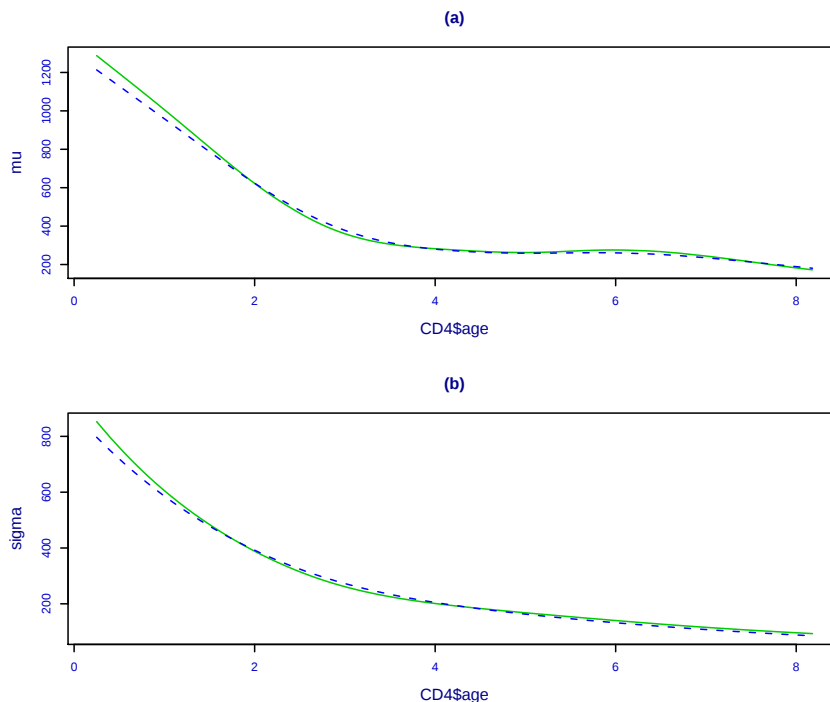


Figure 4.7: A plot of the fitted μ and σ values against `age` for models `m42` (continuous green) and `m31` (dashed blue).

The validation generalized deviance function `VGD()` provides an alternative way of tuning the degrees of freedom in a smoothing situation. It is suitable for large sets of data where we can afford to use part of the data for fitting the model (training) and part for validation. Here we demonstrate how it can be used. First the data are split into training and validation subsets with approximately 60% and 40% of the cases respectively. Then we use the function `optim` to search for the optimum smoothing degrees for μ and σ (which results in a fitted model to the training data which minimizes the validation deviance).

```
> set.seed(1234)
> rSample6040 <- sample(2, length(CD4$cd4), replace = T, prob = c(0.6,
+   0.4))
> fn <- function(p) VGD(cd4 ~ cs(age, df = p[1]), sigma.fo = ~cs(age,
+   df = p[2]), data = CD4, rand = rSample6040)
```

```
> op <- optim(par = c(3, 1), fn, method = "L-BFGS-B", lower = c(1,
+ 1), upper = c(10, 10))
> op$par
```

```
[1] 4.779947 1.376534
```

The resulting smoothing degrees of freedom for μ and σ are (4.78, 1.38), in this instance not very different from the ones we obtained from AIC and SBC methods.

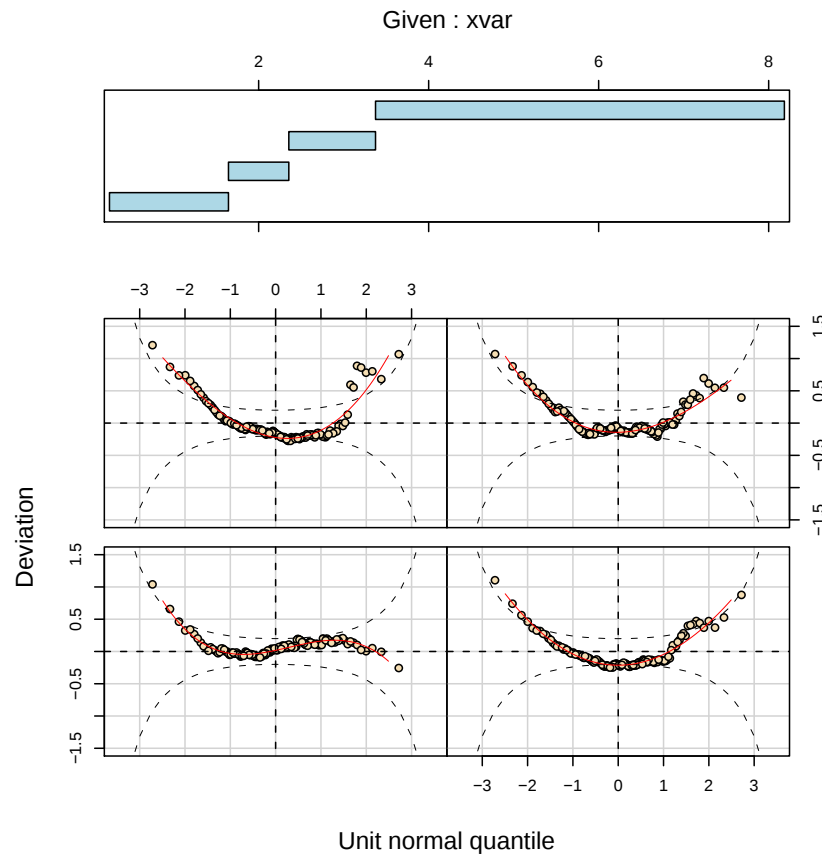


Figure 4.8: A worm plot of the residuals from models `m42`.

Figure 4.8 shows a worm plots from the residuals of model `m42`. Worm plots were introduced by van Buuren and Fredriks (2001). The important point here is that quadratic and cubic shapes in a worm plot indicate the presence of skewness and kurtosis respectively in the residuals (within the corresponding range of the explanatory variable, i.e. `CD4$age`). That is, the normal distribution fitted so far to the data is not appropriate. Now we try to identify a suitable distribution for the response variable. There are zeros in the response so unless we shift them to the right by a small amount, we must model it with distributions accepting zeros. Here we try the t (TF), power exponential (PE), sinh-arcsinh (SHASH) and skew exponential power type 3 (SEP3) distributions. We use the `update` function and start from the normal `m42` model:

```

> library(gamlss.dist)
> m42TF <- update(m42, family = TF)
> m42PE <- update(m42, family = PE)
> m42SEP3 <- update(m42, family = SEP3, method = mixed(30, 100))
> m42SHASH <- update(m42, family = SHASH, method = mixed(20, 100))
> GAIC(m42, m42TF, m42PE, m42SEP3, m42SHASH)

              df      AIC
m42SEP3  11.531173 8690.531
m42SHASH 11.530125 8703.417
m42TF    10.531045 8785.149
m42PE    10.531061 8790.081
m42      9.529738 8790.437

> GAIC(m42, m42TF, m42PE, m42SEP3, m42SHASH, k = log(length(CD4$age)))

              df      AIC
m42SEP3  11.531173 8741.405
m42SHASH 11.530125 8754.286
m42TF    10.531045 8831.611
m42      9.529738 8832.481
m42PE    10.531061 8836.542

```

The SEP3 distribution seems that fit this data best according to both the AIC and SBC criteria. It can be shown that by adding a linear model to the ν model, i.e. `nu.fo= age`, improves the AIC but not the SBC.

4.2.2 The abdominal circumference data

Data summary:

R data file: abdom in package **gamlss** of dimensions 610×2

variables

y : abdominal circumference

x : gestational age

purpose: to find appropriate distribution for the response variable and appropriate degrees of freedom for the smoothing parameters

conclusion: logistic distribution assumption is appropriate

comment: the data were first used in Section [2.3](#)

The data, in Figure [4.9](#) are measurements of abdominal circumference (Y), taken from 663 fetuses during ultrasound scans at Kings College Hospital, London, at gestational ages (X) ranging between 12 and 42 weeks. The data were used to derive reference centile intervals for Y given X by Chitty *et al.* (1994) and also for comparing different reference centile methods by Wright and Royston (1997), who were unable to find a satisfactory distribution for Y and commented that the distribution of residual z-scores obtained from the different fitted models

'has somewhat longer tails than the normal distribution'. Here we input the data and we plot them.

```
> library(gamlss)
> data(abdom)
> plot(y ~ x, data = abdom)
```

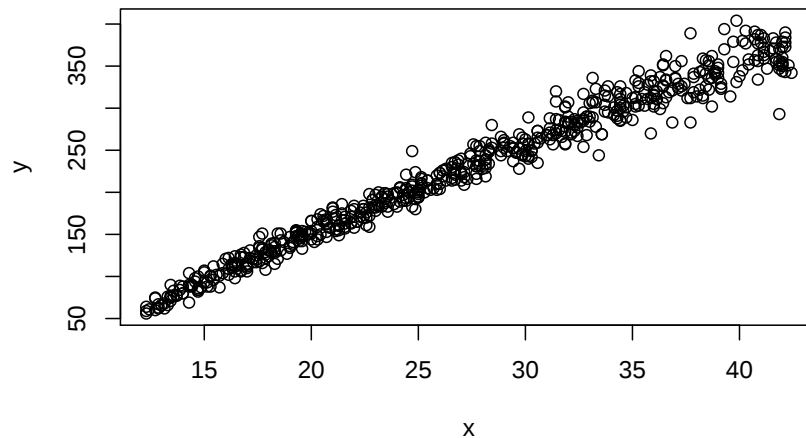


Figure 4.9: The plot of the abdominal data data

Initial model fitting using a normal distribution indicated that a cubic smoothing spline in x , $cs(x,3)$, with 3 extra effective degrees of freedom on top of the constant (denoted by 1) and the linear terms, was adequate for each of the location and log scale parameters. Hence, to determine an appropriate distribution for these data, different distributions were fitted to the data with the location and log scale parameters modelled using model $cs(x,df=3)$ for both μ and $\log \sigma$, and any shape parameters modelled as constants.

Table 4.1 shows the GD, AIC and SBC for the resulting fitted models using different distributions. The logistic (LO) distribution is selected if the SBC is used as a selection criterion, and the Box-Cox t distribution (BCT) if the AIC is used. The table was created using the following R commands:

```
> con <- gamlss.control(trace = F)
> mNO <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = NO, control = con)
> mGA <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = GA(mu.link = identity), control = con)
> mIG <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = IG(mu.link = identity), control = con)
> mGU <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = GU, control = con)
```

```

> mRG <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = RG, control = con)
> mLO <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = LO, control = con)
> mCG <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = BCCG(mu.link = identity), control = con)
> mPE <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = PE, control = con)
> mTF <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3), data = abdom,
+   family = TF, control = con)
> mBCT <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3),
+   data = abdom, family = BCT, control = con)
> mBCPE <- gamlss(y ~ cs(x, df = 3), sigma.fo = ~cs(x, df = 3),
+   data = abdom, family = BCPE, control = con)

```

```

> GAIC(mNO, mGA, mIG, mGU, mRG, mLO, mCG, mPE, mTF, mBCT, mBCPE,
+   k = 0)

```

	df	AIC
mBCT	12.001483	4772.422
mBCPE	12.001513	4774.754
mTF	11.001504	4776.544
mLO	10.001497	4778.974
mPE	10.999854	4779.195
mIG	9.999854	4779.361
mCG	10.999864	4779.529
mGA	9.999858	4780.056
mNO	9.999872	4784.718
mRG	9.999815	4856.258
mGU	10.001505	4947.792

```

> GAIC(mNO, mGA, mIG, mGU, mRG, mLO, mCG, mPE, mTF, mBCT, mBCPE,
+   k = 2)

```

	df	AIC
mBCT	12.001483	4796.425
mTF	11.001504	4798.547
mBCPE	12.001513	4798.757
mLO	10.001497	4798.977
mIG	9.999854	4799.361
mGA	9.999858	4800.056
mPE	10.999854	4801.195
mCG	10.999864	4801.528
mNO	9.999872	4804.718
mRG	9.999815	4876.257
mGU	10.001505	4967.795

```

> GAIC(mNO, mGA, mIG, mGU, mRG, mLO, mCG, mPE, mTF, mBCT, mBCPE,
+   k = log(length(abdom$y)))

```

Table 4.1: Models for the abdominal circumference data

Distributions	GD	AIC	SBC
Normal	4784.7	4804.7	4848.9
Gamma	4780.1	4800.1	4844.2
Inv. Gaussian	4779.4	4799.4	4843.5
Gumbel	4948.2	4968.2	5012.3
Reverse Gumbel	4856.7	4876.7	4920.9
Logistic	4778.8	4798.8	4842.9
Cole and Green	4779.5	4801.5	4850.1
Power Exp.	4779.1	4801.1	4849.6
t family	4776.545	4798.5	4847.1
Box-Cox t	4772.42	4796.43	4849.39
Box-Cox Power Exp.	4774.75	4798.76	4851.73

	df	AIC
mLO	10.001497	4843.118
mIG	9.999854	4843.495
mGA	9.999858	4844.190
mTF	11.001504	4847.102
mNO	9.999872	4848.852
mBCT	12.001483	4849.393
mPE	10.999854	4849.742
mCG	10.999864	4850.076
mBCPE	12.001513	4851.725
mRG	9.999815	4920.391
mGU	10.001505	5011.936

The penalty $k = 2$ above gives the usual Akaike information criterion (AIC), the penalty $k = 0$ gives the global deviance and the penalty $k = \log(n)$ gives the SBC.

Having identified the need for a kurtotic distribution we can now try to fine tune, i.e. simplify, the mean and the dispersion models. Table 4.2 shows the SBC table for the logistic model (LO) classified according to different total (effective) degrees of freedom for the location, (3, 4, 5, 6, 7), and for the scale, (1, 2, 3, 4, 5), models. Corresponding tables were obtained for each of the other distributions in Table 4.1. The overall 'best' model using the SBC criterion is a logistic model with a cubic smoothing spline $\text{cs}(x, 2)$ model for μ and a linear model x for σ , with $\text{SBC}=4825.2$. This was the chosen "best" model. The AIC criterion ultimately lead to the same model, except for model $\text{cs}(x, 5)$ for μ . Note that for the LO distribution, the location parameter μ is both the median and the mean of the distribution.

The R code for creating Table 4.2 is shown below. Note that special control has to be imposed for the first model $\text{MM}[1, 1]$ with the μ model $\text{mu.fo}=\text{cs}(x, 1)$ and the constant σ model given by $\text{sigma.fo}=\sim 1$ otherwise it will not converge:

```
> con <- gamlss.control(trace = F)
> con1 <- gamlss.control(trace = F, sigma.step = 0.1)
> MM <- matrix(0, nrow = 5, ncol = 5)
> MM[1, 1] <- gamlss(y ~ cs(x, 1), sigma.fo = ~1, data = abdom,
+   family = LO, control = con1)$sbc
> for (j in 2:5) MM[1, j] <- gamlss(y ~ cs(x, j), sigma.fo = ~1,
```

	df	mean				
		3	4	5	6	7
scale	1	4925.2	4921.3	4923.6	4927.6	4932.3
	2	4830.0	4825.2	4826.5	4830.2	4835.2
	3	4835.5	4831.1	4832.4	4836.1	4841.0
	4	4841.4	4836.6	4837.7	4841.4	4846.2
	5	4847.1	4842.0	4842.9	4846.5	4851.3

Table 4.2: Abdominal circumference data: SBC for the logistic distribution

```

+ data = abdom, family = L0, control = con1)$sbc
> for (j in 1:5) {
+   for (i in 2:5) {
+     MM[i, j] <- gamlss(y ~ cs(x, df = j), sigma.fo = ~cs(x,
+       df = i - 2), data = abdom, family = L0, , control = con)$sbc
+   }
+ }
+ }
> MM

```

```

      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 4925.255 4921.321 4923.558 4927.629 4932.296
[2,] 4830.015 4825.232 4826.474 4830.243 4835.171
[3,] 4835.700 4831.167 4832.421 4836.180 4841.096
[4,] 4841.628 4836.758 4837.867 4841.536 4846.402
[5,] 4847.405 4842.184 4843.118 4846.687 4851.499

```

Fig. 4.10 displays the data and the fitted median (and mean) parameter μ for the chosen “best” model, together with 2.5% and 97.5% reference centile estimates for Y . For a given age x , the 100 α % reference centile estimate for Y is $\hat{Y}_\alpha = \hat{\mu} + z_\alpha \hat{\sigma}$ where $z_\alpha = F^{-1}(\alpha) = \log[\alpha/(1 - \alpha)]$, where $F(z)$ is the cumulative distribution function of a standard logistic distribution and $\hat{\mu}$ and $\hat{\sigma}$ are the fitted values of the location and scale parameters for Y at $X = x$. Approximate confidence interval bands for the centiles can also be obtained, using the empirical Bayes argument in Rigby and Stasinopoulos (2000). The R commands for the centile plot are as follows:

```

> m2L0 <- gamlss(y ~ cs(x, df = 2), sigma.fo = ~x, data = abdom,
+   family = L0, control = con)
> centiles(m2L0, xvar = abdom$x, cent = c(2.5, 50, 97.5))

% of cases below 2.5 centile is 2.295082
% of cases below 50 centile is 50.98361
% of cases below 97.5 centile is 97.86885

```

A plot of the (normalized quantile) residuals, using the command `plot(mFIN)`, for the chosen model is shown in Fig. 4.11. Panels (a) and (b) plot the residuals against the fitted values and against the case number respectively, while panels (c) and (d) provide a kernel density estimate and QQ plot for residuals respectively. The residuals appear random but the QQ plot indicates that the lower tail of the distribution of the response variable is slightly shorter than the logistic. Nevertheless it provides a reasonable fit to the data.

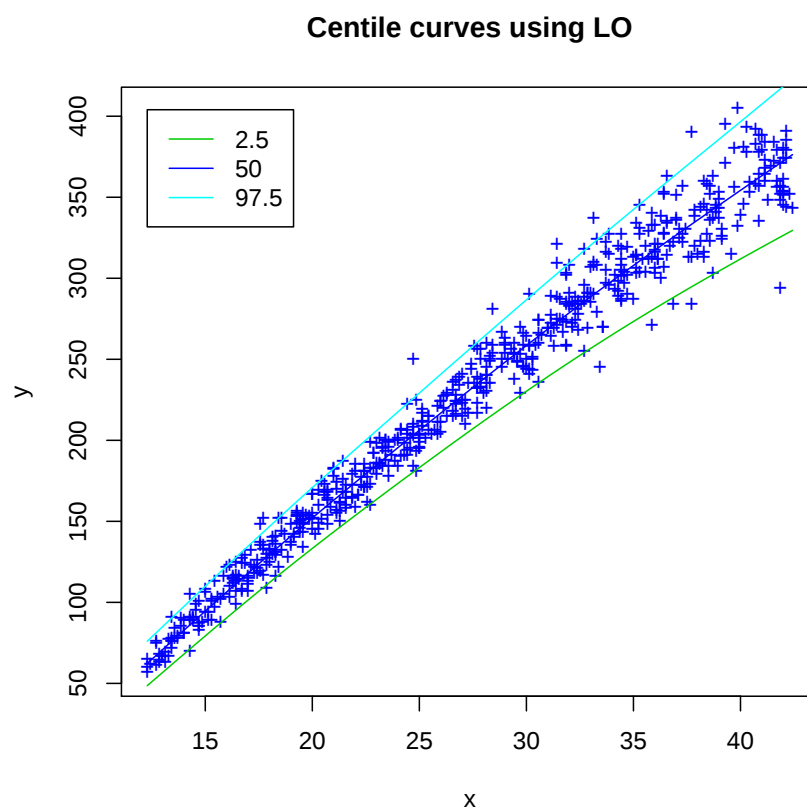


Figure 4.10: The abdominal circumference data with the fitted median curve, (50%) centile, and 2.5% and 97.5% reference centile curves from the chosen m2LO model

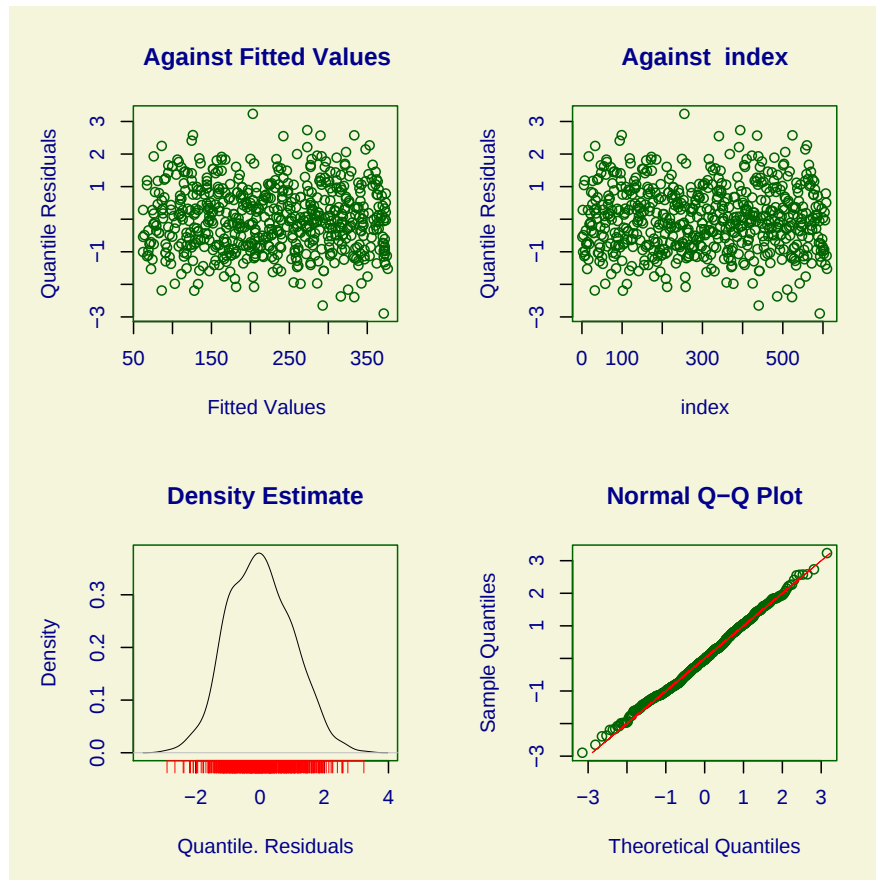


Figure 4.11: The residual plot from model m2LO of the abdominal circumference data

To validate the model, we can randomly spit the data into training (60%) and validation (40%) data sets. Minimizing the GD for the training data can be used for model fitting, while minimizing the GD for the validation data (i.e. VGD) can be used for selecting the distribution and the combination of degrees of freedom for the location and scale models. We provide a function called `VGD()` to help with this. To use the `VGD()` function, you need to create a index which splits the data into training and validation data sets with approximately 60% and 40% of the observations respectively. This is done using the R function `sample()`.

```
> set.seed(101)
> rand <- sample(2, 610, replace = T, prob = c(0.6, 0.4))
> table(rand)/610

rand
      1      2
0.5983607 0.4016393

> VAL <- matrix(0, nrow = 4, ncol = 5)
> for (j in 1:5) {
+   for (i in 1:4) {
+     VAL[i, j] <- VGD(y ~ cs(x, df = j), sigma.fo = ~cs(x,
+       df = i - 1), data = abdom, family = LO, rand = rand)
+   }
+ }
> VAL

      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1918.774 1914.119 1912.946 1913.066 1913.495
[2,] 1923.890 1918.896 1917.655 1917.759 1918.151
[3,] 1926.901 1921.437 1920.059 1920.114 1920.477
[4,] 1928.452 1922.714 1921.231 1921.247 1921.595
```

In the commands above *LO* distribution models with different degrees of freedom for its μ and $\log\sigma$ models are fitted to the training data and the resulting validation global deviance is calculated from the validation data. Note that in general the best choice of model using the VGD criterion would vary according to what actual proportion in the sample is chosen for fitting (training) and validation. Here the model with the minimum $VGD = 1912.946$ is close to the one chosen by the SBC criterion earlier, except that it selects `cs(x,3)` for μ instead of `cd(x,2)` previously. Note that the argument of the function `VGD()`, `rand` should be the same for all searches for the best model.

4.3 Bibliography

Exercises for Chapter 4

- **Q1 Visual analog scale (VAS) data:** The Visual analog scale is used to measure pain and quality of life. For example patients are required to indicate in a scale from 0 to 100 the amount of discomfort they have. This can be easily translated to a value from 0 to 1 and consequently analyzed using the beta distribution. Unfortunately if 0's or 100's are recorded the beta distribution is not appropriate since the values 0 and 1 are not allowed in

the definition of the beta distribution. Here we use the inflated beta distribution allowing values at 0 and 1. This is a mixed distribution (continuous and discrete) having four parameters, ν for modelling the probability at zero $p(Y = 0)$ relative to $p(0 < Y < 1)$, τ for modelling the probability at one $p(Y = 1)$ relative to $p(0 < Y < 1)$, and μ and σ for modelling the between values, $0 < Y < 1$, using a beta distributed variable $BE(\mu, \sigma)$ with mean μ and variance $\sigma\mu(1 - \mu)$.

In the original data 368 patients, measured at 18 times after treatment with one of 7 drug treatments (including placebo), plus a baseline measure (time=0) and one or more pre-baseline measures (time=-1). Here for illustration we will ignore the repeated measure nature of the data and we shall use data from time 5 only (364 observations). The VAS scale response variable, Y , is assumed to be distributed as $Y \sim BEINF(\mu, \sigma, \nu, \tau)$ where any of the distributional parameters μ , σ , ν and τ are modelled as a constant or as a function of the treatment, (*treat*).

- (a) Fit all 16 possible combinations of models i.e.

```

vas5 <- dget("D:/gamlss/data/palermo/vas-5")
mod01 <- gamlss(vas/100 ~ 1, data = vas5, family = BEINF)
mod11 <- gamlss(vas/100 ~ treat, data = vas5, family = BEINF)
mod12 <- gamlss(vas/100 ~ 1, sigma.fo = ~treat, data = vas5,
               family = BEINF)
mod13 <- gamlss(vas/100 ~ 1, nu.fo = ~treat, data = vas5,
               family = BEINF)
mod14 <- gamlss(vas/100 ~ 1, tau.fo = ~treat, data = vas5,
               family = BEINF)
mod21 <- gamlss(vas/100 ~ treat, sigma.fo = ~treat, data = vas5,
               family = BEINF)
mod22 <- gamlss(vas/100 ~ treat, nu.fo = ~treat, data = vas5,
               family = BEINF)
mod23 <- gamlss(vas/100 ~ treat, tau.fo = ~treat, data = vas5,
               family = BEINF)
mod24 <- gamlss(vas/100 ~ 1, sigma.fo = ~treat, nu.fo = ~treat,
               data = vas5, family = BEINF)
mod25 <- gamlss(vas/100 ~ 1, sigma.fo = ~treat, tau.fo = ~treat,
               data = vas5, family = BEINF)
mod26 <- gamlss(vas/100 ~ 1, nu.fo = ~treat, tau.fo = ~treat,
               data = vas5, family = BEINF)
mod31 <- gamlss(vas/100 ~ treat, sigma.fo = ~treat, nu.fo = ~treat,
               data = vas5, family = BEINF)
mod32 <- gamlss(vas/100 ~ treat, sigma.fo = ~treat, tau.fo = ~treat,
               data = vas5, family = BEINF)
mod33 <- gamlss(vas/100 ~ treat, nu.fo = ~treat, tau.fo = ~treat,
               data = vas5, family = BEINF)
mod34 <- gamlss(vas/100 ~ 1, sigma.fo = ~treat, nu.fo = ~treat,
               tau.fo = ~treat, data = vas5, family = BEINF)
mod41 <- gamlss(vas/100 ~ treat, sigma.fo = ~treat, nu.fo = ~treat,
               tau.fo = ~treat, data = vas5, family = BEINF)

```

- (b) Use the AIC function with each of the penalties $k = 2, 3.8$ and $7.96 = \log(2868)$,

[corresponding to criteria AIC, $\chi^2_{1,0.05}$ and SBC respectively], in order to select the best model.

- (c) For you best model plot the seven different fitted distributions (one for each treatment. The following R commands can be used:

```
> mod22 <- gamlss(vas/100 ~ treat, nu.fo = ~treat, data = vas5,
+   family = BEINF)
> op <- par(mfrow = c(4, 2))
> lev <- c(1, 2, 3, 4, 5, 6, 7)
> ind <- c(3, 2, 9, 12, 1, 7, 4)
> j <- 0
> for (i in ind) {
+   j = 1 + j
+   xlab <- paste("treatment = ", eval(substitute(lev[j])))
+   ylab <- paste("p(y)")
+   plotBEINF(mu = fitted(mod22)[i],
+             sigma = fitted(mod22, "sigma")[i],
+             nu = fitted(mod22, "nu")[i],
+             tau = fitted(mod22, "tau")[i],
+             from = 0, to = 1, n = 101, xlab = xlab, ylab = ylab)
+ }
> term.plot(mod22, se = T)
> par(op)
```


Chapter 5

Discrete response: count data

5.1 Introduction

By count data we mean data where the theoretical distribution of the response variable can take values at $0, 1, 2, \dots, \infty$. The classical approach to model this type of data especially if the counts are relatively small is using the Poisson distribution. The problem is that very often count data which are modelled using a Poisson distribution exhibit overdispersion. Overdispersion is defined as the extra variation occurred in modelling count data which is not explained by the Poisson distribution alone.

Overdispersion has been recognized for a long time as a potential problem within the literature of generalized linear models, (Nelder and Wedderburn, 1972) which originally modelled only the mean of the distribution of the response. Over the years several solutions to the problem of overdispersion have been suggested, see e.g. Consul (1989) and Dossou-Gbété and Mizère (2006). Here we consider three major categories:

- (i) *Ad-hoc* solutions
- (ii) Discretized continuous distributions
- (iii) Random effect at the observation level solutions.

We refer here to *ad-hoc* solutions as those that have been implemented in the past, mainly for their computational convenience (and some also for good asymptotic properties for the estimation of the mean regression function), but which do *not* assume an explicit proper distribution for the response variable. The quasi-likelihood function approach proposed by Wedderburn (1974), for example, requires assumptions on the first two moments of the response variable. The quasi-likelihood approach is incapable of modelling the second moment parameter, the dispersion, as a function of explanatory variables, therefore the extended quasi-likelihood (EQL) was proposed by Nelder and Pregibon (1987). Alternatively approaches are the pseudo-likelihood (PL) method introduced by Carroll and Ruppert (1982) and Efron's double exponential (EDE) family, Efron (1986). The PL method effectively approximates the probability function by a normal distribution with a chosen variance-mean relationship, but does not properly maximize the resulting likelihood. See Davidian and Carroll (1988) and Nelder and Lee (1992) for a comparison of the EQL and the PL. The problem with all these methods is that, while they work well with moderate overdispersion, they have difficulty modelling long tails in the distribution of the response variable. They also suffer from the fact, that, for a given set of data,

the adequacy of the fit of those methods cannot be compared using a properly maximized log likelihood function $\hat{\ell}$ and criteria based on $\hat{\ell}$, e.g. the (generalized) Akaike information criterion $AIC = -2\hat{\ell} + \sharp.df$, where $\sharp$ is the penalty and df denotes the total (effective) degrees of freedom used in the model. The problem is that they do not properly fit a discrete distribution. For the EQL and EDE methods the distribution probabilities do not add up to one, see for example Stasinopoulos (2006). Note that with increasing computer power the constant of summation, missing from the EQL and EDE methods, can be calculated so that they represent proper distributions resulting in a true likelihood function that can be maximized. However these models are still computational slow to fit to large data sets, the true probability function cannot be expressed explicitly (except by including an infinite sum for the constant of summation) and their flexibility is limited by usually having at most two parameters. See Lindsey (1999) for a similar criticism of the *ad-hoc* methods.

By discretized continuous distributions, category (ii) solutions, we refer to methods which use continuous distributions to create a discrete one. For example, let $F_W(w)$ to be the cumulative distribution function of a continuous random variable W defined in $\mathbb{R}^+$ then $f_Y(y) = F_W(y+1) - F_W(y)$ is a discrete distribution defined on $y = 0, 1, 2, \dots, \infty$. Alternatively let $f_Y(0) = F_W(.5)$ and $f_Y(y) = F_W(y + 0.5) - F_W(y - 0.5)$ for $y = 1, 2, \dots, \infty$. Distributions of this kind can be fitted easily using the `gamlss.cens` package. One potential criticism of the above methods of generating discrete distributions is the fact that if the parameter μ_W is the mean of the continuous random variable W , then the mean of the discrete random variable Y will not in general be exactly μ_W . Another example of a discretized continuous distribution is the gamma count distribution described in Winkelmann (1997) pp. 47-51. The distribution is derived from a random process where Y is modelled as the total number of events occurring within a fixed time interval, where the intervals between events have a gamma distribution, see Lindsey (1999) pp 30-31.

Note that both methods (i) and (ii) described above can cope with underdispersion as well as overdispersion in count data. Category (iii) solutions described below can only deal with overdispersion.

The random effect at the observation level, category (iii), solutions account for the overdispersion by including an extra random effect variable. They generally assume that, given a random effect variable γ , the response variable Y has a discrete conditional probability function $f(y|\gamma)$ and marginally γ has probability (density) function $f_\gamma(\gamma)$. Then the marginal probability function of Y is given by $f_Y(y) = \int f(y|\gamma)f_\gamma(\gamma)d\gamma$. Within the random effect at the observation level models, category (iii) above, we distinguish three different types:

- (a) when an explicit continuous mixture distribution, $f_Y(y)$, exists.
- (b) when a continuous mixture distribution, $f_Y(y)$, is not explicit but is approximated by integrating out the random effect using approximations, e.g. Gaussian quadrature or Laplace approximation.
- (c) when a 'non-parametric' mixture (effectively a finite mixture) is assumed for the response variable.

In this chapter we shall mainly concentrate on models of type (a). Assume that the conditional distribution of Y given γ is Poisson, i.e. $f(y|\gamma) = PO(\mu\gamma)$. Table 5.1 shows a variety of (marginal) overdispersed Poisson count data distributions, $f_Y(y)$ used in applied statistics and their corresponding mixing distribution $f_\gamma(\gamma)$. The last two distribution of Table 5.1 are not explicitly implemented yet in the `gamlss` packages. Note that zero inflated mixing distribution lead to zero inflated marginal distribution (as shown in Table 5.1 for the gamma distribution).

Table 5.1: Discrete gamlss family distributions for count data (derived from Poisson mixtures)

Distributions	R Name	mixing distribution for γ
Poisson	$PO(\mu)$	-
Negative binomial type I	$NBI(\mu, \sigma)$	$GA(1, \sigma^{\frac{1}{2}})$
Negative binomial type II	$NBII(\mu, \sigma)$	$GA(1, \sigma^{\frac{1}{2}}/\mu)$
Poisson-inverse Gaussian	$PIG(\mu, \sigma)$	$IG(1, \sigma^{\frac{1}{2}})$
Sichel	$SICHEL(\mu, \sigma, \nu)$	$GIG(1, \sigma^{\frac{1}{2}}, \nu)$
Delaporte	$DEL(\mu, \sigma, \nu)$	$SG(1, \sigma^{\frac{1}{2}}, \nu)$
Zero inflated Poisson	$ZIP(\mu, \sigma)$	$BI(1, 1 - \sigma)$
Zero inflated Poisson 2	$ZIP2(\mu, \sigma)$	$(1 - \sigma)^{-1}BI(1, 1 - \sigma)$
Zero inflated neg. binomial	-	zero inflated gamma
Poisson-Tweedie	-	Tweedie family

Note also that the probability function for the Poisson-Tweedie is given by Hougaard *et al.* (1997).

Hinde (1982) was the first to apply the Gaussian quadrature approach of type (b) to approximate a Poisson-normal mixture. Lee and Nelder (1996, 2006) use the Laplace approximation approach in their hierarchical and double hierarchical models. However their conditional distribution $f(y|\gamma)$ is, in general, *not* a proper distribution and consequently their marginal likelihood function is not a proper likelihood function, so we do not consider these methods further here.

Aitkin (1996, 1999) is an advocate of the "non-parametric" mixture approach of type (c) where the Poisson distribution is mixed with a non-parametric distribution. The non-parametric distribution involves unknown mass points and probabilities which have to be estimated from the data. This approach is referred to as "non-parametric maximum likelihood".

All the types of observation level random effect models in category (iii) above can be fitted using **gamlss** packages. The software also generalizes type (b) and (c) models above by allowing more general conditional distributions $f(y|\gamma)$ to be used rather than Poisson, e.g. a negative binomial distribution (resulting in a negative binomial-normal mixture model and a negative binomial non-parametric mixture model for Y respectively).

The next section discusses the explicit continuous mixture distributions in more detail.

5.2 Explicit continuous mixture distributions

Suppose, given a random variable γ , that Y has a Poisson distribution with mean $\mu\gamma$, i.e. $Y|\gamma \sim PO(\mu\gamma)$, where $\mu > 0$, and suppose that γ has probability density function $f_\gamma(\gamma)$ defined on $\mathbb{R}^+$, then the (marginal) distribution of Y is a mixed Poisson distribution. Provided γ has mean 1, then Y has mean μ . The model can be considered as a multiplicative Poisson random effect model, provided the distribution of γ does not depend on μ .

Many parameterizations of mixed Poisson distributions [e.g. the Sichel and Delaporte distributions, see Johnson, Kotz and Kemp (2005) and Wimmer and Altmann (1999)] for a discrete count random variable Y have been defined such that none of the parameters of the distribution is the mean of Y , and indeed the mean of Y is often a complex function of the distribution parameters, making the distribution difficult to use for regression models.

Table 5.2: Discrete gamlss family distributions for count data

Distributions	R Name	params	mean	variance
Poisson	$PO(\mu)$	1	μ	μ
Negative binomial type I	$NBI(\mu, \sigma)$	2	μ	$\mu + \sigma\mu^2$
Negative binomial type II	$NBII(\mu, \sigma)$	2	μ	$\mu + \sigma\mu$
Poisson-inverse Gaussian	$PIG(\mu, \sigma)$	2	μ	$\mu + \sigma\mu^2$
Sichel	$SICHEL(\mu, \sigma, \nu)$	3	μ	$\mu + h(\sigma, \nu)\mu^2$
Delaporte	$DEL(\mu, \sigma, \nu)$	3	μ	$\mu + \sigma(1 - \nu)^2\mu^2$
Zero inflated Poisson	$ZIP(\mu, \sigma)$	2	$(1 - \sigma)\mu$	$(1 - \sigma)\mu + \sigma(1 - \sigma)\mu^2$
Zero inflated Poisson type 2	$ZIP2(\mu, \sigma)$	2	μ	$\mu + \frac{\sigma}{(1 - \sigma)}\mu^2$

Here we consider several mixed Poisson distribution defined so that the mean μ is a parameter of the distribution. This allows easier interpretation of models for μ and generally provides a more orthogonal parameterization.

Specifically the following distributions with mean exactly equal to μ are considered: Poisson, negative binomial type I and type II, Poisson-inverse Gaussian, Sichel and Delaporte distributions. The distributions are continuous mixtures of Poisson distributions.

Table 5.2 shows the distributions for count data currently available in **gamlss** packages, together with their R name within **gamlss**, their number of parameters, mean, variance. Table 5.1 shows the mixing distribution for γ . The probability functions for all the distributions in Tables 5.2 and 5.1 are given in the Appendix A, (except for SG distribution defined later in this section).

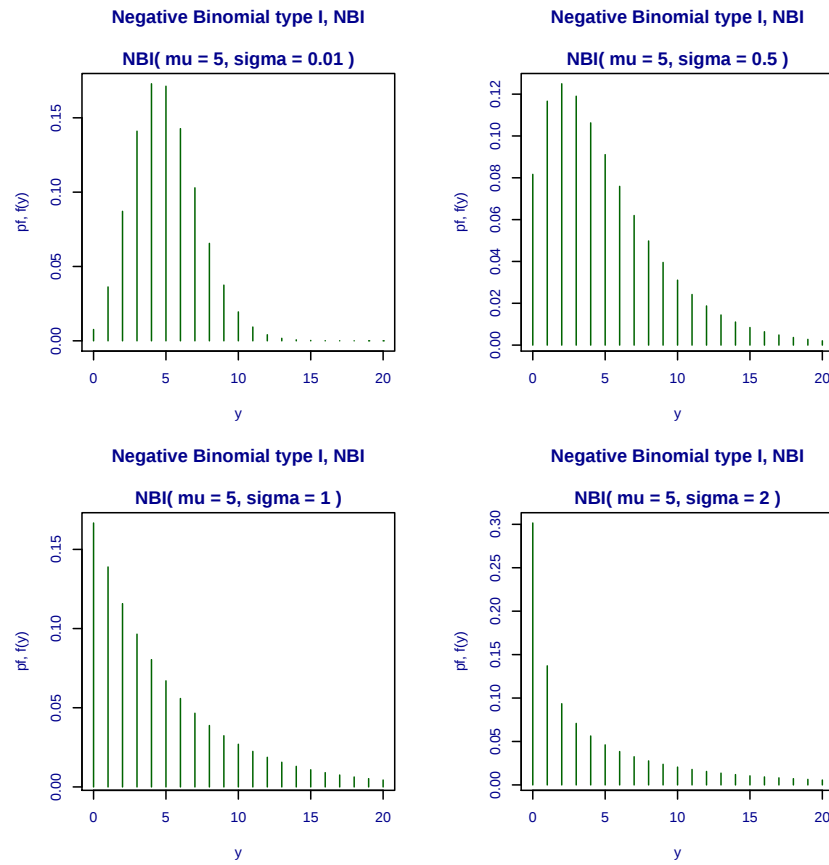
In Table 5.2 $\mu > 0$ and $\sigma > 0$ for all distributions, while $-\infty < \nu < \infty$ for the Sichel distribution and $0 < \nu < 1$ for the Delaporte distribution.

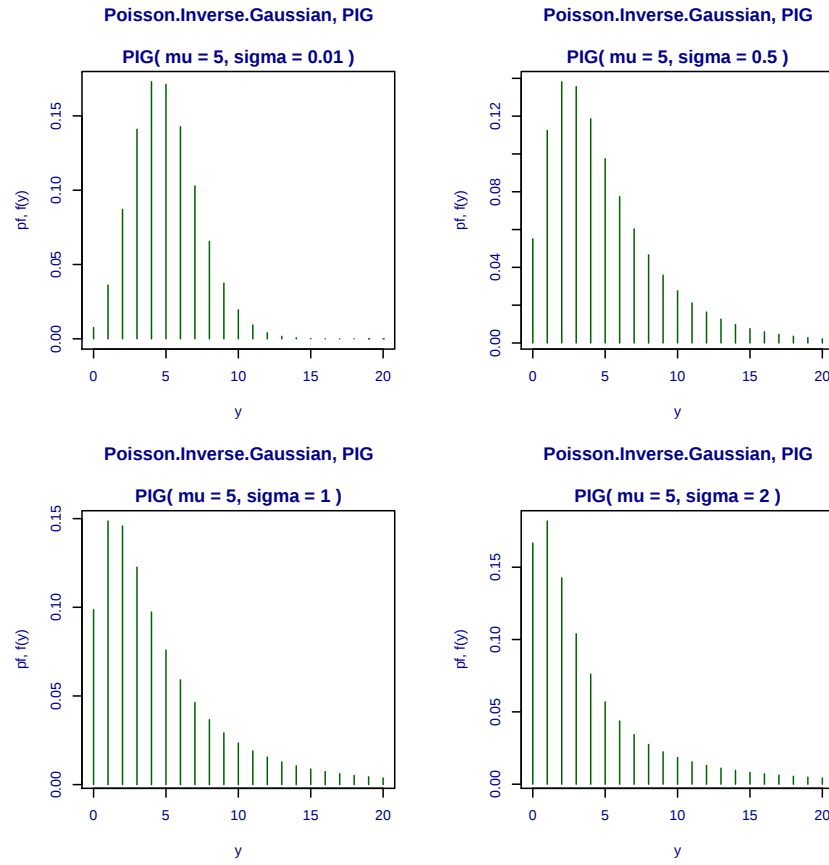
The Poisson-inverse Gaussian (PIG) is a special case of the Sichel where $\nu = -0.5$. The Poisson is a limiting case of the other distributions as $\sigma \rightarrow 0$.

5.2.1 Negative binomial distribution

The negative binomial type I distribution (denoted NBI in **gamlss** package is a mixed Poisson distribution obtained as the marginal distribution of Y when $Y|\gamma \sim PO(\mu\gamma)$ and $\gamma \sim GA(1, \sigma^{\frac{1}{2}})$, i.e. γ has a gamma distribution with mean 1 and scale parameter $\sigma^{\frac{1}{2}}$ (and hence has dispersion σ). Figure 5.1 plots the negative binomial type I distribution, $NBI(\mu, \sigma)$, for $\mu = 5$ and $\sigma = (0.01, 0.5, 1, 2)$. [The plot was created using the command `pdf.plot(family="NBI", mu=5, sigma=c(0.01, 0.5, 1, 2), min=0, max=20, step=1)`.] Note that plot for $\sigma = 0.01$ is close to a Poisson, $PO(5)$, distribution which corresponds to $\mu = 5$ and $\sigma \rightarrow 0$ in the $NBI(\mu, \sigma)$ distribution.

The negative binomial type II distribution (denoted NBII in **gamlss**) is a mixed Poisson distribution obtained as the marginal distribution of Y when $Y|\gamma \sim PO(\mu\gamma)$ and $\gamma \sim GA(1, (\sigma/\mu)^{\frac{1}{2}})$.

Figure 5.1: NBI distribution for $\mu = 5$ and $\sigma = (0.01, 0.5, 1, 2)$

Figure 5.2: PIG distribution for $\mu = 5$ and $\sigma = (0.01, 0.5, 1, 2)$

This is a reparameterization of the NBI distribution obtained by replacing σ by σ/μ .

The NBI and NBII models differ when there are explanatory variables for μ and/or σ . The negative binomial distribution can be highly positively skewed, unlike the Poisson distribution, which is close to symmetric for moderate μ and even closer as μ increases. The extra σ parameter allows the variance to change for a fixed mean, unlike the Poisson distribution for which the variance is fixed equal to the mean. Hence the negative binomial allows modelling of the variance as well as of the mean.

5.2.2 Poisson-inverse Gaussian

The Poisson-inverse Gaussian distribution (denoted **PIG** in **gamlss**) is a mixed Poisson distribution obtained as the marginal distribution of Y when $Y|\gamma \sim PO(\mu\gamma)$ and $\gamma \sim IG(1, \sigma^{\frac{1}{2}})$, an inverse Gaussian mixing distribution. This allows for even higher skewness, i.e. longer upper tail, than the negative binomial distribution. Figure 5.2 plots the Poisson-inverse Gaussian distribution, $PIG(\mu, \sigma)$, for $\mu = 5$ and $\sigma = (0.01, 0.5, 1, 2)$. Note that plot for $\sigma = 0.01$ is close to a Poisson, $PO(5)$, distribution. [The plot was created using the command `pdf.plot(family="NBI", mu=5, sigma=c(0.01, 0.5, 1, 2), min=0, max=20, step=1)`

5.2.3 Sichel distribution

The Sichel distribution has been found to provide a useful three parameter model for over-dispersed Poisson count data exhibiting high positive skewness, e.g. Sichel (1992). In the parametrization below μ is the mean of the Sichel distribution, while the two remaining parameters σ and ν jointly define the scale and shape of the Sichel distribution. In particular the three parameters of the Sichel allow **different** shapes (in particular the level of positive skewness) of the distribution for a fixed mean and variance, unlike the Poisson, negative binomial and Poisson-inverse Gaussian distributions. The Sichel distribution therefore allows modelling of the mean, variance **and** skewness.

The Sichel distribution (denoted **SICHEL** in **gamlss**) is a mixed Poisson distribution obtained as the marginal distribution of Y when $Y|\gamma \sim PO(\mu\gamma)$ and $\gamma \sim GIG(1, \sigma^{\frac{1}{2}}, \nu)$, a generalized inverse Gaussian mixing distribution with probability density function given by

$$f_{\gamma}(\gamma) = \frac{c^{\nu} \gamma^{\nu-1}}{2K_{\nu}(\frac{1}{\sigma})} \exp \left[-\frac{1}{2\sigma} \left(c\gamma + \frac{1}{c\gamma} \right) \right] \quad (5.1)$$

for $\gamma > 0$, where $\sigma > 0$ and $-\infty < \nu < \infty$.

This parameterization of the GIG ensures that $E[\gamma] = 1$. The mean and variance of Y , are given by $E[Y] = \mu$ and $V(Y) = \mu + \mu^2 [2\sigma(\nu + 1)/c + 1/c^2 - 1]$ respectively. For the Sichel distribution $h(\sigma, \nu) = [2\sigma(\nu + 1)/c + 1/c^2 - 1]$ in Table 5.2 where $c = R_{\nu}(1/\sigma)$ and $R_{\lambda}(t) = K_{\lambda+1}(t)/K_{\lambda}(t)$ and $K_{\lambda}(t) = \frac{1}{2} \int_0^{\infty} x^{\lambda-1} \exp[-\frac{1}{2}t(x + x^{-1})] dx$ is the modified Bessel function of the third kind.

5.2.4 Delaporte distribution

The Delaporte distribution (denoted **DEL** in **gamlss**) is a mixed Poisson distribution obtained as the marginal distribution of Y when $Y|\gamma \sim PO(\mu\gamma)$ and $\gamma \sim SG(1, \sigma^{\frac{1}{2}}, \nu)$, a shifted gamma mixing distribution with probability density function given by

$$f_{\gamma}(\gamma) = \frac{(\gamma - \nu)^{\frac{1}{\sigma} - 1}}{\sigma^{1/\sigma}(1 - \nu)^{1/\sigma}\Gamma(1/\sigma)} \exp \left[-\frac{(\gamma - \nu)}{\sigma(1 - \nu)} \right] \quad (5.2)$$

for $\gamma > \nu$, where $\sigma > 0$ and $0 \leq \nu < 1$. This parameterization ensures that $E[\gamma] = 1$.

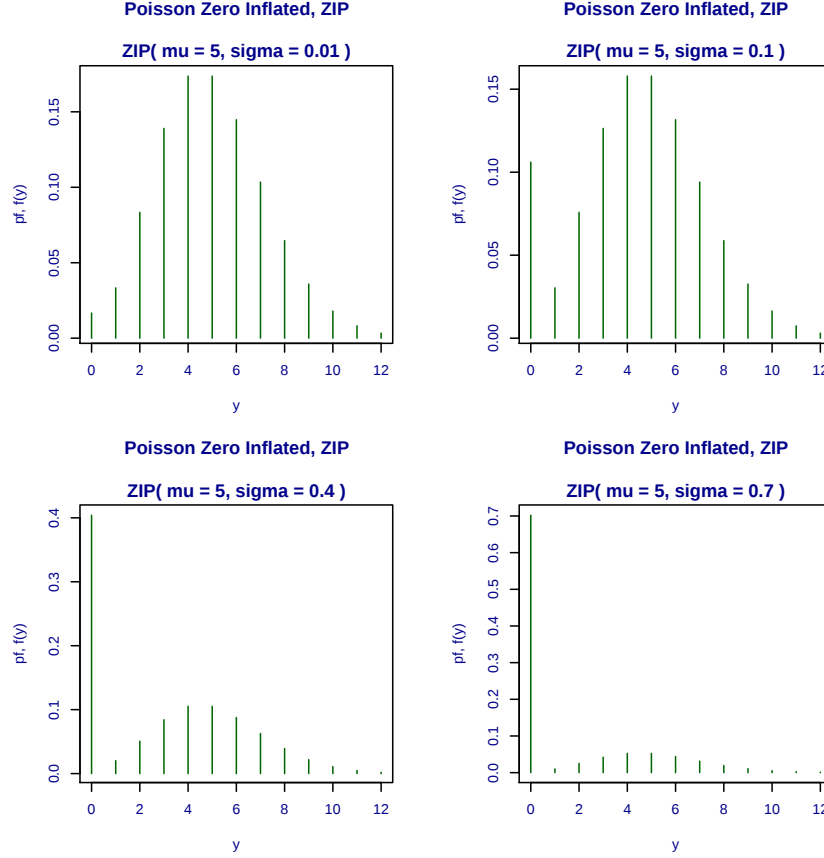


Figure 5.3: ZIP distribution for $\mu = 5$ and $\sigma = (0.01, 0.1, 0.4, 0.7)$

5.2.5 Zero inflated Poisson

The zero inflated Poisson distribution (denoted ZIP in **gamlss**) is a discrete mixture of two components: value 0 with probability σ and a Poisson distribution with mean μ with probability $1 - \sigma$.

This can be viewed as a discrete mixed Poisson distribution defined by the marginal distribution of Y where $Y|\gamma \sim PO(\mu\gamma)$ and $\gamma \sim BI(1, 1 - \sigma)$, i.e. $\gamma = 0$ with probability σ and $\gamma = 1$ with probability $1 - \sigma$. Note however that γ has mean $1 - \sigma$ in this formulation and Y has mean $\mu/(1 - \sigma)$.

An alternative parametrization, the zero inflated Poisson type 2 distribution (denoted in **gamlss** as ZIP2) is the marginal distribution for Y where $Y|\gamma \sim PO(\mu\gamma)$ and $\gamma \sim (1 - \sigma)^{-1}BI(1, 1 - \sigma)$. Hence γ has mean 1 and Y has mean μ .

Figure 5.3 plots the zero inflated Poisson distribution, $\text{ZIP}(\mu, \sigma)$, for $\mu = 5$ and $\sigma = (0.01, 0.1, .4, 0.7)$. Note that plot for $\sigma = 0.01$ is close to a Poisson, $\text{PO}(5)$, distribution. [The plot was created using the command `pdf.plot(family="ZIP", mu=5, sigma=c(0.01, 0.1, 0.4, 0.7), min=0, max=12, step=1).`]

5.2.6 Comparison of the marginal distributions

Marginal distributions for Y can be compared using a (ratio moment) diagram of their skewness and kurtosis, given in the Appendix to this chapter and obtained from Appendix B of Rigby *et al.* (2008). Figure 5.4 displays the skewness-kurtosis combinations for different marginal distributions of Y , where Y has fixed mean 1 and fixed variance 2.

The zero-inflated Poisson (ZIP), negative binomial (NB) and Poisson-inverse Gaussian (PIG) distributions each have two parameters, so fixing the mean and variance of Y results in a single combination of skewness-kurtosis, displayed as a circle.

The Sichel, Poisson-Tweedie and Delaporte distributions each have three parameters, so their possible skewness-kurtosis combinations are represented by curves. The three curves meet at the skewness-kurtosis point of the negative binomial which is a limiting case of the Sichel, an internal special case of the Poisson-Tweedie and a boundary special case of the Delaporte. The Poisson-Tweedie curve alone continues (as its power parameter decreases from two to one) and stops at the circle between ZIP and NB. [Note also that the PIG is a special case of both the Sichel and the Poisson-Tweedie distributions.]

The zero-inflated negative binomial distribution (ZINB) skewness-kurtosis curve (shown in Figure 5.4 but not labeled) is the line from the skewness-kurtosis of the ZIP to that of the NB. The zero-inflated Poisson reciprocal Gamma (ZIPRG) curve has the highest kurtosis for a given skewness.

The Poisson-shifted generalized inverse Gaussian (PSGIG) is a four parameter distribution and has skewness-kurtosis combinations covering the region between the Sichel and Delaporte curves, while the zero-inflated Sichel (ZISichel) covers the region between the ZIPRG and Sichel curves. Similar figures were obtained for other combinations of fixed mean and variance of Y .

5.2.7 Families modelling the variance-mean relationship

The multiplicative Poisson random effect model defined in Section 5.2 leads to a variance-mean relationship for Y given by $V[Y] = \mu + \mu^2 V[\gamma]$ where in general $V[\gamma] = v(\sigma, \nu, \tau)$ is a function of the parameters σ , ν and τ of the mixing distribution $f_\gamma(\gamma)$. Hence in particular the negative binomial type I, the Poisson-inverse Gaussian, Sichel, Delaporte and PSGIG distributions all have this quadratic variance-mean relationship. Alternative variance-mean relationships can be obtained by reparametrization. [Note, however that Theorem 1 in the Appendix A of Rigby *et al.* (2008) will no longer hold, since the resulting $f_\gamma(\gamma)$ depends on μ .]

For example consider the negative binomial type I distribution with probability function given by

$$p_Y(y) = \frac{\Gamma(y + \frac{1}{\sigma})}{\Gamma(\frac{1}{\sigma})\Gamma(y+1)} (\sigma\mu)^y (1 + \sigma\mu)^{-(y+1/\sigma)} \quad (5.3)$$

for $y = 0, 1, 2, \dots$ where $\mu > 0$ and $\sigma > 0$ with mean μ and variance $V[Y] = \mu + \sigma\mu^2$. If σ is reparameterized to σ_1/μ then $V[Y] = (1 + \sigma_1)\mu$ giving a negative binomial type II distribution. If σ is reparameterized to $\sigma_1\mu$ then $V[Y] = \mu + \sigma_1\mu^3$. Note that modelling σ as function of explanatory variables models the excess variance on top of the variance-mean relationship.

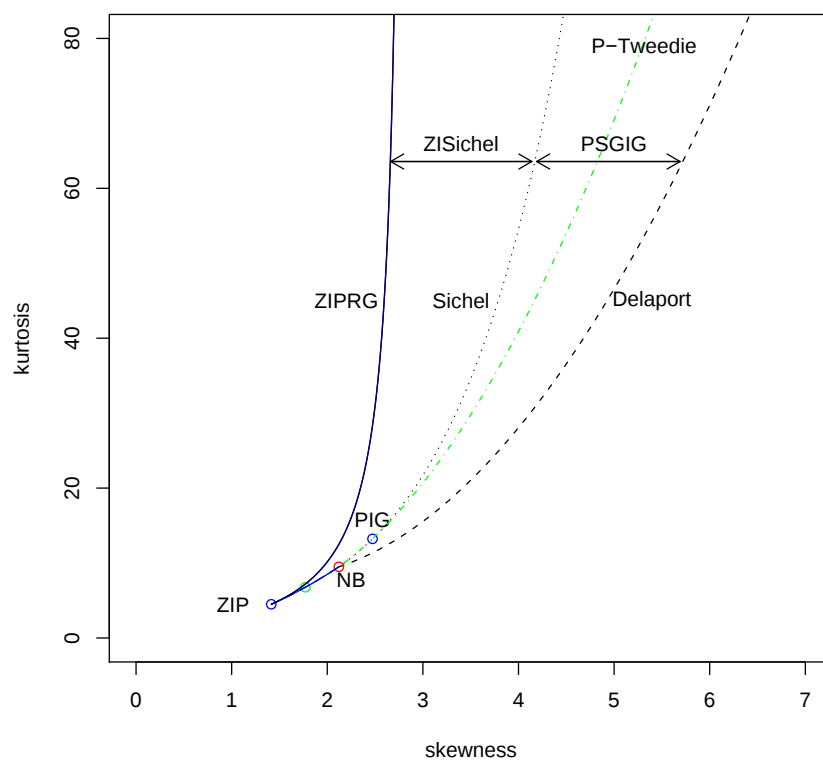


Figure 5.4: Skewness-kurtosis combinations for different distributions for Y (for fixed mean 1 and variance 2 for Y)

More generally a family of reparameterizations of the negative binomial type I distribution can be obtained by reparameterizing σ to $\sigma_1 \mu^{\nu-2}$ giving $V(Y) = \mu + \sigma_1 \mu^\nu$. This gives a three parameter model with parameters μ , σ_1 and ν . The model can be fitted by maximum likelihood estimation. Note that a family of reparameterizations can be applied to any multiplicative Poisson random effect model as defined in Section 5.2. In particular the Poisson-inverse Gaussian, Sichel, Delaporte and PSGIG, can all be extended to reparameterization families using an extra parameter.

5.3 Examples: fitting a distribution

5.3.1 The computer failure data

Data summary:

R data file: computer in package **gamlss.dist** of dimensions 128×1

source: Hand *et al.* (1994)

variables

failure : the number of computers that broke down.

purpose: to demonstrate the fitting of a parametric discrete distribution to the data.

conclusion a PIG distribution fits best

The following data relate to DEC-20 computers which operated at the Open University in the 1980. They give the number of computers that broke down in each of the 128 consecutive weeks of operation, starting in late 1983, see Hand *et al.* (1994) page 109 data set 141. Here we use four different count data distributions and choose between them using the Akaike information criterion (AIC):

```
> graphics.off()

> library(gamlss.dist)
> data(computer)
> op <- par(mfrow = c(2, 2))
> mPO <- histDist(computer$failure, "PO", main = "PO", trace = FALSE)
> mNBI <- histDist(computer$failure, "NBI", main = "NBI", trace = FALSE)
> mPIG <- histDist(computer$failure, "PIG", main = "PIG", trace = FALSE)
> mSI <- histDist(computer$failure, "SICHEL", main = "SICHEL",
+   trace = FALSE)
> AIC(mPO, mNBI, mPIG, mSI)

      df      AIC
mPIG  2 636.4159
mNBI  2 636.8405
mSI   3 638.0551
mPO   1 771.9487

> par(op)
```

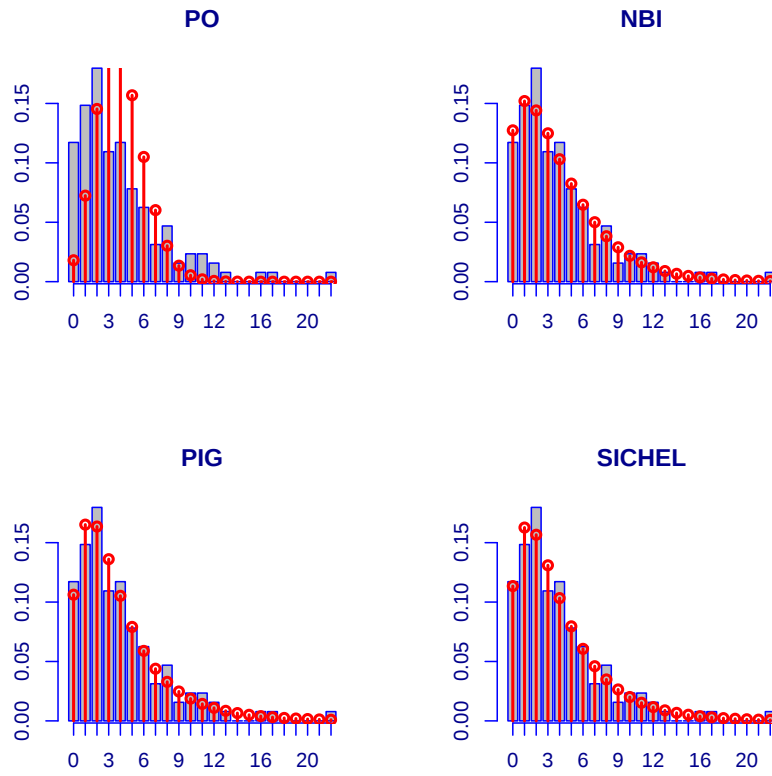


Figure 5.5: The computer failure data fit with (a) Poisson , (b) negative binomial (c) Poisson inverse gaussian and (d) Sichel distributions respectively

From the GAIC table above we conclude that the PIG model is the appropriate model. Now we refit the model and display a summary of the final model.

```
> mod1 <- gamlss(failure ~ 1, data = computer, family = PIG, trace = FALSE)
> summary(mod1)
```

```
*****
Family:  c("PIG", "Poisson.Inverse.Gaussian")

Call:  gamlss(formula = failure ~ 1, family = PIG, data = computer,
             trace = FALSE)

Fitting method: RS()

-----
Mu link function:  log
Mu Coefficients:
              Estimate Std. Error  t value  Pr(>|t|)
(Intercept)    1.390      0.08355   16.64  1.058e-33

-----
Sigma link function:  log
Sigma Coefficients:
              Estimate Std. Error  t value  Pr(>|t|)
(Intercept)  -0.3489      0.2387   -1.461   0.1464

-----
No. of observations in the fit: 128
Degrees of Freedom for the fit: 2
      Residual Deg. of Freedom: 126
                        at cycle: 3

Global Deviance:    632.4159
              AIC:    636.4159
              SBC:    642.12
*****
```

Hence the fitted PIG model for the computer failure data is given by $Y \sim PIG(\hat{\mu}, \hat{\sigma})$ where $\hat{\mu} = \exp(1.390) = 4.015$ and $\hat{\sigma} = \exp(-0.3489) = 0.7055$, with fitted mean $\hat{E}(Y) = \hat{\mu} = 4.015$ and fitted variance $\hat{V}(Y) = \hat{\mu} + \hat{\sigma}\hat{\mu}^2 = 15.39$.

5.3.2 The lice data

Data summary:

R data file: lice in package **gamlss.dist** of dimensions 71×2

source: Williams (1944)

variables

head : the number of head lice

freq : the frequency of prisoners with the number of head lice

purpose: to demonstrate the fitting of a parametric discrete distribution to the data.

conclusion a SICHEL distributions fits best

The following data come from Williams (1944) and they are frequencies (**freq**) of prisoners with number of head lice (**head**), for Hindu male prisoners in Cannamore, South India, 1937-1939. We fit four different distributions to **head** and choose between them using AIC:

```
> library(gamlss.dist)
> con <- gamlss.control(trace = FALSE, n.cyc = 50)
> data(lice)
> mPO <- gamlss(head ~ 1, data = lice, family = PO, weights = freq,
+   trace = FALSE)
> mNBI <- gamlss(head ~ 1, data = lice, family = NBI, weights = freq,
+   trace = FALSE)
> mPIG <- gamlss(head ~ 1, data = lice, family = PIG, weights = freq,
+   trace = FALSE)
> mSI <- gamlss(head ~ 1, data = lice, family = SICHEL, weights = freq,
+   n.cyc = 50, trace = FALSE)
> AIC(mPO, mNBI, mPIG, mSI)
```

	df	AIC
mSI	3	4646.214
mNBI	2	4653.687
mPIG	2	4756.275
mPO	1	29174.823

We conclude that the Sichel model explains the data best. The summary of the final fitted model is shown below:

```
> summary(mSI)

*****
Family:  c("SICHEL", "Sichel")

Call:
gamlss(formula = head ~ 1, family = SICHEL, data = lice, weights = freq,
  n.cyc = 50, trace = FALSE)
```

```

Fitting method: RS()

-----
Mu link function:  log
Mu Coefficients:
              Estimate Std. Error t value    Pr(>|t|)
(Intercept)    1.927     0.07952   24.23 5.965e-104

-----
Sigma link function:  log
Sigma Coefficients:
              Estimate Std. Error t value    Pr(>|t|)
(Intercept)    4.806     0.2034   23.63 6.974e-100

-----
Nu link function:  identity
Nu Coefficients:
              Estimate Std. Error t value    Pr(>|t|)
(Intercept) -0.004787     0.01595  -0.3002    0.7641

-----
No. of observations in the fit: 1083
Degrees of Freedom for the fit: 3
      Residual Deg. of Freedom: 1080
                        at cycle: 21

Global Deviance:    4640.214
      AIC:          4646.214
      SBC:          4661.177
*****

```

Hence the fitted SICHEL distribution for the number of head lice ($Y = \text{head}$) is given by $Y \sim \text{SICHEL}(\hat{\mu}, \hat{\sigma}, \hat{\nu})$ where $\hat{\mu} = \exp(1.927) = 6.869$ and $\hat{\sigma} = \exp(4.806) = 122.24$ and $\hat{\nu} = -0.0047$, with fitted mean $\hat{E}(Y) = \hat{\mu} = 6.869$ and fitted variance $\hat{V}(Y) = 432.25$, (obtained using the code `VSICHEL(mSI)[1]`). Figure 5.6 shows the fitted negative binomial and Sichel models created by the following R commands. Note that Figure 5.6 only plots the data and fitted distributions up to $y = 10$.

```

> op <- par(mfrow = c(2, 1))
> m1 <- histDist(lice$head, "NBI", freq = lice$freq, xlim = c(0,
+ 10), main = "NBI distribution", trace = FALSE)
> m2 <- histDist(lice$head, "SICHEL", freq = lice$freq, xlim = c(0,
+ 10), main = "Sichel distribution", trace = FALSE, n.cyc = 50)
> par(op)

```

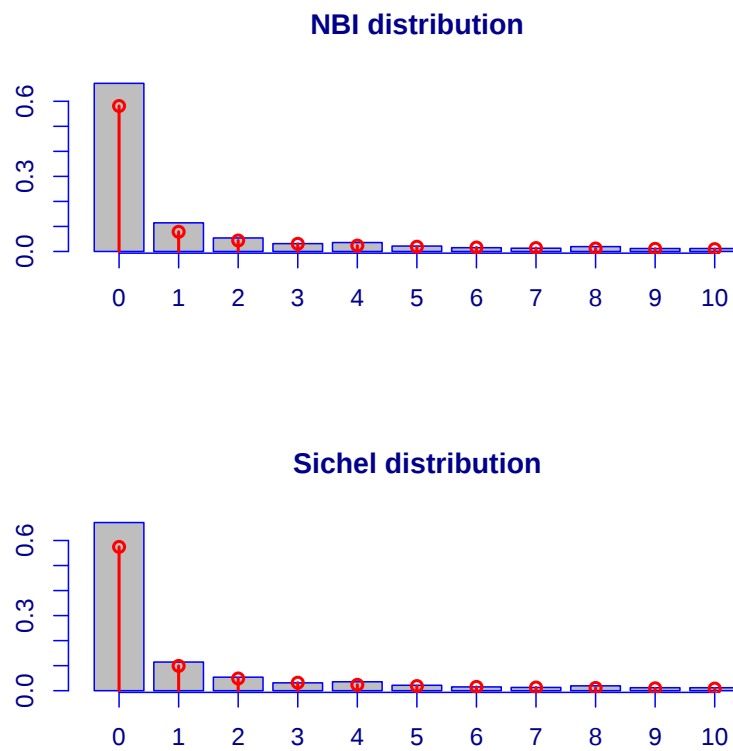


Figure 5.6: The lice data with the fitted (a) negative binomial and (b) Sichel distributions respectively

5.3.3 A stylometric application

Data summary:

R data file: `stylo` in package `gamlss.dist` of dimensions 64×2

source: Dr Mario Corina-Borja

variables

word : is the number of times a word appears in a single text

freq : the frequency of the number of times a word appears in a text

purpose: to demonstrate the fitting of a truncated discrete distribution to the data.

conclusion the truncated SICHEL distributions fits best

The data are from a stylometric application, the discipline which tries to characterize the style of a text, Chappas and Corina-Borja (2006). Here the response variable, **word**, is the number of times a word appears in a single text. The variable **freq** records the frequency of the **word** (i.e. frequency **freq** is the number of different words which occur exactly **word** times in the text). Possible values of **word** are $y = 1, 2, 3, \dots$, and since the objective here is to fit an appropriate distribution to the data we are looking for a zero truncated discrete distribution. In the specific data we are using, the maximum times that any word appears in the text is 64 (**word=64**), while the most frequent value of **word** is 1 with frequency 947. We first input and plot the data:

```
> library(gamlss.dist)
> library(gamlss.tr)
> data(stylo)
> plot(freq ~ word, data = stylo, type = "h", xlim = c(0, 22),
+       xlab = "no of times", ylab = "frequencies", col = "blue")
```

Note that for plotting we restrict the upper x-limit to 22 since the most of the frequencies after that have zero values. We will now generate several truncated discrete distributions, using the function `gen.trunc()` from the package `gamlss.tr`, to fit them to the data. Specifically we generate i) truncated Poisson, ii) truncated negative binomial type II iii) truncated Depalorte and iv) truncated Sichel. [Note that the truncated negative binomial type I model takes more than 300 iterations to converge and eventually give the same result as the truncated negative binomial type II.]

```
> library(gamlss.tr)
> gen.trun(par = 0, family = PO, type = "left")
```

A truncated family of distributions from PO has been generated
and saved under the names:

`dP0tr` `pP0tr` `qP0tr` `rP0tr` `P0tr`

The type of truncation is left and the truncation parameter is 0

```
> gen.trun(par = 0, family = NBII, type = "left")
```

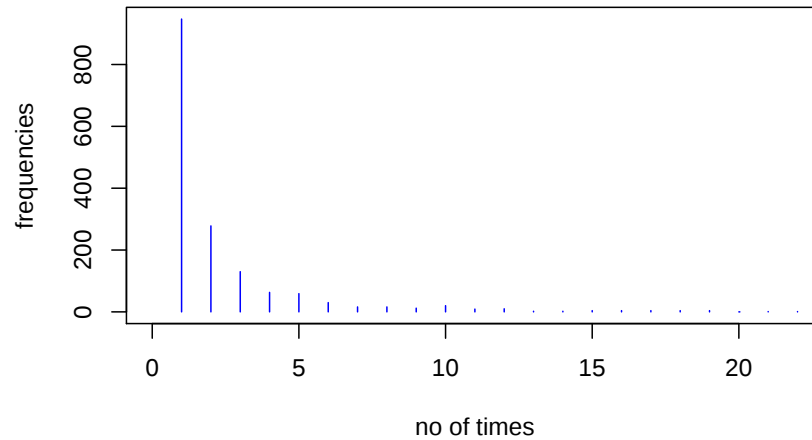


Figure 5.7: The stylometric data: number of time a word appear in a text against the frequencies

A truncated family of distributions from NBII has been generated and saved under the names:

```
dNBIItr pNBIItr qNBIItr rNBIItr NBIItr
```

The type of truncation is left and the truncation parameter is 0

```
> gen.trun(par = 0, family = DEL, type = "left")
```

A truncated family of distributions from DEL has been generated and saved under the names:

```
dDELtr pDELtr qDELtr rDELtr DELtr
```

The type of truncation is left and the truncation parameter is 0

```
> gen.trun(par = 0, family = SICHEL, type = "left", delta = 0.001)
```

A truncated family of distributions from SICHEL has been generated and saved under the names:

```
dSICHELtr pSICHELtr qSICHELtr rSICHELtr SICHELtr
```

The type of truncation is left and the truncation parameter is 0

We new fit the distributions to the data and choose betwing them using AIC:

```
> mPO <- gamlss(word ~ 1, weights = freq, data = stylo, family = P0tr,
+   trace = FALSE)
> mNBII <- gamlss(word ~ 1, weights = freq, data = stylo, family = NBIItr,
+   n.cyc = 50, trace = FALSE)
> mDEL <- gamlss(word ~ 1, weights = freq, data = stylo, family = DELtr,
+   n.cyc = 50, trace = FALSE)
> mSI <- gamlss(word ~ 1, weights = freq, data = stylo, family = SICHELtr,
```

```
+      n.cyc = 50, trace = FALSE)
> GAIC(mPO, mNBII, mDEL, mSI)
```

	df	AIC
mSI	3	5148.454
mDEL	3	5160.581
mNBII	2	5311.627
mPO	1	9207.459

The best fitted model according to the AIC is the truncated Sichel model. The Depalorte model performed better than the negative binomial type II model. Figure 5.8 shows all of the fitted models above. The fit of the (zero truncated) Poisson distribution is shown in part (a) of the Figure 5.8. This is not a very good fit to the data and an improved fit is achieved by using the (truncated) negative binomial distribution II in part (b) and the Delaporte distribution in (c). The (truncated) Sichel in panel (d) is a superior fit according to both AIC and SBC. Figure 5.8 was produced using the following code:

```
> op <- par(mfrow = c(2, 2))
> tabley <- with(stylo, table(rep(word, freq)))
> mNO <- histDist(stylo$word, family = P0tr, freq = stylo$freq,
+   main = "(b) Poisson", ylim = c(0, 0.65), xlim = c(1, 26),
+   trace = FALSE)
> mNBII <- histDist(stylo$word, family = NBIItr, freq = stylo$freq,
+   main = "(c) negative binomial II", ylim = c(0, 0.65), xlim = c(1,
+   26), start.from = mNBII, trace = FALSE)
> mDEL <- histDist(stylo$word, family = DELtr, freq = stylo$freq,
+   main = "(c) Delaporte", ylim = c(0, 0.65), xlim = c(1, 26),
+   start.from = mDEL, trace = FALSE)
> mSI <- histDist(stylo$word, family = SICHELtr, freq = stylo$freq,
+   main = "(d) Sichel", ylim = c(0, 0.65), xlim = c(1, 26),
+   start.from = mSI, trace = FALSE)
> par(op)
```

5.4 Examples: regression analysis

5.4.1 The fish species data

Data summary: the fish species data

R data file: species in package `gamlss.dist` of dimensions 70×2

variables

fish : the number of different species in 70 lakes in the world

lake : the lake area

purpose: to demonstrate the fitting of count data distributions

conclusion:

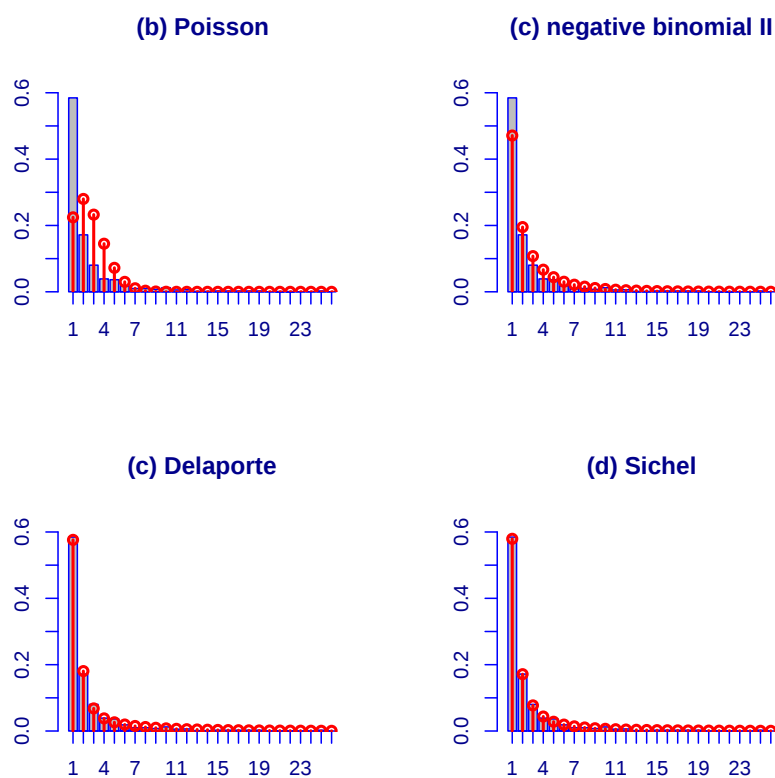


Figure 5.8: The stylometric data fits with (a) Poisson (b) negative binomial type II (c) Delaporte and (d) Sichel distributions respectively

The number of different fish species (**fish**) was recorded for 70 lakes of the world together with explanatory variable $x = \log(\text{lake})$, i.e. $x = \log$ lake area. The data are plotted in Figure 5.9.

```
> library(gamlss.dist)
> data(species)
> plot(fish ~ log(lake), data = species)
```

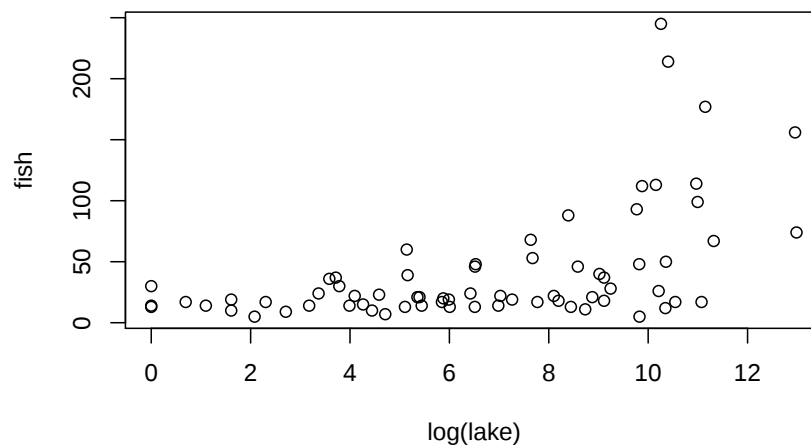


Figure 5.9: The fish species data

The data are given and analyzed by Stein and Juritz (1988) using a Poisson inverse Gaussian, $PIG(\mu, \sigma)$ distribution for **fish** with a linear model in $\log(\text{lake})$ for $\log \mu$ parameter and a constant for $\log \sigma$.

Rigby *et al.* (2008), when analyzing this data set, identified the following questions that need to be answered. Note that the same questions could apply to any regression type situation where the response variable is counts and where $\mathbf{x}$ represents a set explanatory variables.

- How does the mean of the response variable depend on $\mathbf{x}$?
- Is the response variable overdispersed Poisson?
- How does the variance of the response variable depend on its mean?
- What is the distribution of the response variable given $\mathbf{x}$?
- Do the scale and shape parameters of the response variable distribution depend on $\mathbf{x}$?

Here we will model the data using different discrete distributions and consider flexible models for the distributional parameters, where any or all of them can possibly depend on the explanatory variable $\log(\text{lake})$.

We start by fitting six different count distributions to the data [Poisson (PO), negative binomial type I and II (NBI, NBII), poisson inverse Gaussian (PIG), Delaporte (DEL) and Sichel (SICHEL)] using a first a linear and then a quadratic polynomial in $x = \log(\text{lake})$. The AIC of each model is then printed for comparison:

```
> species$x <- log(species$lake)
> fam <- c("PO", "NBI", "NBII", "PIG", "DEL", "SICHEL")
> m.l <- m.q <- list()
> for (i in 1:6) {
+   m.l[[fam[i]]] <- gamlss(fish ~ x, data = species, family = fam[i],
+     n.cyc = 60, trace = FALSE)$aic
+ }
> for (i in 1:6) {
+   m.q[[fam[i]]] <- GAIC(gamlss(fish ~ poly(x, 2), data = species,
+     family = fam[i], n.cyc = 60, trace = FALSE))
+ }
> unlist(m.l)
```

PO	NBI	NBII	PIG	DEL	SICHEL
1900.1562	625.8443	647.5359	623.4638	626.2330	625.4000

```
> unlist(m.q)
```

PO	NBI	NBII	PIG	DEL	SICHEL
1855.2965	622.3173	645.0129	621.3460	623.5816	623.1018

The Poisson model has a very large AIC compared to the rest of the distributions so we can conclude that the data are overdispersed. The quadratic polynomial in x seems to fit better than the linear term across the different count distributions. The best model at this stage is the Poisson inverse Gaussian (PIG) model with a quadratic polynomial in x . We now compare the AIC of a PIG model with a cubic smoothing spline in x instead of a quadratic polynomial in x . The total “effective” degrees of freedom for x in the cubic spline model (including the constant and linear term) is 5 compared to 3 in the quadratic model.

```
> GAIC(gamlss(fish ~ cs(x), data = species, family = PIG, trace = FALSE))
[1] 623.9339
```

The cubic smoothing spline does not seem to improve the model, so we keep the quadratic polynomial in x . We shall now try to model $\log(\sigma)$ as a linear function of x in the five remaining count distributions.

```
> fam <- c("NBI", "NBII", "PIG", "DEL", "SICHEL")
> m.q1 <- list()
> for (i in 1:5) {
+   m.q1[[fam[i]]] <- GAIC(gamlss(fish ~ poly(x, 2), data = species,
+     sigma.fo = ~x, family = fam[i], n.cyc = 60, trace = FALSE))
+ }
> unlist(m.q1)
```

NBI	NBII	PIG	DEL	SICHEL
614.9565	615.1250	612.3684	614.6059	613.7347

Modelling $\log(\sigma)$ as a linear function of x improves the AIC for all models. The PIG model is still the “best”. Since the Sichel and the Delaporte distributions have three parameters we will try to model the third parameter ν as a linear function of x . The Sichel uses the `identity` as the default link for ν while the Delaporte uses the `logit`.

```
> fam <- c("DEL", "SICHEL")
> m.qll <- list()
> for (i in 1:2) {
+   m.qll[[fam[i]]] <- GAIC(gamlss(fish ~ poly(x, 2), data = species,
+     sigma.fo = ~x, nu.fo = ~x, family = fam[i], n.cyc = 60,
+     trace = FALSE))
+ }
> unlist(m.qll)

      DEL      SICHEL
614.7376 611.6365
```

Modelling the ν as a linear function of x improves the Sichel model (which now has lower AIC than the PIG model) but not the Delaporte model. A further simplification of the Sichel model can be achieved by dropping the linear terms in x for the $\log(\sigma)$ model which given the linear model in x for ν does not seem to contribute anything to the fit (a least according to the AIC):

```
> GAIC(gamlss(fish ~ poly(x, 2), data = species, sigma.fo = ~1,
+   nu.fo = ~x, family = SICHEL, n.cyc = 60, trace = FALSE))

[1] 609.7299
```

The fitted parameters of the “best” Sichel model are shown below. They are obtained by refitting the model using this time an ordinary quadratic polynomial in x for $\log(\mu)$ model rather than the orthogonal quadratic polynomial produced by `poly(x,2)`:

```
> mSI <- gamlss(fish ~ x + I(x^2), sigma.fo = ~1, nu.fo = ~x, data = species,
+   family = SICHEL, n.cyc = 60)
```

```
GAMLSS-RS iteration 1: Global Deviance = 613.576
GAMLSS-RS iteration 2: Global Deviance = 602.8399
GAMLSS-RS iteration 3: Global Deviance = 598.3117
GAMLSS-RS iteration 4: Global Deviance = 597.7534
GAMLSS-RS iteration 5: Global Deviance = 597.74
GAMLSS-RS iteration 6: Global Deviance = 597.7313
GAMLSS-RS iteration 7: Global Deviance = 597.7275
GAMLSS-RS iteration 8: Global Deviance = 597.7275
```

```
> summary(mSI)
```

```
*****
Family:  c("SICHEL", "Sichel")
```

```
Call:
```

```
gamlss(formula = fish ~ x + I(x^2), sigma.formula = ~1, nu.formula = ~x,
```

```

family = SICHEL, data = species, n.cyc = 60)

Fitting method: RS()

-----
Mu link function:  log
Mu Coefficients:
              Estimate Std. Error  t value  Pr(>|t|)
(Intercept)   2.788804   0.173170  16.10446 1.001e-24
x             -0.006788   0.068115  -0.09966 9.209e-01
I(x^2)         0.013972   0.005782   2.41639 1.841e-02

-----
Sigma link function:  log
Sigma Coefficients:
              Estimate Std. Error  t value  Pr(>|t|)
(Intercept)    0.3693     0.4361   0.8468    0.4

-----
Nu link function:  identity
Nu Coefficients:
              Estimate Std. Error  t value  Pr(>|t|)
(Intercept)  -10.843     3.3234  -3.263   0.001728
x              1.048     0.3535   2.965   0.004166

-----
No. of observations in the fit:  70
Degrees of Freedom for the fit:  6
      Residual Deg. of Freedom: 64
                        at cycle: 8

Global Deviance:    597.7275
      AIC:          609.7275
      SBC:          623.2185
*****

> plot(fish ~ log(lake), data = species)
> lines(species$x[order(species$lake)], fitted(mSI)[order(species$lake)],
+       col = "red")

```

The fitted model μ together with the data are shown in Figure 5.10. Figures 5.11(a) and 5.11(b) give the fitted distribution of the number of fish species for observation 40, with lake area of 165 and $(\hat{\mu}, \hat{\sigma}, \hat{\nu}) = (22.64, 1.44, -5.68)$, and observation 67, with lake area 8264 and $(\hat{\mu}, \hat{\sigma}, \hat{\nu}) = (47.78, 1.44, -1.2)$, respectively.

```
> pdf.plot(mSI, c(40, 67), min = 0, max = 110, step = 1)
```

Table 5.3, taken from Rigby *et al.* (2008), gives the deviance (DEV), AIC and SBC for specific models fitted to the fish species data, and is used to answer the questions at the start of this section. The terms 1, x and x<2> indicate constant, linear and quadratic terms respectively,

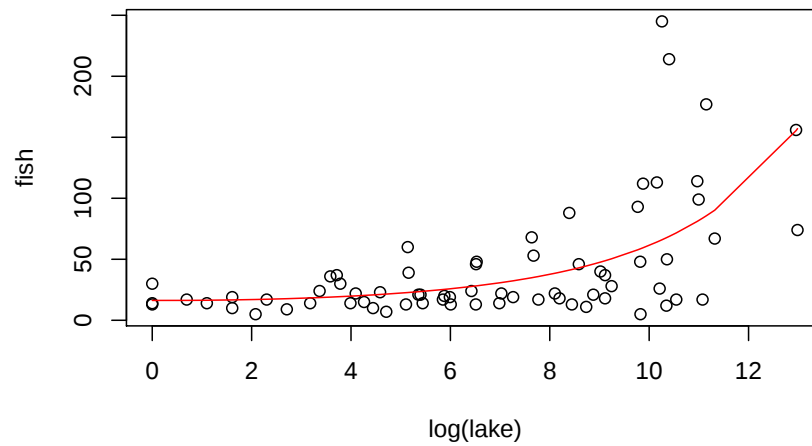
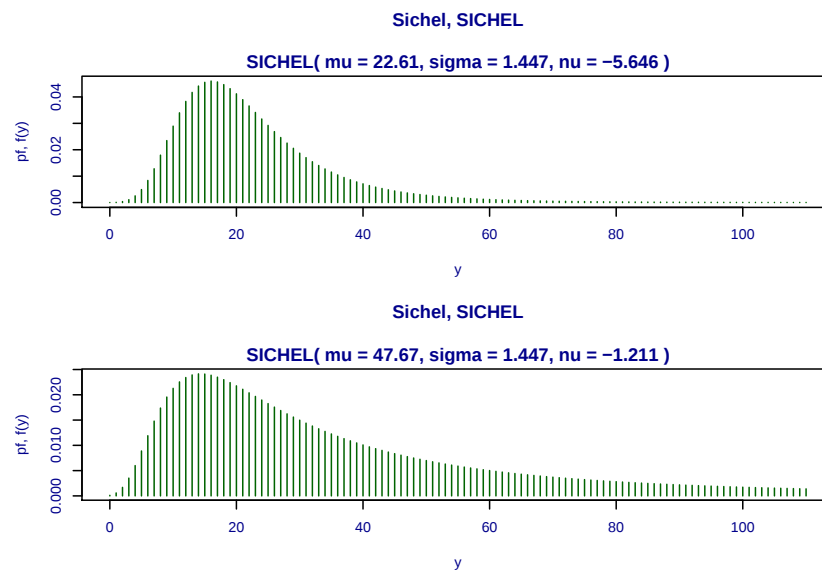
Figure 5.10: Fitted μ (the mean number of fish species) against log lake area

Figure 5.11: Fitted Sichel distributions for observations (a) 40 and (b) 67

while the term `cs(x,2)` indicates a cubic smoothing spline with two degrees of freedom on top of the linear term `x`.

The following analysis is from Rigby *et al.* (2008). Comparing models 2, 3 and 4 indicates that a quadratic model for $\log \mu$ is found to be adequate (while the linear and the cubic spline models were found to be inappropriate here). Comparing model 1 and 3 indicates that Y has a highly overdispersed Poisson distribution. Comparing model 3 with models 5 and 6 shows that either a linear model in x for $\log(\sigma)$ or a different variance-mean relationship from that of the negative binomial (NBI) [i.e. $V[Y] = \mu + \sigma\mu^2$] is required. In particular the estimated ν parameter in the negative binomial family (NBF) of model 6 is $\hat{\nu} = 2.9$ suggesting a possible variance-mean relationship $V[Y] = \mu + \sigma\mu^3$. [The binomial family `NBF()` function is an experimental `gamlss.family` function, defined on the ideas of Section 5.2.7, and it is not fully implemented yet in `gamlss`]. Modelling σ in the NBF did not improve the fit greatly, as shown by model 7. A search of alternative mixed Poisson distributions (type (a) in Section 5.1), included the Poisson-inverse Gaussian (PIG), the Sichel (SI) and the Delaporte (DEL). The models with the best AIC for each distribution were recorded in Table 5.3 models 8 to 11. A normal random effect mixture distribution (type (b) in Section 5.1) was fitted using 20 Gaussian quadrature to the Poisson and NBI conditional distributions giving models 12 and 13, i.e. Poisson-Normal and NBI-Normal respectively. 'Non-parametric' random effects (effectively finite mixtures) (NPFM), type (i) in Section 5.1, were also fitted to Poisson and NBI conditional distributions giving models 14 and 15, i.e. PO-NPFM(6) and NB-NPFM(2) with 6 and 2 components respectively. >From category (i) solutions, Efron's double exponential (Poisson) distribution was fitted giving model 16 (doublePO). >From category (ii) the best discretized continuous distribution fitted was a discrete inverse Gaussian distribution giving model 17 (IGdisc), again suggesting a possible cubic variance-mean relationship.

Overall the best model according to Akaike information criterion (AIC) is model 9, the Sichel model, following closely by model 11, a Delaporte model. According to the Schwarz Bayesian criterion (SBC) the best model is model 17, the discretized inverse Gaussian distribution, again followed closely by model 11.

The model in Table 5.3 with the minimum AIC value 609.7 was selected, i.e. model 9, a Sichel, $SI(\mu, \sigma, \nu)$, model fitted earlier in this section, with $\log \hat{\mu} = 2.790 - 0.00679x + 0.0140x^2$, $\hat{\sigma} = 1.447$ and $\hat{\nu} = -10.843 + 1.048x$. For comparison model 11 gives the Delaporte, $DEL(\mu, \sigma, \nu)$, model (with lowest AIC). Note in model 11 that $\sigma = 1$ is fixed in the Delaporte distribution (??) corresponding to a Poisson-shifted exponential distribution, giving fitted model $\log \hat{\mu} = 2.787 - 0.004207x + 0.013959x^2$, $\sigma = 1$ (fixed) and $\text{logit } \hat{\nu} = 1.066 - 0.2854x$.

The following code can be used to reproduce the results of Table 5.3. Model 16 is not fitted here since it requires Jim Lindsay's package `rmutil`. For completeness we refit models we fitted earlier:

```
> library(gamlss.mx)
> m1 <- gamlss(fish ~ poly(x, 2), data = species, family = PO,
+   trace = FALSE)
> m2 <- gamlss(fish ~ x, data = species, family = NBI, trace = FALSE)
> m3 <- gamlss(fish ~ poly(x, 2), data = species, family = NBI,
+   trace = FALSE)
> m4 <- gamlss(fish ~ cs(x, 3), data = species, family = NBI, trace = FALSE)
> m5 <- gamlss(fish ~ poly(x, 2), sigma.fo = ~x, data = species,
+   family = NBI, trace = FALSE)
> m6 <- gamlss(fish ~ poly(x, 2), sigma.fo = ~1, data = species,
+   family = NBF, n.cyc = 200, trace = FALSE)
```

Table 5.3: Comparison of models for the fish species data

Model	$f_Y(y)$	μ	σ	ν	DEV	df	AIC	SBC
1	PO	$x < 2 >$	-	-	1849.3	3	1855.3	1862.0
2	NBI	x	1	-	619.8	3	625.8	632.6
3	NBI	$x < 2 >$	1	-	614.3	4	622.3	631.3
4	NBI	$cs(x, 3)$	1	-	611.9	6	623.9	637.4
5	NBI	$x < 2 >$	x	-	605.0	5	615.0	626.2
6	NBI-family	$x < 2 >$	1	1	606.1	5	616.1	627.4
7	NBI-family	$x < 2 >$	x	1	604.9	6	616.9	630.4
8	PIG	$x < 2 >$	1	-	613.3	4	621.3	630.3
9	SI	$x < 2 >$	1	x	597.7	6	609.7	623.2
10	DEL	$x < 2 >$	1	x	600.7	6	612.7	626.2
11	DEL	$x < 2 >$	-	x	600.6	5	610.6	621.9
12	PO-Normal	$x < 2 >$	1	-	615.2	4	623.2	632.2
13	NBI-Normal	$x < 2 >$	x	1	603.7	6	615.7	629.2
14	PO-NPFM(6)	$x < 2 >$	-	—	601.9	13	627.9	657.2
15	NB-NPFM(2)	$x < 2 >$	1	—	611.9	6	623.9	637.4
16	doublePO	$x < 2 >$	x	-	616.4	5	626.4	637.6
17	IGdisc	$x < 2 >$	1	-	603.3	4	611.3	620.3

```

> m7 <- gamlss(fish ~ poly(x, 2), sigma.fo = ~x, data = species,
+   family = NBF, n.cyc = 100, trace = FALSE)
> m8 <- gamlss(fish ~ poly(x, 2), data = species, family = PIG,
+   trace = FALSE)
> m9 <- gamlss(fish ~ poly(x, 2), nu.fo = ~x, data = species, family = SICHEL,
+   trace = FALSE)
> m10 <- gamlss(fish ~ poly(x, 2), nu.fo = ~x, data = species,
+   family = DEL, n.cyc = 50, trace = FALSE)
> m11 <- gamlss(fish ~ poly(x, 2), nu.fo = ~x, data = species,
+   family = DEL, sigma.fix = TRUE, sigma.start = 1, n.cyc = 50,
+   trace = FALSE)
> m12 <- gamlssNP(fish ~ poly(x, 2), data = species, mixture = "gq",
+   K = 20, family = PO)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..104 ..105 ..
EM algorithm met convergence criteria at iteration 105
Global deviance trend plotted.

> m13 <- gamlssNP(fish ~ poly(x, 2), sigma.fo = ~x, data = species,
+   mixture = "gq", K = 20, family = NBI)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..45 ..46 ..
EM algorithm met convergence criteria at iteration 46
Global deviance trend plotted.

> m14 <- gamlssNP(fish ~ poly(x, 2), data = species, mixture = "np",
+   K = 6, family = PO)

```

```

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..
EM algorithm met convergence criteria at iteration 12
Global deviance trend plotted.
EM Trajectories plotted.

> m15 <- gamlssNP(fish ~ poly(x, 2), data = species, mixture = "np",
+   K = 2, family = NBI)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..51 ..52 ..
EM algorithm met convergence criteria at iteration 52
Global deviance trend plotted.
EM Trajectories plotted.

> library(gamlss.cens)
> m17 <- gamlss(Surv(fish, fish + 1, type = "interval2") ~ x +
+   I(x^2), sigma.fo = ~1, data = species, family = cens(IG,
+   type = "interval"))

GAMLSS-RS iteration 1: Global Deviance = 603.2793
GAMLSS-RS iteration 2: Global Deviance = 603.2793

> GAIC(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13,
+   m14, m15, m17)

      df      AIC
m9    6.00000 609.7299
m11    5.00000 610.6493
m17    4.00000 611.2793
m10    6.00000 612.6593
m5     5.00000 614.9565
m13    6.00000 615.7281
m6     5.00000 616.0828
m7     6.00000 616.9229
m8     4.00000 621.3460
m3     4.00000 622.3173
m12    4.00000 623.2455
m15    6.00000 623.8794
m4     5.99924 623.9085
m2     3.00000 625.8443
m14   13.00000 627.9431
m1     3.00000 1855.2965

```

5.5 Bibliography

Appendix of Chapter 5: Skewness and kurtosis for the (mixed) Poisson distributions.

Let $Y|\gamma \sim \text{PO}(\mu\gamma)$ and γ have a distribution with cumulative generating function $K_\gamma(t)$, then the cumulative generating function of the marginal distribution of Y , $K_Y(t)$, is given by

$$K_Y(t) = K_\gamma[\mu(e^t - 1)]$$

and hence, assuming that γ has mean 1, the cumulants of Y and γ are related by $E(Y) = \mu$, $V(Y) = \mu + \mu^2 V(\gamma)$,

$$\begin{aligned}\kappa_{3Y} &= \mu + 3\mu^2 V(\gamma) + \mu^3 \kappa_{3\gamma}, \\ \kappa_{4Y} &= \mu + 7\mu^2 V(\gamma) + 6\mu^3 \kappa_{3\gamma} + \mu^4 \kappa_{4\gamma},\end{aligned}$$

where κ_{3Y} and κ_{4Y} are the third and fourth cumulants of Y

The skewness and kurtosis of Y are $\sqrt{\beta_1} = \kappa_{3Y} / [V(Y)]^{1.5}$ and $\beta_2 = 3 + \left\{ \kappa_{4Y} / [V(Y)]^2 \right\}$ respectively. Specific marginal distributions for Y are considered below.

Poisson distribution

If Y has a Poisson, $PO(\mu)$, distribution then the mean, variance, skewness and kurtosis of Y are respectively given by $E(Y) = \mu$, $V(Y) = \mu$, $\sqrt{\beta_1} = \mu^{-0.5}$, $\beta_2 = 3 + (1/\mu)$.

Negative binomial distribution

If Y has a negative binomial type I, $NBI(\mu, \sigma)$, distribution then $E(Y) = \mu$, $V(Y) = \mu + \sigma\mu^2$, $\sqrt{\beta_1} = (1 + 2\mu\sigma) / [\mu(1 + \mu\sigma)]^{0.5}$ and $\beta_2 = 3 + (1 + 6\mu\sigma + 6\mu^2\sigma^2) / [\mu(1 + \mu\sigma)]$.

Delaporte distribution

If Y has a Delaporte, $DEL(\mu, \sigma, \nu)$, distribution then $E(Y) = \mu$, $V(Y) = \mu + \sigma\mu^2(1 - \nu)^2$, $\sqrt{\beta_1} = \mu [1 + 3\mu\sigma(1 - \nu)^2 + 2\mu^2\sigma^2(1 - \nu)^3] / [V(Y)]^{1.5}$ and

$$\beta_2 = 3 + \left\{ \mu [1 + 7\mu\sigma(1 - \nu)^2 + 12\mu^2\sigma^2(1 - \nu)^3 + 6\mu^3\sigma^3(1 - \nu)^4] / [V(Y)]^2 \right\}.$$

Poisson-shifted generalized inverse Gaussian (PSGIG) distribution

If Y has a PSGIG(μ, σ, ν, τ) distribution then the mean, variance skewness and kurtosis of Y are obtained from the equations given at the start of this Appendix, where the cumulants of the mixing distribution $\gamma \sim \text{SGIG}(1, \sigma^{1/2}, \nu, \tau)$, defined by $Z = (\gamma - \tau)/(1 - \tau) \sim \text{GIG}(1, \sigma^{1/2}, \nu)$ and equation 5.1, are given by $E(\gamma) = 1$,

$$\begin{aligned}V(\gamma) &= (1 - \tau)^2 V(Z) = (1 - \tau)^2 g_1, \\ \kappa_{3\gamma} &= (1 - \tau)^3 \kappa_{3Z} = (1 - \tau)^3 [g_2 - 3g_1], \\ \kappa_{4\gamma} &= (1 - \tau)^4 \kappa_{4Z} = (1 - \tau)^4 (g_3 - 4g_2 + 6g_1 - 3g_1^2),\end{aligned}$$

where

$$\begin{aligned}g_1 &= [1/c^2 + 2\sigma(\nu + 1)/c - 1], \\ g_2 &= 2\sigma(\nu + 2)/c^3 + [4\sigma^2(\nu + 1)(\nu + 2) + 1] / c^2 - 1, \\ g_3 &= [1 + 4\sigma^2(\nu + 2)(\nu + 3)] / c^4 + [8\sigma^3(\nu + 1)(\nu + 2)(\nu + 3) + 4\sigma(\nu + 2)] / c^3 - 1,\end{aligned}$$

obtained from $Z = (\gamma - \tau)/(1 - \tau) \sim \text{GIG}(1, \sigma^{1/2}, \nu)$, where the cumulant generating functions of γ and Z are related by $K_\gamma(t) = \tau t + K_Z[(1 - \tau)t]$.

The corresponding results for the Sichel distribution for Y are given by setting $\tau = 0$ in the above results for the PSGIG distribution for Y .

Exercises for Chapter 5

- Q1 Gupta et al. (1996) present the following data giving the number of Lamb foetal movements y observed with frequency f recorded by ultrasound over 240 consecutive five second intervals:

y	0	1	2	3	4	5	6	7
f	182	41	12	2	2	0	0	1

- Fit each of the following distributions for y to the data (using different model names e.g. `mPO` etc. for later comparison): $PO(\mu)$, $NBI(\mu, \sigma)$, $NBII(\mu, \sigma)$, $PIG(\mu, \sigma)$, $SICHEL(\mu, \sigma, \nu)$, $DEL(\mu, \sigma, \nu)$ and $ZIP(\mu, \sigma)$. [Note that the default fitting method `RS` may be slow for the Sichel distribution, so try using e.g. `method=mixed(2,100)`, which performs 2 iterations of the `RS` algorithm, followed by (up to) 100 iterations of the `CG` algorithm.]
- Use the `AIC` command with each of the penalties $k = 2, 3.8$ and $5.48 = \log(240)$, [corresponding to criteria `AIC`, $\chi^2_{1,0.05}$ and `SBC` respectively], in order to select a distribution model. Output the parameter estimates for your chosen model. [Note that the residuals for frequency data are not currently implemented.]

References: Gupta, P.L., Gupta, R.C. and Tripathi, R.C. (1996) Analysis of zero-adjusted count data. *Computational Statistics and Data Analysis*, 23, 207-218.

- Q2 The USA National AIDS Behavioural Study recorded y , the number of times individuals engaged in risky sexual behaviour during the previous six months, together with two explanatory factors sex of individual (male or female) and whether they has a risky partner risky (no or yes), giving the following frequency distributions:

y	0	1	2	3	4	5	5	7	10	12	15	20	30	37	50
male,no	541	19	17	16	3	6	5	2	6	1	0	3	1	0	0
male,yes	102	5	8	2	1	4	1	0	0	0	1	0	0	1	0
female,no	238	8	0	2	1	1	1	1	0	0	1	0	0	0	0
female,yes	103	6	4	2	0	1	0	0	0	0	0	0	0	0	1

The data were previously analysed by Heilbron (1994).

- Read the above frequencies (corresponding to the male yes, male no, female yes, female no rows of the above table) into a variable `f`. Read the corresponding count values into `y`. buy using `y<-rep(c(0:7),10,12,15,20,30,37,50),4)`. Generate a single factor type for type of individual with four levels (corresponding to male yes, male no, female yes, female no) by `type<-gl(4,15)`.
- Fit each of the following distributions for y to the data (using different model names for later comparison): $PO(\mu)$, $NBI(\mu, \sigma)$, $NBII(\mu, \sigma)$, $PIG(\mu, \sigma)$, $SICHEL(\mu, \sigma, \nu)$, $DEL(\mu, \sigma, \nu)$ and $ZIP(\mu, \sigma)$, using factor type for the mean model and a constant scale (and shape).
- Use the `AIC` command with each of the penalties $k = 2, 3$ and 4 , in order to select a distribution model.
- Check whether your chosen distribution model needs the factor type in its scale (and shape) models. Check whether the factor type is needed in the mean model.

- (e) Output the parameter estimates for your chosen model.

References: Heilbron, D.C. (1994) Zero-Altered and Other Regression Models for Count Data with Added Zeros. *Biometrical Journal*, 36, 531-547.

Chapter 6

Discrete response: binary and binomial data

6.1 Available distributions

The binomial distribution is denoted $\text{BI}(n, \mu)$ in **gamlss** for $y = 0, 1, \dots, n$, where $0 < \mu < 1$ and n is a known positive integer called the binomial denominator, (**bd** in the R code). The binomial distribution has mean $n\mu$ and variance $n\mu(1 - \mu)$.

The beta binomial distribution, is denoted $\text{BB}(n, \mu, \sigma)$ in **gamlss** for $y = 0, 1, \dots, n$, where $0 < \mu < 1$, $\sigma > 0$ and n is a known positive integer, has mean $n\mu$ and variance $n\mu(1 - \mu) [1 + \sigma(n - 1)/(1 + \sigma)]$ and hence provides a model for overdispersed binomial data.

6.2 Examples of fitting binomial data

6.2.1 The alveolar-bronchiolar adenomas data

Data summary:

R data file: alveolar in package **gamlss.dist** of dimensions 23×2

source: Tamura and Young (1987), and Hand *et al.* (1994)

variables

r : number of mice having alveolar-bronchiolar

n : total number of mice

purpose: to demonstrate the fitting of a binomial distribution to the data.

conclusion a binomial distribution is adequate

Here we consider the alveolar-bronchiolar adenomas data used by Tamura and Young (1987) and also reproduced in Hand *et al.* (1994), data set 256. The data are the number of mice having alveolar-bronchiolar adenomas out of total numbers of mice (i.e. the “binomial denominator”) in 23 independent groups.

For binomial type of data with no explanatory variables the `histDist()` can still be used but with a limited scope. The plot works fine if the 'binomial denominator' is constant for all observations. In this case we can plot a histogram of y against the number of events from zero to the constant 'binomial denominator', and then superimpose the fitted probabilities from the fitted binomial distribution. When the binomial denominator is not constant for all observations then `histDist()` plots a histogram of the proportions (which may be of some interest) and then indicates where the fitted proportion lies:

```
> library(gamlss.dist)
> data(alveolar)
> alveolar$y <- with(alveolar, cbind(r, n - r))
> con <- gamlss.control(trace = F)
> m1 <- gamlss(y ~ 1, data = alveolar, family = BI, control = con)
> m2 <- gamlss(y ~ 1, data = alveolar, family = BB, control = con)
> GAIC(m1, m2)
```

	df	AIC
m1	1	73.1292
m2	2	75.0665

```
> m3 <- with(alveolar, histDist(y, "BI", xlim = c(0, 0.3)))
```

GAMLSS-RS iteration 1: Global Deviance = 71.1292
 GAMLSS-RS iteration 2: Global Deviance = 71.1292

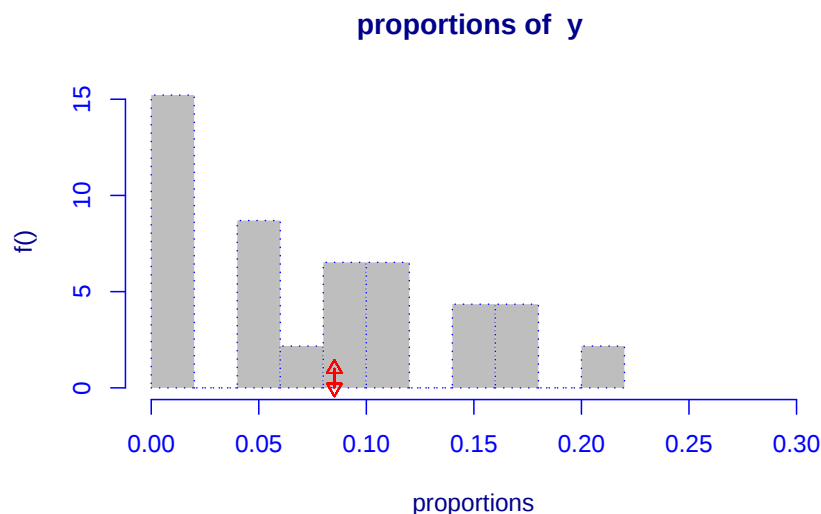


Figure 6.1: The proportion of alveolar-bronchiolar adenomas

With the two models having similar deviances there is no support from the data to favour the beta binomial model instead of the binomial one. Figure 6.1 shows the histogram of proportion of mice having alveolar-bronchiolar adenomas and the red vertical line indicates the fitted probability.

In order to demonstrate what would have happen in the plot if the binomial denominator was the same for all observations we fix it in the above data to be 10.

```
> alveolar$yy <- with(alveolar, cbind(r, 10 - r))
> m1 <- gamlss(yy ~ 1, data = alveolar, family = BI, control = con)
> m2 <- gamlss(yy ~ 1, data = alveolar, family = BB, control = con)
> GAIC(m1, m2)
```

	df	AIC
m2	2	88.9301
m1	1	104.7267

```
> m3 <- histDist(alveolar$yy, "BB")
```

```
GAMLSS-RS iteration 1: Global Deviance = 87.4166
GAMLSS-RS iteration 2: Global Deviance = 85.0155
GAMLSS-RS iteration 3: Global Deviance = 84.9312
GAMLSS-RS iteration 4: Global Deviance = 84.9301
GAMLSS-RS iteration 5: Global Deviance = 84.9301
```

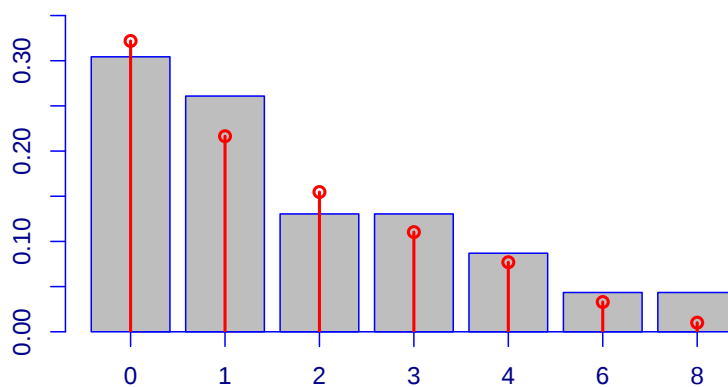


Figure 6.2: The proportion and fitted distribution to the artificial alveolar-bronchiolar adenomas data

6.2.2 The first year student examination results data

Data summary:

R data file: `students` created here of dimensions 8×2

source: Karlis and Xekalaki (2008)

variables

r : number of exams first year students passed out of 8 in total

freq : the frequency (i.e. the number of students) for the number of exams passed

purpose: to demonstrate the fitting of a binomial distribution to the data when the binomial denominator is fixed.

conclusion a beta binomial distribution is adequate

Here we demonstrate the fitting of a binomial type data response variable given that the binomial denominator is constant. The data shown in Table 6.1 (first used by Karlis and Xekalaki, 2008) refer to the numbers of courses passed, **r**, and their frequency, **freq**, of a class of 65 first year students. The students enrolled for a 8 course during the year. The variable **n.r** in table 6.1 is defined as $8 - r$.

	r	n.r	freq
1	0.00	8.00	1.00
2	1.00	7.00	4.00
3	2.00	6.00	4.00
4	3.00	5.00	8.00
5	4.00	4.00	9.00
6	5.00	3.00	6.00
7	6.00	2.00	8.00
8	7.00	1.00	12.00
9	8.00	0.00	13.00

Table 6.1: The first year student examination results data where the binomial denominator is constant at 8.

Now we create the data and fit a binomial and a beta binomial distribution. We then select a model using AIC.

```
> r <- 0:8
> freq <- c(1, 4, 4, 8, 9, 6, 8, 12, 13)
> y <- cbind(r, 8 - r)
> colnames(y) <- c("r", "n-r")
> students <- data.frame(y, freq)
> m1 <- gamlss(y ~ 1, weights = freq, data = students, family = BI,
+   trace = FALSE)
> m2 <- gamlss(y ~ 1, weights = freq, data = students, family = BB,
+   trace = FALSE)
> GAIC(m1, m2)
```

```

      df      AIC
m2  2 273.4987
m1  1 339.6467

```

The beta binomial model has a superior fit. This is also demonstrated in Figure 6.3 where the fitted probabilities from the two models are plotted. The figures were produced using the following code:

```

> op <- par(mfrow = c(2, 1))
> m3 <- with(students, histDist(cbind(rep(y[, 1], freq), rep(y[,
+   2], freq)), family = BI, ylim = c(0, 0.3), xlab = "number of courses passed",
+   ylab = "probability", main = "(a) binomial"))

GAMLSS-RS iteration 1: Global Deviance = 337.6467
GAMLSS-RS iteration 2: Global Deviance = 337.6467

> m4 <- with(students, histDist(cbind(rep(y[, 1], freq), rep(y[,
+   2], freq)), family = BB, ylim = c(0, 0.3), xlab = "number of courses passed",
+   ylab = "probability", main = "(b) beta binomial"))

GAMLSS-RS iteration 1: Global Deviance = 270.2592
GAMLSS-RS iteration 2: Global Deviance = 269.4993
GAMLSS-RS iteration 3: Global Deviance = 269.4987

> par(op)

```

6.2.3 The hospital stay data

Data summary:

R data file: aep in package **gamlss** of dimensions 1383×8

source: Gange *et al.* (1996)

variables

los : total number of days

noinap : number of inappropriate days patient stay in hospital

loglos : the log of los/10

sex : the gender of patient

ward : type of ward in the hospital (medical, surgical or other)

year : 1988 or 1990

age : age of the patient subtracted from 55

y : the response variable, a matrix with columns **noinap**, **los-noinap**

purpose: to demonstrate the fitting of a beta binomial distribution to the data.

conclusion a beta binomial distribution is needed

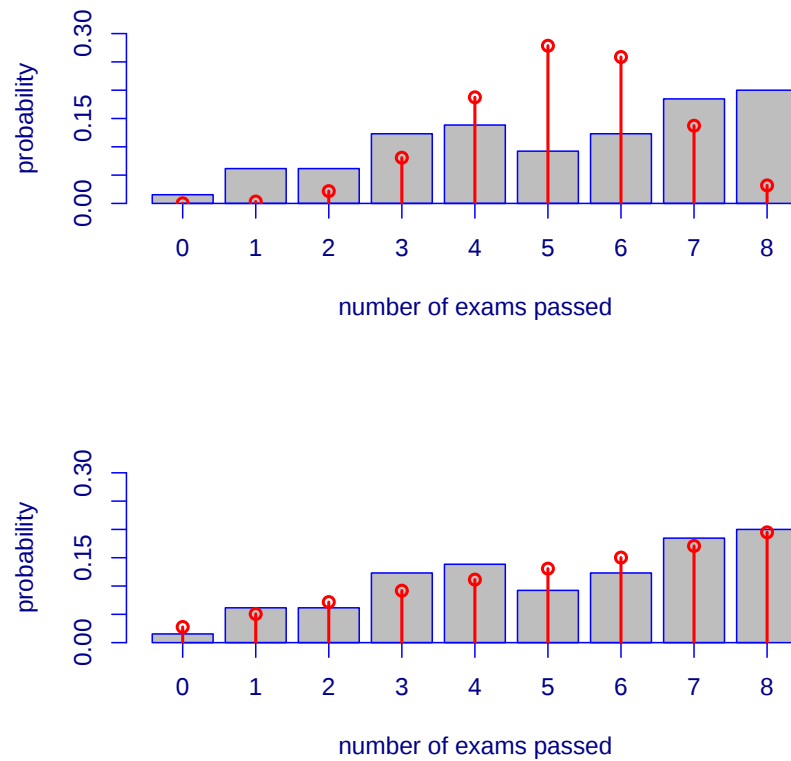


Figure 6.3: The first year student examination results data: Fitted probabilities (a) binomial (b) beta binomial distributions

The data, 1383 observations, are from a study at the Hospital del Mar, Barcelona during the years 1988 and 1990, see Gange *et al.* (1996). The response variable is the number of inappropriate days (`noinap`) out of the total number of days (`los`) patients spent in hospital. Each patient was assessed for inappropriate stay on each day by two physicians who used the appropriateness evaluation protocol (AEP), see Gange *et al.* (1996) and their references for more details. The following variables were used as explanatory variables, `age`, `sex`, `ward`, `year` and `loglos`.

A plot of the inappropriateness rates `ninap/los` against `age`, `sex`, `ward` and `year` are shown in Figure 6.4 obtained by:

```
> data(aep)
> prop <- with(aep, noinap/los)
> op <- par(mfrow = c(2, 2))
> plot(prop ~ age, data = aep, cex = los/30)
> plot(prop ~ sex, data = aep)
> plot(prop ~ ward, data = aep)
> plot(prop ~ year, data = aep)
> par(op)
```

Gange *et al.* (1996) used a logistic regression model for the number of inappropriate days, with binomial and beta binomial errors and found that the later provided a better fit to the data. They modelled both the mean and the dispersion of the beta binomial distribution (BB) as functions of explanatory variables using the epidemiological package EGRET, Cytel Software Corporation (2001), which allowed them to fit a parametric model using a `logit` link for the mean and an identity link for the dispersion. Their final model was a beta binomial model $BB(\mu, \sigma)$, with terms `ward`, `year` and `loglos` in the model for $\text{logit}(\mu)$ and term `year` for model for σ .

First we fit their final model, equivalent to model I in Table 6.2. Although we use a log link for the dispersion σ in Table 6.2, this does not affect model I since `year` is a factor. Table 6.2 shows the GD, AIC and SBC for model I, 4519.4, 4533.4 and 4570.08 respectively. Here we are interested in whether we can improve the model using the flexibility of GAMLSS. For the dispersion parameter model we found that the addition of `ward` improves the fit (see model II in Table 6.2 with $AIC = 4501.02$, $SBC = 4548.11$) but no other term was found to be significant. Non-linearities in the mean model for the terms `loglos` and `age` were investigated using cubic smoothing splines (`cs`) in models III and IV. There is strong support for including a smoothing term for `loglos` as indicated by the reduction in the AIC and SBC for model III compared to model II. The inclusion of a smoothing term for `age` is not so clear cut since while there is some marginal support from the AIC it is rejected strongly from SBC, when comparing model III to model IV. The R script for fitting the models in Table 6.2 is shown below:

```
> mI <- gamlss(y ~ ward + year + loglos, sigma.fo = ~year, family = BB,
+   data = aep, trace = FALSE)
> mII <- gamlss(y ~ ward + year + loglos, sigma.fo = ~year + ward,
+   family = BB, data = aep, trace = FALSE)
> mIII <- gamlss(y ~ ward + year + cs(loglos, 1), sigma.fo = ~year +
+   ward, family = BB, data = aep, trace = FALSE)
> mIV <- gamlss(y ~ ward + year + cs(loglos, 1) + cs(age, 1), sigma.fo = ~year +
+   ward, family = BB, data = aep, trace = FALSE)
> GAIC(mI, mII, mIII, mIV, k = 0)
```

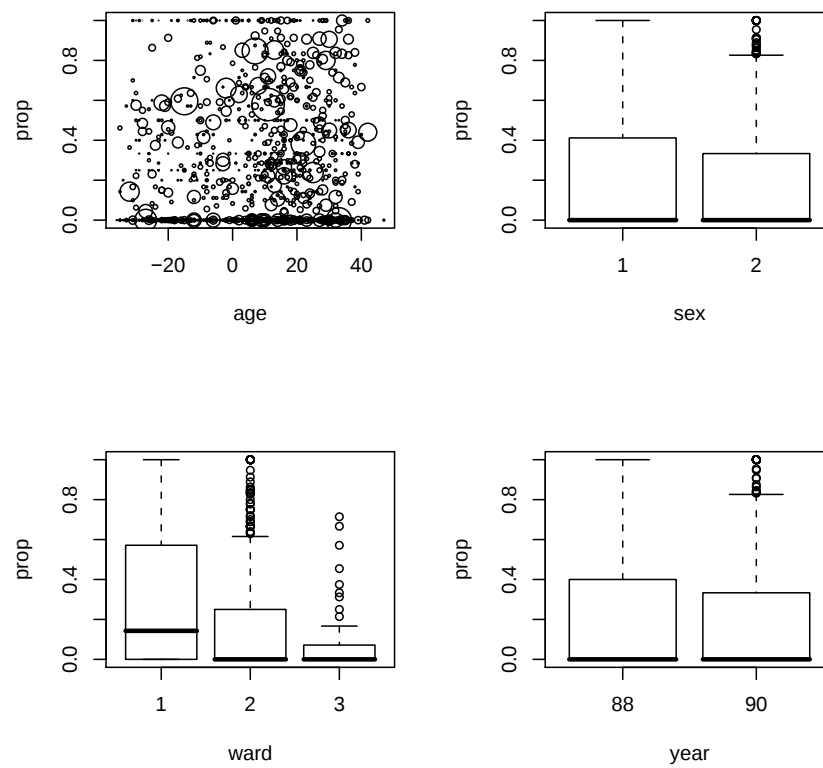
Figure 6.4: The rate of appropriateness against `age`, `sex`, `ward` and `year`

Table 6.2: Models for the AEP data

Models	Links	Terms	GD (AIC) [SBC]
I	logit(μ) log(σ)	1+ward+loglos+year 1+year	4519.4 (4533.4) [4570.1]
II	logit(μ) log(σ)	1+ward+loglos+year 1+year+ward	4483.0 (4501.0) [4548.1]
III	logit(μ) log(σ)	1+ward+cs(loglos,2)+year 1+year+ward	4459.4 (4479.4) [4531.8]
IV	logit(μ) log(σ)	1+ward+cs(loglos,2)+year+cs(age,2) 1+year+ward	4454.4 (4478.4) [4541.2]

```

      df      AIC
mIV 12.00010 4454.362
mIII 10.00045 4459.427
mII  9.00000 4483.020
mI   7.00000 4519.441

```

```
> GAIC(mI, mII, mIII, mIV)
```

```

      df      AIC
mIV 12.00010 4478.362
mIII 10.00045 4479.427
mII  9.00000 4501.020
mI   7.00000 4533.441

```

```
> GAIC(mI, mII, mIII, mIV, k = log(length(aep$age)))
```

```

      df      AIC
mIII 10.00045 4531.750
mIV  12.00010 4541.147
mII   9.00000 4548.108
mI    7.00000 4570.065

```

Note also that the model IV can also be improved marginally by changing the logistic link for the mean to a probit link giving $GD = 4452.36$, $AIC = 4476.36$ and $SBC = 4539.14$ as shown below:

```

> (mIV1 <- gamlss(y ~ ward + year + cs(loglos, 1) + cs(age, 1),
+   sigma.fo = ~year + ward, family = BB(mu.link = "probit"),
+   data = aep, trace = FALSE))

Family:  c("BB", "Beta Binomial")
Fitting method: RS()

```

```

Call:  gamlss(formula = y ~ ward + year + cs(loglos, 1) + cs(age, 1),
  sigma.formula = ~year + ward, family = BB(mu.link = "probit"),
  data = aep, trace = FALSE)

Mu Coefficients:
(Intercept)      ward2      ward3      year90  cs(loglos, 1)
-0.667316    -0.244238    -0.473429    0.151170    0.240327
cs(age, 1)
0.002647
Sigma Coefficients:
(Intercept)      year90      ward2      ward3
0.2953    -0.3729    -0.7172    -1.1713

Degrees of Freedom for the fit: 12.00011 Residual Deg. of Freedom   1371
Global Deviance:      4452.36
AIC:      4476.36
SBC:      4539.14

```

The fitted functions for all the terms for μ in model IV are shown in Figure 6.5. The fitted terms for σ are shown in Figure 6.6. They have been obtained using the function `term.plot()` as follows:

```

> op <- par(mfrow = c(2, 2))
> term.plot(mIV, se = T)
> par(op)
> op <- par(mfrow = c(2, 1))
> term.plot(mIV, "sigma", se = T)
> par(op)

```

Figure 6.7 displays six instances of the normalized randomised quantile residuals (see Section 2.1.5) from model IV. The residuals seem to be satisfactory. The figure is generated using the function `rqres.plot()`:

```

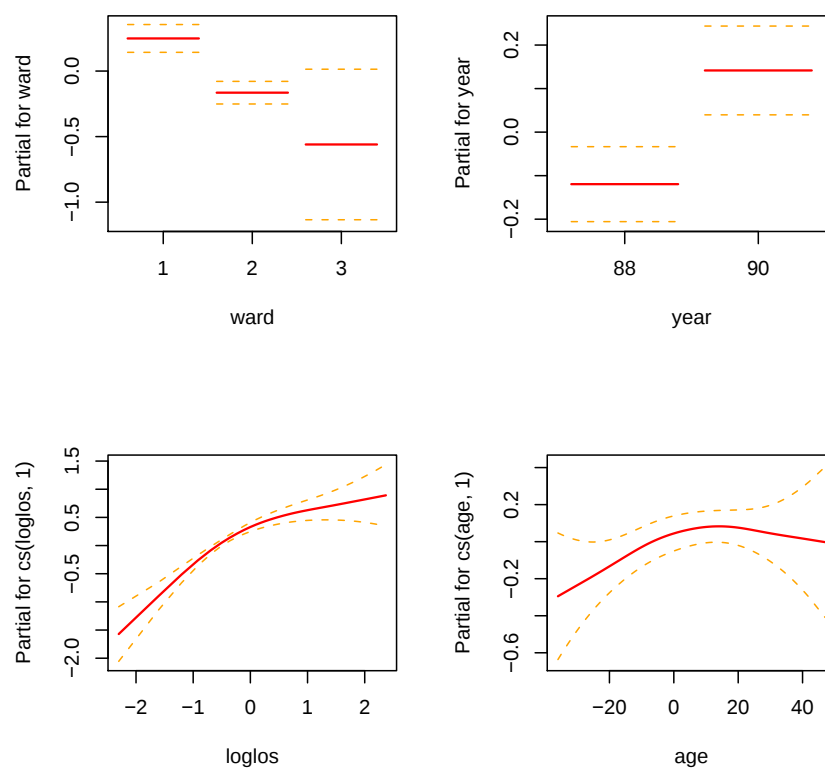
> rqres.plot(mIV)

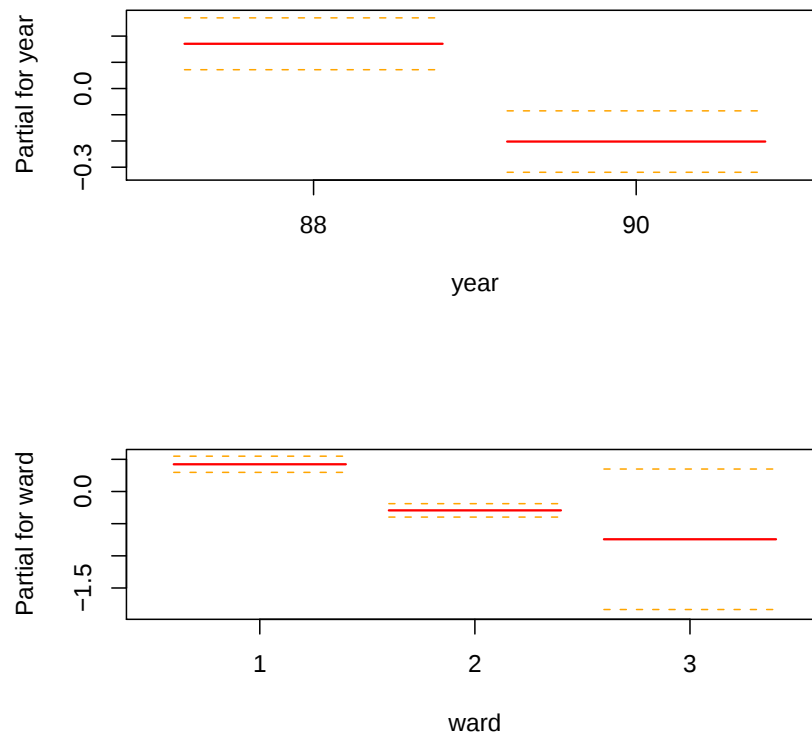
```

6.3 Bibliography

6.4 Appendix of Chapter 6

6.5 Exercises for Chapter 6

Figure 6.5: The fitted terms for μ in model IV

Figure 6.6: The fitted terms for σ in model IV

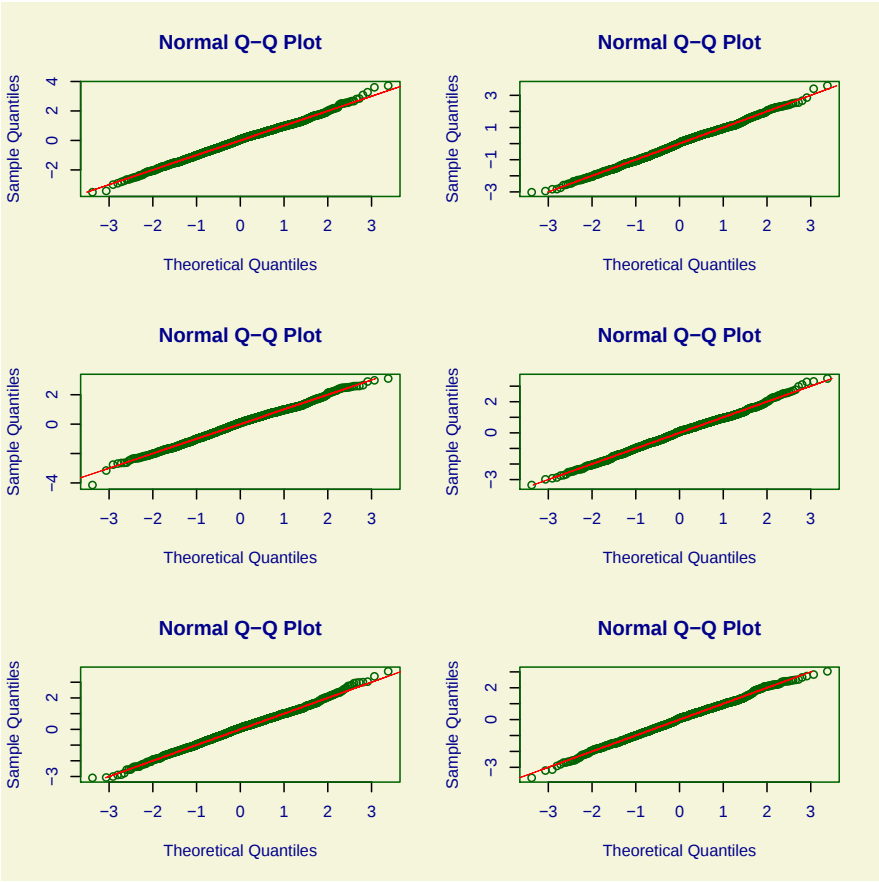


Figure 6.7: Six instances of the normalized randomised quantile residuals for model

Chapter 7

Fitting Finite Mixture Distributions

This Chapter covers finite mixtures within GAMLSS. In general finite mixture distributions are fitted within GAMLSS using the EM algorithm. Certain specific mixtures distributions are explicitly available in **gamlss** packages. The zero inflated Poisson (**ZIP** and **ZIP2**) the zero adjusted (inflated) inverse Gaussian (**ZAIG**), and the four parameter beta inflated at zero and one (**BEINF**).

7.1 Introduction to finite mixtures

Suppose that the random variable Y comes from component k , having probability (density) function $f_k(y)$, with probability π_k for $k = 1, 2, \dots, K$, then the (marginal) density of Y is given by

$$f_Y(y) = \sum_{k=1}^K \pi_k f_k(y) \quad (7.1)$$

where $0 \leq \pi_k \leq 1$ is the prior (or mixing) probability of component k , for $k = 1, 2, \dots, K$ and $\sum_{k=1}^K \pi_k = 1$.

More generally the probability (density) function $f_k(y)$ for component k may depend on parameters θ_k and explanatory variables $\mathbf{x}_k$, i.e. $f_k(y) = f_k(y|\theta_k, \mathbf{x}_k)$.

Similarly $f_Y(y)$ depends on parameters $\psi = (\theta, \pi)$ where $\theta = (\theta_1, \theta_2, \dots, \theta_K)$ and $\pi^T = (\pi_1, \pi_2, \dots, \pi_K)$ and explanatory variables $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K)$, i.e. $f_Y(y) = f_Y(y|\psi, \mathbf{x})$, and

$$f_Y(y|\psi, \mathbf{x}) = \sum_{k=1}^K \psi_k f_k(y|\theta_k, \mathbf{x}_k) \quad (7.2)$$

Subsequently we omit the conditioning on θ_k , $\mathbf{x}_k$ and ψ to simplify the presentation. In Sections 7.2, 7.3 and 7.7 we consider respectively maximum likelihood estimation, the corresponding fitting function **gamlssMX** and examples for finite mixtures models with **no** parameters in common, while in Sections 7.5, 7.6 and 7.7 we consider respectively maximum likelihood estimation, the corresponding fitting function **gamlssNP** and examples for finite mixture models with parameters in common. Throughout this chapter we will assume that all K components of the mixture can be represented by GAMLSS models.

7.2 Finite mixtures with no parameters in common

Here the parameter sets $(\theta_1, \theta_2, \dots, \theta_k)$ are distinct, i.e. no parameter is common to two or more parameters sets. Note that what this means in practice within GAMLSS is that the conditional distribution components in (7.1), $f_k(y)$, can have different `gamlss.family` distributions, e.g. one can be GA and the other IG.

7.2.1 The likelihood function

Given n independent observations y_i for $i = 1, 2, \dots, n$, from finite mixture model (7.2), the likelihood function is given by

$$L = L(\psi, \mathbf{y}) = \prod_{i=1}^n f_{Y_i}(y_i) = \prod_{i=1}^n \left[\sum_{k=1}^K \pi_k f_k(y_i) \right] \quad (7.3)$$

where $\mathbf{y} = (y_1, y_2, \dots, y_n)$, $f_k(y_i) = f_k(y_i | \theta_k, \mathbf{x}_{ki})$, with log likelihood function given by

$$\ell = \ell(\psi, \mathbf{y}) = \sum_{i=1}^n \log \left[\sum_{k=1}^K \pi_k f_k(y_i) \right] \quad (7.4)$$

We wish to maximize ℓ with respect to ψ , i.e. with respect to θ and π . The problem is that the log function between the two summations in (7.4) makes it difficult. One solution, especially for simple mixtures where no explanatory variables are involved, is to use a numerical maximization technique, e.g. function `optim` in R, to maximize the log likelihood in (7.4) numerically, see for example Venables and Ripley (2002) Chapter 16.

7.2.2 Maximizing the likelihood function using the EM algorithm

Here we will use the EM algorithm, (Dempster, Laird and Rubin, 1977) to maximize (7.4) with respect to ψ , treating all the component indicator variables (i.e δ , defined below) as missing variables.

Let

$$\delta_{ik} = \begin{cases} 1, & \text{if observation } i \text{ comes from component } k \\ 0, & \text{otherwise} \end{cases} \quad (7.5)$$

for $k = 1, 2, \dots, K$ and $i = 1, 2, \dots, n$. Let $\delta_i^T = (\delta_{i1}, \delta_{i2}, \dots, \delta_{ik})$ be the indicator vector for observation i . If observation i comes from component k then δ_i is a vector of zeros, except for the k^{th} value which is $\delta_{ik} = 1$. Let $\delta^T = (\delta_1^T, \delta_2^T, \dots, \delta_n^T)$ combine all the indicator variable vectors. Then the complete data, i.e. observed $\mathbf{y}$ and unobserved δ , has complete likelihood function given by

$$\begin{aligned} L_c = L_c(\psi, \mathbf{y}, \delta) &= f(\mathbf{y}, \delta) = \prod_{i=1}^n f(y_i, \delta_i) \\ &= \prod_{i=1}^n f(y_i | \delta_i) f(\delta_i) \\ &= \prod_{i=1}^n \left\{ \prod_{k=1}^K \left[f_k(y_i)^{\delta_{ik}} \pi_k^{\delta_{ik}} \right] \right\}, \end{aligned} \quad (7.6)$$

since if $\delta_{ik} = 1$ and $\delta_{ik'} = 0$ for $k' \neq k$, then

$$\begin{aligned} f(y_i|\boldsymbol{\delta}_i)f(\boldsymbol{\delta}_i) &= f_k(y_i)\pi_k, \\ &= f_k(y_i)^{\delta_{ik}}\pi_k^{\delta_{ik}} \\ &= \prod_{k=1}^K f_k(y_i)^{\delta_{ik}}\pi_k^{\delta_{ik}} \end{aligned}$$

and hence $f(y_i|\boldsymbol{\delta}_i)f(\boldsymbol{\delta}_i) = \prod_{k=1}^K f_k(y_i)^{\delta_{ik}}\pi_k^{\delta_{ik}}$ for all $\boldsymbol{\delta}_i$.

From (7.6) the complete log likelihood is given by

$$\ell_c = \ell_c(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) = \sum_{i=1}^n \sum_{k=1}^K \delta_{ik} \log f_k(y_i) + \sum_{i=1}^n \sum_{k=1}^K \delta_{ik} \log \pi_k \quad (7.7)$$

If $\boldsymbol{\delta}$ were known then, since $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_K$ have no parameter in common, ℓ_c could be maximized over each $\boldsymbol{\theta}_k$ separately, since the likelihood separates.

The EM algorithm alternates between the E-step and the M-step until convergence. Iteration $(r+1)$ of the EM algorithm comprises an E-step followed by an M-step.

E-step

At the $(r+1)^{th}$ iteration, the E-step finds the conditional expectation of the complete data log likelihood (7.7), over the missing $\boldsymbol{\delta}$, given $\mathbf{y}$ and the current parameter estimates $\hat{\boldsymbol{\psi}}^{(r)}$ from iteration r , i.e.

$$\begin{aligned} Q &= E_{\boldsymbol{\delta}} [\ell_c | \mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)}] \\ &= \sum_{k=1}^K \sum_{i=1}^n \hat{w}_{ik}^{(r+1)} \log f_k(y_i) + \sum_{k=1}^K \sum_{i=1}^n \hat{w}_{ik}^{(r+1)} \log \pi_k \end{aligned} \quad (7.8)$$

where

$$\begin{aligned} \hat{w}_{ik}^{(r+1)} &= E [\delta_{ik} | \mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)}] \\ &= [1 * p(\delta_{ik} = 1 | \mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)})] + [0 * p(\delta_{ik} = 1 | \mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)})] \\ &= p(\delta_{ik} = 1 | \mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)}) \\ &= p(\delta_{ik} = 1 | y_i, \hat{\boldsymbol{\psi}}^{(r)}) \\ &= \frac{p(\delta_{ik} = 1, y_i | \hat{\boldsymbol{\psi}}^{(r)})}{f(y_i | \hat{\boldsymbol{\psi}}^{(r)})} \\ &= \frac{f(y_i | \delta_{ik} = 1, \hat{\boldsymbol{\psi}}^{(r)}) p(\delta_{ik} = 1 | \hat{\boldsymbol{\psi}}^{(r)})}{f(y_i | \hat{\boldsymbol{\psi}}^{(r)})} \\ &= \frac{\hat{\pi}_k^{(r)} f_k(y_i | \hat{\boldsymbol{\theta}}_k^{(r)})}{\sum_{k=1}^K \hat{\pi}_k^{(r)} f_k(y_i | \hat{\boldsymbol{\theta}}_k^{(r)})} \end{aligned} \quad (7.9)$$

Note that $\hat{w}_{ik}^{(r+1)} = p(\delta_{ik} = 1 | y_i, \hat{\boldsymbol{\psi}}^{(r)})$ is the posterior probability that observation y_i comes from component k , given y_i and given $\boldsymbol{\psi} = \hat{\boldsymbol{\psi}}^{(r)}$, while $\hat{\pi}_k^{(r)} = p(\delta_{ik} = 1 | \hat{\boldsymbol{\psi}}^{(r)})$ is the prior (or

mixing) probability that observation y_i comes from component k , given $\boldsymbol{\psi} = \hat{\boldsymbol{\psi}}^{(r)}$ only. On convergence, i.e. $r = \infty$, $\hat{w}_{ik}^{(\infty)}$ and $\hat{\pi}_k^{(\infty)}$ are the **estimated** posterior and prior probabilities that the observation y_i comes from component k , respectively, since $\boldsymbol{\psi}$ is estimated by $\hat{\boldsymbol{\psi}}^{(\infty)}$.

M-step

At the $(r+1)^{th}$ iteration, the M step maximizes Q with respect to $\boldsymbol{\psi}$. Since the parameters $\boldsymbol{\theta}_k$ in $f_k(y_i)$ for $k = 1, 2, \dots, K$ are distinct, (i.e. there are no parameters in common to two or more $\boldsymbol{\theta}_k$'s), Q can be maximized with respect to each $\boldsymbol{\theta}_k$ by maximizing separately the k^{th} part of the first term in (7.8), i.e. maximize $\sum_{i=1}^n \hat{w}_{ik}^{(r+1)} \log f_k(y_i)$ with respect to $\boldsymbol{\theta}_k$, for $k = 1, 2, \dots, K$. Assuming, for $k = 1, 2, \dots, K$, that the k component follows a GAMLSS model, this is just a weighted log likelihood for a GAMLSS model with weights $\hat{w}_{ik}^{(r+1)}$ for $i = 1, 2, \dots, n$. Also the parameter $\boldsymbol{\pi}$ only occurs in the second term in (7.8) and so can be estimated by maximizing the second term, subject to $\sum_{k=1}^K \pi_k = 1$, i.e. where $\pi_k = 1 - \sum_{k=1}^{K-1} \pi_k$, leading to $\hat{\pi}_k^{(r+1)} = \frac{1}{n} \sum_{i=1}^n \hat{w}_{ik}^{(r+1)}$ for $k = 1, 2, \dots, K$.

Summary of the $(r+1)^{th}$ iteration of the EM algorithm

E-step Replace δ_{ik} in (7.7) by $\hat{w}_{ik}^{(r+1)}$ [its conditional expectation given $\mathbf{y}$ and given the current estimate $\hat{\boldsymbol{\psi}}^{(r)}$ from iteration r] obtained from (7.9) for $k = 1, 2, \dots, K$ and $i = 1, 2, \dots, n$ to give (7.8).

M-step

- (1) for each $k = 1, 2, \dots, K$, obtain $\hat{\boldsymbol{\theta}}_k^{(r+1)}$ by fitting the GAMLSS model for the k^{th} component to dependent variable $\mathbf{y}$ with explanatory variables $\mathbf{x}_k$ using prior weights $\hat{\mathbf{w}}_k^{(r+1)}$, where $\mathbf{w}_k^T = (w_{1k}, w_{2k}, \dots, w_{nk})$,
- (2) $\hat{\pi}_k^{(r+1)} = \frac{1}{n} \sum_{i=1}^n \hat{w}_{ik}^{(r+1)}$ for $k = 1, 2, \dots, K$,
- (3) $\hat{\boldsymbol{\psi}}^{(r+1)} = [\hat{\boldsymbol{\theta}}^{(r+1)}, \hat{\boldsymbol{\pi}}^{(r+1)}]$ where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_K)$.

7.2.3 Modelling the mixing probabilities

Here we extend the finite mixture model by assuming that the mixing probabilities π_k for $k = 1, 2, \dots, K$ for observations $i = 1, 2, \dots, n$ are not fixed constants but depend on explanatory variables $\mathbf{x}_0$ and parameters $\boldsymbol{\alpha}$, and hence depend on i , so $f_{Y_i}(y_i) = \sum_{k=1}^K \pi_{ik} f_k(y_i)$. We model the mixing probabilities π_{ik} using a multinomial logistic model where $\boldsymbol{\delta}_i$ is a single draw from a multinomial distribution with probability vector $\boldsymbol{\pi}$, i.e. $\boldsymbol{\delta}_i \sim M(1, \boldsymbol{\pi})$ and

$$\log \left[\frac{\pi_{ik}}{\pi_{iK}} \right] = \boldsymbol{\alpha}_k^T \mathbf{x}_{0i} \quad (7.10)$$

for $k = 1, 2, \dots, K$ and $i = 1, 2, \dots, n$. Hence

$$\pi_{ik} = \frac{\exp \{ \boldsymbol{\alpha}_k^T \mathbf{x}_{0i} \}}{\sum_{k=1}^K \exp \{ \boldsymbol{\alpha}_k^T \mathbf{x}_{0i} \}} \quad (7.11)$$

for $k = 1, 2, \dots, K$ and $i = 1, 2, \dots, n$ where $\alpha_K = \mathbf{0}$. Consequently the complete log likelihood is given by replacing π_k by π_{ik} in equation (7.7) to give

$$\ell_c = \ell_c(\psi, \mathbf{y}, \delta) = \sum_{i=1}^n \sum_{k=1}^K \delta_{ik} \log f_k(y_i) + \sum_{i=1}^n \sum_{k=1}^K \delta_{ik} \log \pi_{ik} \quad (7.12)$$

This results in replacing π_k with π_{ik} in equations (7.8) and (7.9) of the EM algorithm, i.e.

$$Q = \sum_{k=1}^K \sum_{i=1}^n \hat{w}_{ik}^{(r+1)} \log f_k(y_i) + \sum_{k=1}^K \sum_{i=1}^n \hat{w}_{ik}^{(r+1)} \log \pi_{ik} \quad (7.13)$$

where

$$\hat{w}_{ik}^{(r+1)} = \frac{\hat{\pi}_{ik}^{(r)} f_k(y_i | \hat{\theta}_k^{(r)})}{\sum_{k=1}^K \hat{\pi}_{ik}^{(r)} f_k(y_i | \hat{\theta}_k^{(r)})} \quad (7.14)$$

Summary of the $(r+1)^{th}$ iteration of the EM algorithm

E-step Replace δ_{ik} in (7.12) by $\hat{w}_{ik}^{(r+1)}$, obtained from (12.6), for $k = 1, 2, \dots, K$ and $i = 1, 2, \dots, n$, to give (7.13).

M-step

- (1) for each $k = 1, 2, \dots, K$, obtain $\hat{\theta}_k^{(r+1)}$, by fitting the GAMLSS model for the k^{th} component to response variable $\mathbf{y}$ with explanatory variables $\mathbf{x}_k$ using weights $\hat{\mathbf{w}}_k^{(r+1)}$, where $\mathbf{w}_k^T = (w_{1k}, w_{2k}, \dots, w_{nk})$,
- (2) obtain $\hat{\alpha}^{(r+1)}$ by fitting multinomial logistic model (7.10) to pseudo multinomial response variable $\mathbf{y}_p$ with expanded explanatory variables $\mathbf{x}_{0e}$ using prior weights $\hat{\mathbf{w}}^{(r+1)}$,
- (3) $\hat{\psi}^{(r+1)} = [(\hat{\theta}^{(r+1)}, \hat{\alpha}^{(r+1)})]$ where $\theta = (\theta_1, \theta_2, \dots, \theta_K)$.

Note that M-step (2) is achieved by expanding the data set K times as shown in Table 7.1. That is, by setting up the pseudo multinomial response variable $\mathbf{y}_p^T$, taking data values $\mathbf{y}_{pik}^T$, where subscript p stands for pseudo and where $\mathbf{y}_{pik}^T = (0, 0, \dots, 0, 1, \dots, 0)$ is a vector of zeros except for one in the k^{th} cell, for $k = 1, 2, \dots, K$ and $i = 1, 2, \dots, n$, prior weight variable $\hat{\mathbf{w}}^{(r+1)}$ [where $\hat{\mathbf{w}}^T = (\hat{\mathbf{w}}_1^T, \hat{\mathbf{w}}_2^T, \dots, \hat{\mathbf{w}}_K^T)$ and $\hat{\mathbf{w}}_k^T = (\hat{w}_{1k}^T, \hat{w}_{2k}^T, \dots, \hat{w}_{nk}^T)$ for $k = 1, 2, \dots, K$] and fitting a multinomial model to $\mathbf{y}_p$ based on expanded explanatory variable $\mathbf{x}_{0e}$ using weights $\hat{\mathbf{w}}^{(r+1)}$.

7.2.4 Zero components

Special cases of the models described above are distributions which we described earlier as type mixed. For example, the zero adjusted inverse Gaussian distribution (ZAIG) described in Appendix A.6.4 can be thought of as a finite mixture where the first component is identically zero, i.e. $y = 0$, with probability 1. Hence

$$f_1(y) = \begin{cases} 1, & \text{if } y=0 \\ 0, & \text{otherwise.} \end{cases} \quad (7.15)$$

i	k	$\mathbf{y}_e$	$\mathbf{x}_{0e}$	$\mathbf{x}_{1e}$	...	$\mathbf{x}_{Ke}$	multinomial response $\mathbf{y}_p$					weights $\hat{\mathbf{w}}^{(r+1)}$
1	1	$\mathbf{y}$	$\mathbf{X}_0$	$\mathbf{X}_1$	...	$\mathbf{X}_K$	1	0	0	...	0	$\hat{\mathbf{w}}_1^{(r+1)}$
2	1											
⋮	⋮											
n	1											
1	2	$\mathbf{y}$	$\mathbf{X}_0$	$\mathbf{X}_1$	...	$\mathbf{X}_K$	0	1	0	...	0	$\hat{\mathbf{w}}_2^{(r+1)}$
2	2											
⋮	⋮											
n	2											
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	K	$\mathbf{y}$	$\mathbf{X}_0$	$\mathbf{X}_1$	...	$\mathbf{X}_K$	0	0	0	...	1	$\hat{\mathbf{w}}_K^{(r+1)}$
2	K											
⋮	⋮											
n	K											

Table 7.1: Table showing the expansion of data for fitting the multinomial model at M step (2) of the EM algorithm

Distributions of this type can be also fitted with the EM algorithm described in the previous section. The EM algorithm only changes in M-step (1) where the fitting of the first component is omitted (since it has no parameters). The rest of the algorithm is unchanged.

7.3 The `gamlssMX()` function

The function to fit finite mixtures with no parameters in common is `gamlssMX()`. In this section we describe how it works. Examples of using the function are given in the next section. The function `gamlssMX()` has the following arguments:

formula This argument should be a single formula (or a list of formulae of length K the number of components in the mixture) for modelling the predictor for the μ parameter of the model. If a single formula is used then the K mixture components have the same predictor for μ , but different parameters in their predictors (since there are no parameters in common to two or more of the K components). Note that modelling the rest of the distributional parameters can be done by using the usual `gamlss()` formula arguments, e.g. `sigma.fo=~x`, which passes the arguments to `gamlss()`. Again either a single common formula or a list of formula of length K is used.

pi.formula This should be a formula for modelling the predictor for prior (or mixing) probabilities as a function of explanatory variables in the multinomial model (7.10). The default model is constants for the prior (or mixing) probabilities. Note that no smoothing or other additive terms are allowed here, only the usual linear terms. The modelling here is done using the `multinom()` function from package `nnet`.

family This should be a `gamlss.family` distribution (or a list of K distributions). Note that if different distributions are used here, it is preferable (but not essential) that their parameters are comparable for ease of interpretation.

- weights** For declaring prior weights if needed.
- K** For declaring the number of components in the finite mixture with default $K=2$
- prob** For setting starting values for the prior probabilities.
- data** The data frame containing the variables in the fit. Note that this is compulsory if `pi.formula` is used for modelling the prior (or mixing) probabilities.
- control** This argument sets the control parameters for the EM iterations algorithm. The default settings are given in the `MX.control` function
- g.control** This argument can be used to pass to `gamlss()` control parameters, as in `gamlss.control`.
- zero.component** This argument declares whether or not there is a zero component, i.e. y identically equal to zero, $y = 0$, in the finite mixture.
- ...** For extra arguments to be passed to `gamlss()`.

7.4 Examples using the `gamlssMX()` function

7.4.1 The enzyme data

Data summary:

R data file: enzyme in package `gamlss.mx` of dimensions 245×1

variables

act : the enzyme activity in the blood.

purpose: to demonstrate the fitting of mixture distribution to a single variable.

conclusion: a two component Reverse Gumble model fits the data adequately

The data comprise independent measurements of enzyme activity (**act**), in the blood of 245 individuals. As reported by McLachlan and Peel (2000), the data were analyzed by Bechtel *et al.* (1993), using a mixture of skew distributions, who identified 2 components, and subsequently by Richardson and Green (1997) and by McLachlan and Peel (2000) who used a mixture of K normal components with different means and standard deviations and identified 3, or possibly 4, components.

Here we model the distribution of the enzyme activity (**act**) as a mixture of K components having no parameter in common. The form of the component distribution may be the same or different. For example with $K = 2$ we may model enzyme as a mixture of two normal density functions or as a mixture of a normal and a gamma density function.

```
> library(gamlss.mx)
```

```
***** GAMLSS Version 1.7-9 *****
```

```
For more on GAMLSS look at http://www.gamlss.com/
```

```
Type gamlssNews() to see new features/changes/bug fixes.
```

```

> data(enzyme)
> library(MASS)
> truehist(enzyme$act, h = 0.1)
> m1 <- gamlssMX(act ~ 1, data = enzyme, family = NO, K = 2)
> m2 <- gamlssMX(act ~ 1, data = enzyme, family = GA, K = 2)
> m3 <- gamlssMX(act ~ 1, data = enzyme, family = RG, K = 2)
> m4 <- gamlssMX(act ~ 1, data = enzyme, family = c(NO, GA), K = 2)
> m5 <- gamlssMX(act ~ 1, data = enzyme, family = c(GA, RG), K = 2)
> AIC(m1, m2, m3, m4, m5)

```

	df	AIC
m3	5	96.29161
m5	5	97.96889
m2	5	102.42911
m4	5	112.89528
m1	5	119.28006

The best model according to AIC is m3, i.e. the reverse Gumbel (RG) model with two components. In order to be sure that we achieved the global (rather than a local) maximum we repeat the fitting process 10 times using random starting values by using the function `gamlssMXfits()`:

```

> set.seed(1436)
> m3 <- gamlssMXfits(n = 10, act ~ 1, data = enzyme, family = RG,
+   K = 2)

model= 1
model= 2
model= 3
model= 4
model= 5
model= 6
model= 7
model= 8
model= 9
model= 10

> m3

Mixing Family:  c("RG", "RG")

Fitting method: EM algorithm

Call:  gamlssMX(formula = act ~ 1, family = RG, K = 2, data = enzyme)

Mu Coefficients for model: 1
(Intercept)
1.127
Sigma Coefficients for model: 1
(Intercept)

```

```

-1.091
Mu Coefficients for model: 2
(Intercept)
  0.1557
Sigma Coefficients for model: 2
(Intercept)
 -2.641

Estimated probabilities: 0.3760176 0.6239824

Degrees of Freedom for the fit: 5 Residual Deg. of Freedom    240
Global Deviance:      86.2916
      AIC:      96.2916
      SBC:      113.798

```

The best (i.e. lowest global deviance) of the fits is saved in `m3` and printed above. Note that the `mu` and `sigma` parameters listed above are for the predictor models for `mu` and `sigma`. The fitted mixture model for Y (i.e. enzyme activity, `act`) is given by

$$f_Y(y) = 0.376 * f_1(y) + 0.624 * f_2(y)$$

where $f_1(y)$ is a reverse Gumbel distribution, $RG(\mu_1, \sigma_1)$ with $\hat{\mu}_1 = 1.127$ and $\hat{\sigma}_1 = \exp(-1.091) = 0.33588$. and $f_2(y)$ is $RG(\mu_2, \sigma_2)$ with $\hat{\mu}_2 = 0.1557$ and $\log(\hat{\sigma}_2) = -2.641$ i.e. $\hat{\sigma}_2 = \exp(-2.641) = 0.07129$

To check whether more than 2 components are needed, we have fitted the reverse Gumbel model, with $K = 3$, ten times using different starting values and using the function `gamlssMXfits()`. The chosen model with $K = 3$ has a global deviance of 82.692 and an AIC of 98.692. This is larger than the AIC of the model with two components which was 96.29161 so it looks that the $K = 2$ component model is adequate.

We leave it to the readers to try different distributions and different K 's. We point out the possibility that the model with `family=c(RG, RG, NO)`, $K=3$, should be for a suitable candidate for consideration.

Here we plot a histogram of the data together with the fitted two component mixture model `m3` (solid line) and a non-parametric density estimator (dash line) with a bandwidth calculated using the "direct plug-in" estimator of Sheather and Jones (1991):

```

> library(MASS)
> truehist(enzyme$act, h = 0.1)
> fyRG <- dMX(y = seq(0, 3, 0.01), mu = list(1.127, 0.1557), sigma = list(exp(-1.091),
+   exp(-2.641)), pi = list(0.376, 0.624), family = list("RG",
+   "RG"))
> lines(seq(0, 3, 0.01), fyRG, col = "red", lty = 1)
> lines(density(enzyme$act, width = "SJ-dpi"), lty = 2)

```

Note that in the `dMX` function above the prior (or mixing) probabilities, define by `pi`, **must** add up **exactly** to one. The residuals of the final fitted model `m3` can be plotted in the usual way using the function `plot`. The residuals are the usual (normalized quantile) residuals, see Section 2.1.5. The fitted model `m3` appears to be adequate.

```

> plot(m3)

```

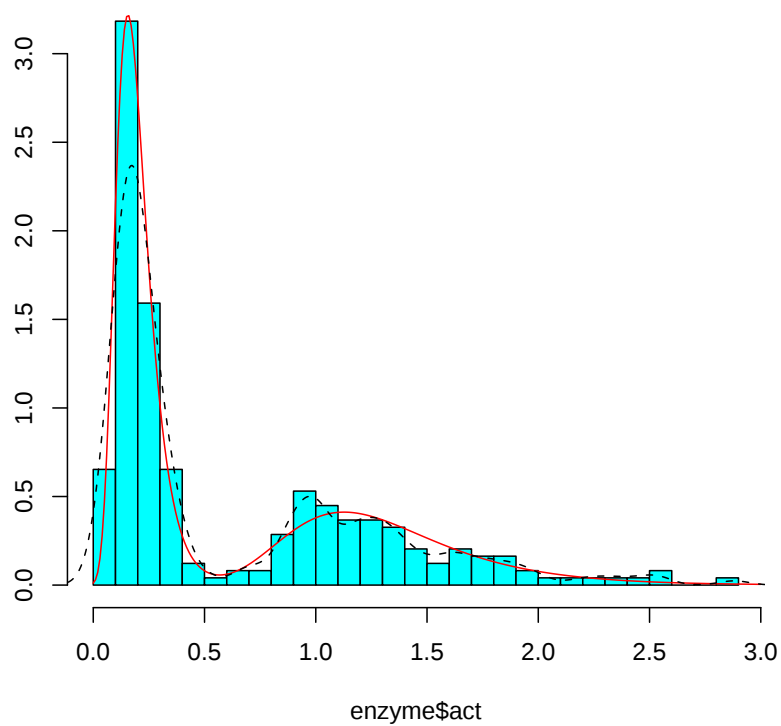


Figure 7.1: A histogram of the enzyme activity data together with a non-parametric density estimator (---) and the fitted two component RG model m3 (—).

```

*****
Summary of the Randomised Quantile Residuals
      mean      = 0.004353063
      variance   = 1.029406
      coef. of skewness = 0.02947350
      coef. of kurtosis = 3.211291
Filliben correlation coefficient = 0.998238
*****

```

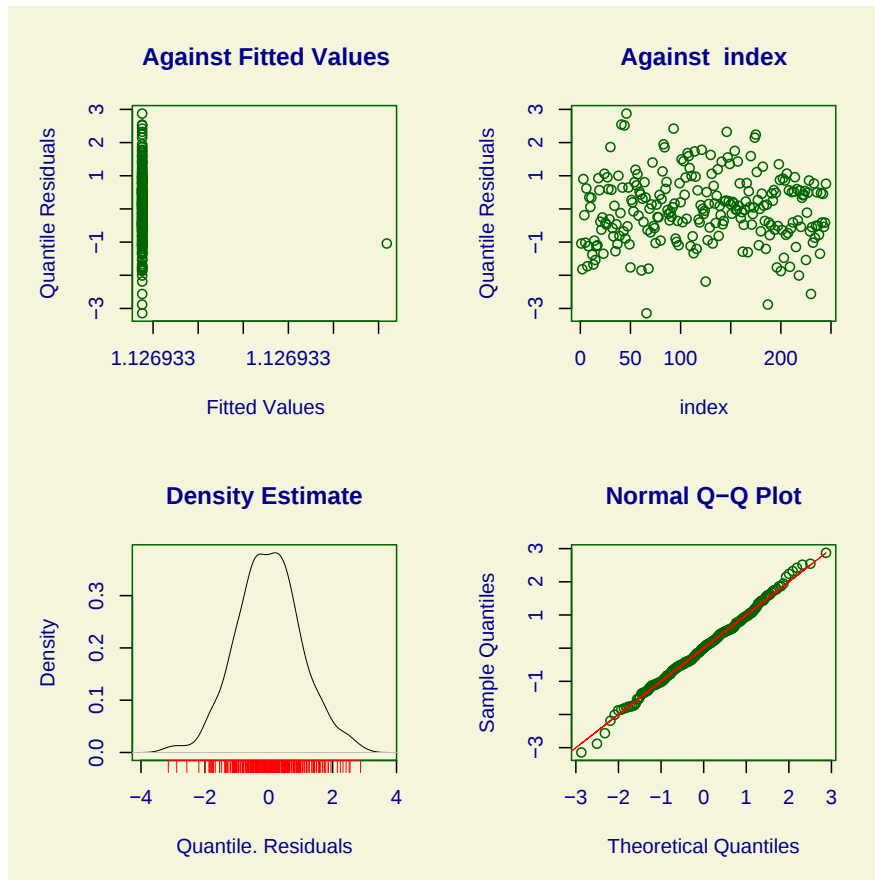


Figure 7.2: The residual plot from the enzyme activity data final two component RG model m3.

7.4.2 The Old Faithful geyser data

The data on the Old Faithful geyser (Azzalini and Bowman, 1990) has two variables, **duration**, the duration of the eruption and **waiting**, the waiting time in minutes until the next eruption. Firstly, the variable **waiting** is used on its own to demonstrate the fitting of a finite mixture to a single response variable. In the second part the data are modified and used as to model of the mixture response variable against an explanatory variable.

Fitting a finite mixture to a single response

Data summary: the old faithful geyser

R data file: `geyser` in package **MASS** of dimensions 299×2

variables

`duration` : the eruption time (in minutes)

`waiting` : the waiting time (in minutes) until the next eruption.

purpose: only the variable `waiting` is used here to demonstrate the fitting of a finite mixture distribution.

conclusion: A two component inverse Gaussian distribution is found to be adequate

Here we study the `waiting` time on its own. We use `waiting` time to demonstrate how to fit a variety of two component mixtures of continuous distributions and then select the ‘best’ using AIC. Two component mixtures of normal, gamma, reverse Gumble, Gumble, logistic and inverse Gaussian distributions are fitted:

```
> library(gamlss.mx)
> library(MASS)
> data(geyser)
> set.seed(1581)
> mNO <- gamlssMX(waiting ~ 1, data = geyser, family = NO, K = 2)
> mGA <- gamlssMX(waiting ~ 1, data = geyser, family = GA, K = 2)
> mRG <- gamlssMX(waiting ~ 1, data = geyser, family = RG, K = 2)
> mGU <- gamlssMX(waiting ~ 1, data = geyser, family = GU, K = 2)
> mLO <- gamlssMX(waiting ~ 1, data = geyser, family = LO, K = 2)
> mIG <- gamlssMX(waiting ~ 1, data = geyser, family = IG, K = 2)
> AIC(mNO, mGA, mRG, mGU, mLO, mIG)
```

	df	AIC
mIG	5	2321.827
mGA	5	2322.764
mRG	5	2323.879
mNO	5	2325.084
mLO	5	2328.147
mGU	5	2420.051

```
> mIG

Mixing Family:  c("IG", "IG")

Fitting method: EM algorithm

Call:  gamlssMX(formula = waiting ~ 1, family = IG, K = 2, data = geyser)

Mu Coefficients for model: 1
(Intercept)
```

```

      4.393
Sigma Coefficients for model: 1
(Intercept)
      -4.642
Mu Coefficients for model: 2
(Intercept)
      4.006
Sigma Coefficients for model: 2
(Intercept)
      -4.304

Estimated probabilities: 0.669591 0.3304090

Degrees of Freedom for the fit: 5 Residual Deg. of Freedom    294
Global Deviance:      2311.83
      AIC:      2321.83
      SBC:      2340.33

```

The best model appears to be `mIG`, the two component inverse Gaussian (IG) model for Y (`=waiting`) given by $f_Y(y) = \hat{\pi}_1 f_1(y) + \hat{\pi}_2 f_2(y) = 0.67 f_1(y) + 0.33 f_2(y)$ where $f_1(y)$ is an inverse Gaussian distribution, $IG(\mu_1, \sigma_1)$ with $\hat{\mu}_1 = \exp(4.393) = 80.88$ and $\hat{\sigma}_1 = \exp(-4.642) = 0.009638$ and $f_2(y)$ is an inverse Gaussian distribution, $IG(\mu_2, \sigma_2)$ with $\hat{\mu}_2 = \exp(4.006) = 54.93$ and $\hat{\sigma}_2 = \exp(-4.304) = 0.01351$. We next plot a histogram of the data together with the fitted two component IG model (solid line) and a non-parametric density estimator (dash line):

```

> truehist(geyser$waiting, h = 2)
> fyIG <- dMX(y = seq(39, 115, 1), mu = list(exp(4.393), exp(4.006)),
+   sigma = list(exp(-4.642), exp(-4.304)), pi = list(0.6695835,
+   0.3304165), family = list("IG", "IG"))
> lines(seq(39, 115, 1), fyIG, col = "red", lty = 1)
> lines(density(geyser$waiting, width = "SJ-dpi"), lty = 2)

```

The residuals of the final fitted model `mIG` are plotted next.

```

> plot(mIG)

*****
      Summary of the Randomised Quantile Residuals
              mean      = -0.001605047
              variance   = 0.994629
      coef. of skewness  = 0.07091106
      coef. of kurtosis  = 2.862159
Filliben correlation coefficient = 0.9968815
*****

```

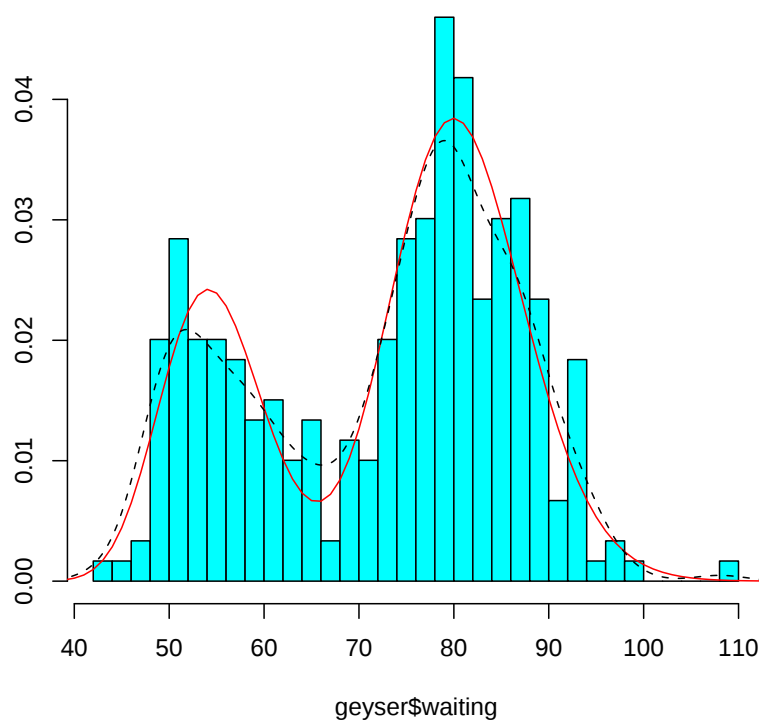


Figure 7.3: A histogram of variable waiting time (to next eruption from the Old Faithful geyser data), together with a non-parametric density estimator (dashed) and the fitted two component IG model (solid).

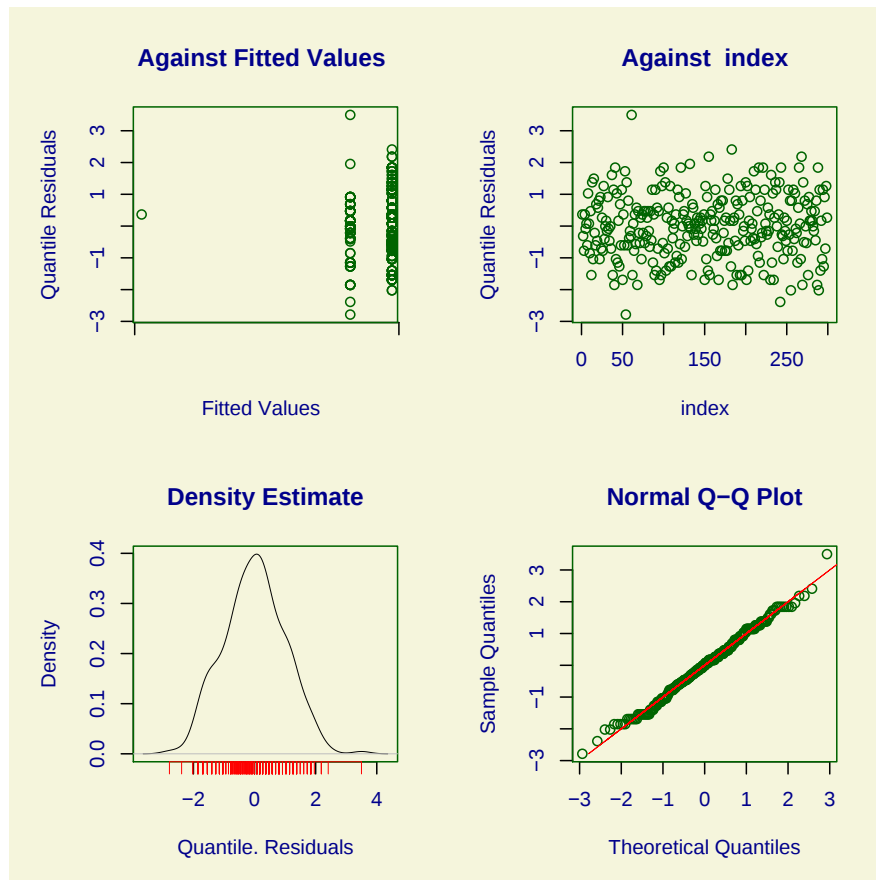


Figure 7.4: The residual plot from the fitted two component IG model for waiting time from the Old Faithful geyser data.

Fitting a finite mixture to a simple regression problem

Data summary: the old faithful geyser

R data file: `geyser2` created from `geyser` of dimensions 298×2

variables

`waiting` : the response variable waiting time (in minutes) until the next eruption.

`duration` : previous duration of the eruption used as explanatory variable

purpose: model the distribution of waiting time given the explanatory variable previous duration

conclusion: The response can be modeled as a mixture of two components each having an inverse Gaussian distribution or a single component inverse Gaussian with smoothing

We now follow Venables and Ripley (2002) p441 and model the probabilities, π 's, of belonging to one of the two mixture components as functions of the previous `duration` of the eruption.

Note that in order to model the π 's, the function `gamlssMX` needs the `data` argument.

We first create a data frame containing the current codewaiting time and the **previous duration** of the eruption. The data are displayed in the left panel of Figure 7.5. Then we fit a normal (NO) two component mixture model, used by Venables and Ripley (2002), and a inverse Gaussian (IG) two component mixture model. First we fit constant models for the predictors of both μ and π , then include duration in the predictor of each of μ and π separately and finally include duration in the predictor of both μ and π . We compare all the models using AIC.

```
> geyser2 <- matrix(0, ncol = 2, nrow = 298)
> geyser2[, 1] <- geyser$waiting[-1]
> geyser2[, 2] <- geyser$duration[-299]
> colnames(geyser2) <- c("waiting", "duration")
> geyser2 <- data.frame(geyser2)
> set.seed(1581)
> mNO1 <- gamlssMX(waiting ~ 1, data = geyser2, family = NO, K = 2)
> mIG1 <- gamlssMX(waiting ~ 1, data = geyser2, family = IG, K = 2)
> mNO2 <- gamlssMX(waiting ~ 1, pi.formula = ~duration, data = geyser2,
+   family = NO, K = 2)
> mIG2 <- gamlssMX(waiting ~ 1, pi.formula = ~duration, data = geyser2,
+   family = IG, K = 2)
> mNO3 <- gamlssMX(waiting ~ duration, pi.formula = ~1, data = geyser2,
+   family = NO, K = 2)
> mIG3 <- gamlssMX(waiting ~ duration, pi.formula = ~1, data = geyser2,
+   family = IG, K = 2)
> mNO4 <- gamlssMX(waiting ~ duration, pi.formula = ~duration,
+   data = geyser2, family = NO, K = 2)
> mIG4 <- gamlssMX(waiting ~ duration, pi.formula = ~duration,
+   data = geyser2, family = IG, K = 2)
> AIC(mNO1, mNO2, mNO3, mNO4, mIG1, mIG2, mIG3, mIG4)
```

```
df      AIC
mIG4 8 1930.034
```

```

mN04  8 1936.679
mN03  7 1953.317
mIG3  7 1961.234
mIG2  6 1970.647
mN02  6 1981.932
mIG1  5 2315.304
mN01  5 2318.472

> mIG4

Mixing Family:  c("IG", "IG")

Fitting method: EM algorithm

Call:  gamlssMX(formula = waiting ~ duration, pi.formula = ~duration,
               family = IG, K = 2, data = geyser2)

Mu Coefficients for model: 1
(Intercept)      duration
   4.09618      0.07007
Sigma Coefficients for model: 1
(Intercept)
   -4.807
Mu Coefficients for model: 2
(Intercept)      duration
   3.6312      0.1935
Sigma Coefficients for model: 2
(Intercept)
   -4.351
model for pi:
              (Intercept) duration
fac.fit2    10.18838 -3.131291

Estimated probabilities:
      pi1      pi2
1 0.91598280 0.0840172
2 0.03058744 0.9694126
3 0.91187830 0.0881217
...

Degrees of Freedom for the fit: 8 Residual Deg. of Freedom    290
Global Deviance:      1914.03
                   AIC:      1930.03
                   SBC:      1959.61

```

The best model using AIC is model `mIG4`. This model is a mixture of two components. In each component waiting time has an inverse Gaussian distribution, with a simple linear regression model in duration for the predictor of the mean and a constant scale. The predictor for the mixing probability is also simple linear regression models in duration. So the final

mixture model `mIG4` is given by

$$f_Y(y) = \hat{\pi}_1 f_1(y) + \hat{\pi}_2 f_2(y)$$

where $f_1(y)$ is an inverse Gaussian distribution $IG(\hat{\mu}_1, \hat{\sigma}_1)$ with

$$\hat{\mu}_1 = \exp \{4.0962 + 0.07008 * \text{duration}\}$$

and

$$\hat{\sigma}_1 = \exp \{-4.807\} = 0.00817$$

and where $f_2(y)$ is also an inverse Gaussian distribution $IG(\hat{\mu}_2, \hat{\sigma}_2)$ with

$$\hat{\mu}_2 = \exp \{3.6313 + 0.1935 * \text{duration}\}$$

and

$$\hat{\sigma}_2 = \exp \{-4.351\} = 0.01289$$

and where

$$\log [\hat{\pi}_2 / (1 - \hat{\pi}_2)] = \boldsymbol{\eta}_\pi = 10.1892 - 3.1318 * \text{duration}$$

Figure 7.5 (a) plots the data together with the fitted means of each of the two components. Figure 7.5 (b) shows the fitted probability of belonging to group 1. As the previous eruption duration increases, the probability that the waiting time will belong to component 2 increases. Figure 7.5 was obtained by the following commands.

```
> op <- par(mfrow = c(1, 2))
> plot(waiting ~ duration, data = geyser2, xlab = "previous duration",
+       ylab = "waiting time", main = "(a)")
> lines(fitted(mIG4$models[[1]])[order(geyser2$duration)] ~
+       geyser2$duration[order(geyser2$duration)],
+       col = "dark green", lty = 3)
> lines(fitted(mIG4$models[[2]])[order(geyser2$duration)] ~
+       geyser2$duration[order(geyser2$duration)],
+       col = "red", lty = 4)
> plot(mIG4$prob[, 1][order(duration)] ~ duration[order(duration)],
+       data = geyser2, xlab = "previous duration", ylab = "probability of component 2",
+       main = "(b)")
> lines(mIG4$prob[, 1][order(duration)] ~ duration[order(duration)],
+       data = geyser2)
> lines(mIG4$prob[, 1][order(duration)] ~ duration[order(duration)],
+       data = geyser2)
> par(op)
```

Figure 7.6 shows the fitted distribution in three dimensions (using the commands below). Figure 7.8 (a) shows it as a `levelplot` (see later for the commands).

```
> grid <- expand.grid(duration = seq(1.5, 5.5, 0.1), waiting = seq(40,
+       110, 0.5))
> etap1 <- 10.19069 - 3.132215 * grid$duration
> etamu1 <- 4.09618 + 0.07007 * grid$duration
```

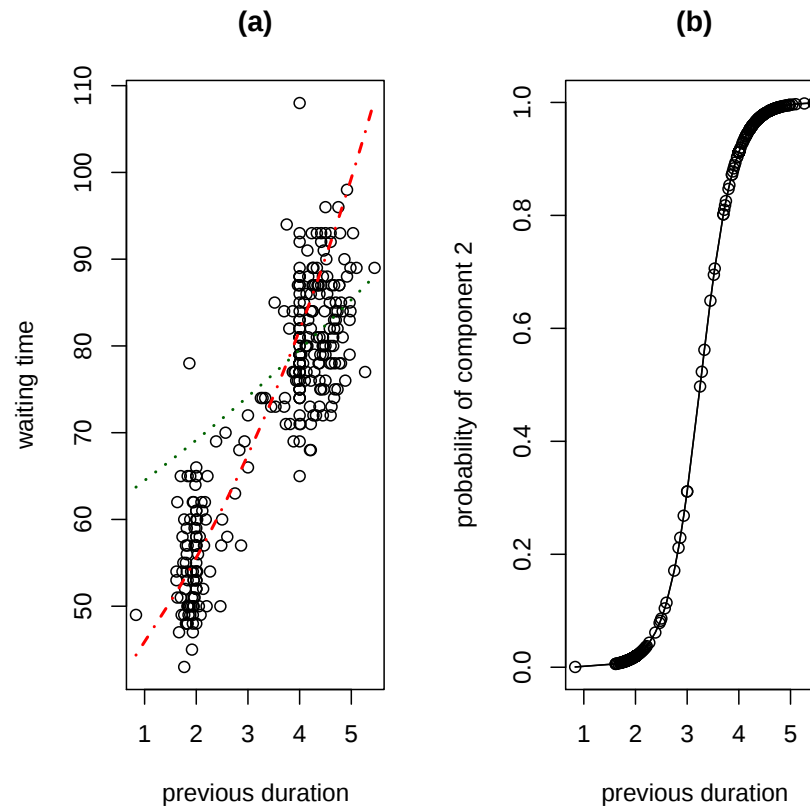


Figure 7.5: (a) A scatter plot of the waiting time against the previous eruption duration from the Old Faithful geyser data together with the fitted values from the two components, (dotted and dashed for component 1 and 2 respectively) (b) a plot of the probability of belonging to component 2 as a function of duration estimated from model `mIG4` .

```

> etamu2 <- 3.6312 + 0.1935 * grid$duration
> pp <- (exp(etapi)/(1 + exp(etapi)))
> grid$f1 <- dMX(y = grid$waiting, mu = list(exp(etamu1), exp(etamu2)),
+   sigma = list(exp(-4.807), exp(-4.351)), pi = list(1 - pp,
+   pp), family = list("IG", "IG"))
> library(lattice)
> wireframe(f1 ~ duration * waiting, data = grid, aspect = c(1,
+   0.5), drape = TRUE)

```

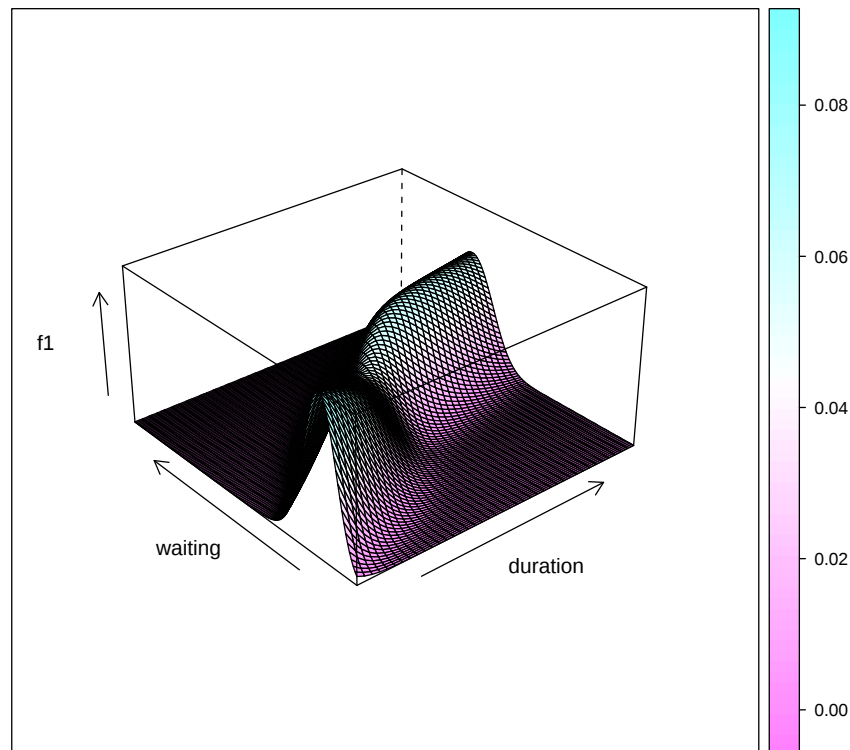


Figure 7.6: Fitted conditional probability density function ($f1$) for waiting time given previous eruption duration for model `mIG4`.

Model `mIG4` provides us with an example of a regression model where the response variable has a mixture distribution with two components and where the probability of belonging to each component of the mixture is modelled as a function of a single explanatory variable. The model is appropriate if modelling the probability of belonging to a component is of interest. If, on the other hand, the interest lies in just modelling the waiting time as a function of the previous duration, a simple GAMLSS model could be appropriate.

We will try here to compare the `mIG4` (finite mixture) model with a single component model (not a mixture) using the inverse Gaussian distribution with a regression models in the previous

duration for both μ and σ . A flexible smoothing cubic spline function as a function of the previous duration is also used for μ and σ .

```
> mIG5 <- gamlss(waiting ~ duration, sigma.formula = ~duration,
+ data = geyser2, family = IG, trace = FALSE)
> mIG6 <- gamlss(waiting ~ cs(duration), sigma.formula = ~duration,
+ data = geyser2, family = IG, trace = FALSE)
> mIG7 <- gamlss(waiting ~ cs(duration), sigma.formula = ~cs(duration),
+ data = geyser2, family = IG, trace = FALSE)
> mIG8 <- gamlss(waiting ~ cs(duration), sigma.formula = ~1, data = geyser2,
+ family = IG, trace = FALSE)
> AIC(mIG4, mIG5, mIG6, mIG7, mIG8)
```

	df	AIC
mIG6	7.000763	1928.512
mIG4	8.000000	1930.034
mIG7	10.000071	1933.061
mIG8	6.000815	1957.221
mIG5	4.000000	1958.542

Model mIG6 is marginally better than model mIG4 in terms of AIC. Figure 7.7 compares the fitted means for the two models. The smooth fitted lines of model mIG6 follows closely the component 1 line of model mIG4 up to duration around 4 and then the component 2 line. The two models behave very similar as far the model of the mean is concerned.

```
> plot(waiting ~ duration, data = geyser2, xlab = "previous duration",
+ ylab = "waiting time")
> lines(fitted(mIG4$models[[1]])[order(geyser2$duration)] ~
+ geyser2$duration[order(geyser2$duration)],
+ col = "green", lty = 3)
> lines(fitted(mIG4$models[[2]])[order(geyser2$duration)] ~
+ geyser2$duration[order(geyser2$duration)],
+ col = "red", lty = 4)
> lines(fitted(mIG6)[order(duration)] ~ duration[order(duration)],
+ data = geyser2, col = "blue", lty = 1)
```

Figures 7.8 (a) and (b) shows levelplots of the conditional probability density function (pdf) for waiting time given the previous eruption time for models (a) mIG4 and (b) mIG6 respectively obtained using the commands below. The plots are similar, although model mIG4 has a higher conditional pdf for waiting time around 50 when previous duration is less than 2.

```
> mu <- predict(mIG6, what = "mu", type = "response", newdata = grid[,
+ c("waiting", "duration")], data = geyser2)
> sigma <- predict(mIG6, what = "sigma", type = "response", newdata = grid[,
+ c("waiting", "duration")], data = geyser2)
> grid$f2 <- dIG(y = grid$waiting, mu = mu, sigma = sigma)
> print(levelplot(f1 ~ duration * waiting, data = grid, colorkey = F,
+ at = seq(0, 0.075, 0.001), xlab = "previous duration", ylab = "waiting time",
+ col.regions = rev(trellis.par.get("regions")$col), main = "(a)",
+ split = c(1, 1, 2, 1), more = TRUE))
```

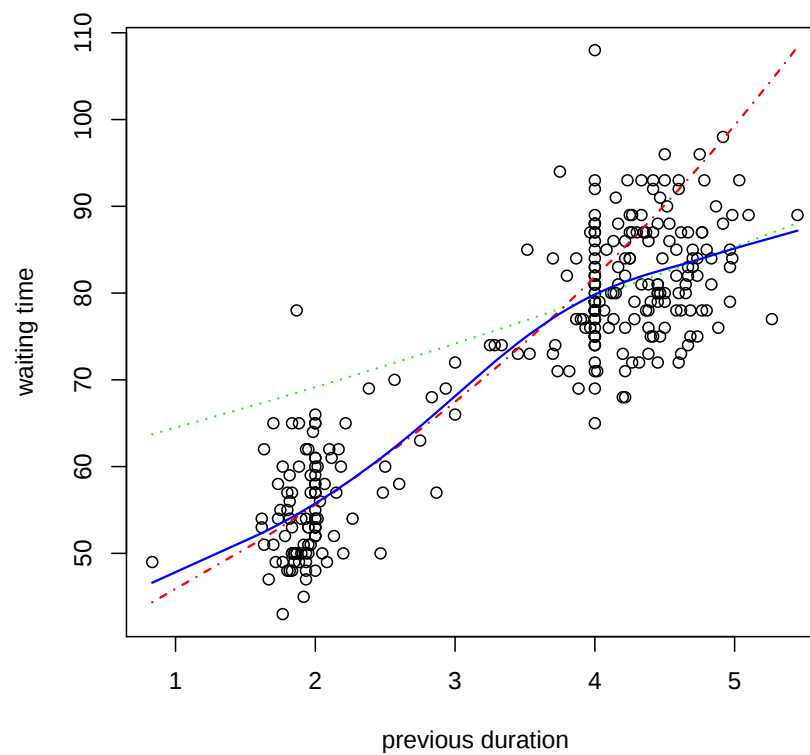


Figure 7.7: Comparison of the fitted values for μ for models mIG4 (dashed and dotted lines) and mIG6 (solid line).

```

> print(levelplot(f2 ~ duration * waiting, data = grid, colorkey = F,
+   at = seq(0, 0.075, 0.001), xlab = "previous duration", ylab = "waiting time",
+   col.regions = rev(trellis.par.get("regions")$col), main = "(b)"),
+   split = c(2, 1, 2, 1))

```

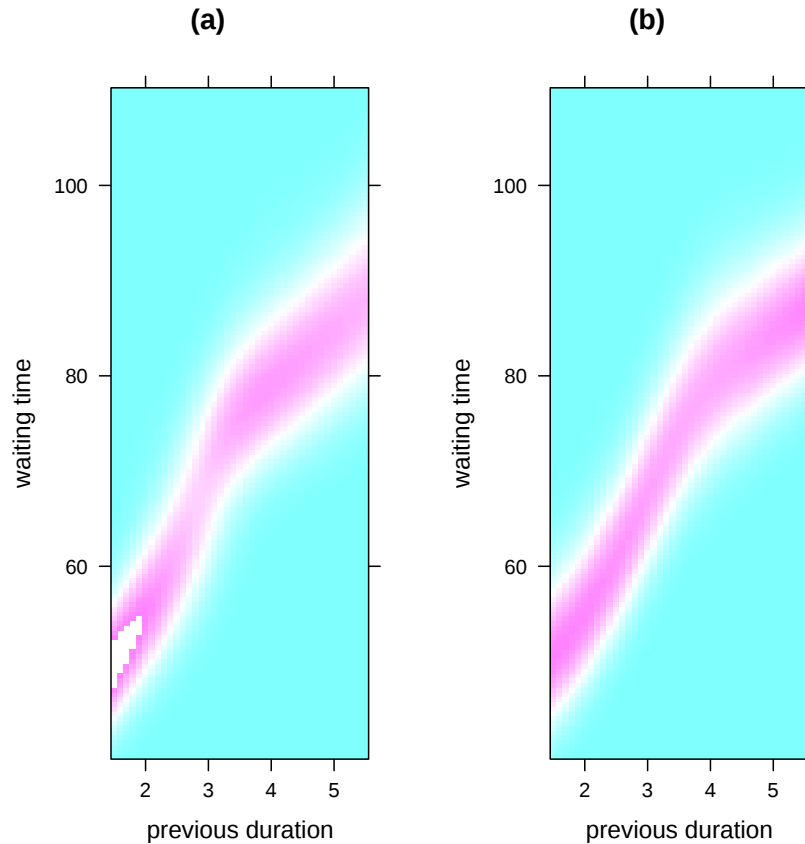


Figure 7.8: Levelplot of the fitted conditional probability density function of the waiting time given the previous eruption time for models (a) `mIG4` and model (b) `mIG6`.

7.5 Finite mixtures with parameters in common

Here the K components of the mixture may have parameters in common, i.e. the parameter sets $(\theta_1, \theta_2, \dots, \theta_k)$ are not disjoint. The prior (or mixing) probabilities are either assumed to be constant (as in function `gamlssNP()`) or may depend on explanatory variables $\mathbf{x}_0$ and parameters α through a multinomial logistic model as in Section 7.2.3. We assume that the K components $f_k(y) = f_k(y|\theta_k, \mathbf{x}_k)$ for $k = 1, 2, \dots, K$ can be represented by GAMLSS models. Note that since some of the parameters may be common to the K components, the distribution used must be the same for all K components. Similarly the link functions of the distribution parameters must be the same for all K components. GAMLSS models have up to four distributional parameters μ ,

σ , ν and τ . In our notation in this Chapter, the parameter vector θ_k contains all the parameters in the (linear) predictor models for μ , σ , ν and τ for component k , for $k = 1, 2, \dots, K$. Here are some examples to clarify this.

Example 1, Mixture of K Poisson regression models: $f(y) = \sum_{k=1}^K \pi_k f_k(y)$ where $f_k(y)$ is $PO(\mu_k)$ for $k = 1, 2, \dots, K$, and where $\log \mu_k = \beta_{0k} + \beta_1 x$. Here the slope parameter β_1 , a predictor parameter for the distribution parameter μ_k , is the same for all K components, but the intercept parameter β_{0k} depends on k , for $k = 1, 2, \dots, K$.

Example 2, Mixture of K negative binomials regression models: Let $f_k(y)$ be $NBI(\mu_k, \sigma_k)$ for $k = 1, 2, \dots, K$, where $\log \mu_k = \beta_{10k} + \beta_{11}x$ and $\log \sigma_k = \log \sigma = \beta_{20} + \beta_{21}x$. Here the predictor slope parameter β_{11} for μ_k and all predictor parameters for σ are the same for all K components, but the predictor intercept parameter β_{10k} for μ_k depends on k , for $k = 1, 2, \dots, K$.

Example 3, Mixture of K BCT models: Let $f_k(y) = BCT(\mu_k, \sigma_k, \nu_k, \tau_k)$ for $k = 1, 2, \dots, K$, where $\log \mu_k = \beta_{10k} + \beta_{11k}x$, $\log \sigma_k = \beta_{20k} + \beta_{21k}x$, $\nu_k = \nu = \beta_{30}$ and $\log \tau_k = \log \tau = \beta_{40}$. Here predictor parameters β_{10k} and β_{11k} for μ and β_{20k} and β_{21k} for σ depend on k for $k = 1, 2, \dots, K$, but parameters β_{30} for ν and β_{40} for τ are the same for all k components.

7.5.1 Maximizing the likelihood using the EM algorithm

As in Section 7.2.3 the complete log likelihood is given by (7.12). The following is a summary of the EM algorithm suitable for dealing with GAMLSS models with common parameters in the mixture.

Summary of the $(r+1)^{th}$ iteration of the EM algorithm

E-step Replace δ_{ik} in (7.12) by $\hat{w}_{ik}^{(r+1)}$, obtained from (12.6) for $k = 1, 2, \dots, K$ and $i = 1, 2, \dots, n$ to give (7.13), i.e. $Q = \sum_{k=1}^K \sum_{i=1}^n \hat{w}_{ik}^{(r+1)} \log f_k(y_i) + \sum_{k=1}^K \sum_{i=1}^n \hat{w}_{ik}^{(r+1)} \log \pi_{ik}$.

M-step

- (1) Since components $f_k(y)$ for $k = 1, 2, \dots, K$ have parameters in common, Q cannot be maximized separately with respect to each θ_k . Obtain $\hat{\theta}^{(r+1)}$ by fitting a single GAMLSS model to an expanded response variable $\mathbf{y}_e$, with expanded explanatory variable design matrix $\mathbf{X}_e$, using weights $\hat{\mathbf{w}}^{(r+1)}$ (see Table 7.2).
- (2) Obtain $\hat{\alpha}^{(r+1)}$ by fitting a multinomial logistic model as in Section 7.2.3.
- (3) $\hat{\psi}^{(r+1)} = [\hat{\theta}^{(r+1)}, \hat{\alpha}^{(r+1)}]$

Note that the M step (1) is achieved by expanding the data set K times as in Table 7.2. This method is identical to the method used in Aitkin *et al.* (2006) but here we are not restricting ourselves to the exponential family. The column headed as **MASS** identifies the K mixture components. This column is declared as a factor in the R implementation of the EM algorithm. If this factor **MASS** is included in the predictor for a distribution parameter μ , σ , ν , or τ , then the predictor intercepts differs between the K components. If an interaction between this factor **MASS** and an explanatory variable x is included in the predictor model for a distribution parameter, then the coefficient of x differ between the K components. Note however that the syntax used in `gamlssNP()` for the interaction between **MASS** and x in the predictor for μ is achieved using the `random=~x` argument (see Section 7.7 for an example).

i	MASS	$\mathbf{y}_e$	$\mathbf{X}_e$	$\hat{\mathbf{w}}^{(r+1)}$
1	1	$\mathbf{y}$	$\mathbf{X}$	$\hat{\mathbf{w}}_1^{(r+1)}$
2	1			
$\vdots$	$\vdots$			
n	1			
1	2	$\mathbf{y}$	$\mathbf{X}$	$\hat{\mathbf{w}}_2^{(r+1)}$
2	2			
$\vdots$	$\vdots$			
n	2			
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	K	$\mathbf{y}$	$\mathbf{X}$	$\hat{\mathbf{w}}_K^{(r+1)}$
2	K			
$\vdots$	$\vdots$			
n	K			

Table 7.2: Table showing the expansion of data use in M-step (1) of the EM algorithm for fitting the common parameter mixture model

7.6 The gamlssNP() function

The function to fit finite mixtures with parameters in common is **gamlssMX**. In this section we describe how it works. Examples of using the function are given in the next section. The function **gamlssMX** has the following arguments:

formula This argument should be a formula defining the response variable and explanatory fixed effects terms for the μ parameter of the model. Note that modelling the rest of the distribution parameters can be done by using the usual formulae, e.g. `sigma.fo = x`, which passes the arguments to **gamlss()**

random This should be a formula defining the random part of the model (for random effect models). This formula is also used for fixed effect mixture models to define interactions of the factor **MASS** with explanatory variables x in the predictor for μ (needed to request different coefficients in x in the predictor of μ for the K components).

family A gamlss family distribution.

data This should be a data frame. Note that this argument is mandatory for this function even if the data are attached. This is because the data frame is used to expand the data as in Table 7.2.

K Declaring the number of mixture components (in fixed effects finite mixture models), or the number of mass points or integration quadrature points (for random effects models)

mixture Defining the mixing distribution, "np" for non-parametric finite mixtures or "gq" for Gaussian quadrature.

tol This defines the tolerance scalar usually between zero and one, used for changing the starting values.

weights For prior weights

control This sets the control parameters for the EM iterations algorithm. The default setting is the `NP.control` function.

g.control This is for controlling the `gamlss` control function, `gamlss.control`, passed to the `gamlss` fit

... For extra arguments

7.7 Examples using the `gamlssNP()` function

7.7.1 The animal brain data

Data summary: the animal brain data

R data file: `brains` in package `gamlss.mx` of dimensions 28×2 (identical to `Animals` in package (`MASS`))

variables

`brain` : brain weight in g.

`body` : body weight in kg.

purpose: To fit a finite mixture model with different intercepts.

conclusion: A three component normal distribution mixture is found to be adequate

The brain size (`brain`) and the body weight (`body`) were recorded for 28 different animals. Since the distribution of both brain size and body weight are highly skewed a log transformation was applied to each variable to give transformed variables `lbrain` and `lbody`. The resulting data are plotted in Figure 7.9.

```
> library(gamlss.mx)
> data(brains)
> brains$lbrain <- log(brains$brain)
> brains$lbody <- log(brains$body)
> with(brains, plot(lbrain ~ lbody, ylab = "log brain", xlab = "log body"))
```

A normal error linear regression model of `lbrain` against `lbody` has a highly significant slope for `lbody` but it is believed that the data may represent different stages of evolution and so a mixture models is fitted to the data. In the mixture model, the evolution stage was represented by a shift in the intercept of the regression equation. Normal mixture models with K equal to 1, 2, 3, 4 are fitted below. Models `br.2`, `br.3` and `br.4` are models with different intercepts for the K components, where $K = 2, 3$ and 4 respectively. Slopes are the same for the K components, so parallel lines are fitted (see later for how different slopes can be incorporated in the model).

```
> br.1 <- gamlss(lbrain ~ lbody, data = brains)
```

```
GAMLSS-RS iteration 1: Global Deviance = 101.2578
```

```
GAMLSS-RS iteration 2: Global Deviance = 101.2578
```

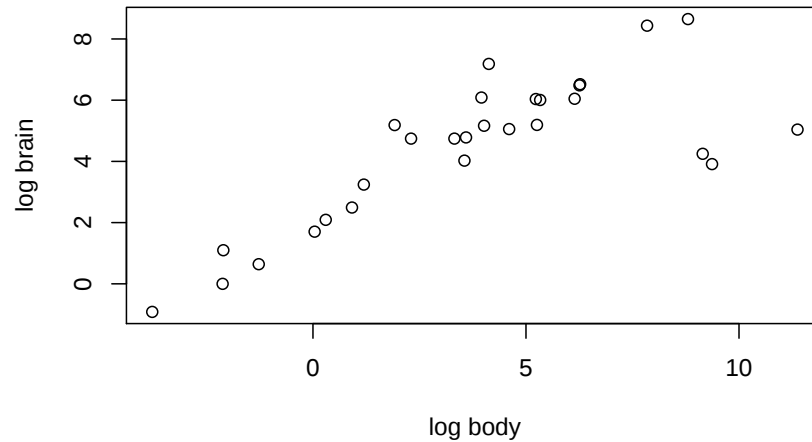


Figure 7.9: A plot of the brain size data.

```
> br.2 <- gamlssNP(formula = lbrain ~ lbody, mixture = "np", K = 2,
+   tol = 1, data = brains, family = NO)
```

```
1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..17 ..18 ..19 ..
EM algorithm met convergence criteria at iteration 33
Global deviance trend plotted.
EM Trajectories plotted.
```

```
> br.3 <- gamlssNP(formula = lbrain ~ lbody, mixture = "np", K = 3,
+   tol = 1, data = brains, family = NO)
```

```
1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..
EM algorithm met convergence criteria at iteration 14
Global deviance trend plotted.
EM Trajectories plotted.
```

```
> br.4 <- gamlssNP(formula = lbrain ~ lbody, mixture = "np", K = 4,
+   tol = 1, data = brains, family = NO)
```

```
1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..17 ..18 ..19 ..
EM algorithm met convergence criteria at iteration 29
Global deviance trend plotted.
EM Trajectories plotted.
```

We compare the models using each of the criteria AIC and SBC:

```
> GAIC(br.1, br.2, br.3, br.4)
```

```

      df      AIC
br.3  7  79.15079
br.4  9  83.15613
br.2  5  85.95938
br.1  3 107.25779

```

```
> GAIC(br.1, br.2, br.3, br.4, k = log(length(brains$body)))
```

```

      df      AIC
br.3  7  88.47622
br.2  5  92.62040
br.4  9  95.14598
br.1  3 111.25440

```

Changing the starting values by trying different values for `tol` (e.g. trying each of the values 0.1, 0.2, ..., 1 in turn), for models `br.2`, `br.3` and `br.4`, did not change the values of AIC and SBC given by the two GAIC commands above. The model `br.3` with three components (i.e. three parallel lines) is selected by both AIC and SBC criteria. We now print model `br.3` and its estimated (fitted) posterior probabilities.

```
> br.3
```

```
Mixing Family:  c("NO Mixture with NP", "Normal Mixture with NP")
```

```
Fitting method: EM algorithm
```

```
Call:  gamlssNP(formula = lbrain ~ lbody, family = NO, data = brains,
               K = 3, mixture = "np", tol = 1)
```

```
Mu Coefficients :
```

```
(Intercept)      lbody      MASS2      MASS3
      -3.072      0.750      4.981      6.553
```

```
Sigma Coefficients :
```

```
(Intercept)
      -0.9387
```

```
Estimated probabilities: 0.1071429 0.7514161 0.1414410
```

```
Degrees of Freedom for the fit: 7 Residual Deg. of Freedom  21
```

```
Global Deviance:      65.1508
                  AIC:      79.1508
                  SBC:      88.4762
```

```
> br.3$post.prob
```

```
[[1]]
      [,1]      [,2]      [,3]
[1,]  0 9.999624e-01 3.760045e-05
[2,]  0 9.999995e-01 4.736429e-07
[3,]  0 9.996309e-01 3.691210e-04
```

```

[4,]    0 9.979683e-01 2.031733e-03
[5,]    0 9.999947e-01 5.254125e-06
[6,]    1 0.000000e+00 0.000000e+00
[7,]    0 9.583487e-01 4.165135e-02
[8,]    0 9.995208e-01 4.792198e-04
[9,]    0 9.999824e-01 1.764759e-05
[10,]   0 1.617020e-01 8.382980e-01
[11,]   0 9.947820e-01 5.217995e-03
[12,]   0 9.999769e-01 2.306099e-05
[13,]   0 9.998409e-01 1.590788e-04
[14,]   0 3.157024e-06 9.999968e-01
[15,]   0 9.997563e-01 2.436742e-04
[16,]   1 0.000000e+00 0.000000e+00
[17,]   0 1.044992e-04 9.998955e-01
[18,]   0 9.999998e-01 2.035525e-07
[19,]   0 9.999978e-01 2.187091e-06
[20,]   0 9.999398e-01 6.024621e-05
[21,]   0 9.999799e-01 2.013594e-05
[22,]   0 9.992899e-01 7.101261e-04
[23,]   0 9.999975e-01 2.489188e-06
[24,]   0 6.263055e-02 9.373694e-01
[25,]   1 0.000000e+00 0.000000e+00
[26,]   0 9.999977e-01 2.336595e-06
[27,]   0 8.662450e-01 1.337550e-01
[28,]   0 9.999999e-01 6.645917e-08

```

So model `br.3` can be presented as $Y \sim NO(\hat{\mu}, \hat{\sigma})$ where

$$\hat{\mu} = \begin{cases} -3.072 + 0.750x, & \text{with probability 0.107} \\ 1.909 + 0.750x, & \text{with probability 0.751} \\ 3.481 + 0.750x, & \text{with probability 0.141} \end{cases} \quad (7.16)$$

and $\hat{\sigma} = 0.391$. [Note that the intercept for the second component in (7.16) is obtained from the estimated parameter coefficients for μ by $1.909 = -3.072 + 4.981$, since MASS2 gives the adjustment to the intercept for the second mixture component; similarly for MASS3.] The output given by `br.3$post.prob` contains the estimated posterior probabilities of each of the observations in the data set belonging to each of the 3 components. These are the fitted weights $\hat{w}_{ik}$ given by (7.9) on convergence of the EM algorithm.

A plot of the data together with the fitted values for the μ parameter of model `br.3` are shown in Figure 7.10. Each observation of the data was allocated to the component for which it had the highest posterior probability and the observations are plotted in the command below with circles (colour red), squares (colour green) and diamonds (colour blue) representing allocation to each of the 3 components. Note that since the parameter μ in this (normal distribution) case is the mean of the distribution the lines are the fitted means of the conditional distributions $f_k(y)$ for $k = 1, 2, 3$. Figure 7.10 is obtained by :

```

> with(brains, plot(lbody, lbrain, pch = c(21, 22, 23)[max.col(br.3$post.prob[[1]])],
+   bg = c("red", "green3", "blue")[max.col(br.3$post.prob[[1]])]))
> for (k in 1:3) {
+   with(brains, lines(fitted(br.3, K = k)[order(lbody)] ~ lbody[order(lbody)]),

```

```
+      lty = k))
+ }
```

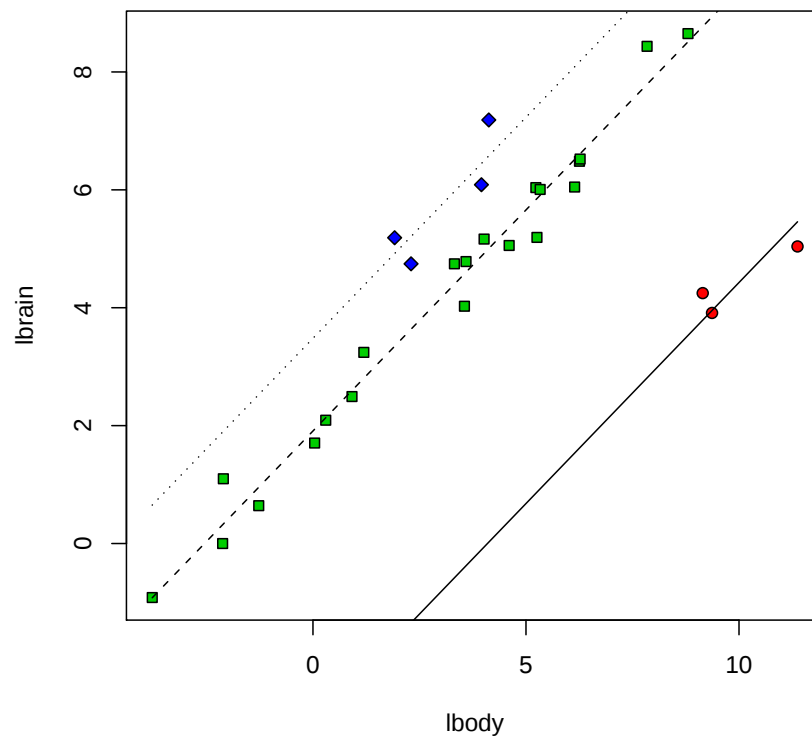


Figure 7.10: A plot of the brain size data together with a plot of the three component fitted means of log brain size (`lbrain`) against log body size (`lbody`), (solid, dashed and dotted for component 1,2 and 3 respectively).

The weighted average for the (conditional) parameters $\hat{\mu}$ for the $K(=3)$ components for each observation, i.e. $\sum_{k=1}^K \hat{\pi}_k \hat{\mu}_{ik}$ can be obtained using the command `fitted(br.3, K=0)`. Since the parameter μ is, in this case, the mean of the normal distribution, this gives the marginal mean of the response variable `lbrain` given the explanatory variable `lbody`.

Note how the marginal mean, using the function `fitted()`, is obtained here compared to the conditional means. If the argument `K` of the `fitted()` function has any value in the range 1, 2, 3, (that is the range of permissible values for the model `br.3`), then the conditional parameters is given. For any other value the average μ is given. This will be the marginal mean only if parameter μ is the mean of the conditional distribution for each component.

A residual plot of the finite mixture model is obtained the usual way using the function `plot()`.

```
> plot(br.3)
```

```

*****
Summary of the Randomised Quantile Residuals
      mean      = -0.004003875
      variance   = 1.052469
      coef. of skewness = 0.1668313
      coef. of kurtosis = 2.739025
Filliben correlation coefficient = 0.9962244
*****

```

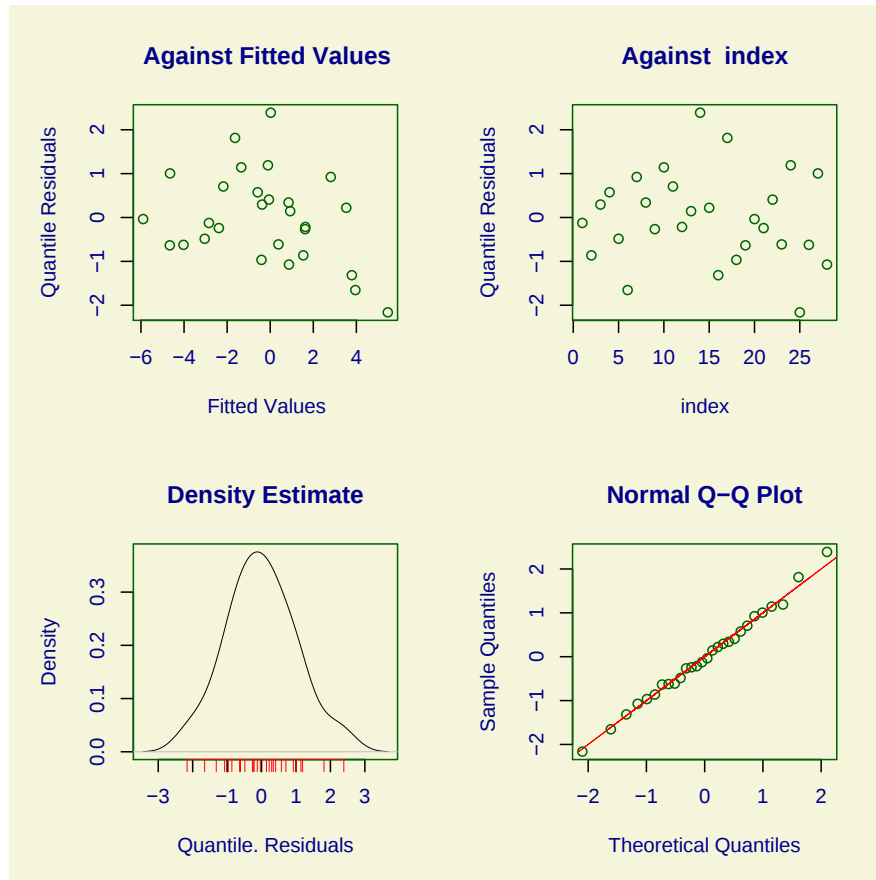


Figure 7.11: The residual plot of model `br.3` for the animal brain size data.

There are several different models that we could fit here depending on which parameters are common to the $K = 3$ components in the model. Table 7.3 shows possible alternative models and the code below shows how to fit them:

```

> br.31 <- gamlssNP(formula = lbrain ~ lbody, sigma.fo = ~MASS,
+   mixture = "np", K = 3, tol = 1, data = brains, family = NO)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..17 ..18 ..19 ..
EM algorithm met convergence criteria at iteration    28

```

model	μ intercept	μ slope	σ
br.3	different	same	same
br.31	different	same	different
br.32	different	different	same
br.33	different	different	diferent

Table 7.3: Possible alternative models for the animal brain data

Global deviance trend plotted.

EM Trajectories plotted.

```
> br.32 <- gamlssNP(formula = lbrain ~ lbody, random = ~lbody,
+   sigma.fo = ~1, mixture = "np", K = 3, tol = 1, data = brains,
+   family = NO)
```

```
1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..
```

EM algorithm met convergence criteria at iteration 16

Global deviance trend plotted.

EM Trajectories plotted.

```
> br.33 <- gamlssNP(formula = lbrain ~ lbody, random = ~lbody,
+   sigma.fo = ~MASS, mixture = "np", K = 3, tol = 1, data = brains,
+   family = NO)
```

```
1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..17 ..
```

EM algorithm met convergence criteria at iteration 17

Global deviance trend plotted.

EM Trajectories plotted.

We compare the models using each of the criteria AIC and SBC:

```
> GAIC(br.3, br.31, br.32, br.33)
```

```
      df      AIC
br.32  9 77.31133
br.3   7 79.15079
br.33 11 80.26824
br.31  9 81.93037
```

```
> GAIC(br.3, br.31, br.32, br.33, k = log(length(brains$lbody)))
```

```
      df      AIC
br.3   7 88.47622
br.32  9 89.30117
br.31  9 93.92021
br.33 11 94.92249
```

Model **br.3** has the smallest SBC. [Note model **br.32** has the smallest AIC, however with so many parameters in the model and so few data points it is not sensible to try to interpreted this model.] Note also that since model **br.33** has components with no parameters in common it could also be fitted using the **gamlssMX** function.

7.8 Bibliographic notes

There is an extensive literature on mixture distributions and their use in modelling data. Everitt and Hand (1981), Titterton et al. (1985), Lindsay (1995), Böhning (1999) and McLachlan and Peel (2000) are some of the books dedicated exclusively to mixture distributions. Aitkin, Francis and Hinde (2005) include useful chapters related to mixture distributions.

Chapter 8

Model selection

8.1 Introduction on model selection

Let $\mathcal{M} = \{\mathcal{D}, \mathcal{G}, \mathcal{T}, \boldsymbol{\lambda}\}$ represent the GAMLSS model (2.1) where the components: (i) $\mathcal{D}$ specifies the distribution of the response variable (ii) $\mathcal{G}$ the set of link functions $[g_1(), g_2(), g_3(), g_4()]$ for parameters (μ, σ, ν, τ) (iii) $\mathcal{T}$ the set of predictor terms $(t_\mu, t_\sigma, t_\nu, t_\tau,)$ for predictors $(\eta_\mu, \eta_\sigma, \eta_\nu, \eta_\tau)$ and (iv) $\boldsymbol{\lambda}$ the set of hyperparameters.

For a specific data set, the GAMLSS model building process consists of comparing many different competing models for which different combinations of components $\mathcal{M} = \{\mathcal{D}, \mathcal{G}, \mathcal{T}, \boldsymbol{\lambda}\}$ are tried.

Inference about quantities of interest can be made either conditionally on a single selected 'final' model or by averaging between selected models. If the purpose of the study is to describe the data parsimoniously, then a single 'final' model is usually sufficient. Conditioning on a single final model was criticized by Draper (1995) and Madigan and Raftery (1994) since it ignores model uncertainty and generally leads to the underestimation of the uncertainty about quantities of interest. Averaging between selected models can reduce this underestimation, Hjort and Claeskens (2003).

As with all scientific inferences the determination of the adequacy of any model depends on the substantive question of interest and requires subject specific knowledge.

For parametric GAMLSS models (2.3) or (2.5) each fitted GAMLSS model $\mathcal{M}$ can be assessed by its fitted global deviance, GD , given by $GD = -2\ell(\hat{\boldsymbol{\theta}})$ where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4) = (\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\nu}, \boldsymbol{\tau})$ and $\ell(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^n \ell(\hat{\boldsymbol{\theta}}^i)$ is the fitted (or maximized) log-likelihood function and $\boldsymbol{\theta}^i = (\mu_i, \sigma_i, \nu_i, \tau_i)$. Two nested parametric GAMLSS models, $\mathcal{M}_0$ and $\mathcal{M}_1$, with fitted global deviances GD_0 and GD_1 and error degrees of freedom df_{e0} and df_{e1} respectively may be compared using the (generalized likelihood ratio) test statistic $\Lambda = GD_0 - GD_1$ which has an asymptotic Chi-squared distribution under $\mathcal{M}_0$, with degrees of freedom $d = df_{e0} - df_{e1}$, (given that the usual regularity conditions are satisfied). For each model $\mathcal{M}$ the error degrees of freedom df_e is defined by $df_e = n - \sum_{k=1}^p df_{\theta_k}$, where df_{θ_k} is the degrees of freedom used in the predictor model for distribution parameter θ_k for $k = 1, \dots, 4$.

When the GAMLSS models $\mathcal{M}_0$ and $\mathcal{M}_1$ contain nonparametric additive terms as in (2.2) and (2.4), the same test can be used as a guide to fitted model selection in the same way that Hastie and Tibshirani (1990, Ch 3.9) compare 'nested' Generalized Additive Models (GAM) fits. The degrees of freedom used here is the trace of the smoothing matrix S_{jk} in the fitting algorithm, called the 'effective' degrees of freedom, Hastie and Tibshirani (1990).

For comparing non-nested GAMLSS models, to penalize over-fitting the generalized Akaike Information Criterion (GAIC), Akaike (1983), can be used. This is obtained by adding to the fitted deviance a fixed penalty $\sharp$ for each effective degree of freedom used in a model, i.e. $\text{GAIC}(\sharp) = \text{GD} + \sharp \cdot df$, where df denotes the total effective degrees of freedom used in the model and GD is the fitted global deviance. The model with the smallest value of the criterion $\text{GAIC}(\sharp)$ is then selected. The Akaike information criterion (AIC), Akaike (1974), and the Schwartz Bayesian criterion (SBC), Schwarz (1978), are special cases of the $\text{GAIC}(\sharp)$ criterion corresponding to $\sharp = 2$ and $\sharp = \log(n)$ respectively. The two criteria, AIC and SBC, are asymptotically justified as predicting the degree of fit in a new data set, i.e. approximations to the average predictive error. Justification for the use of SBC comes also as a crude approximation to Bayes factors, Raftery (1996, 1999). In practice it is usually found that while the original AIC is very generous in model selection the SBC is too restrictive. Our experience is that a value of the penalty $\sharp$ in the range $2.5 \leq \sharp \leq 3$ works well for most data. Kin and Gu (2004) suggested using $\sharp \approx 2.8$. A selection of different values of $\sharp$ e.g. $\sharp = 2, 2.5, 3, 3.5, 4$ could be used in turn to investigate the sensitivity or robustness of the model selection to the choice of the value of the penalty $\sharp$. Claeskens and Hjort (2003) consider a focused information criterion (FIC) in which the criterion for model selection depends on the objective of the study, in particular on the specific parameter of interest. Using $\text{GAIC}(\sharp)$ allows different penalties $\sharp$ to be tried for different modelling purposes. The sensitivity of the selected model to the choice of $\sharp$ can also be investigated.

For small data sets, the full data sample is usually used for both model fitting (minimizing GD) and for model selection (minimizing a penalized criterion, e.g. AIC or SBC). For very large data sets, the data could be split into (i) training, (ii) validation and (iii) test data sets. This split is now routinely available in some statistical packages such as SAS Enterprise Miner, SAS Institute Inc. (2000). Some of these procedure are now implemented in the **gamlss** packages and they will be described later in this chapter.

Within the GAMLSS framework (i) the training data could be used for model fitting (minimizing its GD) (ii) the validation data could be used for model selection, in particular selection of the distribution, link functions, predictor terms and smoothing parameters (by minimizing its GD , denoted by VGD) and (iii) the test data could be used for the assessment of the predictive power of the model chosen by (ii) and fitted by (i) and applied to the test data (again using its GD , denoted by TGD).

Different model selection strategies can be used to build a GAMLSS model but more importantly the determination of the **model adequacy should always be carried out with respect to the substantive questions of interest** and not in isolation. This means that different problems could possibly require different model strategies.

Section 8.3 show how the functions `addterm`, `dropterm`, `stepGAIC()`, `stepGAIC.VR()` and `stepGAIC.CH()` can be used to select (or eliminate) terms from a model formula. Section 8.5.1 discusses a simple way of selecting the hyper-parameters of a GAMLSS model.

8.2 Data sets

8.2.1 The US pollution data

Data summary: US pollution data

R data file: `usair` in package `gamlss` of dimensions 41×7

variables

`y` : sulphur dioxide concentration in air in mgs. per cubic meter

`x1` : average annual temperature in degrees F

`x2` : number of manufacturers employing > 20 workers

`x3` : population size in thousands

`x4` : average annual wind speed in miles per hour

`x5` : average annual rainfall in inches

`x6` : average number of days rainfall per year

purpose: to demonstrate term selection techniques

The US pollution data set taken from Hand *et al.* (1994) data set 26, USAIR.DAT. The data are from 41 cities in the USA and have 7 continuous variables. Preliminary analysis has shown that it is better to model the distribution of the response variable Y using the gamma rather the normal distribution.

8.2.2 The AIDS data

Data summary: The AIDS data

R data file: `aids` in package `gamlss` of dimensions 45×3 , quarterly reported AIDS cases in the U.K. from January 1983 to March 1994 obtained from the Public Health Laboratory Service, Communicable Disease Surveillance Centre, London.

variables

`y` : the number of quarterly aids cases in England and Wales

`x` : time in quarters from January 1983

`qrt` : a factor for the quarterly seasonal effect

purpose: to demonstrate hyperparameters selection techniques

8.3 Selecting explanatory variables using `addterm`, `dropterm`, and `stepGAIC`

There are five functions within GAMLSS to assist with selecting explanatory variable terms. The first two are the functions `addterm` and `dropterm` which allow the addition or removal

of a term in a model respectively. Those two functions are building blocks for the functions `stepGAIC.VR()` and `stepGAIC.CH()` suitable for stepwise selection of models. Both functions perform the stepwise model selection using a Generalized Akaike Information Criterion. The function `stepGAIC.VR()` is based on the function `stepAIC` given in the package MASS of Venables and Ripley (2002), (where more details and examples of the function can be found), with the additional property that it allows selection of terms for any selected distribution parameter. The function `stepGAIC.CH` is based on the S function `step.gam()` (see Chambers and Hastie (1992), for more information) and it is more suited for models with smoothing additive terms in them. Again the function `stepGAIC.CH` is generalized here so it can be used for any distribution parameter within GAMLSS. The main difference between `stepGAIC.VR()` and `stepGAIC.CH()` lies on the use of the `scope` argument. The function `stepGAIC()` combines the two functions by having an extra argument `additive` which when set to `TRUE` the `stepGAIC.CH()` function is used, and when set to `FALSE` the `stepGAIC.VR()` function is used. `stepGAIC.VR()` is used by default.

8.3.1 The addterm and dropterm functions

The functions `addterm` and `dropterm` are generic functions with their original definitions defined at the package MASS of Vendable and Ripley (2002). This package has to be attached, (i.e. `library(MASS)`), before their method for classes `gamlss` can be used. The functions `stepGAIC()`, `stepGAIC.VR()` and `stepGAIC.CH()` can be used without attaching MASS.

The `dropterm` and `addterm` functions in GAMLSS have the following arguments

<code>object</code>	a <code>gamlss</code> object.
<code>scope</code>	a formula giving terms which might be dropped or added. For the function <code>dropterm</code> the default is the model formula. For the function <code>addterm</code> the <code>scope</code> is a formula specifying a maximal model which should include the current one. Only terms that can be dropped or added while maintaining marginality are actually tried.
<code>scale</code>	scale is not used in <code>gamlss</code>
<code>test</code>	it takes value <code>"none"</code> for no test and <code>"Chisq"</code> for a χ^2 test statistic relative to the original model.
<code>k</code>	the multiple of the degrees of freedom used for the penalty in the GAIC. Note <code>k = 2</code> gives the original AIC, <code>k = log(n)</code> gives SBC.
<code>sorted</code>	If <code>TRUE</code> the results are sorted in the order of the GAIC from the lowest (the best model) to the highest (the worst model).
<code>trace</code>	if <code>'TRUE'</code> additional information may be given on the fits as they are tried.
<code>...</code>	: arguments passed to or from other methods.

In order to demonstrate how `dropterm` and `addterm` is working consider the US pollution data set taken from Hand *et al.* (1994). Preliminary analysis has shown that it is better to model the distribution of the response variable Y using the gamma rather the normal distribution. We start by fitting the full linear model for μ including all six explanatory variables:

8.3. SELECTING EXPLANATORY VARIABLES USING ADDTERM, DROPTERM, AND STEPGAIC193

```
> data(usair)
> mod1 <- gamlss(y ~ ., data = usair, family = GA)
```

GAMLSS-RS iteration 1: Global Deviance = 303.1603

GAMLSS-RS iteration 2: Global Deviance = 303.1602

Now we use the `dropterm` function to check whether any linear terms can be dropped.

```
> library(MASS)
> dropterm(mod1, test = "Chisq")
```

Single term deletions for
mu

Model:

```
y ~ x1 + x2 + x3 + x4 + x5 + x6
      Df    AIC      LRT  Pr(Chi)
<none>    319.16
x1      1 327.58    10.42 0.001244 **
x2      1 326.92     9.76 0.001788 **
x3      1 321.39     4.23 0.039717 *
x4      1 324.08     6.92 0.008501 **
x5      1 320.57     3.41 0.064642 .
x6      1 317.16 0.001716 0.966960
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

The above output gives the test for removing each of the six variables from the full model. Given all other linear terms in the model, the variable x_6 is the first to be dropped since it has the highest p-values, 0.967, given by column `Pr(Chi)`, and so is the least significant. To demonstrate the function `addterm` consider adding a two way interaction term into the model `mod1`. Note that the `scope` argument has to be defined explicitly here.

```
> addterm(mod1, scope = ~(x1 + x2 + x3 + x4 + x5 + x6)^2, test = "Chisq")
```

Single term additions for
mu

Model:

```
y ~ x1 + x2 + x3 + x4 + x5 + x6
      Df    AIC      LRT  Pr(Chi)
<none>    319.16
x1:x2    1 320.09    1.07 0.3012045
x1:x3    1 319.40    1.76 0.1843028
x1:x4    1 320.60    0.56 0.4533271
x1:x5    1 316.94    4.22 0.0398901 *
x1:x6    1 320.93    0.24 0.6277906
x2:x3    1 320.48    0.68 0.4100846
x2:x4    1 319.75    1.41 0.2344256
x2:x5    1 318.17    2.99 0.0839194 .
```

```

x2:x6    1 321.13    0.03 0.8603147
x3:x4    1 317.38    3.78 0.0519200 .
x3:x5    1 320.19    0.97 0.3253680
x3:x6    1 320.85    0.31 0.5800599
x4:x5    1 307.07   14.09 0.0001745 ***
x4:x6    1 320.33    0.83 0.3609322
x5:x6    1 318.74    2.42 0.1198894
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Among the two way interactions `x4:x5` is highly significant with a p-value of less than 0.001.

8.3.2 The `stepGAIC` function

In order to build a model we need the functions `stepGAIC()`, `stepGAIC.VR()` and `stepGAIC.CH`. The last two functions have the following arguments

<code>object</code>	a <code>gamlss</code> object. This is used as the initial model in the stepwise search
<code>scope</code>	defines the range of models examined in the stepwise search. For the function <code>stepGAIC.VR()</code> this should be either a single formula, or a list containing components <code>upper</code> and <code>lower</code> , both formulae. For the function <code>stepGAIC.CH</code> the <code>scope</code> defines the range of models examined in the stepwise search. It is a list of formulas, with each formula corresponding to a single term in the model. A 1 in the formula allows the additional option of leaving the term out of the model entirely.
<code>direction</code>	the mode of stepwise search, can be one of <code>both</code> , <code>backward</code> , or <code>forward</code> , with a default of <code>both</code> which performs forward stepwise model selection. If the <code>scope</code> argument is missing the default for <code>direction</code> is <code>backward</code>
<code>trace</code>	if positive, information is printed during the running of <code>stepGAIC</code> . Larger values may give more information on the fitting process
<code>keep</code>	a filter function whose input is a fitted model object and the associated AIC statistic, and whose output is arbitrary. Typically <code>keep</code> will select a subset of the components of the object and return them. The default is not to keep anything.
<code>steps</code>	the maximum number of steps to be considered. The default is 1000 (essentially as many as required). It is typically used to stop the process early.
<code>scale</code>	scale is not used in <code>gamlss</code>
<code>what</code>	which distribution parameter is being modelled, default <code>what="mu"</code>
<code>k</code>	the multiple of the degrees of freedom used for the penalty in the GAIC criterion. <code>k = 2</code> gives the genuine Akaike information criterion (AIC), while <code>k = log(n)</code> gives the Schwartz Bayesian criterion (SBC).
<code>...</code>	any additional arguments to <code>extractAIC</code> . (None are currently used).

The `stepGAIC()` function has the same argument as the functions above but with an extra argument `additive=TRUE` or `FALSE` for selecting `stepGAIC.CH()` or `stepGAIC.VR()` respectively with default `FALSE`.

8.3.3 The `stepGAIC.VR` function

The set of models searched by `stepGAIC.VR()` is determined by the `scope` argument and its `lower` and `upper` components. The `lower` and `upper` are model formulae. The terms defined by the formula in `lower` component are always included in the model. The formula in `upper` is the most complicated model that the procedure would consider. The fitted model specified in the `object` option should lie between those two models. If the `scope` is missing then a backward elimination starts from the model define by the `gamlss` object. In the following example a backward elimination is performed on the model given by `mod1`. Note that `mod2` has a new component called `anova` showing the steps taken in the search of the model.

```
> mod2 <- stepGAIC.VR(mod1)

Distribution parameter: mu
Start: AIC= 319.16
y ~ x1 + x2 + x3 + x4 + x5 + x6

      Df    AIC
- x6    1 317.16
<none>    319.16
- x5    1 320.57
- x3    1 321.39
- x4    1 324.08
- x2    1 326.92
- x1    1 327.58

Step: AIC= 317.16
y ~ x1 + x2 + x3 + x4 + x5

      Df    AIC
<none>    317.16
- x3    1 319.39
- x4    1 322.48
- x5    1 324.14
- x2    1 324.92
- x1    1 336.11

> mod2$anova

Stepwise Model Path
Analysis of Deviance Table

Initial
mu
Model:
y ~ x1 + x2 + x3 + x4 + x5 + x6
```

```
Final
mu
Model:
y ~ x1 + x2 + x3 + x4 + x5
```

	Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
	1			33	303.1602	319.1602
	2 - x6	1	0.001715758	34	303.1619	317.1619

Note that the same result is produced using either `mod2<-stepGAIC.VR(mod1)` or `mod2<-stepGAIC(mod1, additive=FALSE)`.

The above backward search procedure confirms the fact that, if we want to include only linear additive terms in the model, the variable `x6` is not needed. The default penalty for the procedure is $\sharp = 2$, i.e. a genuine original AIC selection procedure. Using the SBC results in the same regression model selection as shown below:

```
> mod2 <- stepGAIC(mod1, k = log(41))
```

```
Distribution parameter: mu
Start: AIC= 332.87
y ~ x1 + x2 + x3 + x4 + x5 + x6
```

	Df	AIC
- x6	1	329.16
- x5	1	332.57
<none>		332.87
- x3	1	333.39
- x4	1	336.08
- x2	1	338.91
- x1	1	339.58

```
Step: AIC= 329.16
y ~ x1 + x2 + x3 + x4 + x5
```

	Df	AIC
<none>		329.16
- x3	1	329.67
- x4	1	332.76
- x5	1	334.43
- x2	1	335.20
- x1	1	346.39

As an example of using the `scope` argument explicitly we consider whether two way interactions between the explanatory variables are needed in the model. The simplest model we are considered here is with only a constant, i.e. `lower= 1`, and the most complicated is the one with all two way interactions. The final model will be something between those two.

```
> mod3 <- stepGAIC(mod1, scope = list(lower = ~1, upper = ~(x1 +
+ x2 + x3 + x4 + x5 + x6)^2))
```

8.3. SELECTING EXPLANATORY VARIABLES USING ADDTERM, DROPTERM, AND STEPGAIC197

Distribution parameter: μ

Start: AIC= 319.16

$y \sim x1 + x2 + x3 + x4 + x5 + x6$

	Df	AIC
+ x4:x5	1	307.07
+ x1:x5	1	316.94
- x6	1	317.16
+ x3:x4	1	317.38
+ x2:x5	1	318.17
+ x5:x6	1	318.74
<none>		319.16
+ x1:x3	1	319.40
+ x2:x4	1	319.75
+ x1:x2	1	320.09
+ x3:x5	1	320.19
+ x4:x6	1	320.33
+ x2:x3	1	320.48
- x5	1	320.57
+ x1:x4	1	320.60
+ x3:x6	1	320.85
+ x1:x6	1	320.93
+ x2:x6	1	321.13
- x3	1	321.39
- x4	1	324.08
- x2	1	326.92
- x1	1	327.58

Step: AIC= 307.07

$y \sim x1 + x2 + x3 + x4 + x5 + x6 + x4:x5$

	Df	AIC
+ x1:x6	1	300.94
+ x4:x6	1	301.65
+ x1:x4	1	302.09
- x6	1	305.12
+ x3:x5	1	306.94
<none>		307.07
+ x2:x5	1	307.78
+ x2:x4	1	307.94
+ x3:x4	1	308.13
+ x3:x6	1	308.56
+ x1:x2	1	308.65
+ x2:x3	1	308.76
+ x1:x5	1	308.89
+ x2:x6	1	309.02
+ x1:x3	1	309.06
+ x5:x6	1	309.06

```

- x3      1 310.66
- x2      1 317.09
- x4:x5   1 319.16
- x1      1 325.97

```

```

....
....
....
....

```

Step: AIC= 292.72

```

y ~ x1 + x2 + x3 + x4 + x5 + x6 + x4:x5 + x1:x6 + x4:x6 + x3:x4 +
    x2:x4 + x2:x3 + x3:x6 + x2:x6

```

	Df	AIC
<none>		292.72
+ x1:x4	1	293.55
+ x1:x5	1	293.95
+ x2:x5	1	294.08
- x2:x6	1	294.19
+ x5:x6	1	294.54
+ x3:x5	1	294.55
+ x1:x2	1	294.71
+ x1:x3	1	294.72
- x1:x6	1	295.18
- x3:x6	1	296.41
- x2:x3	1	297.34
- x3:x4	1	300.27
- x2:x4	1	300.41
- x4:x6	1	307.60
- x4:x5	1	328.13

```
> mod3$anova
```

Stepwise Model Path
Analysis of Deviance Table

Initial

mu

Model:

```
y ~ x1 + x2 + x3 + x4 + x5 + x6
```

Final

mu

Model:

```
y ~ x1 + x2 + x3 + x4 + x5 + x6 + x4:x5 + x1:x6 + x4:x6 + x3:x4 +
    x2:x4 + x2:x3 + x3:x6 + x2:x6
```

8.3. SELECTING EXPLANATORY VARIABLES USING ADDTERM, DROPTERM, AND STEPGAIC199

	Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
1				33	303.1602	319.1602
2	+	x4:x5	1 14.086994	32	289.0732	307.0732
3	+	x1:x6	1 8.133304	31	280.9399	300.9399
4	+	x4:x6	1 4.786482	30	276.1534	298.1534
5	+	x3:x4	1 2.082757	29	274.0706	298.0706
6	+	x2:x4	1 4.460528	28	269.6101	295.6101
7	+	x2:x3	1 2.886924	27	266.7232	294.7232
8	+	x3:x6	1 2.532079	26	264.1911	294.1911
9	+	x2:x6	1 3.468607	25	260.7225	292.7225

Model `mod3` is a rather complicated interaction model. Note that the variable `x6` is included in the model `mod3` since higher interactions involving `x6` are selected in the model. More than two way interactions are not permitted for continuous variables which is the case in our example. A plot of the residuals of model `mod3` indicates possible heterogeneity in the variation of Y . We shall deal with this problem later.

8.3.4 The `stepGAIC.CH` function

Now we consider the `stepGAIC.CH` function. For the function `stepGAIC.CH()` each of the formulas in `scope` specifies a "regimen" of candidate forms in which the particular term may enter the model. For example, a term formula might be `1 + x1 + cs(x1, df=3)`. This means that `x1` could either be omitted from the model, appear linearly, or as a cubic smoothing spline function `cs()` estimated non-parametrically. Every term in the model is described by such a term formula, and the final model is built up by selecting a component from each formula. The function `gamlss.scope` is similar to the S '`gam.scope()`' in Chambers and Hastie (1991) and can be used to create term formulae automatically from specified data or model frames. The supplied model object is used as the starting model, and hence there is the requirement that one term from each of the term formulas be present in the formula of the distribution parameter. This also implies that any terms in the formula of the distribution parameter not contained in any of the term formulas will be forced to be present in every model considered. Below we use the `gamlss.scope` function to create a `scope` for the function `stepGAIC.CH`.

```
> gs <- gamlss.scope(model.frame(y ~ x1 + x2 + x3 + x4 + x5 + x6,
+   data = usair))
> gs

$х1
~1 + x1 + cs(x1)

$х2
~1 + x2 + cs(x2)

$х3
~1 + x3 + cs(x3)

$х4
~1 + x4 + cs(x4)
```

```
$x5
~1 + x5 + cs(x5)
```

```
$x6
~1 + x6 + cs(x6)
```

The function `gamlss.scope` has the following arguments:

frame	a data or model frame
response	which variable is the response in the data or model frame, the default is the first
smoother	what type smoother to use, with the default being cubic smoothing spline, <code>cs</code>
arg	any additional arguments required by the smoother, (for example <code>df</code> for <code>cs</code>)
form	should a formula be returned (default), or else a character version of the formula

Lets us experiment with the function `gamlss.scope` and use a different smoother `lo` with a `span=.7`

```
> gs1 <- gamlss.scope(model.frame(y ~ x1 + x2 + x3 + x4 + x5 +
+   x6, data = usair), smoother = "lo", arg = "span=.7", form = TRUE)
> gs1
```

```
$x1
~1 + x1 + lo(x1, span = 0.7)
```

```
$x2
~1 + x2 + lo(x2, span = 0.7)
```

```
$x3
~1 + x3 + lo(x3, span = 0.7)
```

```
$x4
~1 + x4 + lo(x4, span = 0.7)
```

```
$x5
~1 + x5 + lo(x5, span = 0.7)
```

```
$x6
~1 + x6 + lo(x6, span = 0.7)
```

Next we use the `stepGAIC.CH` to find a suitable additive model using a cubic smoothing spline as a smoother.

```
> mod5 <- gamlss(y ~ 1, data = usair, family = GA)
```

```
GAMLSS-RS iteration 1: Global Deviance = 349.7146
```

```
GAMLSS-RS iteration 2: Global Deviance = 349.7146
```

8.3. SELECTING EXPLANATORY VARIABLES USING ADDTERM, DROPTERM, AND STEPGAIC201

```
> mod6 <- stepGAIC(mod5, gs, additive = TRUE)

Distribution parameter: mu
Start: y ~ 1; AIC= 353.7146
Trial: y ~ x1 + 1 + 1 + 1 + 1 + 1 + 1; AIC= 338.0354
Trial: y ~ 1 + x2 + 1 + 1 + 1 + 1 + 1; AIC= 343.0487
Trial: y ~ 1 + 1 + x3 + 1 + 1 + 1 + 1; AIC= 349.2046
Trial: y ~ 1 + 1 + 1 + x4 + 1 + 1 + 1; AIC= 355.0206
Trial: y ~ 1 + 1 + 1 + 1 + x5 + 1 + 1; AIC= 355.4256
Trial: y ~ 1 + 1 + 1 + 1 + 1 + x6; AIC= 343.9733
Step : y ~ x1 ; AIC= 338.0354

Trial: y ~ cs(x1) + 1 + 1 + 1 + 1 + 1 + 1; AIC= 337.3823
Trial: y ~ x1 + x2 + 1 + 1 + 1 + 1 + 1; AIC= 328.781
Trial: y ~ x1 + 1 + x3 + 1 + 1 + 1 + 1; AIC= 332.942
Trial: y ~ x1 + 1 + 1 + x4 + 1 + 1 + 1; AIC= 339.6497
Trial: y ~ x1 + 1 + 1 + 1 + x5 + 1 + 1; AIC= 335.1107
Trial: y ~ x1 + 1 + 1 + 1 + 1 + x6; AIC= 335.9902
Step : y ~ x1 + x2 ; AIC= 328.781

Trial: y ~ cs(x1) + x2 + 1 + 1 + 1 + 1 + 1; AIC= 328.8904
Trial: y ~ x1 + cs(x2) + 1 + 1 + 1 + 1 + 1; AIC= 333.3071
Trial: y ~ x1 + x2 + x3 + 1 + 1 + 1 + 1; AIC= 325.665
Trial: y ~ x1 + x2 + 1 + x4 + 1 + 1 + 1; AIC= 327.6493
Trial: y ~ x1 + x2 + 1 + 1 + x5 + 1 + 1; AIC= 324.4413
Trial: y ~ x1 + x2 + 1 + 1 + 1 + x6; AIC= 325.5355
Step : y ~ x1 + x2 + x5 ; AIC= 324.4413
...
...
...
Trial: y ~ 1 + x2 + x3 + cs(x4) + cs(x5) + 1; AIC= 311.8572
Trial: y ~ cs(x1) + x2 + x3 + cs(x4) + cs(x5) + 1; AIC= 305.8586
Trial: y ~ x1 + 1 + x3 + cs(x4) + cs(x5) + 1; AIC= 315.1205
Trial: y ~ x1 + cs(x2) + x3 + cs(x4) + cs(x5) + 1; AIC= 305.2431
Trial: y ~ x1 + x2 + 1 + cs(x4) + cs(x5) + 1; AIC= 309.6543
Trial: y ~ x1 + x2 + cs(x3) + cs(x4) + cs(x5) + 1; AIC= 307.9507
Trial: y ~ x1 + x2 + x3 + cs(x4) + x5 + 1; AIC= 305.1266
Trial: y ~ x1 + x2 + x3 + cs(x4) + cs(x5) + x6; AIC= 305.3415

> mod6$anova
```

	From	To	Df	Deviance	Resid. Df	Resid. Dev	AIC
1			NA	NA	39.00000	349.7146	353.7146
2		x1	-1.000000	-17.679187	38.00000	332.0354	338.0354
3		x2	-1.000000	-11.254434	37.00000	320.7810	328.7810
4		x5	-1.000000	-6.339658	36.00000	314.4413	324.4413
5	x5	cs(x5)	-2.999638	-13.806499	33.00036	300.6348	316.6341
6		x3	-1.000000	-3.178930	32.00036	297.4559	315.4552
7		x4	-1.000000	-2.833874	31.00036	294.6220	314.6213
8	x4	cs(x4)	-2.999290	-16.786533	28.00107	277.8355	303.8334

```

> mod6

Family:  c("GA", "Gamma")
Fitting method: RS()

Call:  gamlss(formula = y ~ x1 + x2 + x3 + cs(x4) + cs(x5), family = GA,
             data = usair, trace = F)

Mu Coefficients:
(Intercept)          x1          x2          x3          cs(x4)          cs(x5)
  6.5319736  -0.0508509   0.0013516  -0.0009606  -0.1195133   0.0160507

Sigma Coefficients:
(Intercept)
  -1.199

Degrees of Freedom for the fit: 12.99893 Residual Deg. of Freedom   28.00107
Global Deviance:          277.836
              AIC:          303.833
              SBC:          326.108

```

The algorithm took eight different steps (we only included the model trials for the first, second, third and eighth). In the first step, all the linear terms are tried and the variable x_1 is selected for inclusion. In the second step, since x_1 is already in the model, $cs(x_1)$ is tried together with all the linear terms from the rest of the variable. In this second step the variable x_2 is selected. The important thing to notice here is that there is an hierarchy in the inclusion of the terms in the model according to its scope, e.g. the component of the `gamlss.scope` for x_1 is `1 + x1 + cs(x1)` requiring 1, (i.e. no x_1 variable), x_1 , (linear in x_1) and $cs(x_1)$, (smooth term in x_1), to be tested in sequence. The terms for x_1 in the `gamlss.scope` can only move one step up or down from the current term in x_1 in the sequence. Hence, for example, the model in x_1 can change from 1 only to x_1 , but from x_1 to either 1 or $cs(x_1)$, and from $cs(x_1)$ only to x_1 .

8.3.5 Selecting model for σ using the `stepGAIC.VR` function

We shall now try to include linear terms in the σ model. Note that with only 41 observations and with a reasonably complicated model μ , it **not** advisable to try smoothing terms for σ . Here we check whether including linear terms in the model for σ will improve the model, i.e. reduce AIC using the `stepGAIC` function.

```

mod7<-stepGAIC(mod6, what="sigma", scope=~x1+x2+x3+x4+x5+x6)
Distribution parameter:  sigma Start:  AIC= 303.83
~1

              Df      AIC
+ x3    1.00198 299.93
+ x2    1.00098 302.09
<none>              303.83
+ x4    0.99944 304.37
+ x5    1.00240 305.59

```

8.3. SELECTING EXPLANATORY VARIABLES USING ADDTERM, DROPTERM, AND STEPGAIC203

```
+ x1    0.99962 305.63
+ x6    1.00108 305.83
```

```
Step:  AIC= 299.93
~x3
```

```
      Df    AIC
+ x4  0.99774 299.50
<none>          299.93
+ x6  1.00029 300.85
+ x5  0.99906 301.22
+ x1  0.99731 302.09
+ x2  0.99900 302.79
- x3  1.00198 303.83
```

```
Step:  AIC= 299.5
~x3 + x4
```

```
      Df    AIC
<none>          299.50
- x4  0.99774 299.93
+ x5  1.00124 300.13
+ x6  1.00211 301.17
+ x1  1.00112 301.25
+ x2  1.00243 302.38
- x3  1.00027 304.37
```

There were 13 warnings (use warnings() to see them)

```
> mod7$anova
Stepwise Model Path
Analysis of Deviance Table
```

```
Initial
sigma
Model:
~1
```

```
Final
sigma
Model:
~x3 + x4
```

Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
1			28.00107	277.8355	303.8334
2 + x3	1.0019760	5.908179	26.99910	271.9273	299.9291
3 + x4	0.9977373	2.427305	26.00136	269.5000	299.4973

According to criterion AIC the model needs **x3+x4** in the formula for σ . The warnings given during the execution of the **stepGAIC** function are due to the fact that the algorithm has not

converged occasionally in 20 iteration. In order to increase the number of iterations you have to go back to the fitting of model `mod5`. This is not needed on this occasion since the results remain the same even if we do just that.

8.4 Data example

8.4.1 The third party claims

Data summary:

R data file: `LGAcclaims` in package `gamlss.dist` of dimensions 176×11

variables

LGA : the 176 geographical areas (local government areas, in New South Wales, Australia).

Claims : the number of third party claims (the response variable)

SD : the areas grouped into thirteen statistical divisions

Accidents : the number of accidents

KI : the number of people killed or injured

Population : population

Pop_density : population density

L_Population : the log population

L_Accidents : the log of the number of accidents

L_KI : the log of the number of people killed or injured

L_Popdensity : the log of population density

purpose: to demonstrate the selection of terms in `gamlss`

conclusion: A NBI distribution is adequate

The data used here are provided by Gillian Heller and can be found in de Jong and Heller (2007). They are third party insurance data. Third party is a compulsory insurance for vehicle owners in Australia. It insures vehicle owners against injury caused to other drivers, passengers or pedestrians, as a result of an accident. This data set records the number of third party claims, **Claims**, in a twelve month period between 1984-1986 in each of 176 geographical areas (local government areas, **LGA**) in New South Wales, Australia. Areas are grouped into thirteen statistical divisions (**SD**). Other recorded variables are the number of accidents, **Accidents**, the number of people killed or injured (**KI**), population density (**Pop_density**) and population (**Population**) with all variables classified according to area. In most of the analysis here we will use the log values for **Population**, **Accidents**, **KI** and **Pop_density** which are denoted as **L_Population**, **L_Accidents**, **L_KI** and **L_Popdensity** respectively. Figure 8.1 shows the numbers of claims against the rest of the continuous variables in the data.

```
> library(gamlss.dist)
> data(LGAcclaims)
```

```
> with(LGAclaims, plot(data.frame(Claims, L_Popdensity, L_KI, L_Accidents,
+   L_Population)))
```

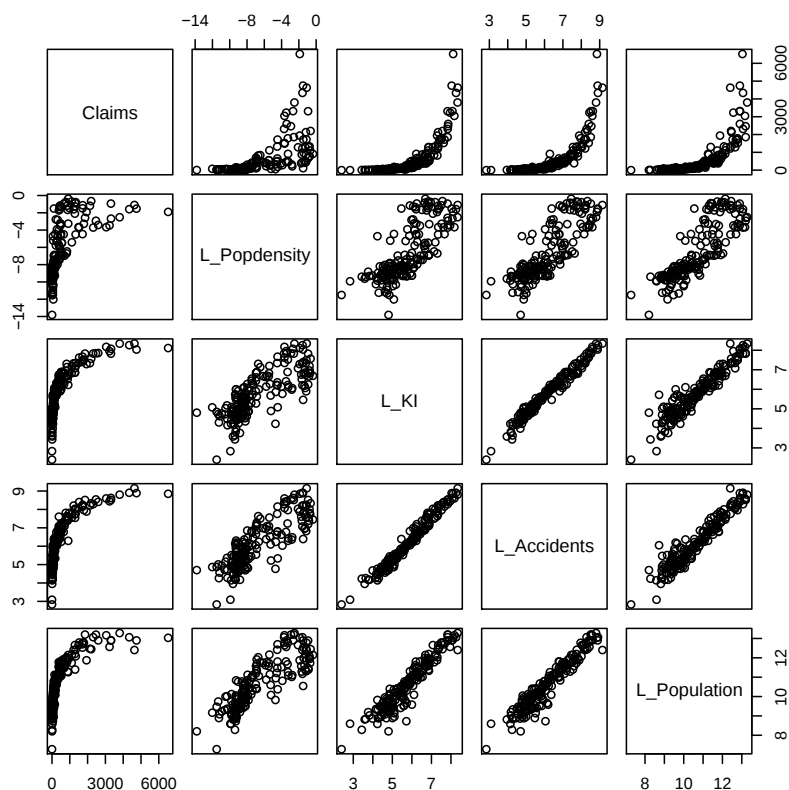


Figure 8.1: The plot of the third party insurance data

We start with a model for μ including all the explanatory variables. To check overdispersion we compare the Poisson distribution model (PO) against the negative binomial type I model (NBI). Since the reduction in deviance between those two models is enormous (4606.169) with only one extra degree of freedom, for the rest of the chapter we will use the negative binomial distribution model in order to choose terms for the μ and (possibly) for the σ models.

```
> m0 <- gamlss(Claims ~ factor(SD) + L_Popdensity + L_KI + L_Accidents +
+   L_Population, data = LGAclaims, family = PO)
```

GAMLSS-RS iteration 1: Global Deviance = 6487.73

GAMLSS-RS iteration 2: Global Deviance = 6487.73

```
> m1 <- gamlss(Claims ~ factor(SD) + L_Popdensity + L_KI + L_Accidents +
+   L_Population, data = LGAclaims, family = NBI)
```

```
GAMLSS-RS iteration 1: Global Deviance = 1883.942
GAMLSS-RS iteration 2: Global Deviance = 1881.561
GAMLSS-RS iteration 3: Global Deviance = 1881.561
```

```
> deviance(m0) - deviance(m1)
```

```
[1] 4606.169
```

Selection of variables

We shall now use the `dropterm` to check if model `m1` can be simplified by dropping any of the existing terms in μ and the function `addterm` to check whether two way interactions of the existing terms are needed.

```
> library(MASS)
> mD <- dropterm(m1, test = "Chisq")
> mD
```

Single term deletions for
mu

Model:

```
Claims ~ factor(SD) + L_Popdensity + L_KI + L_Accidents + L_Population
```

	Df	AIC	LRT	Pr(Chi)
<none>		1917.56		
factor(SD)	12	1931.45	37.89	0.000160 ***
L_Popdensity	1	1970.84	55.28	1.045e-13 ***
L_KI	1	1961.00	45.44	1.576e-11 ***
L_Accidents	1	1920.63	5.07	0.024362 *
L_Population	1	1923.02	7.46	0.006313 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
> mA <- addterm(m1, scope = ~(factor(SD) + L_Popdensity + L_KI +
+   L_Accidents + L_Population)^2, test = "Chisq")
> mA
```

Single term additions for
mu

Model:

```
Claims ~ factor(SD) + L_Popdensity + L_KI + L_Accidents + L_Population
```

	Df	AIC	LRT	Pr(Chi)
<none>		1917.56		
factor(SD):L_Popdensity	12	1927.46	14.10	0.29424
factor(SD):L_KI	12	1921.44	20.12	0.06485 .
factor(SD):L_Accidents	12	1923.62	17.94	0.11764
factor(SD):L_Population	12	1923.94	17.62	0.12759
L_Popdensity:L_KI	1	1919.46	0.10	0.74753
L_Popdensity:L_Accidents	1	1919.25	0.32	0.57410

```

L_Popdensity:L_Population  1 1918.21    1.36 0.24430
L_KI:L_Accidents           1 1919.51    0.05 0.82555
L_KI:L_Population          1 1919.29    0.27 0.60173
L_Accidents:L_Population   1 1919.28    0.28 0.59547
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Based on the Chi square tests no terms can be left out and no two way interaction is needed. Since we established that adding or dropping terms in μ is not beneficial there is no point using `stepGAIC.VR()` for modelling the μ parameter with linear terms. Instead we will use (`stepGAIC.CH()`) trying to establish if smoothing terms are needed in the μ model. The function `gamlss.scope()` is used here to create the different models to be fitted.

```

> gs <- gamlss.scope(model.frame(Claims ~ factor(SD) + L_Popdensity +
+   L_KI + L_Accidents + L_Population, data = LGAclaims))
> gs

$`factor(SD)`
~1 + factor(SD)

$L_Popdensity
~1 + L_Popdensity + cs(L_Popdensity)

$L_KI
~1 + L_KI + cs(L_KI)

$L_Accidents
~1 + L_Accidents + cs(L_Accidents)

$L_Population
~1 + L_Population + cs(L_Population)

> m2 <- stepGAIC.CH(m1, scope = gs, k = 2)

Distribution parameter: mu
Start: Claims ~ factor(SD) + L_Popdensity + L_KI + L_Accidents
      + L_Population; AIC= 1917.561
Trial: Claims ~ 1 + L_Popdensity + L_KI + L_Accidents
      + L_Population; AIC= 1931.452
Trial: Claims ~ factor(SD) + 1 + L_KI + L_Accidents
      + L_Population; AIC= 1970.842
Trial: Claims ~ factor(SD) + cs(L_Popdensity) + L_KI
      + L_Accidents + L_Population; AIC= 1917.225
Trial: Claims ~ factor(SD) + L_Popdensity + 1
      + L_Accidents + L_Population; AIC= 1960.999
Trial: Claims ~ factor(SD) + L_Popdensity + cs(L_KI)
      + L_Accidents + L_Population; AIC= 1917.716
Trial: Claims ~ factor(SD) + L_Popdensity + L_KI + 1
      + L_Population; AIC= 1920.630
Trial: Claims ~ factor(SD) + L_Popdensity + L_KI + cs(L_Accidents)

```

```

+ L_Population; AIC= 1913.966
Trial: Claims ~ factor(SD) + L_Popdensity + L_KI
+ L_Accidents + 1; AIC= 1923.020
Trial: Claims ~ factor(SD) + L_Popdensity + L_KI
+ L_Accidents + cs(L_Population); AIC= 1917.606
Step : Claims ~ factor(SD) + L_Popdensity + L_KI + cs(L_Accidents)
+ L_Population ; AIC= 1913.966

Trial: Claims ~ 1 + L_Popdensity + L_KI + cs(L_Accidents)
+ L_Population; AIC= 1925.391
Trial: Claims ~ factor(SD) + 1 + L_KI + cs(L_Accidents)
+ L_Population; AIC= 1963.121
Trial: Claims ~ factor(SD) + cs(L_Popdensity) + L_KI
+ cs(L_Accidents) + L_Population; AIC= 1913.232
Trial: Claims ~ factor(SD) + L_Popdensity + 1
+ cs(L_Accidents) + L_Population; AIC= 1952.037
Trial: Claims ~ factor(SD) + L_Popdensity + cs(L_KI)
+ cs(L_Accidents) + L_Population; AIC= 1918.475
Trial: Claims ~ factor(SD) + L_Popdensity + L_KI
+ cs(L_Accidents) + 1; AIC= 1921.038
Trial: Claims ~ factor(SD) + L_Popdensity + L_KI
+ cs(L_Accidents) + cs(L_Population); AIC= 1913.002
Step : Claims ~ factor(SD) + L_Popdensity + L_KI + cs(L_Accidents)
+ cs(L_Population) ; AIC= 1913.002

Trial: Claims ~ 1 + L_Popdensity + L_KI + cs(L_Accidents)
+ cs(L_Population); AIC= 1924.935
Trial: Claims ~ factor(SD) + 1 + L_KI + cs(L_Accidents)
+ cs(L_Population); AIC= 1954.897
Trial: Claims ~ factor(SD) + cs(L_Popdensity) + L_KI + cs(L_Accidents)
+ cs(L_Population); AIC= 1912.463
Trial: Claims ~ factor(SD) + L_Popdensity + 1 + cs(L_Accidents)
+ cs(L_Population); AIC= 1950.536
Trial: Claims ~ factor(SD) + L_Popdensity + cs(L_KI) + cs(L_Accidents)
+ cs(L_Population); AIC= 1917.920
Step : Claims ~ factor(SD) + cs(L_Popdensity) + L_KI + cs(L_Accidents)
+ cs(L_Population) ; AIC= 1912.463

Trial: Claims ~ 1 + cs(L_Popdensity) + L_KI + cs(L_Accidents)
+ cs(L_Population); AIC= 1927.276
Trial: Claims ~ factor(SD) + cs(L_Popdensity) + 1 + cs(L_Accidents)
+ cs(L_Population); AIC= 1947.168
Trial: Claims ~ factor(SD) + cs(L_Popdensity) + cs(L_KI) + cs(L_Accidents)
+ cs(L_Population); AIC= 1917.415
Trial: Claims ~ factor(SD) + cs(L_Popdensity) + L_KI + L_Accidents
+ cs(L_Population); AIC= 1918.366

```

```
> m2$anova
```

	From	To	Df	Deviance	Resid. Df	Resid. Dev
1			NA	NA	158.0000	1881.561
2	L_Accidents	cs(L_Accidents)	-3.000838	-9.596529	154.9992	1871.964
3	L_Population	cs(L_Population)	-3.000782	-6.965547	151.9984	1864.999
4	L_Popdensity	cs(L_Popdensity)	-3.000727	-6.540878	148.9977	1858.458

AIC

1	1917.561
2	1913.966
3	1913.002
4	1912.463

```
> formula(m2, "mu")
```

```
Claims ~ factor(SD) + cs(L_Popdensity) + L_KI + cs(L_Accidents) +
      cs(L_Population)
```

The resulting `gamlss` object has an extra component `anova` which summarizes the selection process. The best model includes smoothing terms for `L_Popdensity`, `L_Accidents` and `L_Population` but not for `L_KI`. Plotting the smoothing additive functions can be achieved using the function `term.plot()`, see Figure 8.2.

```
> op <- par(mfrow = c(3, 2))
> term.plot(m2, se = T, partial = T)
> par(op)
```

Given that we have establish a good model for μ , we proceed to find a good model for σ . We start first with linear terms but we exclude the factor `SD` since some of the levels of the factor have very few observations.

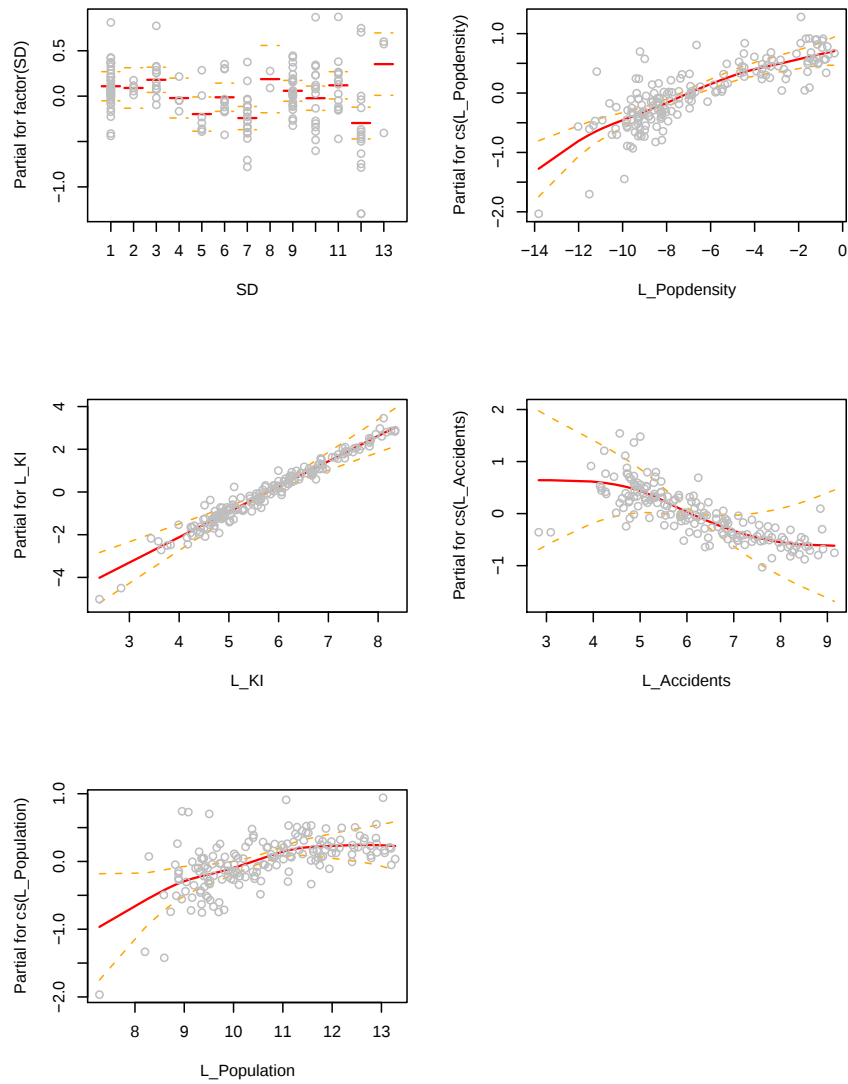
```
> m11 <- stepGAIC.VR(m2, scope = ~L_Popdensity + L_KI + L_Accidents +
+   L_Population, what = "sigma", k = 2)
```

```
Distribution parameter: sigma
Start: AIC= 1912.46
~1
```

	Df	AIC
+ L_Population	0.9984	1906.0
+ L_KI	1.0001	1910.2
+ L_Accidents	1.0001	1910.8
<none>		1912.5
+ L_Popdensity	1.0001	1913.8

```
Step: AIC= 1905.96
~L_Population
```

	Df	AIC
+ L_Accidents	1.0000	1899.4
+ L_Popdensity	1.0016	1900.5
+ L_KI	1.0013	1902.7
<none>		1906.0

Figure 8.2: The additive parameter plots for the μ model

```
- L_Population 0.9984 1912.5
```

```
Step: AIC= 1899.39
```

```
~L_Population + L_Accidents
```

	Df	AIC
+ L_Popdensity	0.99990	1898.3
<none>		1899.4
+ L_KI	0.99998	1901.4
- L_Accidents	0.99999	1906.0
- L_Population	0.99827	1910.8

```
Step: AIC= 1898.3
```

```
~L_Population + L_Accidents + L_Popdensity
```

	Df	AIC
+ L_KI	1.00006	1898.1
<none>		1898.3
- L_Popdensity	0.99990	1899.4
- L_Accidents	0.99832	1900.5
- L_Population	0.99829	1910.1

```
Step: AIC= 1898.09
```

```
~L_Population + L_Accidents + L_Popdensity + L_KI
```

	Df	AIC
- L_Accidents	1.00002	1896.4
<none>		1898.1
- L_KI	1.00006	1898.3
- L_Popdensity	0.99998	1901.4
- L_Population	0.99833	1912.0

```
Step: AIC= 1896.36
```

```
~L_Population + L_Popdensity + L_KI
```

	Df	AIC
<none>		1896.4
+ L_Accidents	1.00002	1898.1
- L_KI	0.99836	1900.5
- L_Popdensity	0.99868	1902.7
- L_Population	0.99835	1910.6

```
> m11$anova
```

```
Stepwise Model Path
Analysis of Deviance Table
```

```
Initial
sigma
```

```

Model:
~1

Final
sigma
Model:
~L_Population + L_Popdensity + L_KI

```

	Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
1				148.9977	1858.458	1912.463
2	+ L_Population	0.9983966	8.4984558	147.9993	1849.960	1905.961
3	+ L_Accidents	0.9999946	8.5717312	146.9993	1841.388	1899.389
4	+ L_Popdensity	0.9999033	3.0910247	145.9994	1838.297	1898.298
5	+ L_KI	1.0000618	2.2042417	144.9993	1836.093	1898.094
6	- L_Accidents	1.0000190	0.2617682	145.9993	1836.354	1896.356

Note that the argument `what` is used here to determine which distribution parameter is to be modelled. Here variables `L_Population`, `L_KI` and `L_Accidents` were found important in explaining the σ parameter. The model chosen using AIC appears over complicated. Maybe a higher penalty for GAIC would be more appropriate here.

8.5 Selecting hyperparameters

8.5.1 Selecting hyperparameters using `find.hyper`

This function appears to work well in searching for the optimum degrees of freedom for smoothing and/or non-linear parameters (e.g. a power parameter ξ used to transform x to x^ξ).

The function `find.hyper` selects the values of hyperparameters (and/or non-linear parameters) in a GAMLSS model. It uses the R function `optim` which then minimizes the generalized Akaike information criterion (GAIC) with a user defined penalty.

The arguments of the function `find.hyper` are

model	this is a quoted (<code>((quote()))</code>) GAMLSS model in which the required hyperparameters are denoted by <code>p[number]</code> , e.g. <code>quote(gamlss(y~cs(x,df=p[1]),sigma.fo=~ cs(x,df=p[2]),data=abdom))</code>
parameters	the starting parameter values in the search for the optimum hyperparameters and/or non-linear parameters, e.g. <code>parameters=c(3,3)</code>
other	this is used to optimize non-linear parameter(s), for example a transformation of the explanatory variable of the kind $x^{p[3]}$, e.g. <code>others=quote(nx<-x^p[3])</code> where <code>nx</code> is now in the model formula
penalty	specifies the penalty in the GAIC, (the default is 2) e.g. <code>penalty=3</code>
steps	the steps in the parameter(s) taken during the optimization procedure (see for example the <code>ndeps</code> option in the control function for <code>optim()</code>), by default set to 0.1 for all hyper parameters and non-linear parameters

lower	the lower bounds on the permissible values of the parameters e.g. for two parameters <code>lower=c(1,1)</code> . This does not apply if a method other than the default method "L-BFGS-B" is used
upper	the upper bounds on the permissible values of the parameters e.g. for two parameters <code>upper=c(30,10)</code> . This does not apply if a method other than the default method "L-BFGS-B" is used
method	the method used in <code>optim()</code> to numerically minimize the GAIC over the hyperparameters and/or non-linear parameters. By default this is "L-BFGS-B" to allow box-restriction on the parameters
...	this can be used for extra arguments in the <code>control</code> argument of the R function <code>optim()</code>

The function `find.hyper` returns the same output as the R function `optim()`.

The Aids data

As an example of using the function `find.hyper()` (but also of some difficulties arising from it) consider the AIDS data of Section 8.2.2:

```
> data(aids)
> with(aids, plot(x, y, pch = 21, bg = c("red", "green3", "blue",
+   "yellow")[unclass(qrt)]))
> aids.1 <- gamlss(y ~ cs(x, df = 7) + qrt, data = aids, family = NBI)

GAMLSS-RS iteration 1: Global Deviance = 365.8195
GAMLSS-RS iteration 2: Global Deviance = 361.9924
GAMLSS-RS iteration 3: Global Deviance = 362.1084
GAMLSS-RS iteration 4: Global Deviance = 362.1114
GAMLSS-RS iteration 5: Global Deviance = 362.1123

> lines(fitted(aids.1) ~ aids$x)
```

The model for μ the mean number of AIDS cases is a non-parametric cubic spline with 7 extra degrees of freedom for smoothing (on top of the constant and linear terms) together with a quarterly seasonal effect. Hence the hyperparameter (degrees of freedom for smoothing) was fixed at 7 for model `aids.1`. The data and the fitted parameter $\hat{\mu}$ from model `aids.1` are shown in Figure 8.3.

Here we would like to see if we could automate the process of finding the degrees of freedom. First we have to declare the model using the `quote` R function. For each hyperparameter to be estimated we put `p[.]` with the appropriate number in the square brackets. It is advisable to use `trace=FALSE` to switch off the printing of the global deviance in each `gamlss()` iteration.

The function `find.hyper()` below minimizes GAIC with `penalty=2.8`. The initial degrees of freedom parameter, `p[1]`, for the search is set to 3 (i.e. `parameters=c(3)`), the minimum value for `p[1]` for the search is set to 1 (i.e. `lower=c(1)`) and the steps in `p[1]` used within the `optim()` search to 0.1 (i.e. `steps=c(0.1)`). [The default method used by `optim()` within `find.hyper()` is the "L-BFGS-B" procedure which starts with the initial parameter value(s), changes each parameter in turn by $\pm$ step for that parameter, and then jumps to new value(s) for the set of parameter(s). This is repeated until convergence. See the help on the R function `optim()` for details.]

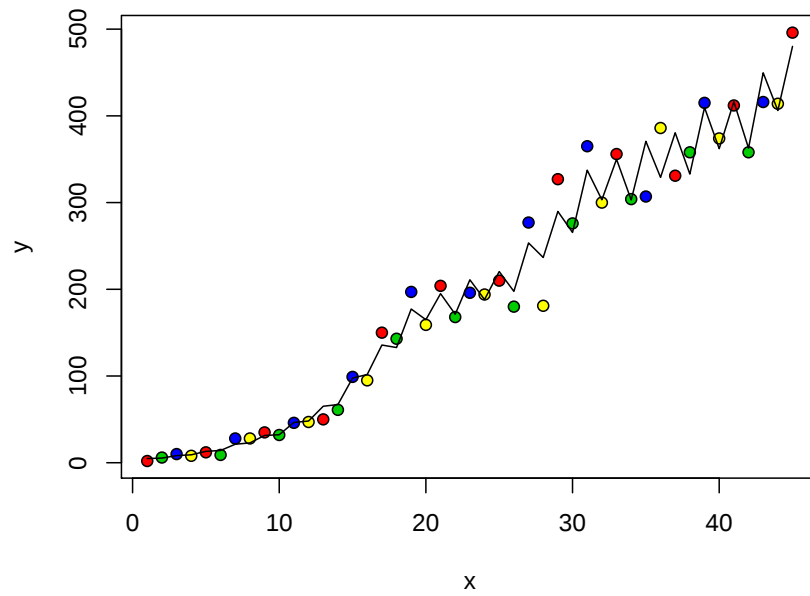


Figure 8.3: AIDS data: data and fitted values from the `aids.1` model

```

> mod1 <- quote(gamlss(y ~ cs(x, df = p[1]) + qrt, family = NBI,
+   data = aids, trace = FALSE))
> op <- find.hyper(model = mod1, par = c(3), lower = c(1), steps = c(0.1),
+   pen = 2.8, trace = FALSE)

par 3 crit= 405.0762 with pen= 2.8
par 3.1 crit= 404.6292 with pen= 2.8
par 2.9 crit= 405.5637 with pen= 2.8
par 4 crit= 401.9336 with pen= 2.8
par 4.1 crit= 401.7271 with pen= 2.8
...
...
par 7.752983 crit= 398.4014 with pen= 2.8
par 7.552983 crit= 398.4014 with pen= 2.8

> op

$par
[1] 7.652983

$value
[1] 398.3985

$counts
function gradient
      12      12

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"

```

Alternatively if the user is familiar with the R function `optim()` the following code will produce the same result:

```

> fn <- function(p) {
+   GAIC(gamlss(y ~ cs(x, df = p[1]) + qrt, family = NBI, data = aids,
+     trace = FALSE), k = 2.8)
+ }
> op1 <- optim(c(3), fn, method = "L-BFGS-B")
> op1

$par
[1] 7.647214

$value
[1] 398.3987

$counts

```

```
function gradient
      7      7
```

```
$convergence
```

```
[1] 0
```

```
$message
```

```
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

So according to the GAIC with penalty 2.8 the optimal value for the degrees of freedom is 7.65 which looks reasonable. Unfortunately this is not the end of the story. Figure 8.4 shows four plots, (generated using the function `prof.term`, see below), where different criteria have been plotted against the degrees of freedom for smoothing in the AIDS data: a) the left top figure shows the global deviance (which we expect always to decrease for increasing degrees of freedom) b) the right top shows the AIC (GAIC with penalty 2) c) the bottom left the GAIC with penalty 2.8 (the one we have just minimized) and d) the SBC (GAIC with penalty $\log(45) = 3.8$).

For the GAIC penalty=2.8 the function `find.hyper()` found a local minimum, which in this case is a reasonable solution (since the alternative around 30 degrees of freedom is too excessive for only 45 observations). Using the SBC in this case would have resulted to a much clearer minimum as shown in Figure 8.4(d). The optimum smoothing degrees of freedom in the model for μ using criterion SBC was 5.7 as shown below:

```
> fn <- function(p) {
+   GAIC(gamlss(y ~ cs(x, df = p[1]) + qrt, family = NBI, data = aids,
+   trace = FALSE), k = 3.8)
+ }
> op1 <- optim(c(3), fn, method = "L-BFGS-B")
> op1
```

```
$par
```

```
[1] 5.718166
```

```
$value
```

```
[1] 411.0879
```

```
$counts
```

```
function gradient
      7      7
```

```
$convergence
```

```
[1] 0
```

```
$message
```

```
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

So in this case the optimum smoothing degrees of freedom for μ (with minimum value of criterion $SBC \equiv GAIC(3.8)$) is 5.7.

As an explanation of why the problem arises in this specific data we compare the fitted values from the two different models, one with the degrees of freedom 10 and the other with 30.

```

> mod2 <- quote(gamlss(y ~ cs(x, df = this) + qrt, data = aids,
+   family = NBI))
> layout(matrix(c(1, 2, 3, 4), byrow = TRUE, ncol = 2))
> prof.term1(mod2, min = 1, max = 40, step = 1, criterion = "GD")
> prof.term1(mod2, min = 1, max = 40, step = 1, criterion = "IC",
+   pen = 2)
> prof.term1(mod2, min = 1, max = 40, step = 1, criterion = "IC",
+   pen = 2.8)
> prof.term1(mod2, min = 1, max = 40, step = 1, criterion = "IC",
+   pen = 3.8)

```

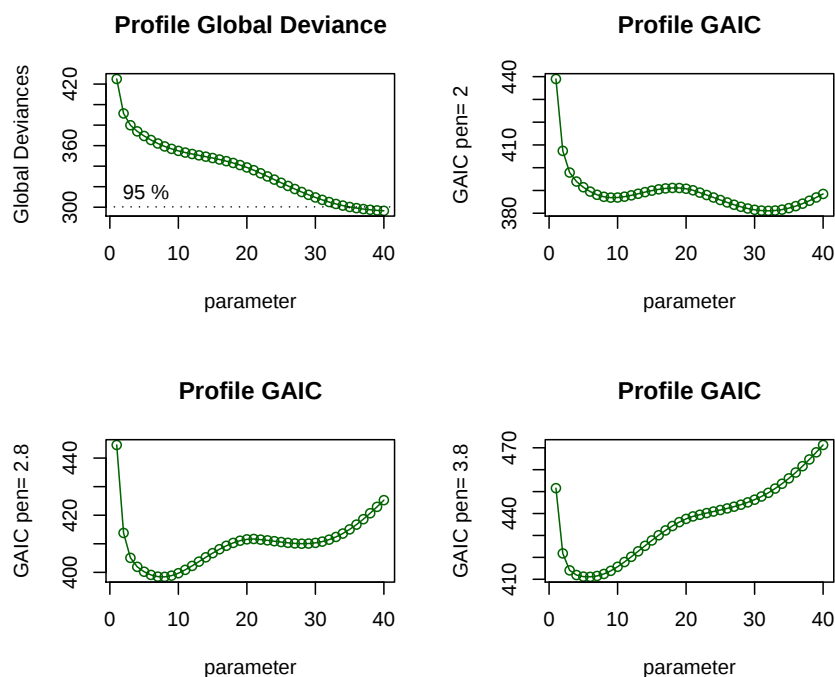


Figure 8.4: Profile global deviance and GAIC for different smoothing degrees of freedom fitted in the NBI model to the AIDS data. (a) global deviance (b) GAIC with penalty $\sharp = 2$, (c) $\sharp = 2.8$ (d) $\sharp = 3.8$, plotted against the smoothing degrees of freedom in the model for μ

Figure 8.5 shows that many of the extra degrees of freedom used in the $df = 30$ fit are there to counteract the apparent misfit of observations 28, 29, 35, 36 and 37.

```
> mod1 <- gamlss(y ~ cs(x, df = 10) + qrt, family = NBI, data = aids)

GAMLSS-RS iteration 1: Global Deviance = 358.9069
GAMLSS-RS iteration 2: Global Deviance = 354.8459
GAMLSS-RS iteration 3: Global Deviance = 354.9021
GAMLSS-RS iteration 4: Global Deviance = 354.9064
GAMLSS-RS iteration 5: Global Deviance = 354.9069

> mod2 <- gamlss(y ~ cs(x, df = 30) + qrt, family = NBI, data = aids)

GAMLSS-RS iteration 1: Global Deviance = 320.2859
GAMLSS-RS iteration 2: Global Deviance = 309.4075
...
...
GAMLSS-RS iteration 15: Global Deviance = 309.5181
GAMLSS-RS iteration 16: Global Deviance = 309.519

> plot(aids$x, aids$y)
> lines(aids$x, fitted(mod1), col = "red", lty = 1)
> lines(aids$x, fitted(mod2), col = "blue", lty = 2)
```

The abdominal data

Let us move to a different example using the abdominal circumference data first considered in Section 4.2.2. We model the abdominal circumference, Y , using a $BCT(\mu, \sigma, \nu, \tau)$ distribution where the μ and σ predictors are modelled using cubic smoothing splines in a transformed variable $t(x) = x^\xi$, where x =gestational age of the baby in weeks. Let us assume that we are interested in estimating the hyperparameters $\lambda = (df_\mu, df_\sigma, \xi)$. In the R code we use `cs(tx, df=p[1])` and `cs(tx, df=p[2])` respectively for μ and σ where $tx = x^{p[3]}$:

```
> data(abdom)
> mod1 <- quote(gamlss(y ~ cs(tx, df = p[1]), sigma.fo = ~cs(tx,
+   df = p[2]), family = BCT, data = abdom, control = gamlss.control(trace = FALSE)))
> op <- find.hyper(model = mod1, other = quote(tx <- x^p[3]), par = c(3,
+   1, 0.5), lower = c(1, 1, 0.001), steps = c(0.05, 0.05, 0.001), penalty=2.5)

par 3 1 0.5 crit= 4798.759 with pen= 2.5
par 3.05 1 0.5 crit= 4798.79 with pen= 2.5
par 2.95 1 0.5 crit= 4798.732 with pen= 2.5
par 3 1.05 0.5 crit= 4798.671 with pen= 2.5
par 3 1 0.5 crit= 4798.76 with pen= 2.5
par 3 1 0.501 crit= 4798.762 with pen= 2.5
par 3 1 0.499 crit= 4798.758 with pen= 2.5
par 2.696051 1.916867 0.2412168 crit= 4798.392 with pen= 2.5
par 2.746051 1.916867 0.2412168 crit= 4798.411 with pen= 2.5
...
...
```

```
> plot(aids$x, aids$y)
> lines(aids$x, fitted(mod1), col = "red")
> lines(aids$x, fitted(mod2), col = "blue")
```

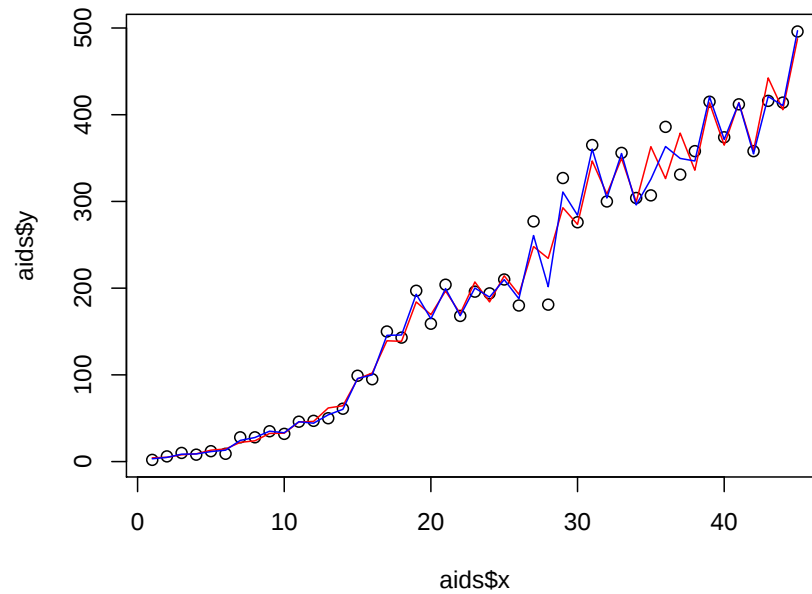


Figure 8.5: AIDS data: fitted NBI models with $df = 10$ (red solid line) and $df = 30$ (blue dashed, line)

```
par 2.606039 1.274147 0.2202409 crit= 4797.948 with pen= 2.5
par 2.606039 1.324147 0.2212409 crit= 4797.948 with pen= 2.5
par 2.606039 1.324147 0.2192409 crit= 4797.944 with pen= 2.5
```

```
> op
```

```
$par
[1] 2.6060394 1.3241473 0.2202409
```

```
$value
[1] 4797.944
```

```
$counts
function gradient
      23      23
```

```
$convergence
[1] 0
```

```
$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

The selected hyperparameters are 2.6 degrees of freedom for smoothing μ , 1.32 degrees of freedom for smoothing the σ and $x^{0.22}$ for the power parameter.

We refit the model and plot its fitted parameters.

```
> tx <- abdom$x^0.22
> m1 <- gamlss(y ~ cs(tx, df = 2.6), sigma.fo = ~cs(tx, df = 1.32),
+   family = BCT, data = abdom)
```

```
GAMLSS-RS iteration 1: Global Deviance = 4774.341
GAMLSS-RS iteration 2: Global Deviance = 4773.194
GAMLSS-RS iteration 3: Global Deviance = 4773.156
GAMLSS-RS iteration 4: Global Deviance = 4773.151
GAMLSS-RS iteration 5: Global Deviance = 4773.15
GAMLSS-RS iteration 6: Global Deviance = 4773.15
```

```
> fitted.plot(m1, x = abdom$x)
```

Figure 8.6 shows all the fitted parameter estimates plotted against x .

8.6 Exercises for Chapter 8

- Q1: Use the aep data introduced in section 6.2.3 :

Start with a binomial null model

```
mbi1<-gamlss(y~1,data=aep, family=BI)
```

and try to find the best two way interaction model using

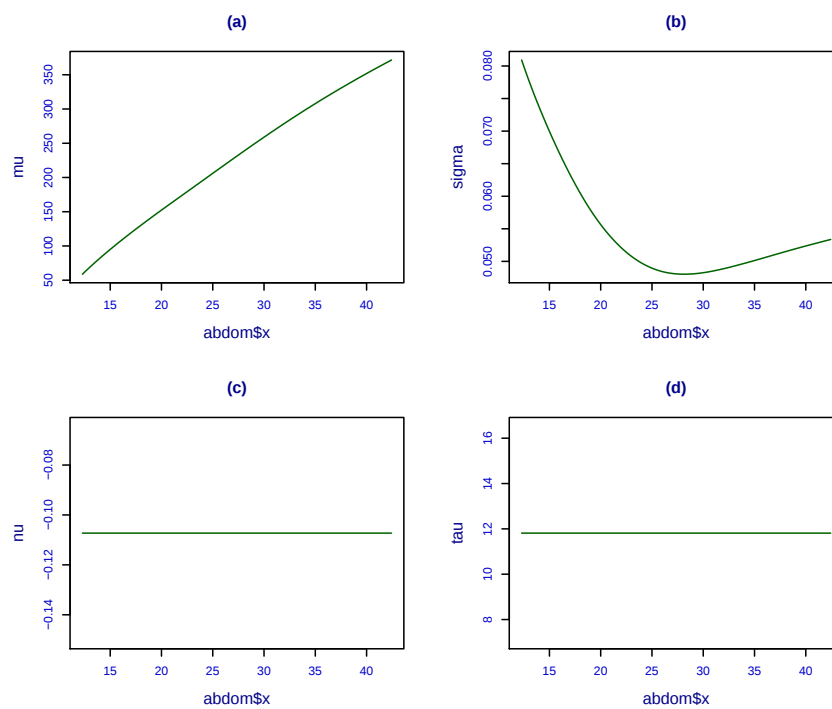


Figure 8.6: Abdominal circumference data: fitted μ , σ , ν and τ against x

```
mbi2<-stepGAIC(mbi1,scope=list(upper=~(loglos+sex+ward+year+age)^2))
```

The component `mbi2$anova` provides you with an anova table of the possible change in the model. Repeat the above procedure by fitting a beta binomial model (BB) this time. Compare the resulting BI and BB models in terms of complexity and AIC.

Use the BB distribution model and try to select a good model for sigma using.

```
mBB21<-stepGAIC(mBB2,what="sigma",
  scope=list(upper=~(loglos+sex+ward+year+age)^2)) year+age)^2))
```

where `mBB2` is the chosen BB model from the previous selection. Now recheck the mu model by backward elimination using

```
mBB22<-stepGAIC(mBB21,what="mu")
```

To check whether smoothing is needed for the continuous variables `loglos` and `age` try the following commands:

```
mBB23stepGAIC(mBB22,scope= gamlss.scope(model.frame(y~loglos+age)),what="mu")
```

Finally try to find the appropriate smoothing degrees of freedom using the `find.hyper` as in section 5.3 using `penalty=3`.

Chapter 9

Smoothing

9.1 Introduction

Suppose we have n measurements of a response variable $\mathbf{y} = (y_1, y_2, \dots, y_n)^\top$ and a single explanatory variable $\mathbf{x} = (x_1, x_2, \dots, x_n)^\top$ and we want to study their relationship. First we would plot the variables against each other. A curve through the data would help us to make up our mind on the kind of relationship existing between the two variables. *Scatterplot Smoothers* are statistical devices which could help us to fit curves in situations like this. A *scatterplot smoother*, or for convenience, a smoother summarizes the trend of the response variable as a function of $\mathbf{x}$. The smoother, in statistical terms, is often an estimate of the conditional mean or expected value of $\mathbf{y}$ given $\mathbf{x}$, $E(\mathbf{y}|\mathbf{x})$, but more generally, can be an estimate of any other location parameter of the distribution of $\mathbf{y}$, e.g. the median. The smoother describes the trend of the response variable $\mathbf{y}$ given the values of $\mathbf{x}$. An important property of a smoother is that it does not assume any parametric functional form for the dependence of $\mathbf{y}$ on $\mathbf{x}$ but lets the data decide. This is the reason why the smoother is a tool in *non-parametric* regression.

Smoothers are generalizations of the simple linear regression model and formally as statistical models can be expressed as

$$\mu_i = a + h(x_i) \tag{9.1}$$

for $i = 1, \dots, n$, where $h(\cdot)$ is an arbitrary function which we assume to exist, a is a constant which most of the time we absorb into the function $h(\cdot)$, so $\mu_i = h(x_i)$, and μ is the trend that we would like to estimate. Typically, for continuous response variables we also assume that $Y_i \sim N(\mu_i, \sigma^2)$ for $i = 1, \dots, n$ and that Y_i 's are independent. The function $h(\cdot)$ is arbitrarily defined but very often, to make it more regular, we assume that it has some properties, for example continuous first and second derivatives. Smoothers can be helpful in situations where the response variable is not continuous. Consider the data in Figure 9.1. Here we have a scatter plot of 10590 observations with the y variable being a binary response variable. The scatter plot (ignoring for the moment the fitted smoothing curve in the middle) is in this case completely uninformative. The data were kindly provided by Prof Brian Francis. The response variable is whether a particular crime was reported ($y=1$) or not ($y=0$) in the media. The explanatory variable is the age of the victim. The smoothing curve in the figure is obtained using a cubic spline smoother and by assuming that the response is a binomially distributed variable. The curve shows the estimated probability of reporting a crime in the media according to age. It shows that the probability that a crime is reported is higher when it is committed on a young

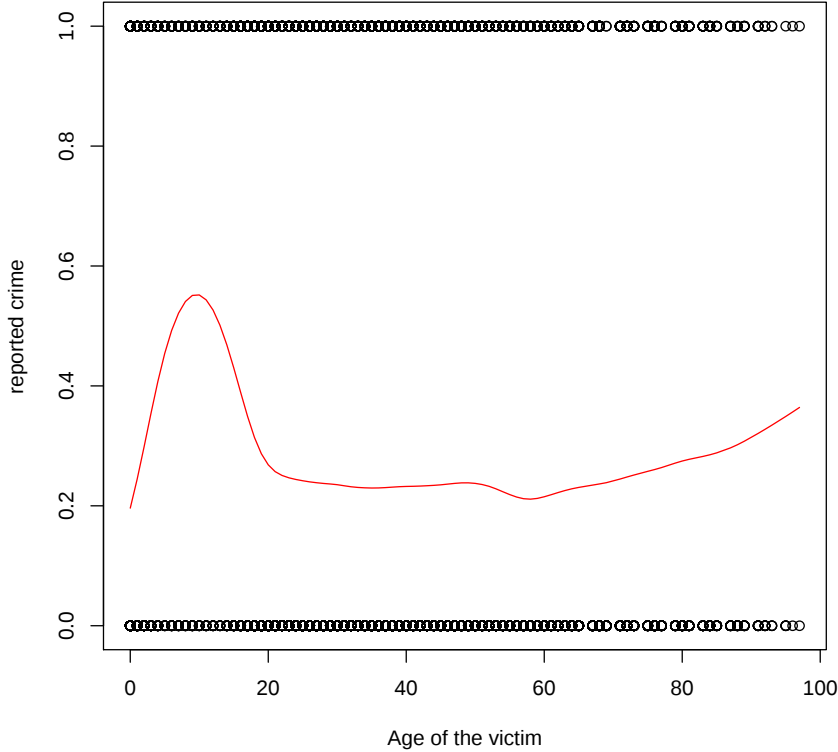


Figure 9.1: Showing a scatter plot of the reported cases by media against the age of the of victims.

person, with a peak at around age ten. The probability then declines until the victim reaches the age of twenty. From then on, the probability, remains constant until the age of sixty after which the reporting rate rises steadily with age.

A scatterplot smoother has two main uses within statistical modelling. It can be used on its own to describe the relation between $\mathbf{y}$ and $\mathbf{x}$ or it can be utilized in an additive model. An *additive model* is of the form

$$\mu = a + \sum_{j=1}^p h_j(x_j) \quad (9.2)$$

where $f_j(\cdot)$'s are univariate functions of the explanatory variables and a is a constant which can be absorbed into one of the $f_j(\cdot)$'s functions. As usual, the additional assumption that the $Y_i \sim N(\mu_i, \sigma^2)$ for $i = 1, \dots, n$ and that they are independent should be made. Additive models are easy to interpret because each function can be viewed separately. It is difficult for us to perceive functions in more than three dimensions (and impossible to plot).

There was an explosion of statistical smoothing techniques at the late 80's and early 90's section and the reader is referred to Hastie and Tibshirani (1990), and Green and Silverman

(1994) for more details.

An advantage of a smoother over a parametric fitting function is its local behaviour. Smoothers are affected by local observations more than by observations far away. This can be achieved by local averaging or by fitting functions locally (for example polynomials) or by fitting piecewise polynomials. When local averaging is involved weighting locally the observation produces more appealing results to the eye than the simple averaging. Also fitting polynomials (rather than constants) locally helps the behaviour of the smoother at the ends of the data. In GAMLSS we use several different smoothing functions as additive terms. Currently we have implemented the following smoothers:

`cs()` Cubic spline

`vc()` Varying coefficient

`ps()` Penalized spline

`lo()` `loess`

The first three are based on piecewise polynomials functions while the last one on locally weighted polynomial fitting. In the next few sections we will explain the use of the smoothers within GAMLSS in more detail. Next we introduce the `rent` data which we will use extensively in this chapter.

9.2 Data sets

In this chapter we will use the following data sets for demonstration.

9.2.1 The crash helmet data

Data summary: the helmet data

R data file: `mcycle` in package **MASS** of dimensions 133×2

variables

`accel` : head acceleration in a simulated motorcycle accident, used to test crash helmets, in g.

`times` : time in milliseconds after impact

purpose: to demonstrate smoothing techniques

The crash helmet data were given by Silverman (1985) and are considered as the classic example for demonstrating smoothing techniques. The data comprise 133 observations of accelerometer readings (`accel`) taken through time (`times`) in an experiment on the efficacy of crash helmets. The data are use in Section [9.5](#).

9.2.2 The rent data

Data summary:

R data file: `rent` in package `gamlss` of dimensions 1969×9 originally given by Fahrmeir *et al.* (1994, 1995).

variables

R : rent response variable, the monthly net rent in DM, i.e. the monthly rent minus calculated or estimated utility cost.

F1 : floor space in square meters

A : year of construction

SP : a two level factor indicating whether the location is above average (550 observations) or not (1419 observations)

SM : a two level factor indicating whether the location is below average (172 obs.) or not (1797 obs.)

B : a two level factor indicating whether there is a bathroom (1925 obs.) or not (44 obs.)

H : a two level factor indicating whether there is central heating (1580 obs.) or not (389 obs.)

L : a two level factor indicating whether the kitchen equipment is above average (161 obs.) or not (1808 obs.)

purpose: to demonstrate the fitting of smoothing functions in `gamlss`

conclusion: A gamma distribution (GA) is appropriate here

A survey was conducted in April 1993 by Infratest Sozialforschung. A random sample of accommodation with new tenancy agreements or increases of rents within the last four years in Munich was selected including: i) single rooms, ii) small apartments, iii) flats, iv) two-family houses. Accommodation subject to price control rents, one family houses and special houses, such as penthouses, were excluded because they are rather different from the rest and are considered a separate market. For the purpose of this study, 1967 observations of the variables listed below were used, i.e. the rent response variable `R` followed by the explanatory variables found to be appropriate for a regression analysis approach by Fahrmeir *et al.* (1994, 1995):

```
> library(gamlss)
> data(rent)
> op <- par(mfrow = c(3, 2))
> plot(R ~ F1, data = rent, xlab = "floor")
> plot(R ~ A, data = rent, xlab = "year")
> plot(R ~ loc, data = rent, xlab = "location")
> plot(R ~ factor(B), data = rent, xlab = "bathroom")
> plot(R ~ factor(H), data = rent, xlab = "heating")
> plot(R ~ factor(L), data = rent, xlab = "kitchen")
> par(op)
```

Each of the five factors above were coded as 0 and 1, with 1 indicating the category with

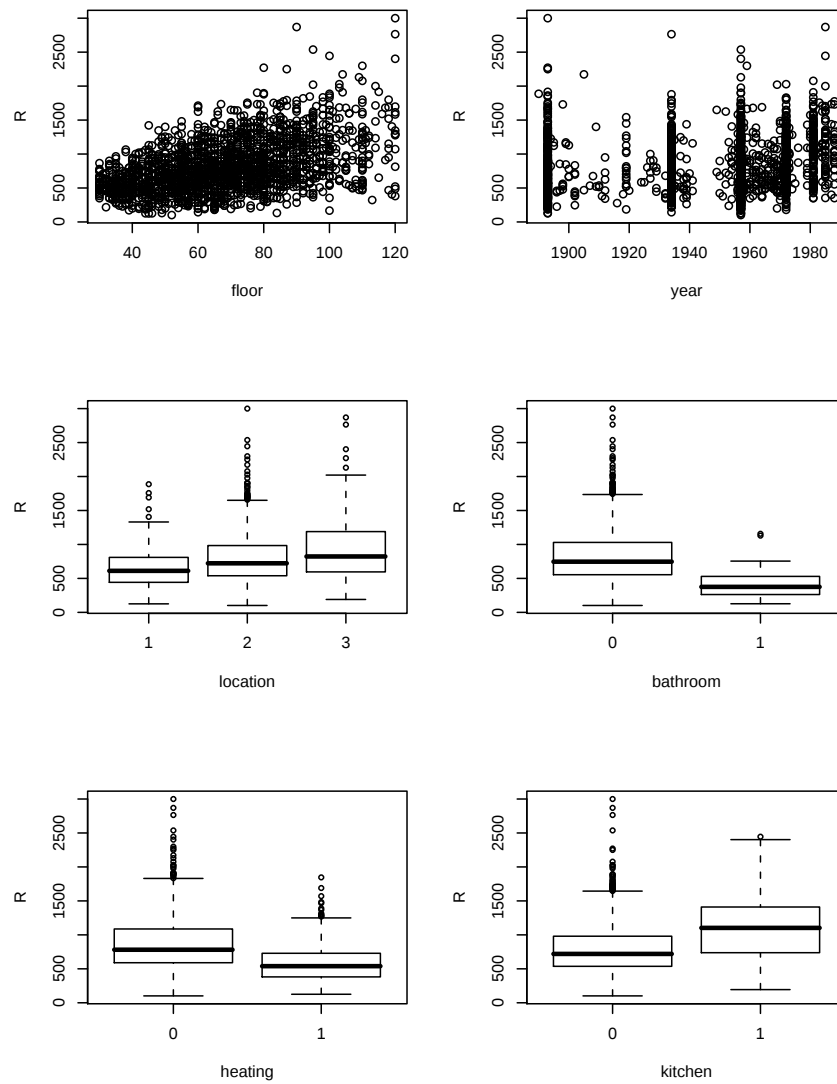


Figure 9.2: Plots of net rent (R) against each of the explanatory variables.

smaller number of observations, so that in the later analysis factor effects relate to the smaller observation groups.

Figure 9.2 shows plots of the net rent R against each of the above explanatory variables. Although these are bivariate exploratory plots and take no account of the interplay between the explanatory variables, they give an indication of the complexity of this data. The first two explanatory variables, $F1$ and A , are continuous. The plot of rent, R , against floor space $F1$ suggests a positive relationship with an increased variation for larger floor spaces. The assumption of homogeneity in the variance of the rent variable appears to be violated here. There is also some indication of positive skewness in the distribution of the rent variable. The peculiarity of the plot of rent R against year of construction A is due to the method of data collection. Many of the observations of A were collected on an interval scale and assigned the value of the interval midpoint, while for the rest the actual year of construction was recorded. The plot suggests that for houses up to 1960 the rent prices are roughly constant but for flats constructed after that year there is an increasing trend in the rent price. The remaining box and whisker plots display how the rent price varies according to the explanatory factors. There are no surprises in the plots here but again the problem of skewness is prominent with non symmetrical boxes about the median and longer upper than lower whiskers.

In summary, any statistical model used for the analysis of the above data should be able to deal with the following statistical problems:

Problem I : The complexity of the relationship between net rent and the explanatory variables. The dependence of rent on floor space and age of construction is nonlinear and non-parametric smoothing functions may be needed. Rent may also depend on nonlinear interactions between the explanatory variables.

Problem II : Non-homogeneity of variance. There is clear indication of non-homogeneity of the variance of rent. The variance of rent may depend on its mean and/or explanatory variables e.g. floor space. A statistical model is needed where this dependence can be modelled explicitly.

Problem III : Skewness in the distribution of the response variable. There is clear indication of skewness in the distribution of net rent and this has to be accounted for within the statistical model.

9.3 Cubic splines, the $cs()$ function

Here we assume in GAMLSS model (2.2) that the functions $h(t)$ are arbitrary twice continuously differentiable functions and we maximize a penalized log likelihood, given by ℓ subject to penalty terms of the form $\lambda \int_{-\infty}^{\infty} [h''(t)]^2 dt$. Following Reinsch (1967), the maximizing functions $h(t)$ are all natural cubic splines, and hence can be expressed as linear combinations of their natural cubic spline basis functions, $B_i(t)$ for $i = 1, 2, \dots, n$, (de Boor, 1978 and Schumaker, 1993), i.e. $h(t) = \sum_{i=1}^n \delta_i B_i(t)$. Let $\mathbf{h} = h(\mathbf{x})$ be the vector of evaluations of the function $h(t)$ at the values $\mathbf{x}$ of the explanatory variable X (assumed to be distinct for simplicity of exposition). Let $\mathbf{N}$ be an $n \times n$ nonsingular matrix containing as its columns the n vectors of evaluations of functions $B_i(t)$, for $i = 1, 2, \dots, n$, at $\mathbf{x}$. Then $\mathbf{h}$ can be expressed using coefficient vector $\boldsymbol{\delta}$ as a linear combination of the columns of $\mathbf{N}$ by $\mathbf{h} = \mathbf{N}\boldsymbol{\delta}$. Let $\boldsymbol{\Omega}$ be the $n \times n$ matrix of inner products of the second derivatives of the natural cubic spline basis functions, with $(r, s)^{th}$ entry given by $\Omega_{rs} = \int B_r''(t) B_s''(t) dt$. The penalty is then given by the quadratic form

$Q(\mathbf{h}) = \lambda \int_{-\infty}^{\infty} [h''(t)]^2 dt = \lambda \boldsymbol{\delta}^T \boldsymbol{\Omega} \boldsymbol{\delta} = \lambda \mathbf{h}^T \mathbf{N}^{-T} \boldsymbol{\Omega} \mathbf{N}^{-1} \mathbf{h} = \lambda \mathbf{h}^T \mathbf{K} \mathbf{h}$, where $\mathbf{K} = \mathbf{N}^{-T} \boldsymbol{\Omega} \mathbf{N}^{-1}$ is a known penalty matrix that depends only on the values of the explanatory vector $\mathbf{x}$ (Hastie and Tibshirani, 1990, Ch. 2). The precise form of the matrix $\mathbf{K}$ can be found in Green and Silverman (1994) Section 2.1.2.

The model can be formulated as a random effects GAMLSS (2.1) by letting $\boldsymbol{\gamma} = \mathbf{h}$, $\mathbf{Z} = \mathbf{I}_n$, $\mathbf{K} = \mathbf{N}^{-T} \boldsymbol{\Omega} \mathbf{N}^{-1}$ and $\mathbf{G} = \lambda \mathbf{K}$, so that $\mathbf{h} \sim N_n(0, \lambda^{-1} \mathbf{K}^{-1})$, a partially improper prior (Silverman, 1985). This amounts to assuming complete prior uncertainty about the constant and linear functions and decreasing uncertainty about higher order functions, see Verbyla *et al.* (1999).

The cubic splines function in GAMLSS is based on the `smooth.spline` function of R and can be used for univariate smoothing. It fits a cubic smoothing spline function, see Hastie and Tibshirani (1990) or Green and Silverman (1994).

As an example of using the smoothing cubic spline function `cs()` we use the rent data first used in section 9.2.2. The response variable `R`, is the monthly net rent (rent minus calculated or estimated utility cost). Here we fit an additive model for floor space (`F1`) and for age (`A`) the year of construction.

```
> r1 <- gamlss(R ~ cs(F1) + cs(A), data = rent, family = GA)
```

```
GAMLSS-RS iteration 1: Global Deviance = 27920.65
```

```
GAMLSS-RS iteration 2: Global Deviance = 27920.74
```

```
GAMLSS-RS iteration 3: Global Deviance = 27920.74
```

```
> op <- par(mfrow = c(2, 1))
> term.plot(r1, se = TRUE, partial = TRUE)
> par(op)
```

The additive curves created by the function `term.plot` are shown in Figure 9.3. We can also plot the additive fitted surface using the R functions `contour()` and `wireframe` (the latest can be found in the package `lattice`).

```
> par(mfrow = c(1, 1))
> newrent <- data.frame(expand.grid(F1 = seq(30, 120, 5), A = seq(1890,
+   1990, 5)))
> newrent$pred <- predict(r1, newdata = newrent, type = "response")
> Fln <- seq(30, 120, 5)
> An <- seq(1890, 1990, 5)
> contour(Fln, An, matrix(newrent$pred, nrow = length(Fln)), nlevels = 30,
+   ylab = "year of construction", xlab = "floor space")
> library(lattice)
> wireframe(pred ~ F1 * A, newrent, aspect = c(1, 0.5), drape = TRUE,
+   colorkey = list(space = "right", height = 0.6))
```

See Figures 9.4 and 9.5 for the two plots respectively.

9.4 Varying coefficient, the `vc()` function

The varying coefficient terms were introduced by Hastie and Tibshirani (1993) to accommodate a special type of interaction between explanatory variables. This interaction takes the form

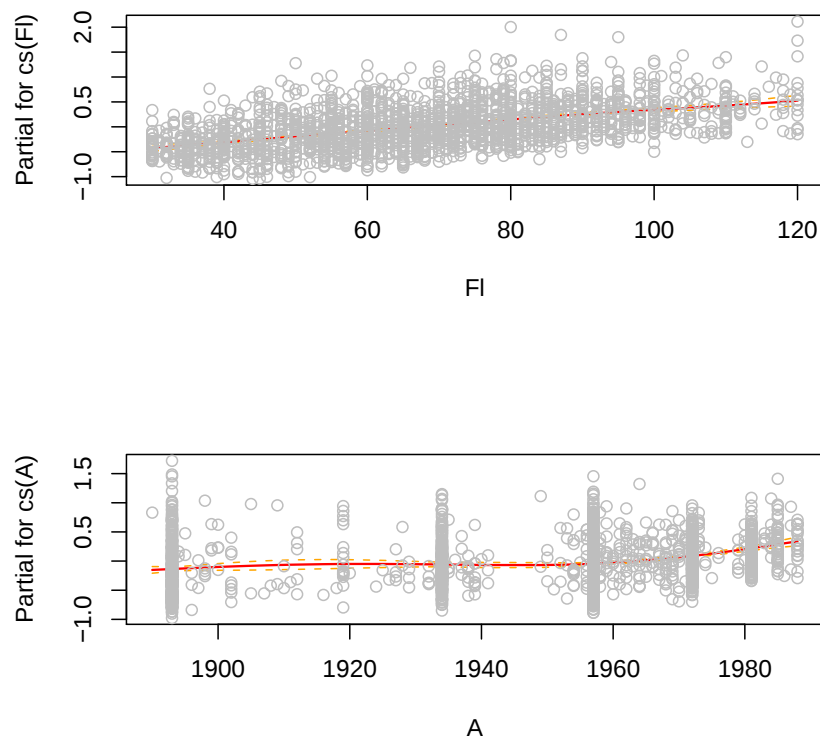


Figure 9.3: Rent data: Additive plots for the cubic splines model

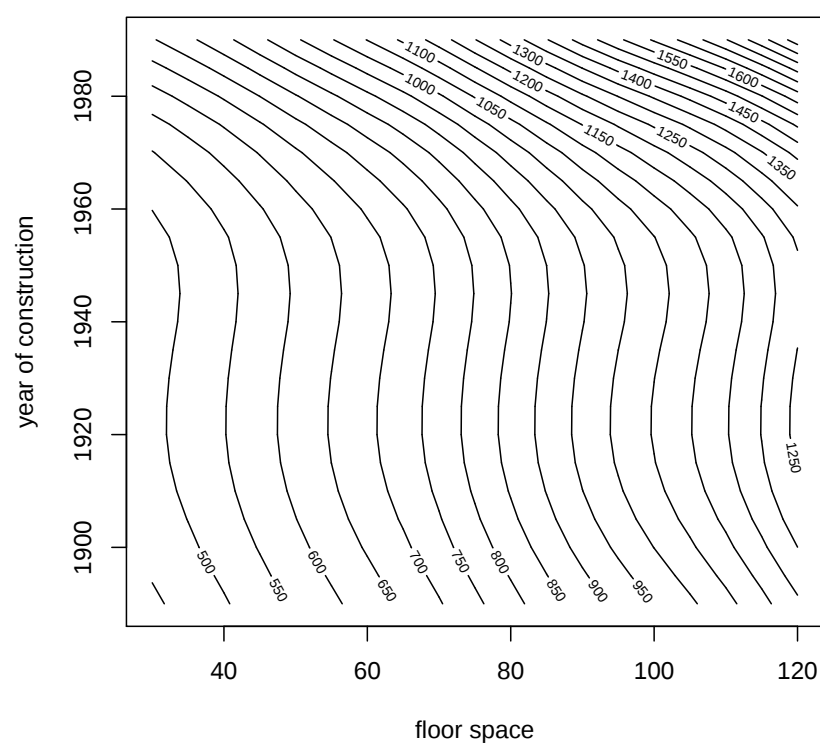


Figure 9.4: Rent data: contour plot for the fitted additive cubic splines model model

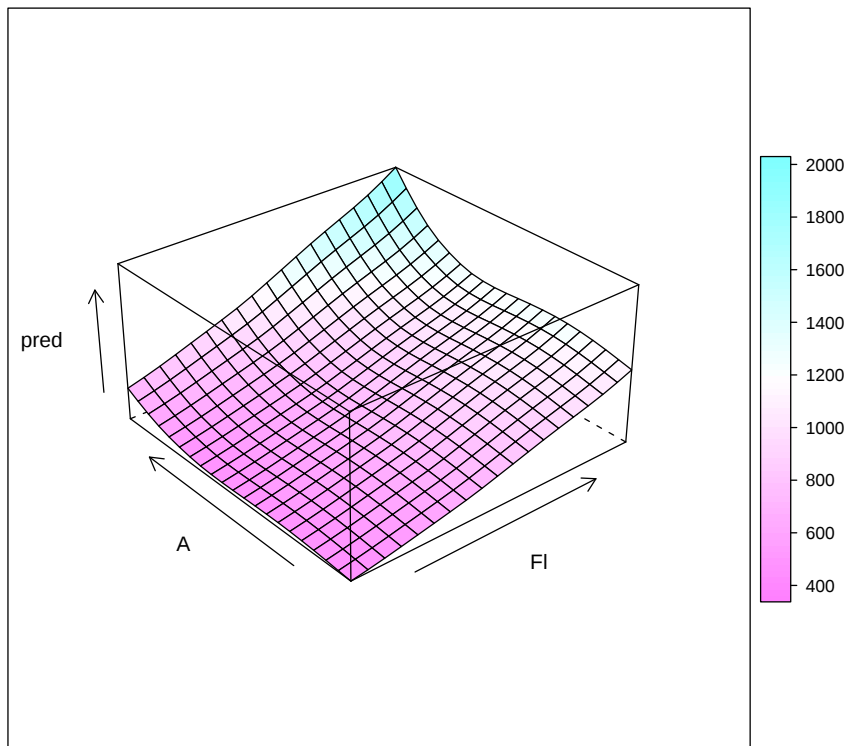


Figure 9.5: Rent data: surface plot for the fitted additive cubic splines model model

of $\beta(R)X$, that is the linear coefficient of the explanatory variable X is changing smoothly according to another explanatory variable R . In many application R will be time. In general R should be a continuous variable while X can be either continuous or categorical. In the current GAMLSS implementation X has to be continuous or a two level factor with levels 0 and 1.

As an example of using the function `vc()` consider the following models for μ in the `rent` data example discussed earlier in Section 9.2.2, using the default 3 additional degrees of freedom for each smoothing term on top of the linear term.

```
> m1 <- gamlss(R ~ cs(Fl) + cs(A), data = rent, family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27920.65
GAMLSS-RS iteration 2: Global Deviance = 27920.74
GAMLSS-RS iteration 3: Global Deviance = 27920.74

> m2 <- gamlss(R ~ cs(Fl) + cs(A) + vc(r = A, x = Fl), data = rent,
+   family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27901.33
GAMLSS-RS iteration 2: Global Deviance = 27901.50
GAMLSS-RS iteration 3: Global Deviance = 27901.50

> AIC(m1, m2)

      df      AIC
m2 14.00192 27929.51
m1 10.00128 27940.74
```

Obviously the varying coefficient model is an improvement over the simple additive model. Note that the function `term.plot()` for plotting additive terms is not producing the right plot for the varying coefficient terms, (for the additive terms it works fine). Also note that while the variables `Fl` and `A` are not centered before the fit `Fl` will be automatically centered, so to plot the fitted varying coefficient function $\beta(A)$ use the following, (see Figure 9.6(a) for the plots):

```
> with(rent, plot(m2$mu.s[, 3]/(Fl - mean(Fl)) ~ A, ylab = expression(beta(A))))
```

Note that we divide the third column of the `mu` smoothing matrix, `mu.s`, which contains the term $\beta(A)(Fl - \bar{Fl})$ by $(Fl - \bar{Fl})$ in order to obtain the right function $\beta(A)$. Also note that the plotted function is the extra contribution after the main effects and interaction have been fitted for the two variables involved, i.e. `Fl*A`, which is automatically put by `vc()` in the linear part of the model. To plot the fitted surface use:

```
> newrent <- data.frame(expand.grid(Fl = seq(30, 120, 5), A = seq(1890,
+   1990, 5)))
> newrent$pred <- predict(m2, newdata = newrent, type = "response",
+   data = rent)
> Fln <- seq(30, 120, 5)
> An <- seq(1890, 1990, 5)
> contour(Fln, An, matrix(newrent$pred, nrow = length(Fln)), nlevels = 30,
+   ylab = "year of construction", xlab = "floor space")
> library(lattice)
> wireframe(pred ~ Fl * A, newrent, aspect = c(1, 0.5), drape = TRUE,
+   colorkey = list(space = "right", height = 0.6))
```

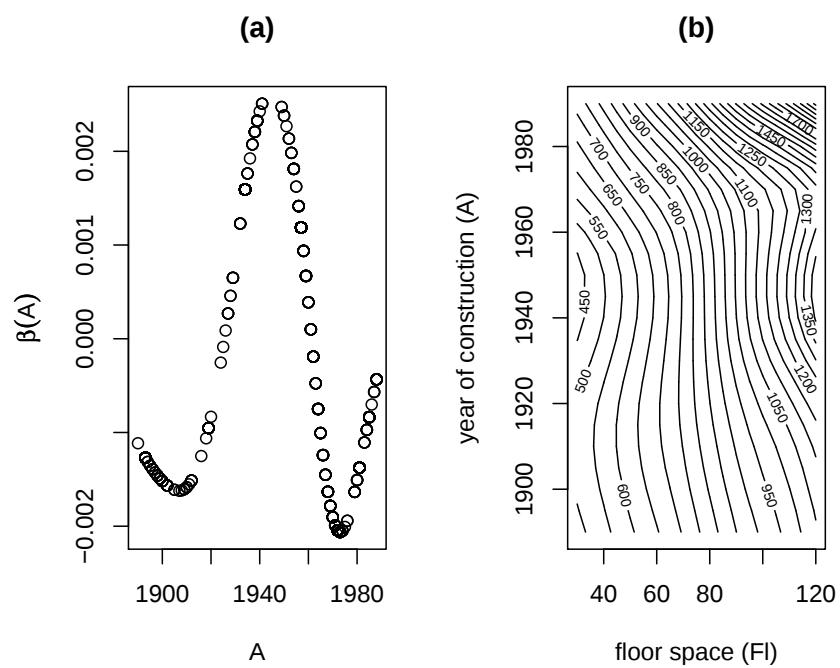


Figure 9.6: Rent data: (a) the $\beta(A)$ function (b) a contour plot for the fitted additive cubic splines model

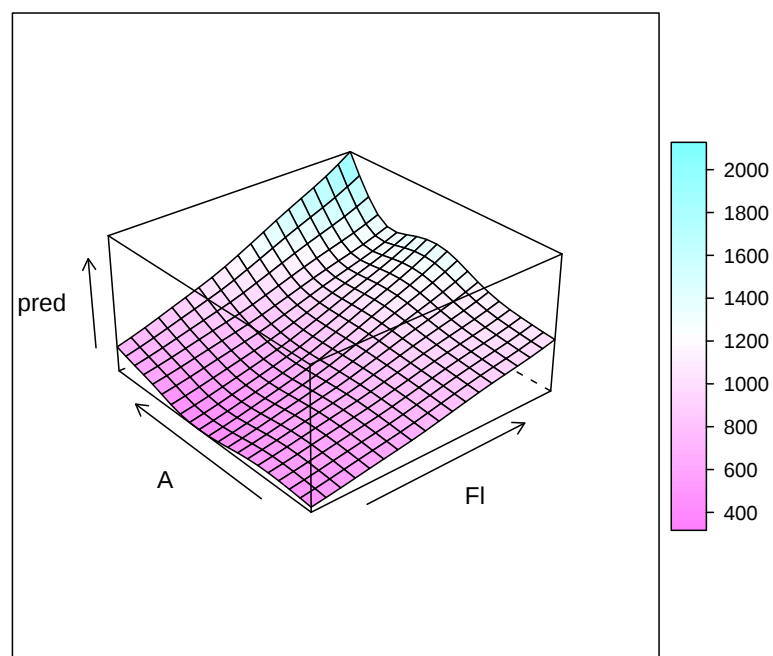


Figure 9.7: Rent data: surface plot for the fitted additive cubic splines model

The contour and surface plots are shown in Figures 9.6(b) and 9.7 respectively.

As an example of using a factor rather than continuous variable for X in the additive function `vc()`, consider the model used in Stasinopoulos *et al.* (2000). The authors have fitted the following models for `mu` and `sigma` respectively as their final model

$$\begin{aligned}\mu = & f_1(Fl) + f_2(A) + Fl * f_3(A) + H * f_4(Fl) \\ & + Fl * (Sp + B) + Sm + L\end{aligned}\tag{9.3}$$

$$\log(\sigma) = h_1(Fl) + h_2(A) + Fl * (Sp + Sm)\tag{9.4}$$

where, the f 's and h 's are smooth functions, the Fl and A are the floor space (in square meters) and year of construction respectively, the H , B and L are factors indicating whether there is central heating, a bathroom and above average kitchen equipment respectively and the Sp and Sm are dummy variables indicating whether the location is above average or below average respectively. Here we first fit the above model using an (identity) link function for `mu`, then we refit it using the default `log` link and finally we fit a model suggested by Prof. John Nelder (2001) in a subsequent published correspondence. All models fitted use the default 3 extra degrees of freedom for smoothing and are compared using a GAIC with penalties 2 (AIC) and $\log(n)$ (SBC).

```
> m3 <- gamlss(R ~ cs(Fl) + cs(A) + vc(r = A, x = Fl) + vc(r = Fl,
+      x = H) + Fl * (Sp + B) + Sm + L, sigma.fo = ~cs(Fl) + cs(A) +
+      Fl * (Sp + Sm), family = GA(mu.link = "identity"), data = rent)
```

```
GAMLSS-RS iteration 1: Global Deviance = 27486.52
GAMLSS-RS iteration 2: Global Deviance = 27481.32
GAMLSS-RS iteration 3: Global Deviance = 27481.28
GAMLSS-RS iteration 4: Global Deviance = 27481.3
GAMLSS-RS iteration 5: Global Deviance = 27481.3
GAMLSS-RS iteration 6: Global Deviance = 27481.3
```

```
> m4 <- gamlss(R ~ cs(Fl) + cs(A) + vc(r = A, x = Fl) + vc(r = Fl,
+      x = H) + Fl * (Sp + B) + Sm + L, sigma.fo = ~cs(Fl) + cs(A) +
+      Fl * (Sp + Sm), family = GA, data = rent)
```

```
GAMLSS-RS iteration 1: Global Deviance = 27481.44
GAMLSS-RS iteration 2: Global Deviance = 27477.9
GAMLSS-RS iteration 3: Global Deviance = 27477.87
GAMLSS-RS iteration 4: Global Deviance = 27477.87
GAMLSS-RS iteration 5: Global Deviance = 27477.87
```

```
> m5 <- gamlss(R ~ cs(Fl) + cs(A) + Sp + B + Sm + L + H + Fl *
+      Sp, sigma.fo = ~cs(A), family = GA, data = rent)
```

```
GAMLSS-RS iteration 1: Global Deviance = 27555.3
GAMLSS-RS iteration 2: Global Deviance = 27553.49
GAMLSS-RS iteration 3: Global Deviance = 27553.48
GAMLSS-RS iteration 4: Global Deviance = 27553.48
```

```
> GAIC(m1, m2, m3, m4, m5)
```

```

      df      AIC
m4 37.00108 27551.87
m3 37.00103 27555.30
m5 20.00054 27593.48
m2 14.00192 27929.51
m1 10.00128 27940.74

```

```
> GAIC(m1, m2, m3, m4, m5, k = log(length(rent$R)))
```

```

      df      AIC
m5 20.00054 27705.19
m4 37.00108 27758.53
m3 37.00103 27761.97
m1 10.00128 27996.60
m2 14.00192 28007.71

```

Model 4 seems superior with the original AIC but with SBC Prof. Nelder's suggestion comes best. [Note that the global deviance reported to in Stasinopoulos *et al* (2000) is different from the one produce by GAMLSS. This is due to the fact that degrees of freedom in Stasinopoulos *et al* (2000) were calculated differently.]

9.5 Penalized splines, the `ps()` function

Penalized Splines (or P-splines) are piecewise polynomials defined by B-spline basis functions in the explanatory variable where the coefficients of the basis functions are penalized to guarantee sufficient smoothness, see Eilers and Marx (1996). More precisely consider the model $g(\boldsymbol{\theta}) = \mathbf{Z}(\mathbf{x})\boldsymbol{\gamma}$ where $\boldsymbol{\theta}$ can be any distributional parameter in a GAMLSS model and $g(\cdot)$ is the link function for $\boldsymbol{\theta}$, $\mathbf{Z}(\mathbf{x})$ is $n \times q$ basis design matrix for the explanatory variable $\mathbf{x}$ defined at $q - \text{degree} - 1$ different knots mostly within the range of $\mathbf{x}$ and $\boldsymbol{\gamma}$ is a $q \times 1$ vector of coefficients which have some stochastic restrictions imposed by assuming that $\mathbf{D}\boldsymbol{\gamma} \sim N(\mathbf{0}, \lambda^{-1}\mathbf{I})$ or equivalently by $\boldsymbol{\gamma} \sim N(\mathbf{0}, \lambda^{-1}\mathbf{K}^{-})$ where $\mathbf{K} = \mathbf{D}^T\mathbf{D}$. The matrix $\mathbf{D}$ is a $(q - r) \times q$ matrix giving r th differences of the q -dimensional vector $\boldsymbol{\gamma}$. So to define a penalized spline we need: i) q the number of knots in the x-axis, defined by argument `ps.intervals` (and of course where to put them, `ps` uses equal spaces in the x-axis), ii) the degree of the piecewise polynomial used in the B-spline basis so we can define $\mathbf{X}$, defined by argument `degree` iii) r the order of differences in the $\mathbf{D}$ matrix indicating the type of the penalty imposed in the the coefficients of the B-spline basis functions, defined by argument `order` and iv) the amount of smoothing required defined either by the desired equivalent degrees of freedom defined by argument `df` [or alternatively by the smoothing parameter defined by argument `lambda`].

The `ps()` function in GAMLSS is based on an S function of Brian Marx obtained from <http://www.stat.lsu.edu/faculty/marx/>

As an example consider the crash helmet data from Silverman (1985). The data comprise 133 observations of accelerometer readings (`accel`) taken through time (`times`) in an experiment on the efficacy of crash helmets. The data need modelling of both the mean and the variance of `accel` as shown in Rigby and Stasinopoulos (1996a), but to start off here we use a model only for the mean.

Figure 9.8 show the basis function of `degree= 0, 1, 2, 3` for `times` having 9 inner knots, i.e. 10 intervals for x , i.e. `ps.intervals=10`. The figure is generated using

```

> library(MASS)
> data(mcycle)
> plotBbasis <- function(x, nknots, degree) {
+   dx <- (max(x) - min(x))/nknots
+   sorder <- degree + 1
+   Aknots <- seq(from = min(x) - degree * dx, to = max(x) +
+     degree * dx, by = dx)
+   basis <- splineDesign(Aknots, x, sorder, 0 * x, outer = T)
+   mmm <- paste("degree =", degree, sep = " ")
+   plot(c(min(x), max(x)), c(0, 1), type = "n", xlab = "x",
+     ylab = "", main = mmm)
+   matlines(x, basis, lty = 1)
+ }
> op <- par(mfrow = c(4, 1))
> plotBbasis(x = mcycle$times, nknots = 10, degree = 0)
> plotBbasis(x = mcycle$times, nknots = 10, degree = 1)
> plotBbasis(x = mcycle$times, nknots = 10, degree = 2)
> plotBbasis(x = mcycle$times, nknots = 10, degree = 3)
> par(op)

```

Now we fix the number of knots to the default value 20, the order to 3 and the degrees of freedom to 15. We vary the `degree` of the piecewise polynomials used to 1, 2, and 3 corresponding to using default value linear, quadratic and cubic polynomial in the basis functions respectively. The resulting fits are shown in Figure 9.9. It is obvious in this example that the different degrees of the piecewise polynomials used in the B-spline basis functions make little difference to the fit, but the `degree` 2 and 3 produce smoother curves than the one when `degree=1`. Do not use `degree=0` as in this example the backfitting does not converge.

```

> op <- par(mfrow = c(3, 1))
> plot(accel ~ times, data = mcycle, main = "degree=1")
> m1 <- gamlss(accel ~ ps(times, df = 15, degree = 1), data = mcycle)

GAMLSS-RS iteration 1: Global Deviance = 1195.225
GAMLSS-RS iteration 2: Global Deviance = 1195.225

> m2 <- gamlss(accel ~ ps(times, df = 15, degree = 2), data = mcycle)

GAMLSS-RS iteration 1: Global Deviance = 1189.215
GAMLSS-RS iteration 2: Global Deviance = 1189.215

> m3 <- gamlss(accel ~ ps(times, df = 15, degree = 3), data = mcycle)

GAMLSS-RS iteration 1: Global Deviance = 1191.953
GAMLSS-RS iteration 2: Global Deviance = 1191.953

> lines(fitted(m1) ~ mcycle$times, col = "red")
> plot(accel ~ times, data = mcycle, main = "degree=2")
> lines(fitted(m2) ~ mcycle$times, col = "green")
> plot(accel ~ times, data = mcycle, main = "degree=3")
> lines(fitted(m3) ~ mcycle$times, col = "blue")
> par(op)

```

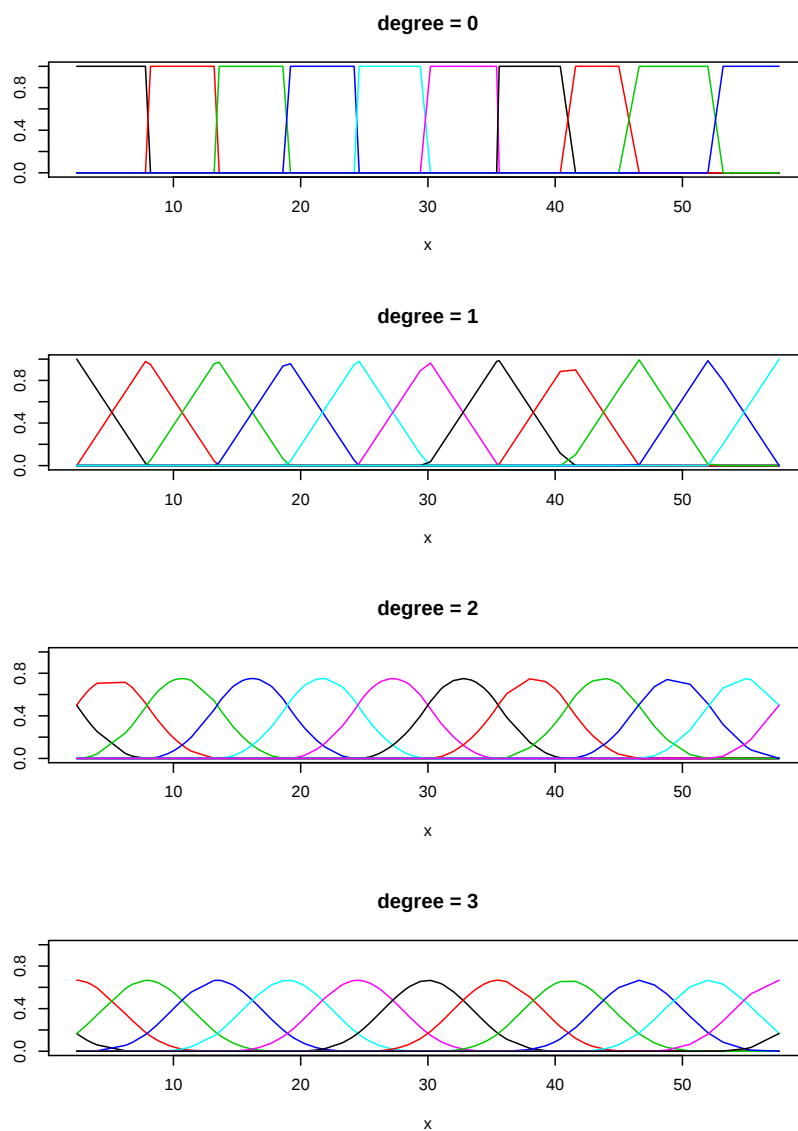


Figure 9.8: B-basis functions for the crash helmet data for times with 9 inner knots.

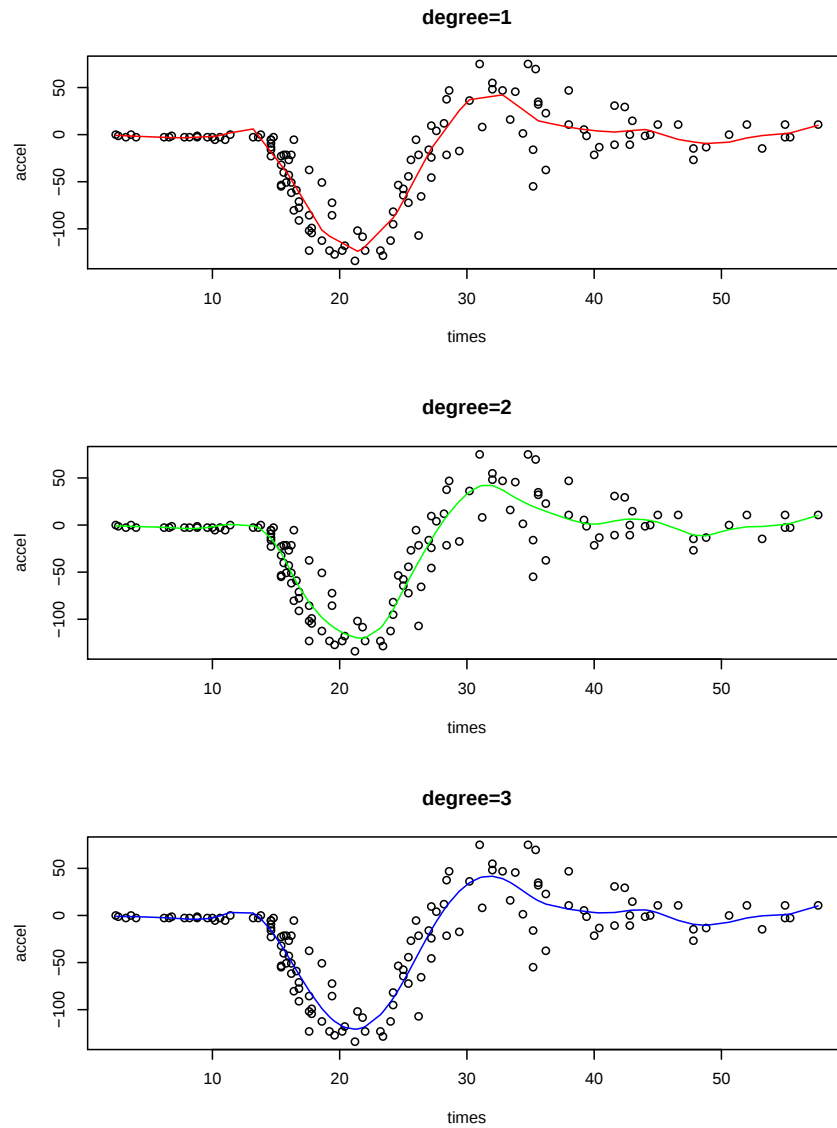


Figure 9.9: The helmet data fit with different degrees 1,2,3 corresponding to red, green, and blue respectively .

As an example on how the `order` effects the smoothness of the coefficients γ consider the following models where we fix the `degree=2` and we varied the `order` to 0,1,2,3.

```
> m20 <- gamlss(accel ~ ps(times, df = 15, degree = 2, order = 0),
+   data = mcycle)

GAMLSS-RS iteration 1: Global Deviance = 1189.380
GAMLSS-RS iteration 2: Global Deviance = 1189.379

> m21 <- gamlss(accel ~ ps(times, df = 15, degree = 2, order = 1),
+   data = mcycle)

GAMLSS-RS iteration 1: Global Deviance = 1188.364
GAMLSS-RS iteration 2: Global Deviance = 1188.364

> m22 <- gamlss(accel ~ ps(times, df = 15, degree = 2, order = 2),
+   data = mcycle)

GAMLSS-RS iteration 1: Global Deviance = 1188.705
GAMLSS-RS iteration 2: Global Deviance = 1188.705

> m23 <- gamlss(accel ~ ps(times, df = 15, degree = 2), order = 3,
+   data = mcycle)

GAMLSS-RS iteration 1: Global Deviance = 1189.215
GAMLSS-RS iteration 2: Global Deviance = 1189.215

> plot(m20$mu.coefSmo[[1]]$coef, type = "l", col = "red")
> lines(m21$mu.coefSmo[[1]]$coef, col = "green")
> lines(m22$mu.coefSmo[[1]]$coef, col = "blue")
> lines(m23$mu.coefSmo[[1]]$coef, col = "purple")
```

9.6 The `loess` function `lo()`

The function `lo()` allows the user to use a `loess` fit in a GAMLSS formula. A `loess` fit is a polynomial (univariate or surface) curve determined by one or more explanatory (continuous) variables, which are fitted locally (see Cleveland *et al.* (1993)). The implementation of the `lo()` function is very similar to the function with the same name in the S-plus implementation of `gam`. However `gamlss lo()` function uses the (R) `loess()` function as its engine and this creates some minor differences between the two `lo()` even when the same model is fitted (see below). `lo()` is the only function currently available in `gamlss` which allows smoothing simultaneously in more than one explanatory (continuous) variables, i.e. multidimensional smoothing.

The complementary `gamlss.lo` function uses the same C interface as `loess` in R. Resulting fits using `loess` and `lo` however are different (even for a normal distribution for the response variable) due to the different way that they are implemented. `gamlss.lo()` calls the C interface after the linear part of the model has been subtracted. This in general would lead to a slightly different smooth fit when the linear part and the smoothing part are added together at the end of the algorithm. The `gamlss.lo()` is closer in principle to the `gam.lo()` of S-plus rather than `lo()` of S-plus.

The `lo()` function has the following arguments:

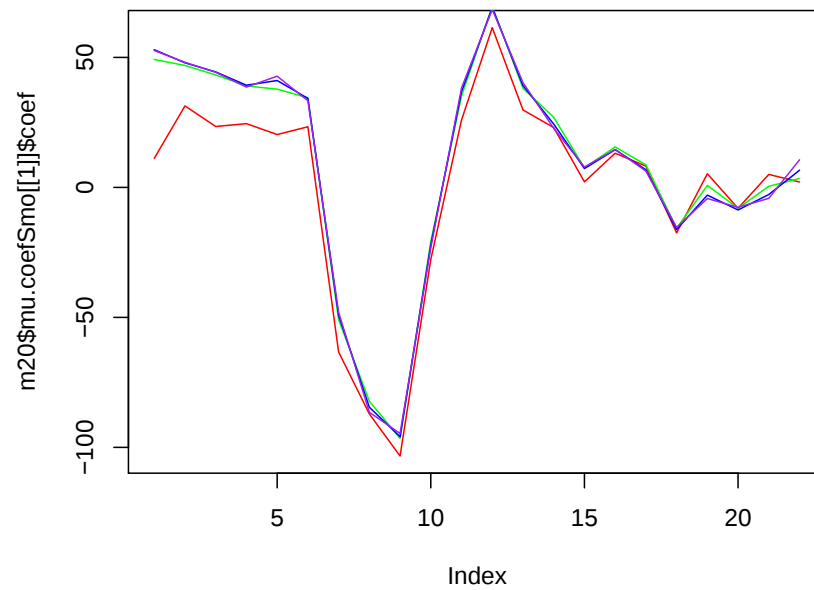


Figure 9.10: The helmet data fitted coefficients with different degrees 0,1,2,3 corresponding to red, green, blue and purple respectively.

- ... the unspecified ... can be a comma-separated list of numeric vectors i.e. (x1,x2,x3), numeric matrix, or expressions that evaluate to either of these. If it is a list of vectors, they must all have the same length.
- span** the parameter which controls the degree of smoothing (it is function of the number of observations in a neighborhood). This is the main smoothing parameter for a loess fit. The default is **span=0.5**
- df** the effective degrees of freedom can be specified instead of span, e.g. **df=5**. This is not a clear cut specification as in cubic splines. Given this parameter an estimated **span** is calculated and passed to the C interface. The default is **df=NULL**.
- degree** the degree of local polynomial to be fit; (the default is linear 1).

As an example of using the `lo()` function in GAMLSS we consider again the rent data. Here we fit four different models. The first two are additive models for floor space (F1) and for age (A) i.e. the year of construction. The third and fourth models fit surfaces for floor space (F1) and age (A). The first and third use a linear local fit while the second and fourth use a quadratic fit. All the models use the default **log** links for both μ and σ . We choose between models using a GAIC with penalty 2.5.

```
> data(rent)
> r11 <- gamlss(R ~ lo(F1) + lo(A), data = rent, family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27921.69
GAMLSS-RS iteration 2: Global Deviance = 27921.69
GAMLSS-RS iteration 3: Global Deviance = 27921.69

> r12 <- gamlss(R ~ lo(F1, degree = 2) + lo(A, degree = 2), data = rent,
+ family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27917.62
GAMLSS-RS iteration 2: Global Deviance = 27917.63
GAMLSS-RS iteration 3: Global Deviance = 27917.63

> r21 <- gamlss(R ~ lo(F1, A, degree = 1), data = rent, family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27908.77
GAMLSS-RS iteration 2: Global Deviance = 27908.98
GAMLSS-RS iteration 3: Global Deviance = 27908.98

> r22 <- gamlss(R ~ lo(F1, A, degree = 2), data = rent, family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27888.84
GAMLSS-RS iteration 2: Global Deviance = 27889.07
GAMLSS-RS iteration 3: Global Deviance = 27889.07

> GAIC(r11, r12, r21, r22, k = 2.5)

      df      AIC
r22 19.08488 27936.78
r21 11.33833 27937.33
r11 10.00128 27946.69
r12 14.12759 27952.95
```

According to the generalized Akaike information criterion with penalty 2.5, $GAIC(k = 2.5)$ the best model is `r22`, fitting a surface with a local polynomial of degree 2. The `span` in all the above models was the default one 0.5. Below we see whether a different `span` improves the fit.

```
> r223 <- gamlss(R ~ lo(Fl, A, degree = 2, span = 0.3), data = rent,
+   family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27874.48
GAMLSS-RS iteration 2: Global Deviance = 27874.59
GAMLSS-RS iteration 3: Global Deviance = 27874.59

> r224 <- gamlss(R ~ lo(Fl, A, degree = 2, span = 0.4), data = rent,
+   family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27882.76
GAMLSS-RS iteration 2: Global Deviance = 27882.98
GAMLSS-RS iteration 3: Global Deviance = 27882.98

> r226 <- gamlss(R ~ lo(Fl, A, degree = 2, span = 0.6), data = rent,
+   family = GA)

GAMLSS-RS iteration 1: Global Deviance = 27897.18
GAMLSS-RS iteration 2: Global Deviance = 27897.41
GAMLSS-RS iteration 3: Global Deviance = 27897.41

> AIC(r22, r223, r224, r226, k = 2.5)

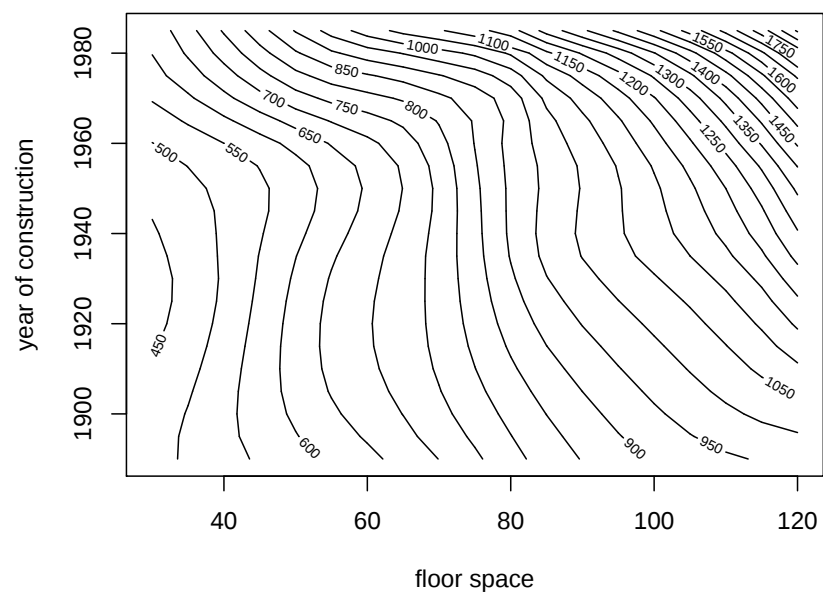
      df      AIC
r22  19.08488 27936.78
r226 16.29977 27938.16
r224 22.46753 27939.15
r223 28.89809 27946.83
```

It looks that the default `span` is fine. Now we plot the fitted surface as a contour plot Figure 9.11 and as wireframe plot in Figure 9.12.

chunk 4

```
> newrent <- data.frame(expand.grid(Fl = seq(30, 120, 5), A = seq(1890,
+   1985, 5)))
> newrent$pred <- predict(r22, newdata = newrent, type = "response")
> Fln <- seq(30, 120, 5)
> An <- seq(1890, 1985, 5)
> contour(Fln, An, matrix(newrent$pred, nrow = length(Fln)), nlevels = 30,
+   ylab = "year of construction", xlab = "floor space")
> library(lattice)
> print(wireframe(pred ~ Fl * A, newrent, aspect = c(1, 0.5), drape = TRUE,
+   colorkey = list(space = "right", height = 0.6)))
```

Note that while the `lo gamlss` implementation is similar to the S-plus implementation of `lo` in `gam`, the results are not always exactly the same even for similar types of models. In fact we have found that for normal error fits the results are identical if more than one variable is used, for example `lo(x1,x2)` but are not if only one variable is used. In this case the smooth fits different slightly and also have slightly different effective degrees of freedom.

Figure 9.11: Rent data: contour plot for the fitted `loess` surface model

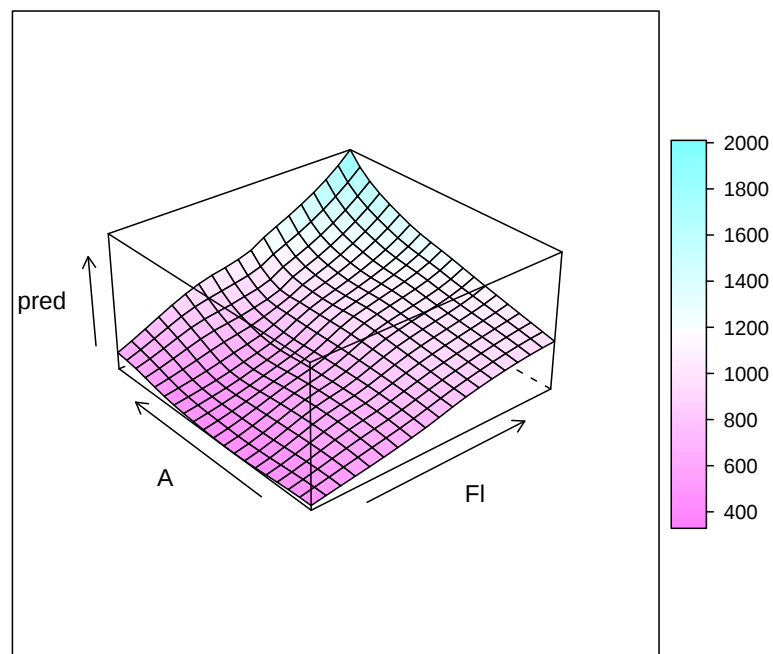


Figure 9.12: Rent data: wireframe plot for the fitted `loess` surface model

Chapter 10

Nonlinear functions in GAMLSS

10.1 Introduction

There are two ways for fitting a nonlinear model within the GAMLSS framework

- by using the function `nlgamlss()`
- by using the additive function `nl()` within the main fitting `gamlss()` function

The function `nlgamlss()` can be applied to any **parametric** models and can be seen as another method (algorithm) for fitting a GAMLSS model. (In fact in future releases it may be incorporated within the `gamlss()` function as an extra algorithmic method for fitting a GAMLSS model in conjunction with the existing methods **RS**, **CG** and **mixed**). Note the emphasis in the above sentence on parametric models. `nlgamlss()` **can not** be used in conjunction with additive terms available within the standard `gamlss` function, i.e. `cs()` or `lo()`. The function `nl()` on the contrary is an extra additive function in `gamlss()`. and therefore it can be used together with other additive term functions. Section 10.2 describes the use of the function `nlgamlss()` while Section 10.3 describe the `nl()` function.

10.2 The function `nlgamlss`

The function `nlgamlss()` is an additional method for fitting GAMLSS models. It is suitable for fitting linear or non linear **parametric** models using the distributions available in the GAMLSS package. The function `nlgamlss()` is based on the function `stablereg()` of the R package **stable** created by Philippe Lambert and Jim Lindsey which can be found in Jim Lindsey's web page <http://popgen.unimaas.nl/~jlindsey/> (see also Lambert, P. and Lindsey, J.K. (1999)). The function `nlgamlss()` is interpreting the formulae for the distributional parameters `mu`, `sigma`, `nu` and `tau` and creates a log-likelihood function to be maximized. The actual maximization is achieved by the nonlinear maximization function `nlm()` in R. The output of the function is an S3 class `nlgamlss` and `gamlss` object.

The interpretation of the formulae is achieved using the function `finterp()`, taken from Jim Lindsey's R package **rmutil**. `help(finterp)`, which will tell you more about how the function is operating. The unpublished article "Data objects and model formulae for the future: some proposals" by Lindsey which can be found at <http://popgen.unimaas.nl/~jlindsey/manuscripts.html> would tell you more about the philosophy behind it.

The function `nlgamlss()` is very general and can be used to fit any parametric models but requires starting values for all the beta parameters of the model, (existing in the individual formulae of the distributional parameters), independently of whether they are linear or non linear. This is the main disadvantage of the method since it is not always easy to find these starting values.

For parametric models the function `nlgamlss` can also be used to give more accurate standard errors, for the parameters of a GAMLSS model than the one given by the `gamlss` object. For a `gamlss` object the standard errors given using `summary()` do not reflect the existing correlation between the distributional parameters `mu`, `sigma`, `nu` and `tau` so they are less reliable.

10.2.1 The data

Data summary:

R data file: `1a` in package `gamlss.nl` of dimensions 251×4 originally given by Lange *et al.* (1989). Only two of the variables are used here.

variables

`bflow` : the blood flow measured invasively using radioactively labelled micro-spheres

`PET60` : is the blood flow measured non-invasively by positron emission tomography using a scan up to 60 seconds

`PETother` : unknown

`PET510` : is the blood flow measured non-invasively by positron emission tomography using a scan up to 510 seconds

purpose: to demonstrate the fitting of non-linear model in `gamlss`

conclusion: A LOGPE distribution fits the data best

As an example of a nonlinear parametric model within GAMLSS we shall use the blood flow data which were previously analyzed by Lange *et al.* (1989), Jones and Faddy (2003) and Rigby and Stasinopoulos (2006). The data are distributed within the `gamlss.nl` package under the name `1a`. There are 251 observations on 4 variables but only two of the variables are used here, i) `bflow`: the blood flow measured invasively using radioactively labelled micro-spheres, and ii) `PET60`: the response variable, which is the blood flow measured non-invasively by positron emission tomography using a scan up to 60 seconds. The distribution of the response variable has been previously modelled by a normal (NO) and a *t* family distribution (TF), Lange *et al.* (1989), by a skew *t* distribution (ST), Jones and Faddy (2003) and by Rigby and Stasinopoulos (2006) using the following distributions: the log normal, (LOGNO), Johnson's Su (JSU) Johnson (1949), skew exponential power (SEP), Azzalini (1986) and DiCiccio and Monti (2004), Box-Cox power exponential (BCPE), Rigby and Stasinopoulos (2004) and Box-Cox *t* (BCT), Rigby and Stasinopoulos (2006). Rigby and Stasinopoulos (2006) found that the best distribution for the data is the log power exponential distribution (LOGPE) which is a special case of the BCPE distribution when $\nu = 0$ (indicating positive skewness since $\nu < 1$).

Lange *et al.* (1989), and Jones and Faddy (2003) used a nonlinear model for the location parameter μ given by $\mu = h_1(x) = x[1 + \beta_{11} \exp(-\beta_{12}/x)]$ and a constant for the scale parameter σ (and any shape parameters ν and τ). Here we use the reparameterization $\mu = h_1(x) = x[1 + (1 - e^{b_{11}}) \exp(-b_{12}/x)]$ to improve model fitting. Note that the nonlinear

function $\mu = h_1(x)$ passes through the origin, i.e. $\mu = 0$ for $x=0$.

10.2.2 Fitting the model

Let us fit the BCPE distribution with the nonlinear model $h_1(x)$ for μ and constant σ , ν and τ and then plot the data together with the fitted values for μ (see figure 10.1).

```
> library(gamlss.nl)
> data(la)
> mod1 <- nlgamlss(y = PET60, mu.fo = ~bflow * (1 - (1 - exp(p1)) *
+   exp(-p2/bflow)), sigma.formula = ~1, mu.start = c(-0.9, 90),
+   sigma.start = log(0.1), nu.start = 0, tau.start = log(2.5),
+   family = BCPE, data = la)
> mod1

Family: c("BCPE", "Box-Cox Power Exponential")
Fitting method: "JL()"

Call:
nlgamlss(y = PET60, mu.fo = ~bflow * (1 - (1 - exp(p1)) * exp(-p2/bflow)),
  sigma.formula = ~1, mu.start = c(-0.9, 90), sigma.start = log(0.1),
  nu.start = 0, tau.start = log(2.5), family = BCPE, data = la)

Mu Coefficients:
      p1      p2
-0.8058  76.2682
Sigma Coefficients:
(Intercept)
      -1.187
Nu Coefficients:
(Intercept)
      -0.2084
Tau Coefficients:
(Intercept)
      0.9488

Degrees of Freedom for the fit: 5 Residual Deg. of Freedom    246
Global Deviance:      2292.81
      AIC:      2302.81
      SBC:      2320.44
```

Note that the formula for μ is defined the way that any nonlinear formula is defined in R. The function `finterp()` which interprets the formula allows also the use of a formula in the Wilkinson and Rogers notation of GLM's or a function. Functions are useful when some interim calculations have to be made before evaluating the formula. We do not need this facility here but as an example consider the following commands using a function instead of a formula to define the model for μ .

```
> funnl <- function(p) bflow * (1 - (1 - exp(p[1])) * exp(-p[2]/bflow))
> mod11 <- nlgamlss(y = PET60, mu.fo = funnl, sigma.formula = ~1,
```

```
> plot(PET60 ~ bflow, data = la)  
> lines(la$bflow, fitted(mod1), col = "red")
```

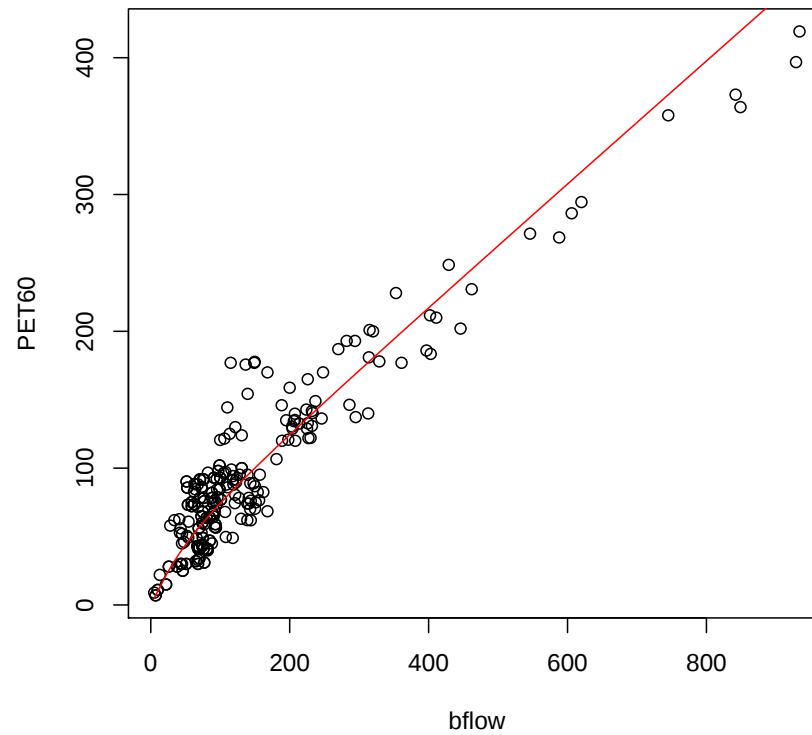


Figure 10.1: Plotting the data together with the fitted values for μ for the fitted model `mod1`.

```
+      mu.start = c(-0.9, 90), sigma.start = log(0.1), nu.start = 0,
+      tau.start = log(2.5), family = BCPE, data = la)
> deviance(mod11)

[1] 2292.809
```

Note that the starting values are for the beta coefficients of the predictors of the distributional parameters. For example to use a starting value of 0.2 for σ , the starting value for $\log \sigma$ the predictor of sigma [because BCPE has a default log link function for σ], must be $\log(0.01)$. Similarly to use a starting value of 2.5 for τ , the starting value for $\log \tau$ must be $\log(2.5)$ [because BCPE has a default log link for τ]. No adjustment are needed for μ or for ν because for the BCPE they both have identity link functions.

For starting values for the parameters we use the following:

mu: $c(-.9, 90)$ is used, (a bit of a cheat since these two values were close to values reported by other researches). In general someone has to look hard for these values and also it is advisable to use different starting values to make sure that the algorithm is converging to the same parameter values. Note it is also possible in general that there may be more than one local maximum, so use different starting values to investigate this and find the global maximum.

sigma: $\log(0.1)$ is used since 0.1 seems a sensible value for a coefficient of variation and **sigma** has a default log link.

nu: 0 is used indicating positive skewness, (the default link here is identity)

tau: $\log(2.5)$ is used with values greater than 2 indicating a platy-kurtotic shape in the distribution, and since **tau** has a default log link.

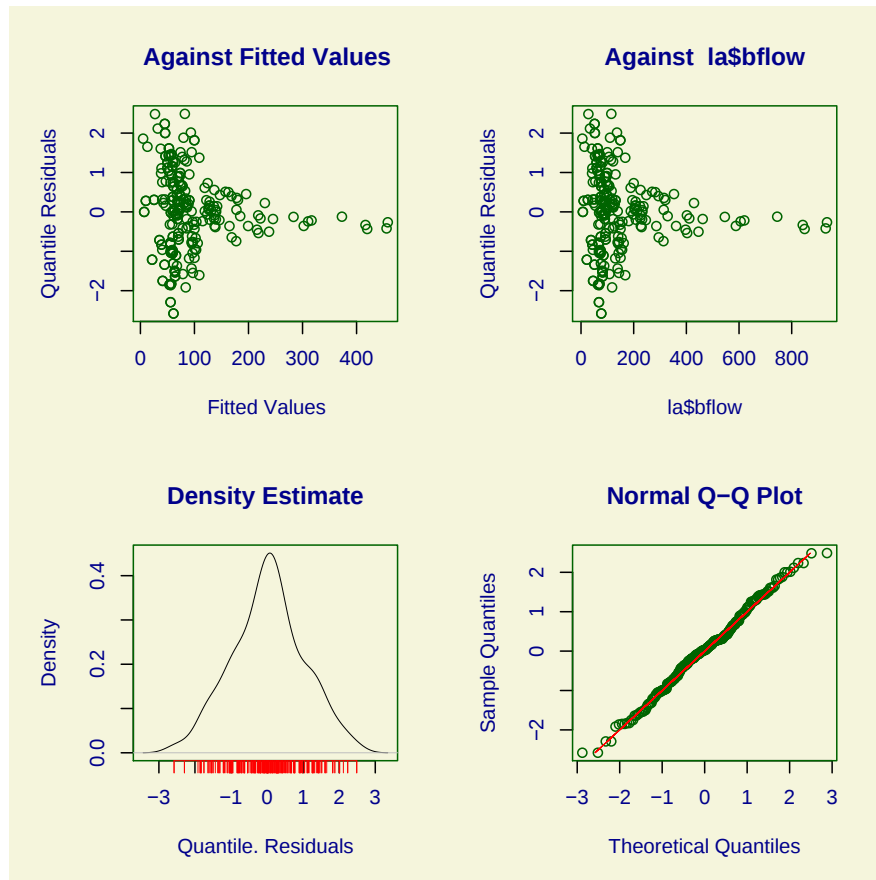
Note that the following generic functions will work with a `nlgamlss` object: `AIC()`, `coef()`, `deviance()`, `extractAIC()`, `fitted()`, `formula()`, `plot()`, `print()`, `residuals()`, `summary()`, `vcov()` and with some care `update()`.

The fitted line of the median parameter **mu** shown in figure 10.1 looks peculiar (in that it misses completely the larger observations in **bflow**). This is mainly due to the fact that the scale parameter **sigma** is not modelled properly. This can be seen in the residual plot of figure 10.2 where the plot of the residuals against the explanatory variable **bflow** in the top right corner (or against the fitted values, in the top left corner) is shown to have a funnel type effect.

```
> plot(mod1, xvar = la$bflow)

*****
      Summary of the Quantile Residuals
              mean   =  0.02553148
              variance =  0.998545
      coef. of skewness = -0.02327884
      coef. of kurtosis =  2.797976
Filliben correlation coefficient =  0.9974858
*****
```

We proceed by fitting a parametric model for $\log \sigma$, an orthogonal polynomial of degree two. Since it is difficult to guess starting values for the beta coefficients of $\log \sigma$ we shall use first a similar GAMLSS model and then use the fitted beta coefficients of this model as starting values. In model `mod2` we fit a cubic spline for μ .

Figure 10.2: The plot of the residuals from model `mod1` .

```

> mod2 <- gamlss(PET60 ~ cs(bflow), sigma.formula = ~poly(bflow,
+ 2), family = BCPE, data = la)

GAMLSS-RS iteration 1: Global Deviance = 2244.250
GAMLSS-RS iteration 2: Global Deviance = 2216.779
GAMLSS-RS iteration 3: Global Deviance = 2211.689
GAMLSS-RS iteration 4: Global Deviance = 2210.43
GAMLSS-RS iteration 5: Global Deviance = 2210.269
GAMLSS-RS iteration 6: Global Deviance = 2210.325
GAMLSS-RS iteration 7: Global Deviance = 2210.396
GAMLSS-RS iteration 8: Global Deviance = 2210.457
GAMLSS-RS iteration 9: Global Deviance = 2210.496
GAMLSS-RS iteration 10: Global Deviance = 2210.518
GAMLSS-RS iteration 11: Global Deviance = 2210.53
GAMLSS-RS iteration 12: Global Deviance = 2210.537
GAMLSS-RS iteration 13: Global Deviance = 2210.541
GAMLSS-RS iteration 14: Global Deviance = 2210.543
GAMLSS-RS iteration 15: Global Deviance = 2210.544
GAMLSS-RS iteration 16: Global Deviance = 2210.545

> coef(mod2, "sigma")

(Intercept) poly(bflow, 2)1 poly(bflow, 2)2
-1.3294835      -8.4106266      0.7984741

> mod3 <- nlglmss(y = PET60, mu.fo = ~bflow * (1 - (1 - exp(p1)) *
+ exp(-p2/bflow)), sigma.formula = ~poly(bflow, 2), mu.start = c(-0.9,
+ 90), sigma.start = coef(mod2, "sigma"), nu.start = 0, tau.start = log(2.5),
+ family = BCPE, data = la)
> summary(mod3)

*****
Family:  c("BCPE", "Box-Cox Power Exponential")

Call:
nlglmss(y = PET60, mu.fo = ~bflow * (1 - (1 - exp(p1)) * exp(-p2/bflow)),
  sigma.formula = ~poly(bflow, 2), mu.start = c(-0.9, 90),
  sigma.start = coef(mod2, "sigma"), nu.start = 0, tau.start = log(2.5),
  family = BCPE, data = la)

Fitting method: "JL()"

-----
Mu link function:  identity
Mu Coefficients:
      Estimate Std. Error t-value  p-value
p1    -0.976    0.02287   -42.67 0.000e+00
p2    98.372    9.09170   10.82 2.767e-27
-----

```

Sigma link function: log

Migma Coefficients:

	Estimate	Std. Error	t-value	p-value
(Intercept)	-1.3267	0.03616	-36.691	1.012e-294
poly(bflow, 2)1	-8.2705	0.56384	-14.668	1.031e-48
poly(bflow, 2)2	0.8434	0.60154	1.402	1.609e-01

Nu link function: identity

Nu Coefficients:

	Estimate	Std. Error	t-value	p-value
(Intercept)	-0.05981	0.1625	-0.368	0.7129

Tau link function: log

Tau Coefficients:

	Estimate	Std. Error	t-value	p-value
(Intercept)	1.239	0.1864	6.648	2.973e-11

No. of observations in the fit: 251

Degrees of Freedom for the fit: 7

Residual Deg. of Freedom: 244

at cycle: 66

Global Deviance: 2212.855

AIC: 2226.855

SBC: 2251.533

Note the differences between the calls for `gamlss()` and `nlgamlss()`. The function `nlgamlss()` has an explicit argument, `y`, for declaring the response variable while in `gamlss()` the response variable is declared implicitly within the `mu` formula argument. Also the starting value arguments in `nlgamlss()` are for the beta coefficients (of the predictors of the distributional parameters `mu`, `sigma`, `nu` and `tau`) while for `gamlss()` the starting values are for the predictors of the distributional parameters μ , σ , ν and τ themselves.

Figure 10.3 shows the residuals for model `mod3` while figure 10.4 shows the fitted values for μ and σ . The funnel effect has now almost disappeared from the residual plots while the fitted values for `mu` are passing through the data points. Model `mod3` is superior to `mod1` in terms of GAIC with a variety of different penalties, even for the more demanding SBC criterion with $k = \log(n)$. Below are the commands to plot figures 10.3 and 10.4:

```
> plot(mod3, xvar = la$bflow)
```

Summary of the Quantile Residuals

mean	=	-0.005030426
variance	=	0.9992643
coef. of skewness	=	0.06283656

```

      coef. of kurtosis = 2.979742
Filliben correlation coefficient = 0.9973569
*****

> op <- par(mfrow = c(2, 1))
> plot(PET60 ~ bflow, data = la, main = "(a)")
> lines(la$bflow, fitted(mod3), col = "red")
> plot(la$bflow, fitted(mod3, "sigma"), type = "l", main = "(b)")
> par(op)
> AIC(mod1, mod2, mod3, k = log(length(la$bflow)))

      df      AIC
mod3 7.000000 2251.533
mod2 9.999481 2265.796
mod1 5.000000 2320.436

```

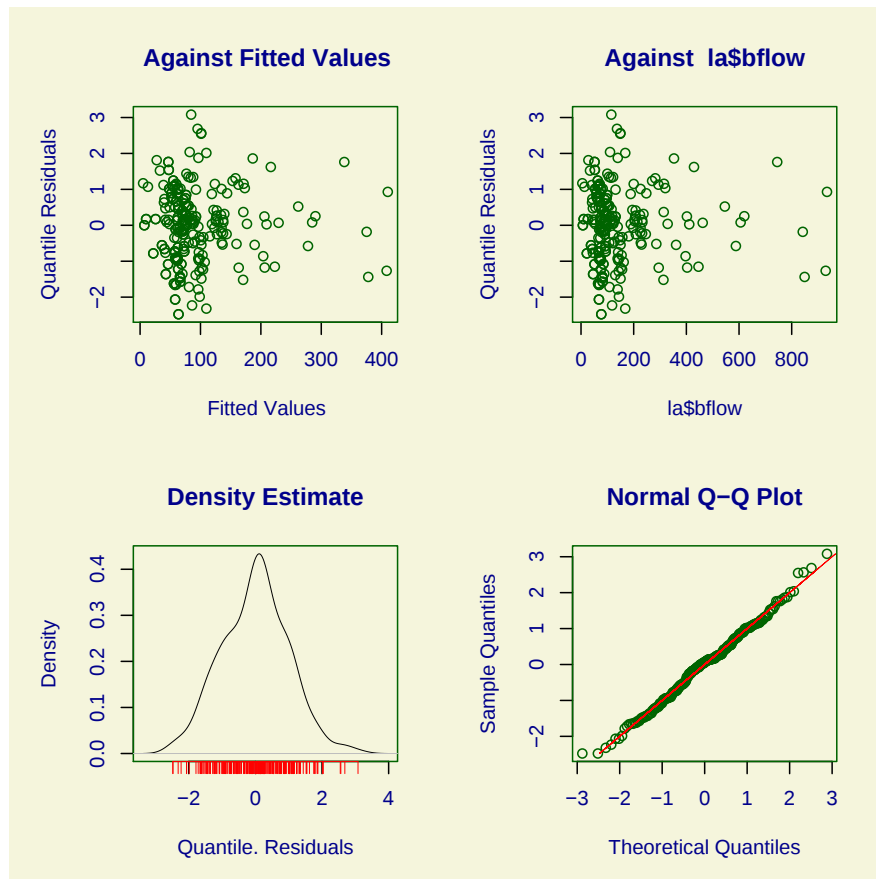


Figure 10.3: The plot of the residuals from model `mod3` using the command `plot(mod3, xvar=la$bflow)`.

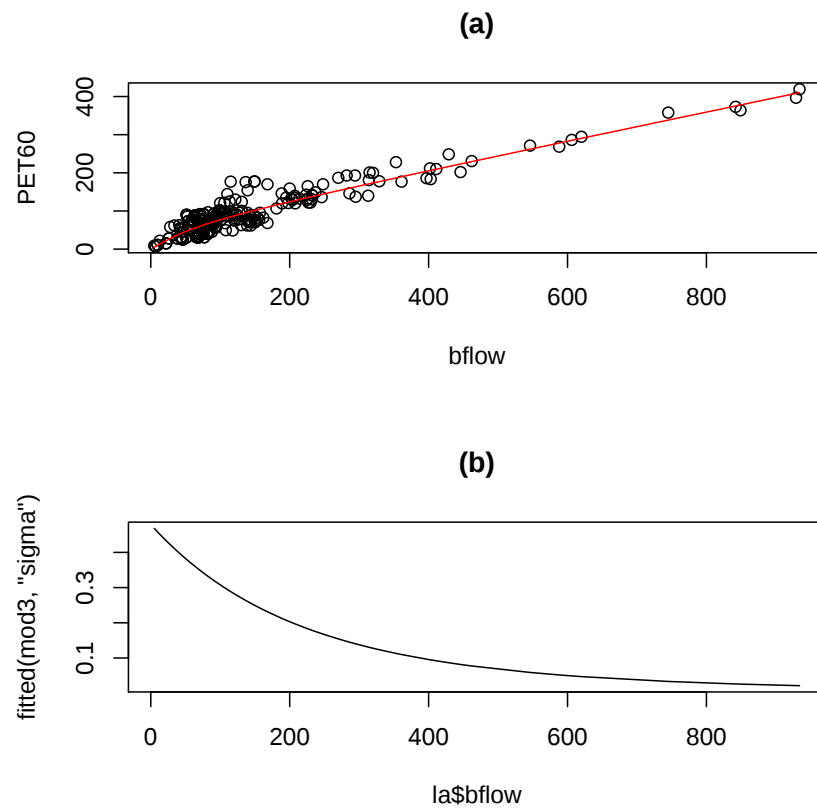


Figure 10.4: Model `mod3` fitted values: (a) the data together with the fitted values for μ , (b) the fitted values for σ

More information and options for the function `nlgamlss()` can be found in the help file for the function. The `nlgamlss` object created from the function is similar to a `gamlss` object. The most notable difference is that a `nlgamlss` fitted object contains the following extra components

- i) `coefficients`
- ii) `se`
- iii) `cov`
- v) `corr`

for the fitted coefficients, their standard errors, the variance covariance matrix and the correlation matrix of the fit respectively. For example to display the correlation matrix type:

```
> mod3$coefficients

[1] -0.97595593 98.37241996 -1.32672494 -8.27053468 0.84343712 -0.05981003
[7] 1.23900303

> mod3$se

[1] 0.02287200 9.09170172 0.03615929 0.56384181 0.60154064 0.16251363 0.18637472

> mod3$corr

      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 1.00000000 -0.87806245 -0.038855688 0.21879511 0.11221156 -0.504319037
[2,] -0.87806245 1.00000000 0.047961738 -0.20370521 0.01478113 0.606850529
[3,] -0.03885569 0.04796174 1.000000000 -0.01605296 -0.06674437 -0.008456865
[4,] 0.21879511 -0.20370521 -0.016052964 1.000000000 -0.15832731 -0.061589927
[5,] 0.11221156 0.01478113 -0.066744373 -0.15832731 1.000000000 -0.021282547
[6,] -0.50431904 0.60685053 -0.008456865 -0.06158993 -0.02128255 1.000000000
[7,] -0.04558620 0.05206620 0.339071672 -0.03146852 -0.20377046 -0.086538197
      [,7]
[1,] -0.04558620
[2,] 0.05206620
[3,] 0.33907167
[4,] -0.03146852
[5,] -0.20377046
[6,] -0.08653820
[7,] 1.00000000
```

The above matrix is the correlation matrix for the 7 beta coefficient parameters in model `mod3`, i.e. 2 for the nonlinear function for μ , 3 for the orthogonal quadratic polynomial for $\log \sigma$, 1 for ν and 1 for $\log \tau$.

10.3 The function `nl()`

The function `nl()` is an additive function in GAMLSS. It can be used to fit nonlinear models for any of the distributional parameters in `gamlss()`. The function is `nl()` takes as argument an

`nlobj` object created by the function `nl.obj()`. The function `nl.obj()` takes as argument the model formula, `formula`, the starting values for the nonlinear parameters, `start`, and the data set where the formula has to be interpreted, `data`. For example for the `la` data the following command is needed.

```
> nlo <- nl.obj(formula = ~bflow * (1 - (1 - exp(p1)) * exp(-p2/bflow)),
+               start = c(-0.9, 90), data = la)
> nlo

function (.p)
eval(attr(.fna, "model"))
<environment: 0x01f19d5c>
attr("formula")
~bflow * (1 - (1 - exp(p1)) * exp(-p2/bflow))
attr("model")
expression(la$bflow * (1 - (1 - exp(.p[1])) * exp(-.p[2]/la$bflow)))
attr("parameters")
[1] "p1" "p2"
attr("covariates")
[1] "bflow"
attr("range")
[1] 1 2
attr("class")
[1] "nlobj"
attr("start")
[1] -0.9 90.0

> class(nlo)

[1] "nlobj"
```

Given that the `nlobj` object is defined we can fit the same model as model `mod3` in Section 10.2.2 by using:

```
> mod4 <- gamlss(PET60 ~ nl(nlo) - 1, sigma.fo = ~poly(bflow, 2),
+               data = la, family = BCPE)

GAMLSS-RS iteration 1: Global Deviance = 2254.306
GAMLSS-RS iteration 2: Global Deviance = 2216.668
GAMLSS-RS iteration 3: Global Deviance = 2213.358
GAMLSS-RS iteration 4: Global Deviance = 2212.933
GAMLSS-RS iteration 5: Global Deviance = 2212.867
GAMLSS-RS iteration 6: Global Deviance = 2212.857
GAMLSS-RS iteration 7: Global Deviance = 2212.856
GAMLSS-RS iteration 8: Global Deviance = 2212.855

> mod4

Family: c("BCPE", "Box-Cox Power Exponential")
Fitting method: RS()
```

```
Call:  gamlss(formula = PET60 ~ nl(nlo) - 1, sigma.formula = ~poly(bflow,
      2), family = BCPE, data = la)
```

```
Mu Coefficients:
```

```
nl(nlo)
```

```
NA
```

```
Sigma Coefficients:
```

```
(Intercept)  poly(bflow, 2)1  poly(bflow, 2)2
```

```
      -1.3267      -8.2706      0.8432
```

```
Nu Coefficients:
```

```
(Intercept)
```

```
-0.05909
```

```
Tau Coefficients:
```

```
(Intercept)
```

```
1.239
```

```
Degrees of Freedom for the fit: 7 Residual Deg. of Freedom  244
Global Deviance:      2212.86
      AIC:      2226.86
      SBC:      2251.53
```

```
> mod4$mu.coefSmo
```

```
[[1]]
```

```
[[1]]$coef
```

```
[1] -0.9760652 98.4311658
```

```
[[1]]$se
```

```
[1] 0.01447233 4.92335746
```

```
[[1]]$varcoeff
```

```
      [,1]      [,2]
```

```
[1,] 0.0002094484 -0.05884772
```

```
[2,] -0.0588477198 24.23944871
```

Note the `-1` in the `mu` formula. In this particular data we want the line to pass through the origin so we have to exclude the constant from the model. [This is not the case in general. In general other model terms, linear and/or additive, can be added to the nonlinear term `nl(nlo)`]. Omitting the constant creates a problem with the function `summary.gamlss` which expects at least one coefficient to appear for each distributional parameter. Here we use instead the function `print` to show the model but this do not print the fitted coefficients for the `mu` model. Since `nl()` is an additive functions the coefficients are stored within the list of `mod4$mu.coefSmo`. Comparing the result of model `mod4` fitted using `nl()` and `mod3` fitted using `nlgamlss` we can see that the actual fitted coefficients are almost identical but their standard errors are considerable different. This is a reflection of the conditional nature of estimation within the `nl()` function where everything is estimated conditionally on the rest of the parameters, so the standard errors are generally lower than their unconditional values and should not be used on their own for inference. To test whether a particular parameter is different from zero, it is better to

compare the global deviance (GD) of the model with the parameter set to zero with the GD of the model with the parameter included, using a χ^2 test. Alternatively the GAIC can be used to compare the models with a specified penalty k .

The nice thing about the function `nl()` is that it can be used with other additive terms. Here we refit the BCPE model using a smoothing cubic spline to fit the scale parameter `sigma` as a function of `bflow` (instead of the quadratic polynomial of model `mod4`). Figure 10.5 shows the fitted values for `sigma` for both models.

```
> mod5 <- gamlss(PET60 ~ nl(nlo) - 1, sigma.fo = ~cs(bflow), data = la,
+   family = BCPE)
```

```
GAMLSS-RS iteration 1: Global Deviance = 2247.776
GAMLSS-RS iteration 2: Global Deviance = 2212.48
GAMLSS-RS iteration 3: Global Deviance = 2208.100
GAMLSS-RS iteration 4: Global Deviance = 2207.643
GAMLSS-RS iteration 5: Global Deviance = 2207.627
GAMLSS-RS iteration 6: Global Deviance = 2207.624
GAMLSS-RS iteration 7: Global Deviance = 2207.624
```

```
> term.plot(mod4, what = "sigma", se = TRUE, partial = TRUE)
> term.plot(mod5, what = "sigma", se = TRUE, partial = TRUE)
```

Different distributions (with the same number of distributional parameters) can be fitted easily using the `update()` function. Here we use the BCT distribution.

```
> update(mod5, family = BCT)
```

```
GAMLSS-RS iteration 1: Global Deviance = 2225.553
GAMLSS-RS iteration 2: Global Deviance = 2218.818
GAMLSS-RS iteration 3: Global Deviance = 2218.801
GAMLSS-RS iteration 4: Global Deviance = 2218.801
```

```
Family: c("BCT", "Box-Cox t")
Fitting method: RS()
```

```
Call: gamlss(formula = PET60 ~ nl(nlo) - 1, sigma.formula = ~cs(bflow),
  family = BCT, data = la)
```

```
Mu Coefficients:
```

```
nl(nlo)
      NA
```

```
Sigma Coefficients:
```

```
(Intercept)    cs(bflow)
   -0.823420   -0.003622
```

```
Nu Coefficients:
```

```
(Intercept)
    0.002038
```

```
Tau Coefficients:
```

```
(Intercept)
    11.85
```

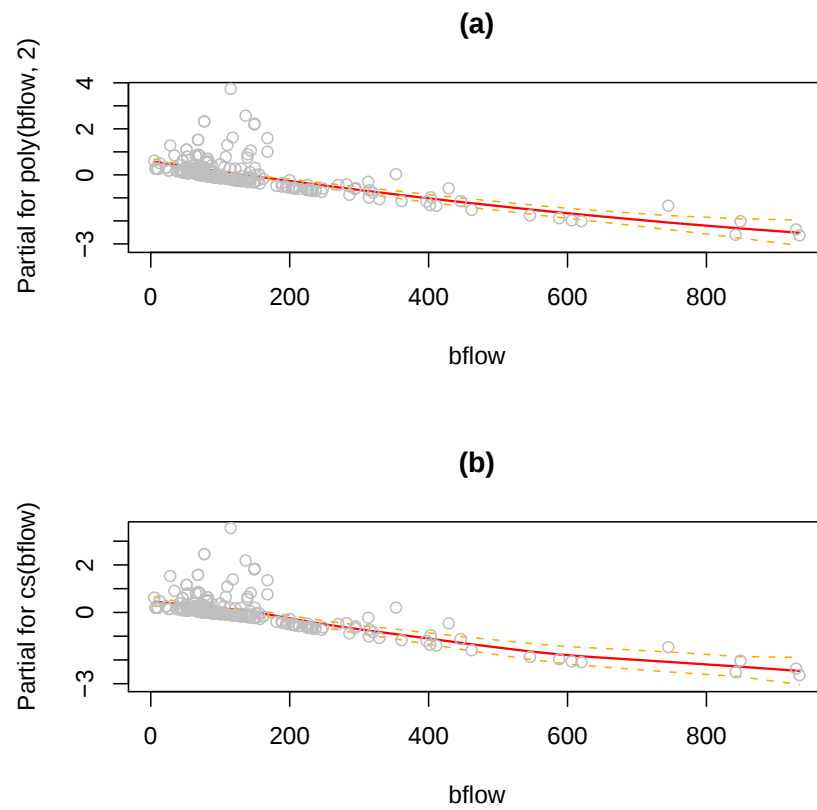


Figure 10.5: Fitted values, Standard errors and partial residuals for the `sigma` parameter model: (a) model `mod4` fitted quadratic polynomials , (b) model `mod5` fitted smoothing cubic spline

```

Degrees of Freedom for the fit: 9.000726 Residual Deg. of Freedom 241.9993
Global Deviance:          2218.8
                        AIC:          2236.8
                        SBC:          2268.53

```

10.4 Exercises

- Q1 The table 11.3 below taken from Rigby and Stasinopoulos (2006), shows the global deviance for different models fitted to the Lange et al (1989) data described in section 7.2.1 where the function for the parameter mu is defined by the formula

```
~ bflow*(1-(1-exp(p1))*exp(-p2/bflow))
```

for identity links function [starting values c(.6, 90)] and

```
~log(bflow)+log(1-(1-exp(p1))*exp(-p2/bflow))
```

for log links [starting values c(-.9, 90)] and the model for log(sigma) is given by $h_2(x)$ is a quadratic function in x.

- i) Reproduce the table using the function `nlgamlss()`
- ii) Reproduce the table using the function `nl()`
- iii) Use `GAIC(#=2,3,4)` to find the best model

Lange, K. L., Little, R. J. A. and Taylor, J. M. G. (1989). Robust statistical modelling using the t distribution. *J. Am. Statist. Ass.*, 84: 881-896. Rigby, R. A. and Stasinopoulos, D. M. (2006). Using the Box-Cox t distribution in GAMLSS to model skewness and kurtosis, *Statistical Modelling*.

Table 10.1: Models for the blood flow data with their fitted global deviance GDEV and generalized information criterion with penalty $\# = 3$ i.e. GAIC(3)

Model	Distribution	μ	$\log \sigma$	ν	τ	GDEV-2200	GAIC(3)-2200
1	LOGNO	$h_1(x)$	1	-	-	93.9	102.9
2	NO	$h_1(x)$	1	-	-	78.7	87.7
3	TF	$h_1(x)$	1	1	-	73.5	85.5
4	ST	$h_1(x)$	1	1	1	66.7	81.7
5	NO	$h_1(x)$	$h_2(x)$	-	-	69.9	85.0
6	TF	$h_1(x)$	$h_2(x)$	1	-	67.4	85.4
7	JSU	$h_1(x)$	$h_2(x)$	1	1	46.1	67.1
8	ST	$h_1(x)$	$h_2(x)$	1	1	45.7	66.7
9	SEP	$h_1(x)$	$h_2(x)$	1	1	33.4	54.4
10	BCT	$h_1(x)$	$h_2(x)$	1	1	21.5	42.6
11	LOGNO	$h_1(x)$	$h_2(x)$	-	-	21.5	36.6
12	LOGPE	$h_1(x)$	$h_2(x)$	-	1	13.0	31.1
13	BCPE	$h_1(x)$	$h_2(x)$	1	1	12.8	33.9

Chapter 11

Centile estimation

11.1 Head circumference data

The Fourth Dutch Growth Study, (Fredriks *et al.* 2000a; Fredriks, van Buuren, Wit and Verloove-Vanhorick 2000b) is a cross-sectional study that measures growth and development of the Dutch population between the ages 0 and 22 years. The study measured, among other variables, height, weight, head circumference and age for 7482 males and 7018 females.

Here the head circumference (y) of the males is analyzed with explanatory variable $x = age^\xi$, the transformed age. There are 7040 observations, as there were 442 missing values for head circumference. The data are plotted in figure 11.1. The data were previously analyzed by van Buuren and Fredriks (2001) who found strong evidence of kurtosis which they were unable to model. The data were subsequently analysed by Rigby and Stasinopoulos (2006) using a BCT distribution to model the kurtosis.

```
> library(gamlss)
> data(db)
> plot(head ~ age, data = db)
```

Given $X = x$, Y is modelled here by a Box-Cox t distribution, $BCT(\mu, \sigma, \nu, \tau)$, defined in the Appendix, where the parameters μ , σ , ν , and τ are modelled, using a special case of the GAMLSS model (2.2), as smooth nonparametric functions of x , i.e. $Y \sim BCT(\mu, \sigma, \nu, \tau)$ where

$$\begin{aligned} g_1(\mu) &= h_1(x) \\ g_2(\sigma) &= h_2(x) \\ g_3(\nu) &= h_3(x) \\ g_4(\tau) &= h_4(x) \end{aligned} \tag{11.1}$$

and, for $k = 1, 2, 3, 4$, $g_k(\cdot)$ are known monotonic link functions, and $h_k(x)$ are smooth nonparametric functions of x .

The model selection procedure comprised of choosing link functions $g_k(\cdot)$, for $k = 1, 2, 3, 4$, ξ in the transformation for age, $x = age^\xi$, and the total (effective) degrees of freedom for the smooth nonparametric cubic spline functions $h_k(x)$ for $k = 1, 2, 3, 4$, denoted df_μ , df_σ , df_ν and df_τ respectively.

Identity link functions were chosen for μ and ν , while log link functions were chosen for σ and τ (to ensure $\sigma > 0$ and $\tau > 0$).

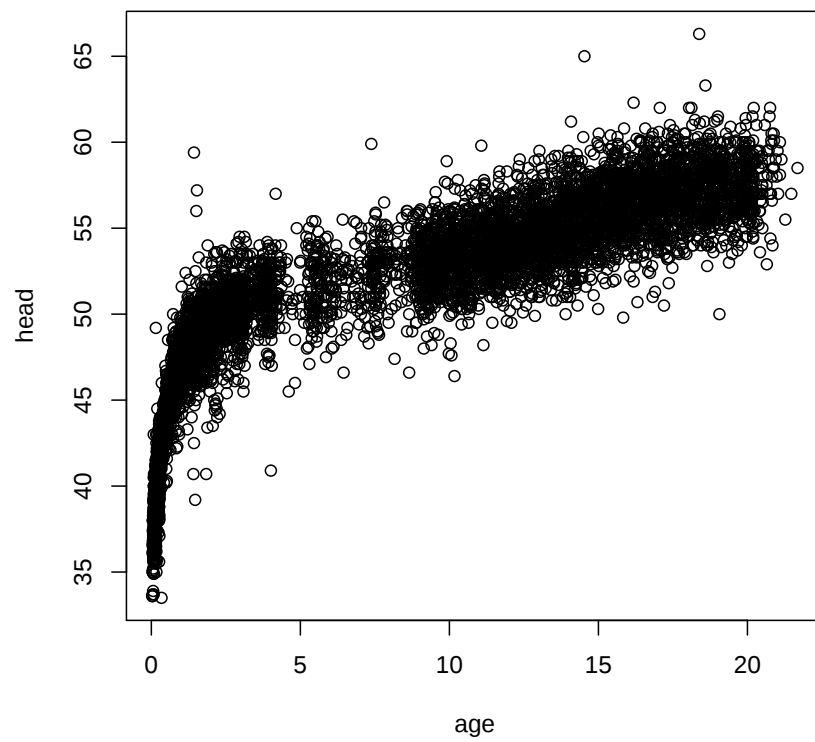


Figure 11.1: Observed Head Circumference of Males (From the Fourth Dutch Growth Study) Against Age.

An automatic procedure, the function `find.hyper()` based on the numerical optimization function `optim` in R, Ihaka and Gentleman (1996), can be used to minimize the $GAIC(\#)$, over the five hyperparameters df_μ , df_σ , df_ν , df_τ and ξ in the BCT model. The results for different values of the penalty $\#$ in $GAIC(\#)$ are shown in Table 11.1. [Note that in general the $GAIC(\#)$ can potentially have multiple local minima (especially for low values of $\#$) and so the automatic procedure should be run with different starting values to ensure a global minimum has been found.]

The following R code can be used to find entries in Table 11.1. Here we show only the $GAIC(\# = 2)$ code. The penalty $\#$ is specified by the `find.hyper()` argument `penalty`. Note also the `c.spar` argument in the cubic spline function `cs()` which is necessary in this case to make sure that the degrees of freedom for smoothing is able to take small values [see the comments on the help file for `cs()`].

```
mod1<-quote(gamlss(head~cs(nage,df=p[1]), sigma.fo=~cs(nage,p[2]),
                    nu.fo=~cs(nage,p[3], c.spar=c(-1.5,2.5)),
                    tau.fo=~cs(nage,p[4], c.spar=c(-1.5,2.5)),
                    data=db, family=BCT, control=gamlss.control(trace=FALSE)))
op<-find.hyper(model=mod1, other=quote(nage<-age~p[5]),
               par=c(10,2,2,2,0.25),
               lower=c(0.1,0.1,0.1,0.1,0.001), steps=c(0.1,0.1,0.1,0.1,0.2), factr=2e9,
               parscale=c(1,1,1,1,0.035), penalty=2 )
```

The procedure takes a long time (approximately one hour!). The final chosen values of the five hyperparameters and the final value of $GAIC(\#)$ are obtained by the components `$par` and `$value` respectively.

```
par 10 2 2 2 0.25 crit= 26792.24 with pen= 2
par 10.1 2 2 2 0.25 crit= 26792.06 with pen= 2
par 9.9 2 2 2 0.25 crit= 26792.43 with pen= 2
par 10 2.1 2 2 0.25 crit= 26792.01 with pen= 2
...
...
par 18.43638 2.679676 0.9969013 6.73205 0.08739384 crit= 26780.56 with pen= 2
par 18.43638 2.679676 0.9969013 6.83205 0.09439384 crit= 26780.57 with pen= 2
par 18.43638 2.679676 0.9969013 6.83205 0.08039384 crit= 26780.56 with pen= 2
>op
$par
[1] 18.43637964 2.67967596 0.99690134 6.83204939 0.08739384

$value
[1] 26780.56

$counts
function gradient
      13      13

$convergence
[1] 0
$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

Note that the degrees of freedom reported in each of the first four components of `$par` in the output do not include the constant and the linear term, so 2 degrees of freedom have to be added to each value to give the total degrees of freedom given in the entries in Table 11.1. If the automatic procedure results in a value of 0.1 for the extra degrees of freedom, that is, the lower boundary of the search, then a further search has to be done to check if the models can be simplify further to either just a linear term or just a constant term. For example, using penalty $\# = 3$ (see row 3 of the Table 11.1) the results of running the automatic procedure were

```
> op
$par
[1] 10.4233233  3.4981990  0.1463869  0.1000000  0.3118194

$value
[1] 26811.75

$counts
function gradient
      12      12

$convergence
[1] 0
```

indicating that the extra degrees of freedom of the τ parameter has reached its lower limit. [also ν has a value close to 0.10]. The model with τ linear in nage was refitted using the commands

```
mod1<-quote(gamlss(head~cs(nage,df=p[1]), sigma.fo=~cs(nage,p[2]),
                  nu.fo=~cs(nage,p[3], c.spar=c(-1.5,2.5)),
                  tau.fo=~nage,
                  data=db, family=BCT, control=gamlss.control(trace=FALSE)))
op<-find.hyper(model=mod1, other=quote(nage<-age^p[4]), par=c(10,4,1,0.25),
               lower=c(0.1,0.1,0.1,0.001), steps=c(0.1,0.1,0.1,0.2), factr=2e9,
               parscale=c(1,1,1,0.035), penalty=3 )
```

The final results are given by

```
>op
$par
[1] 10.3840121  3.5787116  0.1000000  0.3261456 # now nu at 0.2

$value
[1] 26811.67

$counts
function gradient
      6      6

$convergence
[1] 0
```

Given that the ν parameter degrees of freedom has now reached its lower limit of 0.10, we refit the model with a linear function for nage for ν and repeat the process.

Table 11.1: Selected hyperparameters for different penalties $\sharp$ in the BCT model

$\sharp$	df_μ	df_σ	df_ν	df_τ	ξ
2	20.4	4.7	3.0	8.8	0.09
2.5	16.9	4.5	2.7	7.6	0.11
3	12.3	5.7	2	2	0.33
3.84	11.0	2.8	2	2	0.45
$\log(n) = 8.86$	8.9	1	1	2	0.41

```

mod1<-quote(gamlss(head~cs(nage,df=p[1]), sigma.fo=~cs(nage,p[2]),
                  nu.fo=~nage, tau.fo=~nage, data=db, family=BCT,
                  control=gamlss.control(trace=FALSE)))
op<-find.hyper(model=mod1, other=quote(nage<-age~p[3]), par=c(10,4,0.25),
               lower=c(0.1,0.1,0.001), steps=c(0.1,0.1,0.2), factr=2e9,
               parscale=c(1,1,0.035), penalty=3 )

```

The final results are given by

```

> op
$par [1] 10.284879  3.647258  0.331152

$value
[1] 26811.67

$counts
function gradient
      7          7

$convergence
[1] 0

```

This is the the final results in Rigby and Stasinopoulos (2006) and given in row 3 of Table 11.1

Table 11.1 indicates the need for smoothing for μ and possibly also for σ but it is less clear whether smoothing is needed for either ν or τ . The transformation parameter ξ increases for simpler models indicating that this parameter depends partly on the complexity of the fitted model. Below we fit each of the models chosen by $\sharp = 2, 3$ and 8.86 and given in Table 11.1.

```

> nage <- db$age^0.087
> m2 <- gamlss(head ~ cs(nage, df = 18.44), sigma.fo = ~cs(nage,
+   df = 2.68), nu.fo = ~cs(nage, df = 0.99, c.spar = c(-1.5,
+   2.5)), tau.fo = ~cs(nage, df = 6.83), data = db, family = BCT)

GAMLSS-RS iteration 1: Global Deviance = 26878.32
...
...
GAMLSS-RS iteration 9: Global Deviance = 26706.67
GAMLSS-RS iteration 10: Global Deviance = 26706.67

```

```

> nage <- db$age^0.33
> m3 <- gamlss(head ~ cs(nage, df = 10.28), sigma.fo = ~cs(nage,
+   df = 3.65), nu.fo = ~nage, tau.fo = ~nage, data = db, family = BCT,
+   start.from = m2)

GAMLSS-RS iteration 1: Global Deviance = 26749.54
...
...
GAMLSS-RS iteration 5: Global Deviance = 26745.88
GAMLSS-RS iteration 6: Global Deviance = 26745.88

> nage <- db$age^0.414
> m8 <- gamlss(head ~ cs(nage, df = 6.9), sigma.fo = ~1, nu.fo = ~1,
+   tau.fo = ~nage, data = db, family = BCT, start.from = m3)

GAMLSS-RS iteration 1: Global Deviance = 26792.1
...
...
GAMLSS-RS iteration 5: Global Deviance = 26791.48
GAMLSS-RS iteration 6: Global Deviance = 26791.48

> fitted.plot(m3, m2, m8, x = db$age, color = FALSE, line.type = TRUE)

```

Figure 11.2 shows the fitted models for μ , σ , ν and τ with penalties $\sharp = 2$ (AIC), 3 (GAIC) and 8.86 (SBC). The fitted models for μ are very similar but the fitted models for σ , ν and τ vary considerably according to the penalty $\sharp$. The Akaike information criterion (AIC) for model selection appears too liberal in its choice of degrees of freedom. The Schwartz Bayesian Criterion (SBC) for model selection appears too conservative in its choice of degrees of freedom. In general, a higher penalty $\sharp$ leads to a reduction in selected degrees of freedom and hence to a simpler model with smoother fitted μ , σ , ν and τ and smoother resulting fitted centiles. The penalty 3 appears to be a reasonable compromise between the AIC and SBC criteria. In the rest of analysis the penalty $\sharp$ is fixed at 3.

The hyperparameters df_μ , df_σ , df_ν , df_τ and ξ , resulting from minimizing $GAIC(3)$ for seven different distributions (all available in the GAMLSS implementation) are examined in Table 11.2. Of the seven distributions shown, the Box-Cox t (BCT), Johnson's S_u (JSU) Johnson (1949), skew exponential power (SEP) Azzalini (1986) and DiCiccio and Monti (2004), and Box-Cox power exponential (BCPE) Rigby and Stasinopoulos (2004) are all capable of modelling both skewness and kurtosis. The t distribution (TF) is able to model only kurtosis, while the Box-Cox normal distribution, (BCN or BCCG or LMS method), Cole and Green (1992), is able to model only skewness. Table 11.2 shows that the normal distribution (NO), unable to model either skewness or kurtosis, provides the worst fit to the data. The t distribution performs well due to the presence of extreme outliers in the data, but the Box-Cox t distribution performs best as judged GAIC(3). [The models in Table 11.2 were selected using find.hyper using the same techniques described above.]

The conclusion from Table 11.2 is that the BCT model provides the best fit to head circumference, according to criterion GAIC(3), i.e. head circumference requires modelling of both skewness and kurtosis (e.g. using BCT) and is not adequately modelled by modelling either skewness (e.g. using BCN) or kurtosis (e.g. using TF) alone. Hence the *final* chosen model was $BCT(12.3, 5.7, 2, 2, 0.33)$ with global deviance=26745.7.

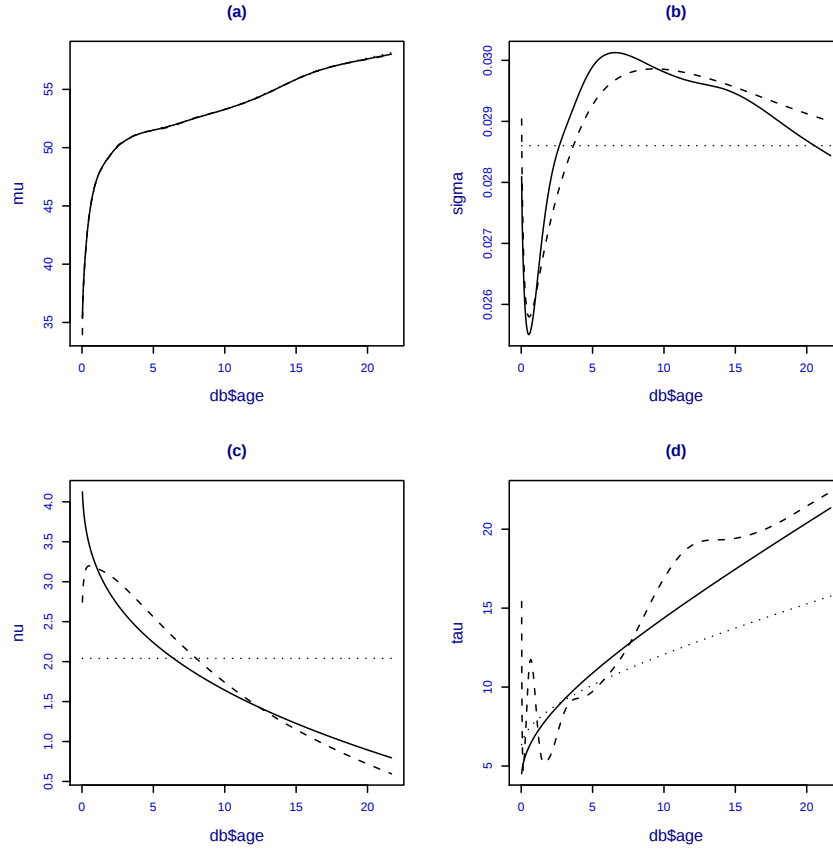


Figure 11.2: The fitted parameters against age for the 'best' BCT models using AIC (---), GAIC(3) (—), and SBC (···), criteria: (a) μ (b) σ (c) ν (d) τ .

Table 11.2: Choosing the distribution using $GAIC(\sharp = 3)$

distribution	GAIC(3)-26814.7	df_{μ}	df_{σ}	df_{ν}	df_{τ}	ξ
NO	221.6	16.4	30	-	-	0.001
BCCG	172.9	16.7	20	14.7	-	0.01
BCPE	81.7	12.2	7.9	2	2	0.34
SEP	71.7	11.7	3.7	2	2	0.40
TF	4.8	13.1	2.9	-	3.1	0.27
JSU	3.4	11.7	3.4	2	2	0.46
BCT	0	12.3	5.7	2	2	0.33

The fitted models for μ , σ , ν , and τ , given by (11.1), for the chosen model $BCT(12.3, 5.7, 2, 2, 0.33)$, are displayed in solid lines in Figure 11.2. The fitted model for μ indicates that the median head circumference of newly born Dutch male babies increases very rapidly until age 1 year, and then increases at a much slower rate (roughly linear from age 3 to 21 years). The fitted centile based coefficient of variation of head circumference is high for newly born Dutch male babies, but decreases until about age 6 months, then increases until age 6 years and then slowly decreases. [This is shown by plotting CV_Y , from Appendix A or Rigby and Stasinopoulos (2006) against age. The plot (not shown here) is similar in shape to the fitted σ plot in figure 11.2(b)].

The fitted model for ν indicates that the distribution of head circumference of newly born Dutch male babies is slightly negatively skew (since $\hat{\nu} > 1$). The negative skewness gradually disappears with age to approximate symmetry at age 19 years (since $\hat{\nu} \approx 1$). The fitted model for τ indicates that the distribution of head circumference of newly born Dutch male babies is highly leptokurtic (since $\hat{\tau}$ is low). The kurtosis gradually reduces with age towards that of a normal distribution (as $\hat{\tau}$ increases).

Figure 11.3 displays the (normalized quantile) residuals, from model $BCT(12.3, 5.7, 2, 2, 0.33)$. Panels (a) and (b) plot the residuals against the fitted values of μ and against age respectively, while panels (c) and (d) provide a kernel density estimate and normal QQ plot for them respectively. The residuals appear random but the QQ plot shows seven extreme outliers (0.1% of the data) in the upper tail of the distribution of y . Nevertheless the Box-Cox t distribution model provides a reasonable fit to the data, substantially better than to the Box-Cox normal distribution (BCCG, LMS) model and preferable to the t distribution (TF) model.

```
> newpar <- par(mfrow = c(2, 2), mar = par("mar") + c(0, 1, 0,
+ 0), col.axis = "blue4", col = "blue4", col.main = "blue4",
+ col.lab = "blue4", pch = "+", cex = 0.45, cex.lab = 1.2,
+ cex.axis = 1, cex.main = 1.2)
> plot(m3, xvar = db$age, par = newpar)

*****
      Summary of the Quantile Residuals
              mean      = -0.0003938235
              variance   = 1.000134
      coef. of skewness  = 0.00893704
      coef. of kurtosis  = 3.057721
Filliben correlation coefficient = 0.9995363
*****

> par(newpar)
```

Figure 11.4 displays detailed diagnostic plots for the residuals using a worm plot developed by van Buuren and Fredriks (2001). In this plot the range of age is split into sixteen contiguous non-overlapping intervals with equal numbers of cases. The sixteen age ranges are displayed in horizontal steps in the chart above the worm plot in Figure 11.4. A detrended normal QQ plot of the residuals in each interval is then displayed. The nineteen outliers are omitted from the worm plot as their deviations lie above the upper limit of the deviation range used in the plots. The worm plot allows detection of inadequacies in the model fit within specific ranges of age. From Figure 11.4, the de-trended QQ plots show adequate fits to the data within most of the 16 age ranges, with only occasional minor inadequacies. van Buuren and Fredriks (2001) proposed fitting cubic models to each of the de-trended QQ plots, with the resulting constant, linear, quadratic and cubic coefficients, $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$ and $\hat{b}_3$ respectively, indicating differences between the

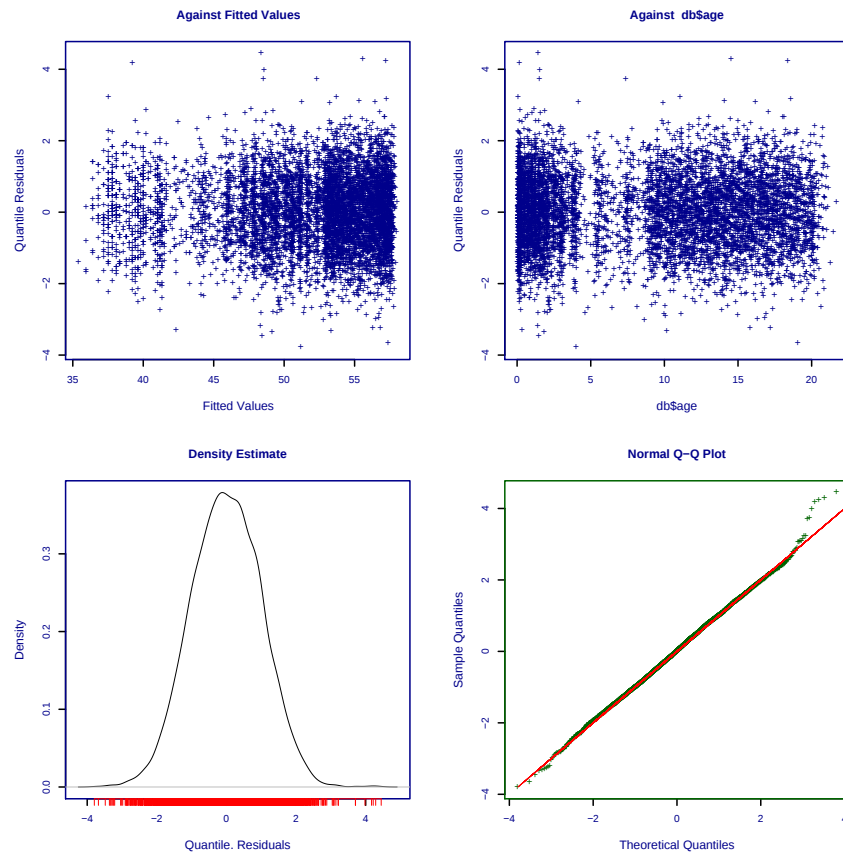


Figure 11.3: The Residuals From Model $BCT(12.3, 5.7, 2, 2, 0.33)$. (a) against fitted values of μ (b) against age (c) kernel density estimate (d) normal QQ plot.

empirical and model residual mean, variance, skewness and kurtosis respectively, within the age range in the QQ plot. They summarize their interpretations in their Table II. For model diagnosis, they categorize absolute values of $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$ and $\hat{b}_3$ in excess of threshold values, 0.10, 0.10, 0.05 and 0.03 respectively, as misfits.

The commands below produce the worm plot in Figure 11.4 together with the number of points missing from each of the 16 detrended Q-Q plots (from the bottom left to top right of Figure 11.4). The sixteen age ranges are given by `$classes`, while `$coef` gives the coefficients $\hat{b}_0$, $\hat{b}_1$, $\hat{b}_2$, and $\hat{b}_3$, of van Buuren and Fredriks (2001).

```
> a<- wp(m3, xvar = db$age, n.inter = 16, ylim.worm = 0.5, cex = 0.3,
+       pch = 20)
```

```
number of missing points from plot= 3 out of 450
number of missing points from plot= 0 out of 430
number of missing points from plot= 1 out of 441
number of missing points from plot= 3 out of 448
number of missing points from plot= 0 out of 433
number of missing points from plot= 1 out of 444
number of missing points from plot= 2 out of 434
number of missing points from plot= 0 out of 441
number of missing points from plot= 0 out of 441
number of missing points from plot= 1 out of 440
number of missing points from plot= 1 out of 438
number of missing points from plot= 1 out of 442
number of missing points from plot= 1 out of 438
number of missing points from plot= 0 out of 444
number of missing points from plot= 3 out of 439
number of missing points from plot= 2 out of 437
```

```
> a
$classes
      [,1] [,2]
[1,] 0.025 0.225
[2,] 0.225 0.695
[3,] 0.695 1.195
[4,] 1.195 1.755
[5,] 1.755 2.545
[6,] 2.545 3.935
[7,] 3.935 7.885
[8,] 7.885 9.995
[9,] 9.995 11.215
[10,] 11.215 12.515
[11,] 12.515 13.655
[12,] 13.655 14.845
[13,] 14.845 16.065
[14,] 16.065 17.375
[15,] 17.375 18.765
[16,] 18.765 21.685
```

```

$coef
      [,1]      [,2]      [,3]      [,4]
[1,] -0.014862811  0.040637477  0.0286515015 -0.0061187442
[2,]  0.014988764 -0.034146717 -0.0286393976 -0.0051177561
[3,]  0.007657586 -0.007530662  0.0204782385 -0.0143011841
[4,] -0.023052142 -0.018047072  0.0312188920  0.0223862354
[5,] -0.013179422  0.054963231 -0.0292996078  0.0008635760
[6,]  0.035064002 -0.056472193 -0.0184655978 -0.0056714383
[7,] -0.014269124  0.018249398 -0.0085894877  0.0070093940
[8,]  0.050715396 -0.007961654 -0.0246738398 -0.0018852659
[9,]  0.008268443  0.041987444 -0.0069455143 -0.0062738689
[10,] -0.057049858 -0.004467303  0.0209612695 -0.0130664534
[11,] -0.079489065 -0.009189481  0.0242124192 -0.0091691084
[12,]  0.016635691  0.040864566  0.0019787469  0.0023046281
[13,]  0.024378092  0.019097741 -0.0003928479 -0.0101459813
[14,]  0.051203931  0.038990419 -0.0315863594 -0.0009339960
[15,]  0.022295461 -0.005969662  0.0368449838 -0.0006083498
[16,] -0.040936746 -0.016087180 -0.0130906981 -0.0036709766

> par(newpar)

```

van Buuren and Fredriks (2001) reported results for the male head circumference data using the LMS model with a 're-scale transformation' of age (which stretches periods of rapid growth and compresses periods of lower growth in y to provide a uniform growth rate on the transformed age scale); see Cole *et al.* (1998) for details. Following this complex transformation of age, they chose 9 degrees of freedom for μ , 5 for σ and a constant value $\nu = 1$. However, they reported a total of 16 violations in the resulting worm plot coefficients from their chosen fitted model (i.e. values of $\hat{b}$'s in excess of their threshold values), indicating that the model does not adequately fit the data within many specific age ranges. In contrast, there are no violations in the worm coefficients from the fitted model $BCT(12.3, 5.7, 2, 2, 0.33)$, indicating an adequate fit to the data within age ranges.

The fit within age groups can be further investigated by calculating Q statistics for testing normality of the residuals within age groups, Royston and Wright (2000).

Let G be the number of age groups and let $\{r_{gi}, i = 1, 2, \dots, n_i\}$ be the residuals in age group g , with mean $\bar{r}_g$ and standard deviation s_g , for $g = 1, 2, \dots, G$. The following statistics $Z_{g1}, Z_{g2}, Z_{g3}, Z_{g4}$ are calculated from the residuals in group g to test whether the residuals in group g have population mean 0, variance 1, skewness 0 and kurtosis 3, where $Z_{g1} = n_g^{1/2} \bar{r}_g$, $Z_{g2} = \left\{ s_g^{2/3} - [1 - 2/(9n_g - 9)] \right\} / \{2/(9n_g - 9)\}^{1/2}$ and Z_{g3} and Z_{g4} are test statistics for skewness and kurtosis given by D'Agostino *et al.* (1990), in their equations (13) and (19) respectively.

The Q statistics of Royston and Wright (2000) are then calculated by $Q_j = \sum_{g=1}^G Z_{gj}^2$ for $j = 1, 2, 3, 4$. Royston and Wright (2000) discuss approximate distributions for the Q statistics under the null hypothesis that the true residuals are normally distributed (although their simulation study was mainly for normal error models) and suggest Chi-squared distributions with adjusted degrees of freedom $G - df_\mu$, $G - [df_\sigma + 1]/2$ and $G - df_\nu$ for Q_1, Q_2 and Q_3 respectively. By analogy we suggest degrees of freedom $G - df_\tau$ for Q_4 . The resulting significance levels should be regarded as providing a guide to model inadequacy, rather than exact formal test results.

Significant Q_1, Q_2, Q_3 or Q_4 statistics indicate possible inadequacies in the models for parameters μ, σ, ν and τ respectively, which may be overcome by increasing the degrees of freedom

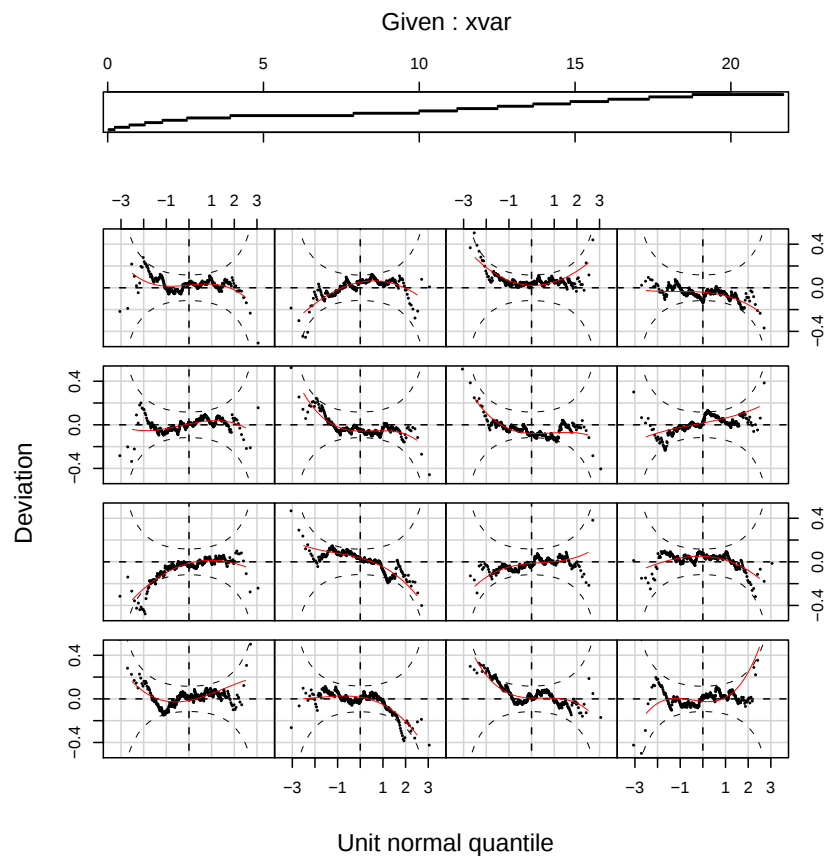


Figure 11.4: Worm Plot of the Residuals From Model $BCT(12.3, 5.7, 2, 2, 0.33)$. The 16 detrended QQ plots, from bottom left to top right plot, correspond to 16 age ranges (displayed in steps above the worm plot from 0 to 22 years).

in the model for the particular parameter. Q-statistics can be obtained using the `Q.stats()` function.

```
> Q.stats(m3, xvar=db$age, n.inter=16)
```

	Z1	Z2	Z3	Z4	AgostinoK2	N
0.02500 to 0.22499	0.29075746	0.76161874	1.49751824	0.41809350	2.41736305	450
0.22499 to 0.69499	-0.28128215	-1.41060390	-1.44149082	-0.30586512	2.17144925	430
0.69500 to 1.195	0.58959281	-1.43372471	1.00651318	-1.47503391	3.18879384	441
1.195 to 1.755	0.17095219	1.57566314	2.04577887	2.90097247	12.60085244	448
1.755 to 2.545	-0.88211304	1.69886701	-1.41571580	0.26905454	2.07664157	433
2.545 to 3.935	0.35088203	-2.17873364	-0.98249826	-0.40098283	1.12609006	444
3.935 to 7.885	-0.47567339	1.16169259	-0.37681407	0.93929578	1.02426540	434
7.885 to 9.995	0.54839023	-0.37698931	-1.25419319	-0.02916264	1.57385101	441
9.995 to 11.215	0.02820869	0.71347006	-0.36923843	-0.46702964	0.35445370	441
11.215 to 12.515	-0.75829127	-1.25102427	1.05517635	-1.24239880	2.65695190	440
12.515 to 13.655	-1.15834867	-1.04467613	1.24497745	-0.69930559	2.03899714	438
13.655 to 14.845	0.39122457	1.46133224	0.28838334	0.88510660	0.86657865	442
14.845 to 16.065	0.50199822	-0.28470210	-0.03779593	-0.81610852	0.66746165	438
16.065 to 17.375	0.41530414	1.10292277	-1.53827925	0.13612365	2.38483271	444
17.375 to 18.765	1.23685873	-0.16111535	2.01054712	0.97951796	5.00175515	439
18.765 to 21.685	-1.12861039	-0.76835437	-0.70354040	0.02422780	0.49555609	437
TOTAL Q stats	7.26743288	23.44758925	23.99114844	16.65474516	40.64589361	7040
df for Q stats	3.71821323	12.67541799	14.00000000	14.00000000	28.00000000	0
p-val for Q stats	0.10373252	0.03212991	0.04593521	0.27504371	0.05785201	0

The Z_{gj} statistic when squared provides the contribution from age group g to the statistic Q_j , and hence helps identify which age groups are causing the Q_j statistic to be significant and therefore in which age groups the model is unacceptable.

Provided the number of groups G is sufficiently large relative to the degrees of freedom in the model for the parameter, then the Z_{gj} values should have approximately standard normal distributions under the null hypothesis that the true residuals are standard normally distributed. We suggest as a rough guide values of $|Z_{gj}|$ greater than 2 be considered as indicative of significant inadequacies in the model. Note that significant positive (or negative) values $Z_{gj} > 2$ (or $Z_{gj} < -2$) for $j = 1, 2, 3$ or 4 indicate respectively that the residuals in age group g have a higher (or lower) mean, variance, skewness or kurtosis than the assumed standard normal distribution. The model for parameter μ, σ, ν or τ may need more degrees of freedom to overcome this. For example if the residual mean in an age group is too high, the model for μ may need more degrees of freedom in order for the fitted μ from the model to increase within the age group.

The following command `centiles.split` obtains the centiles curves given in figure 11.5 for head circumference against age split at age=2.5 years [defined by `c(2.5)` in the command]. The output below compares the sample proportion below each centile curve for each of the two age ranges, i.e. below age 2.4 years and above age 2.5 years.

```
> centiles.split(m3, xvar = db$age, c(2.5), ylab = "HEAD", xlab = "AGE")
```

	0.03 to 2.5	2.5 to 21.68
0.4	0.4599816	0.4110152
2	1.7479301	1.7879162
10	10.3495860	9.8849157
25	26.3109476	25.2774353

50	50.1839926	49.7533909
75	73.7810488	74.3937526
90	90.2023919	89.9712289
98	98.4360626	98.2326346
99.6	99.4940202	99.7944924

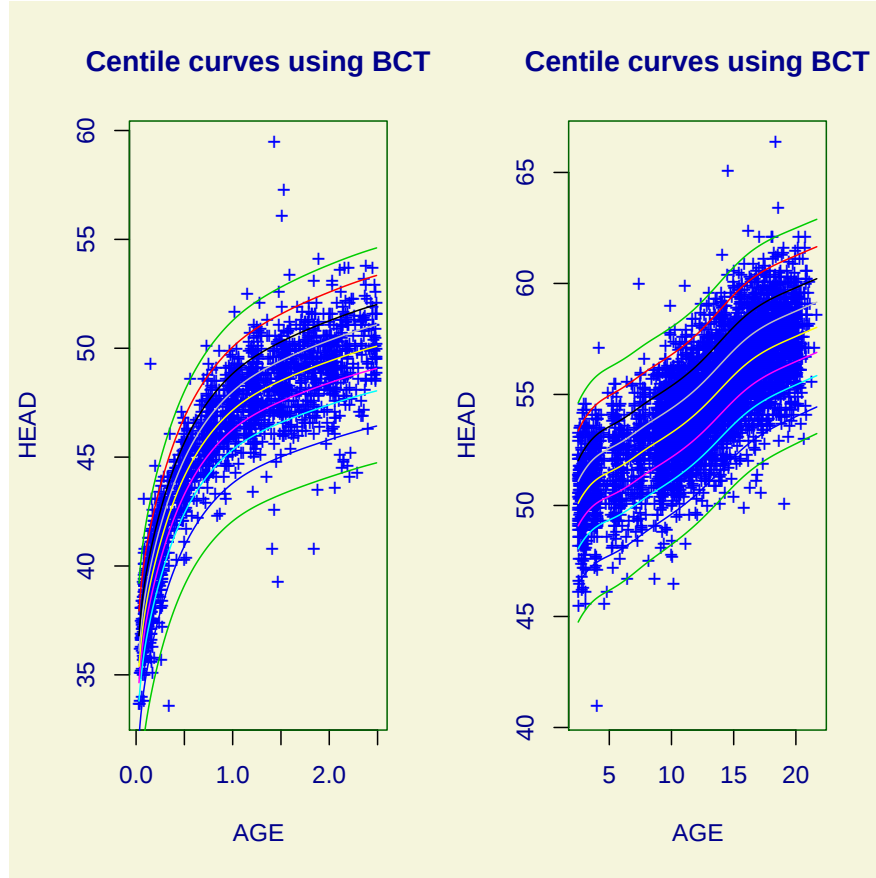


Figure 11.5: Observed Head Circumference With Nine Fitted Model Centile Curves (0.4, 2, 10, 25, 50, 75, 90, 98, 99.6), From Model $BCT(12.3, 5.7, 2, 2, 0.33)$, against Age: (a) 0-2.5 years (b) 2.5-22 years

Figure 11.5 provides nine fitted model centile curves, defined by (11.2) below, for head circumference for model $BCT(12.3, 5.7, 2, 2, 0.33)$, with centiles $100\alpha = 0.4, 2, 10, 25, 50, 75, 90, 98, 99.6$. For each α , the centile curve, y_α against x , is obtained by finding the fitted values $(\hat{\mu}, \hat{\sigma}, \hat{\nu}, \hat{\tau})$ for each x (over a range of values of x) and substituting the values into

$$y_\alpha = \begin{cases} \mu[1 + \sigma\nu t_{\tau,\alpha}]^{1/\nu} & \text{if } \nu \neq 0 \\ \mu \exp[\sigma t_{\tau,\alpha}] & \text{if } \nu = 0, \end{cases} \quad (11.2)$$

where $t_{\tau,\alpha}$ is the 100α centile of t_τ , a standard t distribution with degrees of freedom parameter τ . [Strictly the exact formula for y_α is given in Appendix A of Rigby and Stasinopoulos (2006)]

Table 11.3: Comparison of model and sample percents lying below each centile curve for model $BCT(12.3, 5.7, 2, 2, 0.33)$.

	Percent								
model	0.4	2	10.0	25	50	75	90	98	99.6
sample	0.43	1.78	10.02	25.6	49.88	74.2	90.04	98.29	99.7

]. The resulting centiles are plotted for age range 0 to 2.5 years in Figure 11.5(a) and for age range 2.5 to 22 years in Figure 11.5(b), for clarity of presentation. Table 11.3 compares the sample percent lying below each centile curve with the nominal model 100α percent. The model and sample percents agree reasonably well.

Finally in order to investigate whether skewness and kurtosis were due to the effect of extreme outliers the 14 most extreme observations (7 from the upper and 7 from the lower tail, were removed and the models for each of the seven distributions in Table 11.2 were refitted (after reselecting their optimal df 's and λ). The BCT and JSU distributions still provide the best fits according to $GAIC(3)$, indicating that there is both skewness and kurtosis remaining even after the removal of the extreme outliers. In addition their fits to the data were substantially improved, leading to improved centile estimates. [Finally an adjustment to the resulting centile percentages, for the 0.1 % (7 out of 7040 in each tail) of cases removed, should be made if these cases are believed to be genuine.]

11.2 Exercises

- Q2: Rigby and Stasinopoulos (2004) analysed the body mass index (bmi) of 7294 Dutch boys against age using a Box-Cox Power Exponential (BCPE) distribution for bmi. The data are stored in file dbO.
 - (a) Transform age to a new variable `nage <- age**0.37`.
 - (b) Fit a BCPE distribution to bmi with a cubic smoothing spline in age for the predictors for parameter m with 8.5 smoothing degrees of freedom (on top of constant and linear terms), and a constant for the predictors for s, n and t.
 - (c) Use the fitted values from (b) as starting values for fitting a BCPE distribution to bmi with with cubic smoothing splines in age for the predictors for parameters m, s and n, with 8.5, 3.8 and 3.3 smoothing degrees of freedom respectively, and a linear model for the predictor for t.
 - (d) Plot the fitted parameters for the fitted model in (c) using `fitted.plot`.
 - (e) Obtain a centile plot for the fitted model in (c) using `centiles.split`.
 - (f) Investigate the residuals from the fitted model in (c) using e.g. `plot`, `wp` (worm plot) and `Q.stats`.

Comment: Numerical optimisation (i.e. `find.hyper`) could be used to select the degrees of freedom for the parameters m, s, n and t of the BCPE distribution, together with the power parameter x applied to transform the explanatory variable age, using penalty $\# = 3$ in the generalised Akaike information criterion. The resulting selected model has a

transformation power $x = 0.37$ applied to age and cubic smoothing splines in age for the predictors for parameters m , s and n , with 8.5, 3.8 and 3.3 smoothing degrees of freedom respectively (on top of constant and linear terms), and a linear model for the predictor for t .

References: Rigby, R.A. and Stasinopoulos, D.M. (2004) Smooth centile curves for skew and kurtotic data modelled using the Box-Cox power exponential distribution. *Statistics in Medicine*, 23, 3053-3076.

Chapter 12

Random effect models at the observational level

There are three chapters devoted to random effect models. The current one describes the use of random effects at the observational level, that is, when there are as many random effects as the number of observations in the data. The main application of random effects of this type is to deal with overdispersion, that is, when extra variability is present in the data which can not be explained purely by the use of the distribution of the response variable itself. We distinguish three types of random effect model at the observational level:

- (i) when an explicit continuous mixture distribution exists.
- (ii) when a continuous mixture is not explicit but approximated using Gaussian quadrature points
- (iii) when a 'non-parametric' mixture [effectively an finite mixture] is assumed

These different types are described in Sections 12.2, 12.3 and 12.4 respectively.

Chapter 13 describes the use of random effects at a factor level, that is when we have repeated observation within the same subject and we want the model to take this into the account. Both the current and the next Chapters are dealing mainly with random effects affecting the predictor for the parameter μ of a gamlss family distribution. The more general case where a random effect could be present in any of the distributional parameters is discussed in Sections 12.3.3 and 12.4.3 and in Chapter 14.

Note that the non-parametric random effect models in section

12.1 Introduction

Assume that, given the random effect variable γ , y has conditional probability (density) function $f(y|\gamma)$ and marginally γ has probability (density) function $f(\gamma)$. Then the marginal density of Y is given by

$$f_Y(y) = \int f(y|\gamma)f(\gamma)d\gamma \tag{12.1}$$

Note that γ may be a univariate or multivariate random effect variable. Assume observations $(y_1, y_2, \dots, y_n)$ are conditionally independent given the observational level random effect variables $(\gamma_1, \gamma_2, \dots, \gamma_n)$ and that the random effect variables are a random sample from $f(\gamma)$ and therefor also independent, then marginally $(y_1, y_2, \dots, y_n)$ are independent since

$$\begin{aligned}
 f_Y(\mathbf{y}) &= \int f(\mathbf{y}|\boldsymbol{\gamma})f(\boldsymbol{\gamma})d\boldsymbol{\gamma} \\
 &= \int \left[\prod_{i=1}^n f(y_i|\gamma_i) \right] \left[\prod_{i=1}^n f(\gamma_i) \right] d\boldsymbol{\gamma} \\
 &= \int \prod_{i=1}^n [f(y_i|\gamma_i)f(\gamma_i)] d\gamma_1, d\gamma_2, \dots, d\gamma_n \\
 &= \prod_{i=1}^n \left[\int f(y_i|\gamma_i)f(\gamma_i)d\gamma_i \right] \\
 &= \prod_{i=1}^n f_{Y_i}(y_i)
 \end{aligned} \tag{12.2}$$

where $\mathbf{y}^T = (y_1, y_2, \dots, y_n)$ and $\boldsymbol{\gamma}^T = (\gamma_1, \gamma_2, \dots, \gamma_n)$. Hence the likelihood function is a product of the (marginal) likelihoods of each observation y_i for $i = 1, 2, \dots, n$, which are obtained by integrating out the random effect for each observation.

12.2 Fitting an explicit continuous mixture distributions

For specific combinations of the conditional density function $f(y|\gamma)$ and $f(\gamma)$ the marginal density $f_Y(y)$ can be obtained explicitly and the likelihood maximized directly.

Example 1: Continuous Poisson mixture distribution. Let the conditional distribution of y given the random effect γ be $PO(\gamma\mu)$ and $\gamma \sim GA(1, \sigma^{1/2})$ then $y \sim NBI(\mu, \sigma)$ (see Section 5.2.1). Note that provided γ has mean 1, then μ will be the marginal mean of y , a desirable property for interpretation of the fitted model. Table 5.1 in Section 5 provides other continuous Poisson mixtures, i.e. other continuous mixing distributions for γ , and the resulting marginal distribution for y .

Example 2: Continuous normal mixture distribution. Let $y|\gamma \sim N(\mu, \sigma^2\gamma)$ and $\gamma^{-1} \sim \chi_\nu^2$ then $y \sim TF(\mu, \sigma, \nu)$.

12.3 Fitting non-explicit continuous mixture distributions using Gaussian quadrature

12.3.1 Introduction

For many combinations of $f(y|\gamma)$ and $f(\gamma)$, the marginal density $f_Y(y)$ cannot be obtained explicitly. In this case $f_Y(y)$ can be obtained by numerical integration (at considerable computational cost) or can be obtained approximately by other methods. One approximate method is called Gaussian quadrature, where "Gaussian" refers to the originator of the method and not to any specific (e.g. normal) distribution for $f(\gamma)$. Effectively Gaussian quadrature replace the

continuous distribution $f(\gamma)$ with a discrete distribution taking values γ_k with probability π_k for $k = 1, 2, \dots, K$.

12.3.2 Fitting a normal random effect model in the predictor for μ using Gaussian quadrature

Here we assume that the random effects $\gamma_1, \gamma_2, \dots, \gamma_n$ (at the observational level) are a random sample from a normal distribution. For $i = 1, 2, \dots, n$, let $y_i \sim D(\mu_i, \sigma_i, \nu_i, \tau_i)$ be conditionally independent given random effect γ_i where

$$\begin{aligned} g_1(\boldsymbol{\mu}) &= \boldsymbol{\eta}_1 = \mathbf{X}_1 \boldsymbol{\beta}_1 + \boldsymbol{\gamma} \\ g_2(\boldsymbol{\sigma}) &= \boldsymbol{\eta}_2 = \mathbf{X}_2 \boldsymbol{\beta}_2 \\ g_3(\boldsymbol{\nu}) &= \boldsymbol{\eta}_3 = \mathbf{X}_3 \boldsymbol{\beta}_3 \\ g_4(\boldsymbol{\tau}) &= \boldsymbol{\eta}_4 = \mathbf{X}_4 \boldsymbol{\beta}_4. \end{aligned} \quad (12.3)$$

where $\boldsymbol{\gamma}^T = (\gamma_1, \gamma_2, \dots, \gamma_n)$ and $\gamma \sim N(0, \sigma_\gamma^2)$ independently for $i = 1, 2, \dots, n$. Let $\gamma_i = \sigma_\gamma Z_i$ then $Z_i \sim N(0, 1)$ independently for $i = 1, 2, \dots, n$.

Gaussian quadrature effectively approximates the continuous $N(0, 1)$ distribution for each Z_i by a discrete distribution, i.e.

$$Z_i = z_k \quad \text{with probability} \quad \pi_k \quad \text{for} \quad k = 1, 2, \dots, K. \quad (12.4)$$

See, for example, Figure 12.1 for a visual explanation of the Gaussian quadrature discrete distribution approximation to $N(0, 1)$ when $K=10$. The model (12.3) with (12.4) can now be considered as a finite mixture of K components in which the prior (or mixing) probabilities π_k 's are fixed and known and the z_k 's are also fixed and known (once K , the total number of quadrature points has been chosen). Hence $g_1(\mu_i) = \mathbf{x}_{ik}^T \boldsymbol{\beta}_1 + z_k \sigma_k$ with probability π_k with $k = 1, 2, \dots, K$. Now the likelihood can be maximized using the EM algorithm. Let $\boldsymbol{\delta}_i^T = (\delta_{i1}, \delta_{i2}, \dots, \delta_{iK})$ be the missing indicator variable vector for observation i where

$$\delta_{ik} = \begin{cases} 1, & \text{if observation } i \text{ has random effect value } u_k, \text{ i.e. } \gamma_i = u_k \\ 0, & \text{otherwise} \end{cases} \quad (12.5)$$

Let $\boldsymbol{\delta}^T = (\boldsymbol{\delta}_1^T, \boldsymbol{\delta}_2^T, \dots, \boldsymbol{\delta}_n^T)$ combine all indicator variable vectors. Then the log likelihood of the complete data $(\mathbf{y}, \boldsymbol{\delta})$ is in the form of 7.7. The EM algorithm maximizes the marginal likelihood of data $\mathbf{y}$ with respect to the parameters $\boldsymbol{\psi} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4, \sigma_\gamma)$. Note that $\boldsymbol{\theta}_k = \boldsymbol{\psi}$ for $1, 2, \dots, K$, i.e. all K components of the finite mixture have the same parameters, but the k components have different predictors for μ .

Summary of the $(r+1)^{th}$ iteration of the EM algorithm

E-step Replace δ_{ik} in (7.7) by $\hat{w}_{ik}^{(r+1)}$ [its conditional expectation given $\mathbf{y}$ and given the current estimate $\hat{\boldsymbol{\psi}}^{(r)}$ at iteration r] obtained from

$$\hat{w}_{ik}^{(r+1)} = \frac{\pi_k f_k(y_i | \hat{\boldsymbol{\psi}}^{(r)})}{\sum_{k=1}^K \pi_k f_k(y_i | \hat{\boldsymbol{\psi}}^{(r)})} \quad (12.6)$$

to give (7.8) where $\boldsymbol{\psi} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4, \sigma_\gamma)$.

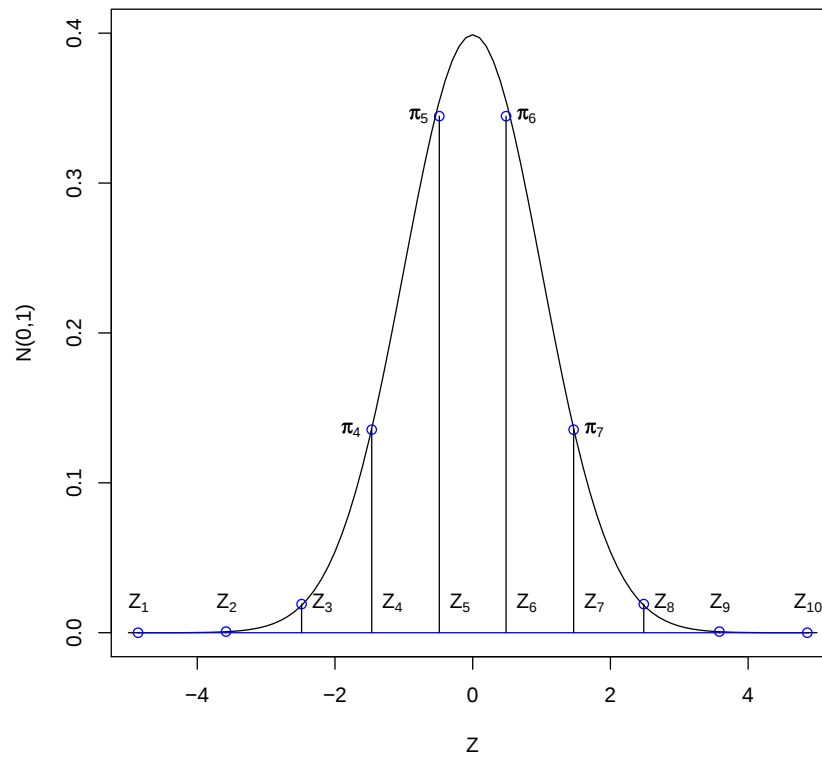


Figure 12.1: Plot showing how the continuous distribution $NO(0,1)$ is approximated by Gaussian quadrature with $K = 10$

Table 12.1: Table showing the expansion of data for fitting the Gauss quadrature model at M step (2) of the EM algorithm

i	k	$\mathbf{y}_e$	$\mathbf{x}_{1e}$	$\mathbf{x}_{2e}$	$\mathbf{x}_{3e}$	$\mathbf{x}_{4e}$	$\mathbf{z}$	weights $\hat{\mathbf{w}}^{(r+1)}$
1	1	$\mathbf{y}$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_1	$\hat{\mathbf{w}}_1^{(r+1)}$
2	1						z_1	
$\vdots$	$\vdots$						$\vdots$	
$\vdots$	$\vdots$						$\vdots$	
n	1						z_1	
1	2	$\mathbf{y}$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_2	$\hat{\mathbf{w}}_2^{(r+1)}$
2	2						z_2	
$\vdots$	$\vdots$						$\vdots$	
$\vdots$	$\vdots$						$\vdots$	
n	2						z_2	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	K	$\mathbf{y}$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_K	$\hat{\mathbf{w}}_K^{(r+1)}$
2	K						z_K	
$\vdots$	$\vdots$						$\vdots$	
$\vdots$	$\vdots$						$\vdots$	
n	K						z_K	

M-step

- (1) Since the components $f_k(y)$, for $k = 1, 2, \dots, K$, have the same parameters (i.e. all parameters in common), Q is maximized with respect to ψ by fitting a single GAMLSS model to an expanded data set, using weights $\hat{\mathbf{w}}^{(r+1)}$ where $\mathbf{w}^T = (\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_K^T)$ and $\mathbf{w}_k^T = (w_{1k}, w_{2k}, \dots, w_{nk})$ for $k = 1, 2, \dots, K$. This is achieved by expanding the data set K times, as in Table 12.1 and fitting a single GAMLSS model to the expanded $\mathbf{y}$, i.e. $\mathbf{y}_e$, where the subscript indicates expanded. The corresponding expanded predictor models for μ_e is $\eta_e = \mathbf{x}_{1e}\beta_1 + \mathbf{z}\sigma_\gamma$. The corresponding expanded predictor models for σ_e , ν_e , and τ_e are $\eta_{se} = \mathbf{x}_{se}\beta_s$ for $s = 2, 3$, and 4 respectively.

Note that since the π 's are fixed and known, they do not need to be estimated.

12.3.3 Fitting a random effect model in the predictor for other distribution parameters.

The normal random effect could have been included in any one of the predictors of the distribution parameters μ , σ , ν or τ in 12.3 and fitted by including the term $\mathbf{z}\sigma_\gamma$ in the corresponding extended predictor in the M-step (1) of the EM algorithm. However if there are more than one (say q) normal random effect effects in the model 12.3 then the Gaussian quadrature fitting method becomes infeasible, because the data need to be expanded k^q times. Chapter 14 considers an alternative fitting procedure for this more general case.

12.4 Non parametric random effects models

12.4.1 Non parametric random intercept in the predictor for μ

Here assume that the random effects $\gamma_1, \gamma_2, \dots, \gamma_n$ in model (12.3) are a random sample from a (non-parametric) distribution which is modelled as a discrete distribution given by

$$\gamma_i = u_k \quad \text{with probability} \quad \pi_k \quad \text{for} \quad k = 1, 2, \dots, K \quad (12.7)$$

for $i = 1, 2, \dots, n$. The u_k 's and π_k 's are assumed to be fixed unknown constants. See Figure 12.2 for an example plot of a 'non-parametric' distribution (12.7) with $K = 5$. The resulting (marginal) model for y_i is just a finite mixture of GAMLSS models with parameters $(\beta_1, \beta_2, \beta_3, \beta_4)$ in common, but different intercept parameters $(u_1, u_2, \dots, u_K)$ in the predictor for μ . The model can be fitted using the EM algorithm of Section 7.5. and using the R function `gamlssNP()`.

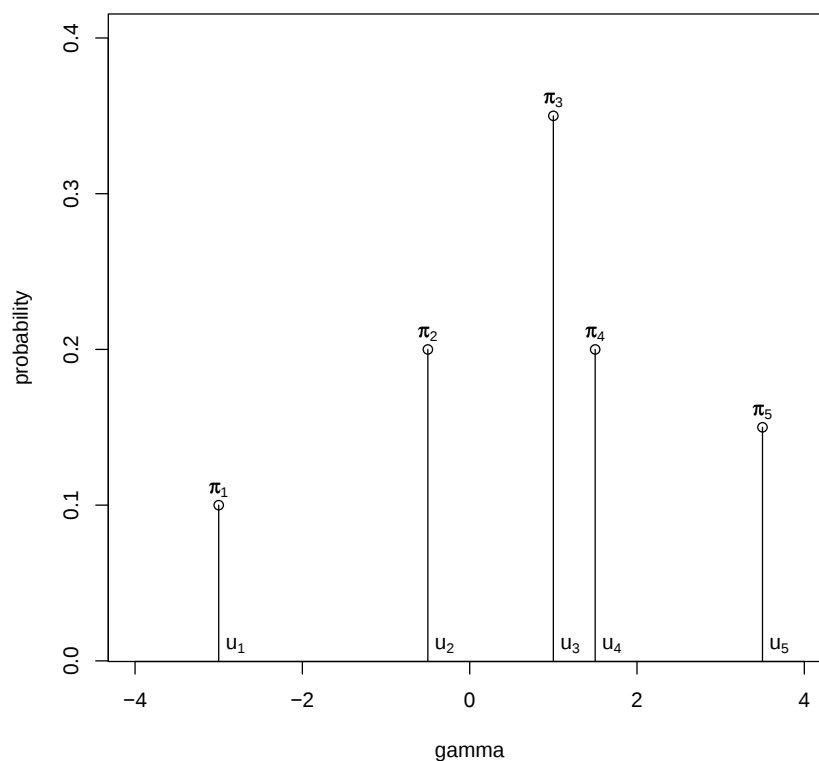


Figure 12.2: Plot showing a non parametric mixture distribution with $K = 5$

12.4.2 Non parametric random coefficients in the predictor for μ

Model (12.3) can be amended to a model with non-parametric random intercept and slope in the predictor for μ , i.e.

$$g_1(\mu_i) = \eta_1 = \mathbf{X}_1^T \beta_1 + \gamma_0 + \gamma_1 x_{1i} \quad (12.8)$$

$$(12.9)$$

where $\gamma_0^T = (\gamma_{01}, \gamma_{02}, \dots, \gamma_{0n})$, $\gamma_1^T = (\gamma_{11}, \gamma_{12}, \dots, \gamma_{1n})$ and $(\gamma_{0i}, \gamma_{1i})$ for $i = 1, 2, \dots, n$ are a random sample from a bivariate 'non-parametric' distribution, taking values (u_{0k}, u_{1k}) with probability π_k for $k = 1, 2, \dots, K$, i.e. $(\gamma_{0i}, \gamma_{1i}) = (u_{0k}, u_{1k})$, with probability π_k for $k = 1, 2, \dots, K$ for $i = 1, 2, \dots, n$. As an example of a two dimensional non-parametric distribution see figure (12.3) where a hypothetical 'non-parametric' distribution is plotted with $K = 10$. The prior probabilities π_k , for $k = 1, 2, \dots, K$, are assumed constant over i . Again the resulting model is a finite mixture model (with parameters in common) and can be fitted using the EM algorithm of Section 7.5. Additional random coefficients can easily be included in the predictor for μ .

The function `gamlssNP()` can be used to fit the model using the argument `random` to declare which of the explanatory variables for μ should have random coefficients.

12.4.3 Non parametric random coefficients in the predictor for all distributional parameters

Model (12.3) can be amended to include 'non-parametric' random coefficients (e.g. intercept and slopes) in the predictor for any one or more of the distributional parameters μ, σ, ν and τ of a GAMLSS model. The model is fitted using the EM algorithm of Section 7.5. For example model (12.3) could be amended to model

$$\begin{aligned} g_1(\mu_i) &= \mathbf{x}_{1i}^T \beta_1 + \gamma_{10i} + \gamma_{11i} x_{1i} \\ g_2(\sigma_i) &= \mathbf{x}_{2i}^T \beta_2 + \gamma_{20i} \\ g_3(\nu_i) &= \mathbf{x}_{3i}^T \beta_3 \\ g_4(\tau_i) &= \mathbf{x}_{4i}^T \beta_4. \end{aligned}$$

for $i = 1, 2, \dots, n$. In the function `gamlssNP()` the above model for σ can be fitted by adding the factor `MASS` in the formula for the predictor of σ , i.e. `sigma.fo=~x+MASS`.

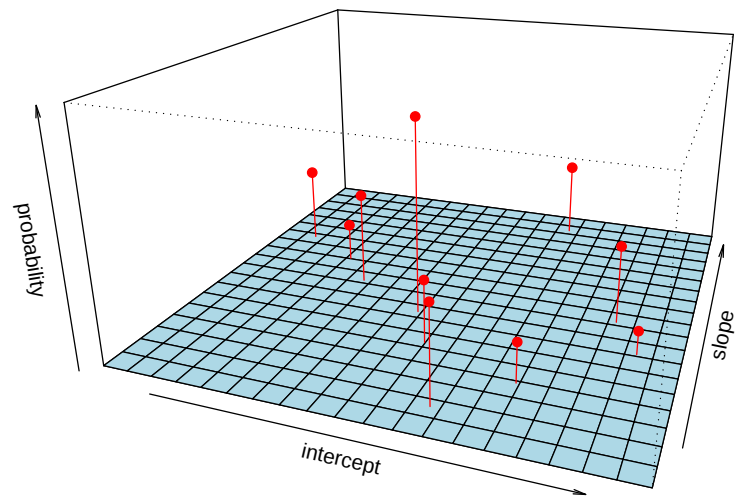


Figure 12.3: Plot showing a non parametric mixture distribution in two dimensions with $K = 10$

school	estimated treatment effect	standard error of effect estimate of σ
A	28	15
B	8	10
C	-3	16
D	7	11
E	-1	9
F	1	11
G	18	10
H	12	18

Table 12.2: The educational testing experiments data

12.5 Examples

12.5.1 Schools example

Data summary:

R data file: `schools` given by Gelman *et al.* (2004) of dimensions 8×3 , (see Table 12.2)

variables

`school` : a factor with 8 levels

`estimate` : the estimated treatment effects

`sd` : the standard error of the treatment effects

purpose: to demonstrate the fitting of simple random effect model in `gamlss`

conclusion: A normal distribution random effect is used

Consider the data in table 12.2 obtained from an educational testing experiment in eight schools, given by Gelman *et al.* (2004), section 5.5. More details on the experiment can be found there. The authors used this example to demonstrate an hierarchical Bayesian analysis. What is important here is that the estimated treatment effects in column two of the table 12.2 are obtained by independent experiments and are assumed to come from a normal distribution with a standard deviation obtained from column 3 of the table 12.2. We start by fitting a null model with a constant mean for the response variable, `estimate`, i.e. $y_i \sim N(\mu, \sigma_i^2)$, and a saturated model with different means for each school, i.e. $y_i \sim N(\mu_i, \sigma_i^2)$, where y_i is the `estimate` at school i and σ_i is given by `sd`. As in Gelman *et al.* (2004) both models are fitted using a fixed standard deviation parameter vector, `sigma`, given in the third column of table 12.2. [Strictly the third column gives estimates of sigmas based on large sample sizes rather than fixed known sigmas.] There are two ways to achieve this in `gamlss`, the first is to use the `gamlss` argument `sigma.fix`. [Note that `sigma` is fixed at its starting value, i.e. `sigma.start=sd`.] The second way is using an `offset` in the formula for `sigma`. [Note that due the default `log` link for `sigma` for the (default) normal distribution, the offset is `log(sd)`. Note also `-1` in the `sigma.fo` removes the constant from the `log(sigma)` model.] We demonstrate both methods below

```

> library(gamlss)

***** GAMLSS Version 1.8-9 *****
For more on GAMLSS look at http://www.gamlss.com/
Type gamlssNews() to see new features/changes/bug fixes.

> library(gamlss.mx)
> school <- c("A", "B", "C", "D", "E", "F", "G", "H")
> estimate <- c(28, 8, -3, 7, -1, 1, 18, 12)
> sd <- c(15, 10, 16, 11, 9, 11, 10, 18)
> schools <- data.frame(school = school, estimate = estimate, sd = sd)
> mnull <- gamlss(estimate ~ 1, sigma.fix = TRUE, sigma.start = sd,
+   data = schools)

GAMLSS-RS iteration 1: Global Deviance = 59.3485
GAMLSS-RS iteration 2: Global Deviance = 59.3485

> msat <- gamlss(estimate ~ school, sigma.fix = TRUE, sigma.start = sd,
+   data = schools)

GAMLSS-RS iteration 1: Global Deviance = 54.6414
GAMLSS-RS iteration 2: Global Deviance = 54.6414

> mnull1 <- gamlss(estimate ~ 1, sigma.fo = ~offset(log(sd)) -
+   1, data = schools)

GAMLSS-RS iteration 1: Global Deviance = 59.4168
GAMLSS-RS iteration 2: Global Deviance = 59.3485
GAMLSS-RS iteration 3: Global Deviance = 59.3485

> msat1 <- gamlss(estimate ~ school, sigma.fo = ~offset(log(sd)) -
+   1, data = schools)

GAMLSS-RS iteration 1: Global Deviance = 54.6414
GAMLSS-RS iteration 2: Global Deviance = 54.6414

> fitted(mnull)

      1      2      3      4      5      6      7      8
7.685617 7.685617 7.685617 7.685617 7.685617 7.685617 7.685617 7.685617

> deviance(mnull1)

[1] 59.34849

> fitted(msat)

 1  2  3  4  5  6  7  8
28 8 -3 7 -1 1 18 12

> deviance(msat1)

```

[1] 54.64141

The global deviance for the null model, `mnull`, is 59.3485 while the fitted values are the mean of the variable `estimate`. The global deviance for the saturated model, `msat`, is 54.6414 while the fitted values are the actual data i.e. variable `estimate`. We can not perform an F test between the model since the degrees of freedom of the saturated model is zero. Nevertheless a chi-square test between the two models result to a value of $\chi^2 = 4.7071$ not significant for the 7 degrees of freedom used. That is, there is no statistical evidence to support the saturated model. A classical statistician would probably stop here and would use the null model for any inference on the data. Gelman *et al.* (2004) argue, quite convincingly, that this maybe is not a good idea and they use an hierarchical Bayesian model to continue their analysis. The GAMLSS framework allows us to go someway along this path. For example, the function `random` allows the user to fit models with fitted values somewhere between those two extreme values of the null and the saturated model. The model fitted by `random` corresponds to the model $y_i \sim N(\mu_i, \sigma_i^2)$ with $\mu_i \sim N(0, \sigma_\gamma^2)$, where $\sigma_\gamma^2 = 1/\lambda$ and λ is the argument `lambda` in the function `random`. By varying `lambda` we can reproduce the saturate model (for $\lambda \rightarrow 0$) or the null model (for $\lambda \rightarrow \infty$). Equivalently we can vary the effective degrees of freedom from zero to 8 (the argument `df` in the function `random`) which is a one-to-one function of the parameter λ .

As an example of the use of the function `random` we are reproducing the null and saturate models and also fit a model with 4 degrees of freedom.

```
> m0 <- gamlss(estimate ~ random(school, df = 1e-05), sigma.fo = ~offset(log(sd)) -
+ 1, data = schools)
```

```
GAMLSS-RS iteration 1: Global Deviance = 59.4168
GAMLSS-RS iteration 2: Global Deviance = 59.3485
GAMLSS-RS iteration 3: Global Deviance = 59.3485
```

```
> m1 <- gamlss(estimate ~ random(school, df = 4), sigma.fo = ~offset(log(sd)) -
+ 1, data = schools)
```

```
GAMLSS-RS iteration 1: Global Deviance = 55.8353
GAMLSS-RS iteration 2: Global Deviance = 55.8787
GAMLSS-RS iteration 3: Global Deviance = 55.8787
```

```
> ms <- gamlss(estimate ~ random(school, df = 8), sigma.fo = ~offset(log(sd)) -
+ 1, data = schools)
```

```
GAMLSS-RS iteration 1: Global Deviance = 54.6414
GAMLSS-RS iteration 2: Global Deviance = 54.6414
```

In fact we can go a step further and see how the fitted values of `mu` vary when we change the degrees of freedom from 0 to 8. The plot on the left of figure 12.4 shows just that. It shows how the fitted values change from a global mean of 7.68 when the degrees of freedom are at zero to the actual data values where the degrees of freedom are at 8. The plot on the right of figure 12.4 (similar to the one given by Gelman *et al.* (2004) on page 142) shows how the fitted values for `mu` vary for different σ_γ , the random effects standard deviations. Both plots in figure 12.4 were produced using the following R commands.

```
> op <- par(mfrow = c(1, 2))
> con <- gamlss.control(trace = FALSE)
```

```

> iii <- seq(0.01, 8, length = 20)
> matfitmu <- matrix(NA, nrow = 20, ncol = 8)
> for (i in 1:20) {
+   mm <- gamlss(estimate ~ random(school, df = iii[i]), sigma.fo = ~offset(log(sd)) -
+   1, data = schools, control = con)
+   matfitmu[i, ] <- fitted(mm)
+ }
> plot(matfitmu[, 1] ~ iii, ylim = c(-4, 29), type = "l", col = 1,
+   xlab = "df", ylab = "fitted mu")
> for (j in 2:8) {
+   lines(iii, matfitmu[, j], col = j)
+ }
> text(rep(iii[14], 8), matfitmu[14, ], schools$school)
> tt <- seq(0.01, 50, length = 20)
> ttt <- 1/tt^2
> matfitmu <- matrix(NA, nrow = 20, ncol = 8)
> for (i in 1:20) {
+   mm <- gamlss(estimate ~ random(school, lambda = ttt[i]),
+   sigma.fo = ~offset(log(sd)) - 1, data = schools, control = con)
+   matfitmu[i, ] <- fitted(mm)
+ }
> plot(matfitmu[, 1] ~ tt, ylim = range(matfitmu), type = "l",
+   col = 1, xlab = "sqrt(1/lambda)", ylab = "fitted mu")
> for (j in 2:8) {
+   lines(tt, matfitmu[, j], col = j)
+ }
> text(rep(tt[14], 8), matfitmu[14, ], schools$school)
> par(op)

```

Now we fit a proper random effect model using the function `gamlssNP()` where the marginal distribution of the response variable is maximized instead of the joint likelihood for fixed λ_γ .

```

> mre <- gamlssNP(estimate ~ 1, sigma.start = sd, sigma.fix = TRUE,
+   data = schools, mixture = "gq", K = 10)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..
EM algorithm met convergence criteria at iteration    7
Global deviance trend plotted.

> mre

Mixing Family:  "NO Mixture with NO"

Fitting method: EM algorithm

Call:
gamlssNP(formula = estimate ~ 1, data = schools, K = 10, mixture = "gq",
  sigma.start = sd, sigma.fix = TRUE)

Mu Coefficients :

```

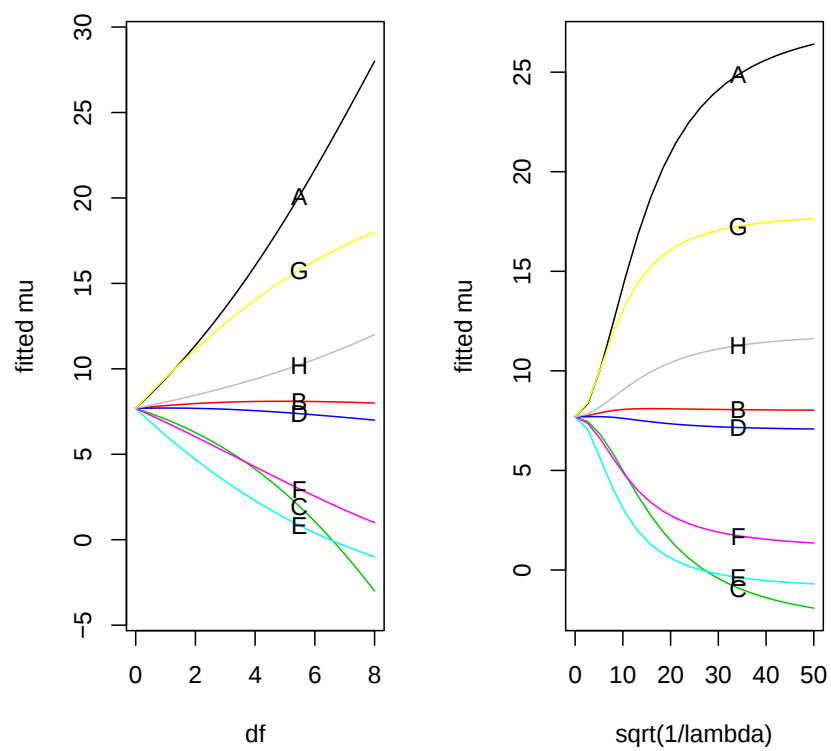


Figure 12.4: schools : plots of the fitted means for different degrees of freedom (on the left) and different standard deviations of the random effect (on the right)

```

(Intercept)          z
      7.6858      0.1275
Sigma Coefficients :
[1] NULL

Degrees of Freedom for the fit: 2 Residual Deg. of Freedom    6
Global Deviance:      59.3489
      AIC:      63.3489
      SBC:      63.5078

> mre$ebp

[1] 7.687309 7.685891 7.685161 7.685748 7.684096 7.684942 7.687518 7.686057

```

The fitted estimate of $\hat{\sigma}_\gamma = 0.1275$, very close to zero, corresponding to a high shrinking estimates of $\hat{\mu}_i$ towards the (weighed) mean value the response variable given by $\hat{\beta}_0 = 7.6858$.

Chapter 13

Random effect models at the factor level

13.1 Non-parametric random intercept in the predictor for μ

Let y_{ij} be observation i on subject j for $i = 1, 2, \dots, n_j$ and $j = 1, 2, \dots, J$. [For example the i^{th} student at j^{th} school, or the i^{th} measurement of j^{th} boy].

Let $y_{ij}|\gamma_j \sim D(\mu_{ij}, \sigma_{ij}, \nu_{ij}, \tau_{ij})$ where

$$\begin{aligned} g_1(\mu_{ij}) &= \mathbf{x}_{1ij}^T \boldsymbol{\beta}_1 + \gamma_j \\ g_2(\sigma_{ij}) &= \mathbf{x}_{2ij}^T \boldsymbol{\beta}_2 \\ g_3(\nu_{ij}) &= \mathbf{x}_{3ij}^T \boldsymbol{\beta}_3 \\ g_4(\tau_{ij}) &= \mathbf{x}_{4ij}^T \boldsymbol{\beta}_4. \end{aligned} \tag{13.1}$$

where γ_j is the random intercept for subject j . In matrix notation we can write (13.1) as

$$\begin{aligned} g_1(\boldsymbol{\mu}) &= \mathbf{X}_1 \boldsymbol{\beta}_1 + \mathbf{Z} \boldsymbol{\gamma} \\ g_2(\boldsymbol{\sigma}) &= \mathbf{X}_2 \boldsymbol{\beta}_2 \\ g_3(\boldsymbol{\nu}) &= \mathbf{X}_3 \boldsymbol{\beta}_3 \\ g_4(\boldsymbol{\tau}) &= \mathbf{X}_4 \boldsymbol{\beta}_4. \end{aligned} \tag{13.2}$$

where

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad (13.3)$$

is $n \times J$ matrix of dummy variables.

Assume γ_j comes from a 'non-parametric' random effect distribution given by

$$\gamma_j = u_k \quad \text{with probability} \quad \pi_k \quad \text{for} \quad k = 1, 2, \dots, K \quad (13.4)$$

and $j = 1, 2, \dots, J$. The u_k 's and π_k 's are assumed to be fixed unknown constants. Let $\mathbf{u}^T = (u_1, u_2, \dots, u_K)$ and $\boldsymbol{\pi}^T = (\pi_1, \pi_2, \dots, \pi_K)$ and the parameters $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\pi})$ where $\boldsymbol{\theta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4, \mathbf{u})$.

Let $\boldsymbol{\delta}_j^T = (\delta_{j1}, \delta_{j2}, \dots, \delta_{jK})$ be the missing indicator variable vector for subject j where

$$\delta_{jk} = \begin{cases} 1, & \text{if subject } j \text{ has random effect value } u_k, \text{ i.e. } \gamma_j = u_k \\ 0, & \text{otherwise} \end{cases} \quad (13.5)$$

Let $\boldsymbol{\delta}^T = (\boldsymbol{\delta}_1^T, \boldsymbol{\delta}_2^T, \dots, \boldsymbol{\delta}_J^T)$ combine all indicator variable vectors.

$$\begin{aligned} L_c = L_c(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) &= f(\mathbf{y}, \boldsymbol{\delta}) = f(\mathbf{y}|\boldsymbol{\delta})f(\boldsymbol{\delta}_j) \\ &= \prod_{j=1}^J \left\{ \left[\prod_{i=1}^{n_j} f(y_{ij}|\boldsymbol{\delta}_j) \right] f(\boldsymbol{\delta}_j) \right\} \\ &= \prod_{j=1}^J \left\{ \left[\prod_{i=1}^{n_j} \prod_{k=1}^K f_k(y_{ij})^{\delta_{jk}} \right] \prod_{k=1}^K \pi_k^{\delta_{jk}} \right\} \\ &= \prod_{j=1}^J \prod_{k=1}^K \left\{ \left[\prod_{i=1}^{n_j} f_k(y_{ij})^{\delta_{jk}} \right] \pi_k^{\delta_{jk}} \right\} \end{aligned} \quad (13.6)$$

Hence the complete log likelihood is given by

$$\ell_c = \ell_c(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) = \sum_{j=1}^J \sum_{k=1}^K \delta_{jk} \log \pi_k + \sum_{j=1}^J \sum_{k=1}^K \sum_{i=1}^{n_j} \delta_{jk} \log f_k(y_{ij}) \quad (13.7)$$

The marginal likelihood $L(\boldsymbol{\psi}, \mathbf{y})$ is maximized with respect to $\boldsymbol{\psi}$ by using the EM algorithm with missing variable $\boldsymbol{\delta}$

E-step

The E step finds the conditional expectation of the complete log likelihood given $\mathbf{y}$ and given the current parameter estimates $\hat{\boldsymbol{\psi}}^{(r)}$ at iteration r , i.e.

$$Q = \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^{n_j} \hat{w}_{jk}^{(r+1)} \log f_k(y_{ij}) + \sum_{k=1}^K \sum_{j=1}^J \hat{w}_{jk}^{(r+1)} \log \pi_k \quad (13.8)$$

where

$$\begin{aligned} \hat{w}_{jk}^{(r+1)} &= E \left[\delta_{jk} | \mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)} \right] \\ &= p(\delta_{jk} = 1 | \mathbf{y}, \hat{\boldsymbol{\psi}}^{(r)}) \\ &= \frac{f(y_{ij} | \delta_{jk} = 1, \hat{\boldsymbol{\psi}}^{(r)}) p(\delta_{jk} = 1 | \hat{\boldsymbol{\psi}}^{(r)})}{f(\mathbf{y} | \hat{\boldsymbol{\psi}}^{(r)})} \\ &= \frac{\hat{\pi}_k^{(r)} \prod_{i=1}^{n_j} f_k(y_{ij} | \hat{\boldsymbol{\theta}}^{(r)})}{\sum_{k=1}^K \left[\hat{\pi}_k^{(r)} \prod_{i=1}^{n_j} f_k(y_{ij} | \hat{\boldsymbol{\theta}}^{(r)}) \right]} \\ &= \frac{\hat{\pi}_k^{(r)} m_{jk}}{\sum_{k=1}^K \hat{\pi}_k^{(r)} m_{jk}} \end{aligned} \quad (13.9)$$

where $m_{jk} = \prod_{i=1}^{n_j} f_k(y_{ij})$.

M-step

The first term in (13.8) is just a weighted log likelihood for a GAMLSS model for $n \times K$ observations (where $n = \sum_{j=1}^J n_j$) with dependent variable $y_{ijk} = y_{ij}$ and predictor for μ given by $g_1(\mu_{ilk}) = \mathbf{x}_{1ij}^T \boldsymbol{\beta}_1 + u_k$ with weights $w_{ijk} = \hat{w}_{jk}^{(r+1)}$ for $i = 1, 2, \dots, n_j$, $j = 1, 2, \dots, J$ and $k = 1, 2, \dots, K$.

13.1.1 Summary of the $(r+1)^{th}$ iteration of the EM algorithm

E-step Replace δ_{jk} in (13.7) by $\hat{w}_{jk}^{(r+1)}$ [its current conditional expectation given $\mathbf{y}$ and given the current estimate $\hat{\boldsymbol{\psi}}^{(r)}$ at iteration r] obtained from (13.9) for $k = 1, 2, \dots, K$ and $j = 1, 2, \dots, J$ to give (13.8), where $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\pi})$ and $\boldsymbol{\theta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4, \mathbf{u})$.

M-step

- (1) Fit a single GAMLSS model to the expanded data given in Table 13.1 using expanded dependent variable $\mathbf{y}_e$ with explanatory variables $\mathbf{x}_1$ and factor MASS (with K levels) in the predictor for μ and $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$ in the predictors for σ , ν and τ respectively, using prior weights $\hat{\mathbf{w}}^{(r+1)}$.
- (2) $\hat{\boldsymbol{\pi}}^{(r+1)} = \frac{1}{J} \sum_{j=1}^J \hat{w}_{jk}^{(r+1)}$

Note that explanatory variables may apply at either the observational level or at the subject level. If there are explanatory variables at the subject level, then non-parametric random coefficients for these variables can be fitted by including interaction terms between these variables and the factor MASS in predictor models for distributional parameters.

Table 13.1: Table showing the expansion of data for fitting the Gauss quadrature model at M step (2) of the EM algorithm. (The column headed by k indicates the factor MASS).

k	i	j	$\mathbf{y}_e$	$\mathbf{x}_{1e}$	$\mathbf{x}_{2e}$	$\mathbf{x}_{3e}$	$\mathbf{x}_{4e}$	z	weights $\hat{\mathbf{w}}^{(r+1)}$
1	1	1	$\mathbf{y}_1$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_1	$\hat{\mathbf{w}}_{11}^{(r+1)}$
1	2	1						z_1	$\hat{\mathbf{w}}_{11}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$						$\vdots$	$\vdots$
1	n_1	1						z_1	$\hat{\mathbf{w}}_{11}^{(r+1)}$
1	1	2	$\mathbf{y}_2$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_1	$\hat{\mathbf{w}}_{21}^{(r+1)}$
1	2	2						z_1	$\hat{\mathbf{w}}_{21}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$						$\vdots$	$\vdots$
1	n_2	2						z_1	$\hat{\mathbf{w}}_{21}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1	1	J	$\mathbf{y}_J$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_1	$\hat{\mathbf{w}}_{j1}^{(r+1)}$
1	2	J						z_1	$\hat{\mathbf{w}}_{j1}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$						$\vdots$	$\hat{\mathbf{w}}_{j1}^{(r+1)}$
1	n_J	J						z_1	$\hat{\mathbf{w}}_{j1}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
K	1	1	$\mathbf{y}_1$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_K	$\hat{\mathbf{w}}_{1K}^{(r+1)}$
K	2	1						z_K	$\hat{\mathbf{w}}_{1K}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$						$\vdots$	$\vdots$
K	n_1	1						z_K	$\hat{\mathbf{w}}_{1K}^{(r+1)}$
K	1	2	$\mathbf{y}_2$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_K	$\hat{\mathbf{w}}_{2K}^{(r+1)}$
K	2	2						z_K	$\hat{\mathbf{w}}_{2K}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$						$\vdots$	$\vdots$
K	n_2	2						z_K	$\hat{\mathbf{w}}_{2K}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
K	1	J	$\mathbf{y}_J$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	z_K	$\hat{\mathbf{w}}_{jK}^{(r+1)}$
K	2	J						z_K	$\hat{\mathbf{w}}_{jK}^{(r+1)}$
$\vdots$	$\vdots$	$\vdots$						$\vdots$	$\hat{\mathbf{w}}_{jK}^{(r+1)}$
K	n_J	J						z_K	$\hat{\mathbf{w}}_{jK}^{(r+1)}$

13.2 Examples

13.2.1 Binomial response with random intercept and slope: A meta analysis

The data here are coming from a statistical meta analysis problem. In meta analysis we combine the evidence from different studies to obtain an overall treatment effect. The data from Silagy *et al.* (2003) consist of different clinical trials of nicotine replacement therapy for smoking cessation. In each trial the patient was randomized into a treatment or control group. The treatment group were given a nicotine gum. In the majority of studies the control group receive the same appearance gum but without the ingredients but in some they were given no gum. The outcome, whether the participant is smoking or not, was observed after six months. The data set `meta`, of dimensions 54×6 consist of the following variables. The data shown in Table 13.2.1 were previously analysed by Aitkin (1999) and by Skrondal and Rebe-Hesketh (2004) Chapter 9.

- `studyname` : the name of the place of the different studies (note that the values of `studyname` is the same for studies at the same place in different years)
- `year` : the year of the study
- `d` : the number of quitters (non-smokers) after six months
- `n` : the total number of participants in the study
- `fac` : a factor with two levels indicating whether control or treatment
- `study` : numeric from 1 to 27 indicating the different studies (that is, the interaction of `studyname` and `year`).

First we input the data and display it graphically by plotting the sample logit i.e $\text{logit}[d/(n-d)]$, against the treatment factor, `fac`, for each of the 27 studies.

```
> library(gamlss.mx)
> meta <- dget("C:/gamlss/data/meta")
> str(meta)

'data.frame':      54 obs. of  6 variables:
 $ studyname: Factor w/ 23 levels "Blondal","Campbell",...: 1 2 3 4 5 6 7 8 8 8 ...
 $ year      : num  1989 1991 1982 1982 1989 ...
 $ d         : num  24 21 23 15 5 17 6 10 14 28 ...
 $ n         : num  90 105 50 172 38 203 46 36 68 103 ...
 $ fac       : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ study     : num   1 2 3 4 5 6 7 8 9 10 ...

> logitd <- with(meta, log(d/(n - d)))
> par.plot(logitd ~ unclass(fac), subjects = factor(study), data = meta)
```

This is a two-level data with `study` as the upper level and patients as the lower level. Following Aitkin (1999) we fit a binomial model with non parametric `study` random effects terms for each of the two levels of `fac`. We set the number of non parametric mixture components to 10 as suggested by Skrondal and Rebe-Hesketh (2004) page 304.

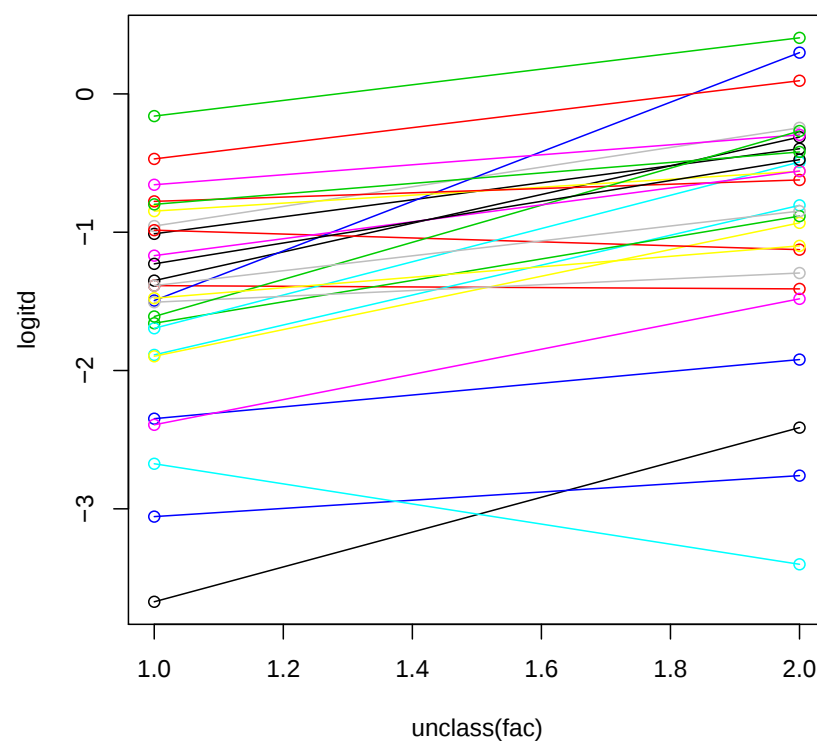


Figure 13.1: Plot of the sample log odds ratios for the different 27 smoking cessation studies

```

> me1 <- gamlssNP(cbind(d, n - d) ~ fac, random = ~fac | study,
+   K = 10, family = BI, data = meta, tol = 0.1)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..16 ..17 ..18 ..19 ..20
EM algorithm met convergence criteria at iteration   37
Global deviance trend plotted.
EM Trajectories plotted.

> summary(me1)

*****
Family:  c("BI Mixture with NP", "Binomial Mixture with NP")

Call:  gamlssNP(formula = cbind(d, n - d) ~ fac, random = ~fac | study,
  family = BI, data = meta, K = 10, tol = 0.1)

Fitting method: RS()

-----
Mu link function:  logit
Mu Coefficients:

```

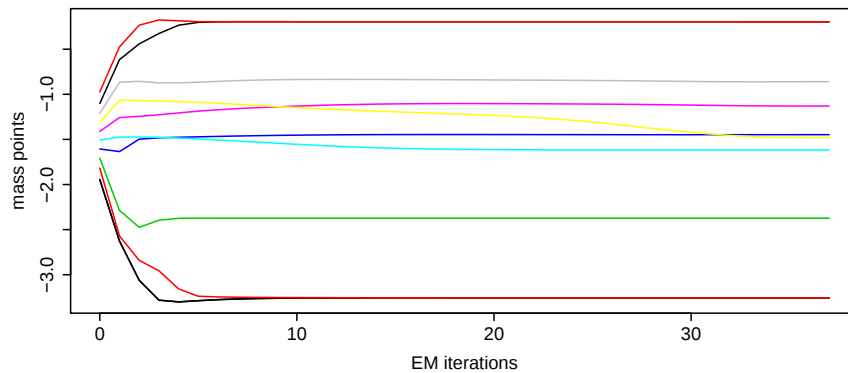
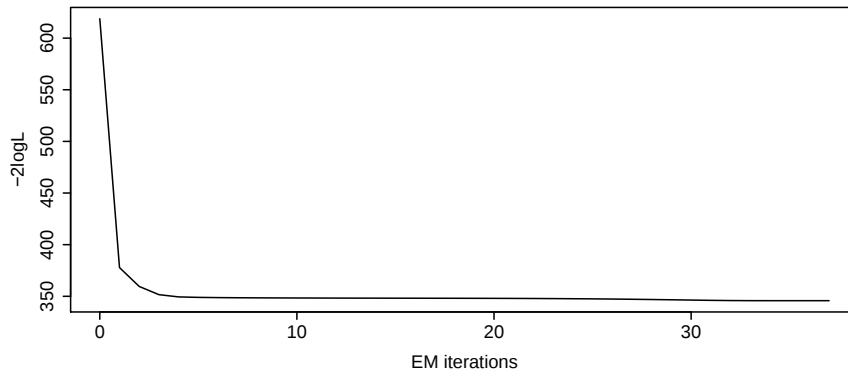
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-3.255e+00	0.6245	-5.213e+00	2.689e-07
fac2	5.758e-01	0.8296	6.940e-01	4.880e-01
MASS2	3.062e-08	0.7468	4.100e-08	1.000e+00
MASS3	8.824e-01	0.6511	1.355e+00	1.759e-01
MASS4	1.808e+00	0.6302	2.869e+00	4.289e-03
MASS5	1.638e+00	0.6478	2.529e+00	1.173e-02
MASS6	2.125e+00	0.6325	3.360e+00	8.363e-04
MASS7	1.776e+00	0.7076	2.510e+00	1.238e-02
MASS8	2.396e+00	0.6365	3.764e+00	1.861e-04
MASS9	3.059e+00	0.7051	4.338e+00	1.725e-05
MASS10	3.059e+00	0.7790	3.927e+00	9.759e-05
fac2:MASS2	-5.178e-08	0.9921	-5.219e-08	1.000e+00
fac2:MASS3	1.914e-01	0.8570	2.234e-01	8.233e-01
fac2:MASS4	-3.723e-01	0.8377	-4.444e-01	6.570e-01
fac2:MASS5	1.764e-01	0.8556	2.062e-01	8.368e-01
fac2:MASS6	5.776e-02	0.8400	6.876e-02	9.452e-01
fac2:MASS7	1.160e+00	0.9348	1.241e+00	2.152e-01
fac2:MASS8	-1.159e-01	0.8441	-1.373e-01	8.908e-01
fac2:MASS9	-5.297e-03	0.9528	-5.559e-03	9.956e-01
fac2:MASS10	-5.297e-03	1.0641	-4.978e-03	9.960e-01

```

-----
No. of observations in the fit:  540
Degrees of Freedom for the fit:  29
  Residual Deg. of Freedom:  25
                        at cycle:  1

```

Global Deviance: 345.7512
 AIC: 403.7512
 SBC: 461.4318



The fitted model is $y_{ij} \sim BI(N_{ij}, \mu_{ij})$ where $\text{logit}(\mu_{ij}) = \gamma_{0j} + \gamma_{ij}$ (if $i = 2$) for $i = 1, 2$ for **fac** level and $j = 1, 2, \dots, 27$ for **study** level where $(\gamma_{0j}, \gamma_{1j}) = (u_{0k}, u_{1k})$ with probability π_k for $k = 1, 2, \dots, 10$ and $j = 1, 2, \dots, 27$.

The mass points and their probability are saved in `me1$mass.points` and `me1$prob` and can be display or plotted. Here we use the purpose made function `pmasses()` to plot them in three dimensions. The fitted non parametric distribution for γ_0 , i.e. probabilities π_k against the values of u_{0k} for $k = 1, 2, \dots, 10$, are plotted as `mass.points[,1]` in Figure 13.2(b). The fitted non parametric distribution for γ_1 , i.e. probabilities π_k against the values of u_{1k} for $k = 1, 2, \dots, 10$, are plotted as `mass.points[,2]` in Figure 13.2(d). Figure 13.2(a) and (c) plot the fitted non-parametric bivariate distribution for (γ_0, γ_1) , i.e. probabilities π_k against the bivariate values (u_{0k}, u_{1k}) , for $k = 1, 2, \dots, 10$.

```
> source("C:/GAMLSS/New Functions/mixtures/plot3d.R")
> op <- par(mfrow = c(2, 2))
> pmasses(me1$mass.points[, 1], me1$mass.points[, 2], me1$prob,
+         main = "(a)")
> plot(me1$mass.points[, 1], me1$prob, type = "h", main = "(b)")
```

```

> points(me1$mass.points[, 1], me1$prob, )
> pmasses(me1$mass.points[, 1], me1$mass.points[, 2], me1$prob,
+         theta = 90, phi = 20, main = "(c)")
> plot(me1$mass.points[, 2], me1$prob, type = "h", main = "(d)")
> points(me1$mass.points[, 2], me1$prob, )
> par(op)

```

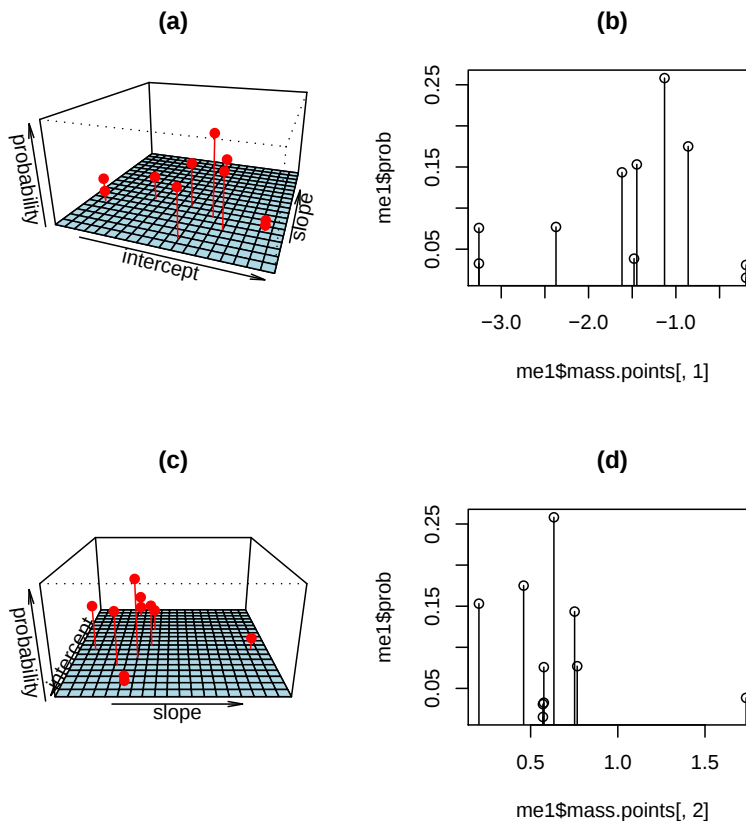


Figure 13.2: Plot of the joint (a) and (c) and marginal (b) and (d) non-parametric distribution for γ_0, γ_1

The means and standard deviations and the correlation of the fitted non-parametric distribution for (γ_0, γ_1) can be found using the commands below. The means (given by \$center) are -1.1498 and 0.598 the standard deviations (given by the square root of the diagonal of the covariance matrix) are 0.8300 and 0.3138 for γ_0 and γ_1 respectively and the correlation is -0.1173 . Hence the overall estimate of the treatment effect (on the logit scale) is 0.598

```

> vM <- cov.wt(me1$mass.points, me1$prob)
> vM
$cov
      [,1]      [,2]

```

```

[1,] 0.68894082 -0.03054541
[2,] -0.03054541 0.09848739

$center
[1] -1.4978969 0.5978631

$n.obs
[1] 10

$wt
      MASS1      MASS2      MASS3      MASS4      MASS5      MASS6      MASS7
0.03262593 0.07586472 0.07704748 0.15312586 0.14349824 0.25814324 0.03851473
      MASS8      MASS9      MASS10
0.17507287 0.03085242 0.01525452

> cov2cor(vM$cov)

      [,1]      [,2]
[1,] 1.0000000 -0.1172640
[2,] -0.1172640 1.0000000

> sqrt(diag(vM$cov))

[1] 0.8300246 0.3138270

```

13.3 Normal random effect at the factor level in the predictor for μ

In model (13.1) let $\gamma_j \sim N(0, \sigma_\gamma^2)$ independently for $j = 1, 2, \dots, J$. Let $\gamma_j = \sigma_\gamma Z_j$ then $Z_j \sim N(0, 1)$ independently for $j = 1, 2, \dots, J$. Gaussian quadrature effectively approximates the continuous $N(0, 1)$ distribution for each Z_j by a discrete distribution, i.e.

$$Z_j = z_k \quad \text{with probability} \quad \pi_k \quad \text{for} \quad i = 1, 2, \dots, K \quad (13.10)$$

Missing indicator variable vectors $\delta_j^T = (\delta_{j1}, \delta_{j2}, \dots, \delta_{jk})$ for $j = 1, 2, \dots, J$ are set up as in Section 13.1 and combined in $\delta^T = (\delta_1^T, \delta_2^T, \dots, \delta_J^T)$. The EM algorithm maximizes the likelihood of the data with respect to the parameters $\psi = (\beta_1, \beta_2, \beta_3, \beta_4, \sigma_\gamma)$. The complete log likelihood of $(\mathbf{y}, \delta)$ is given in (13.7)

13.3.1 Summary of the $(r+1)^{th}$ iteration of the EM algorithm

E-step Replace δ_{jk} in (13.7) by $\hat{w}_{jk}^{(r+1)}$ obtained from $\hat{w}_{jk}^{(r+1)} = \frac{\pi_k m_{jk}}{\sum_{k=1}^K \pi_k m_{jk}}$ where $m_{jk} =$

$$\prod_{i=1}^{n_j} f_k(y_{ij} | \hat{\psi}^{(r)}), \text{ for } j = 1, 2, \dots, J \text{ and } k = 1, 2, \dots, K \text{ to give (13.8).}$$

M-step

- (1) Fit a single GAMLSS model to an expanded data given in Table 13.1 using expanded dependent variable $\mathbf{y}_e$, explanatory variables $\mathbf{x}_{1e}$ and $\mathbf{z}$ for μ and $\mathbf{x}_{2e}$, $\mathbf{x}_{3e}$ and $\mathbf{x}_{4e}$ for σ , ν and τ respectively, using weights $w_{ijk} = w_{jk}^{(r+1)}$ for all i, j and k . Note that since the π 's are fixed and known they do not need to be estimated.

Note also that the normal random effect could have been included (at a factor level) in any one of the predictors for the distribution parameters μ , σ , ν and τ . However if there are more than one (say q) normal random effects in the model, then the Gaussian quadrature method becomes infeasible, because the data need to be expanded k^q times. Rigby and Stasinopoulos (2005) provide a general procedure for maximizing approximately the marginal likelihood of the data over the parameters, where there are multiple (possibly multivariate) normal random effects in the predictors for μ , σ , ν and τ . The only constraint is that the normal random effects for the predictors for μ , σ , ν and τ . are assumed to be independent. This more general approach is discussed in Chapter 14

13.4 Examples

13.4.1 Binary response with random intercept: Respiratory infection in children

This is cohort study of 275 Indonesian preschool children, ($J = 1, 2, \dots, 275$), examined on up to six, ($i = 1, 2, \dots, 6$), consecutive quarters for the presence of respiratory infection. Sommer *et al.* (1983) describe the study, while Zeger and Karim (1991) and Diggle *et al.* (2002) among others analyzed it. The data were also analyzed by Skrondal and Rabe-Hesketh (2004), Section 9.2. The dimension of the data frame containing the data is 1200×14 , with the following 14 variables included.

- id : a factor with 275 levels identifying the individual children
- time : the binary response variable identifying the presence of of respiratory infection
- resp : a vector of ones (not used further)
- age : the age in months (centered around 36)
- xero : a dummy variable for the present of xerophthalmia
- cosine : a cosine term of the annual cycle
- sine : a sin term of the annual cycle
- female : a gender dummy variable, 1=female.
- height : height for age as percent of the National Center for health Statistics standard centered at 90%
- stunted : a dummy variable whether below 85% in height for age
- time.1 : the time that the children has been examine, 1 to 6
- age1 : the age of the child at the fist time of examination
- season : a variable taking the values 1,2,3,4 indicating the season
- time2 : the time in months

Here we fit a binomial random intercept model, $BI(\text{logit}(\mu) = \text{age} + \text{xero} + \text{female} + \text{cosine} + \text{sine} + \text{height} + \text{stunted} + z)$ where $z \sim NO(0, \sigma_z)$ as in Skrondal and Rabe-Hesketh (2004) page 286, hence $Y_{ij}|\gamma_j \sim BI(\mu_{ij})$ where $\text{logit}(\mu_{ij}) = \text{age}_j + \text{xero}_j + \text{female}_j + \text{cosine}_i + \text{sine}_i + \text{height}_j + \text{stunted}_j + \gamma_j$ where $\gamma_j \sim N(0, \sigma_\gamma^2)$ independently for $j = 1, 2, \dots, 275$.

```
> library(gamlss.mx)
> xe <- dget("C:/gamlss/data/xero")
> str(xe)

'data.frame':      1200 obs. of  14 variables:
 $ id      : Factor w/ 275 levels "121013","121113",...: 1 1 1 1 1 1 2 2 2 2 ...
 $ time    : num  0 0 0 0 1 0 0 0 0 0 ...
 $ resp    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ age     : num  31 34 37 40 43 46 -9 -6 -3 0 ...
 $ xero    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ cosine  : num  -1 0 1 0 -1 0 -1 0 1 0 ...
 $ sine    : num  0 -1 0 1 0 -1 0 -1 0 1 ...
 $ female  : num  0 0 0 0 0 0 1 1 1 1 ...
 $ height  : num  -3 -3 -2 -2 -2 -3 2 0 -1 -2 ...
 $ stunted : num  0 0 0 0 0 0 0 0 0 0 ...
 $ time.1  : num  1 2 3 4 5 6 1 2 3 4 ...
 $ age1    : num  31 31 31 31 31 31 -9 -9 -9 -9 ...
 $ season  : num  2 3 4 1 2 3 2 3 4 1 ...
 $ time2   : num  1 4 9 16 25 36 1 4 9 16 ...

> m1 <- gamlssNP(time ~ age + xero + female + cosine + sine + height +
+   stunted, random = ~1 | id, K = 20, mixture = "gq", data = xe,
+   family = BI)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..
EM algorithm met convergence criteria at iteration 13
Global deviance trend plotted.

> summary(m1)

*****
Family: "BI Mixture with NO"

Call: gamlssNP(formula = time ~ age + xero + female + cosine + sine +
  height + stunted, random = ~1 | id, family = BI, data = xe,
  K = 20, mixture = "gq")

Fitting method: RS()

-----
Mu link function: logit
Mu Coefficients:

```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.66973	0.186972	-14.2788	4.585e-46
age	-0.03396	0.006656	-5.1030	3.368e-07

xero	0.62599	0.452307	1.3840	1.664e-01
female	-0.43572	0.226976	-1.9197	5.491e-02
cosine	-0.59358	0.172674	-3.4376	5.880e-04
sine	-0.16484	0.173462	-0.9503	3.420e-01
height	-0.04791	0.024514	-1.9542	5.069e-02
stunted	0.20163	0.405571	0.4971	6.191e-01
z	0.79894	0.114487	6.9784	3.063e-12

No. of observations in the fit: 24000

Degrees of Freedom for the fit: 9

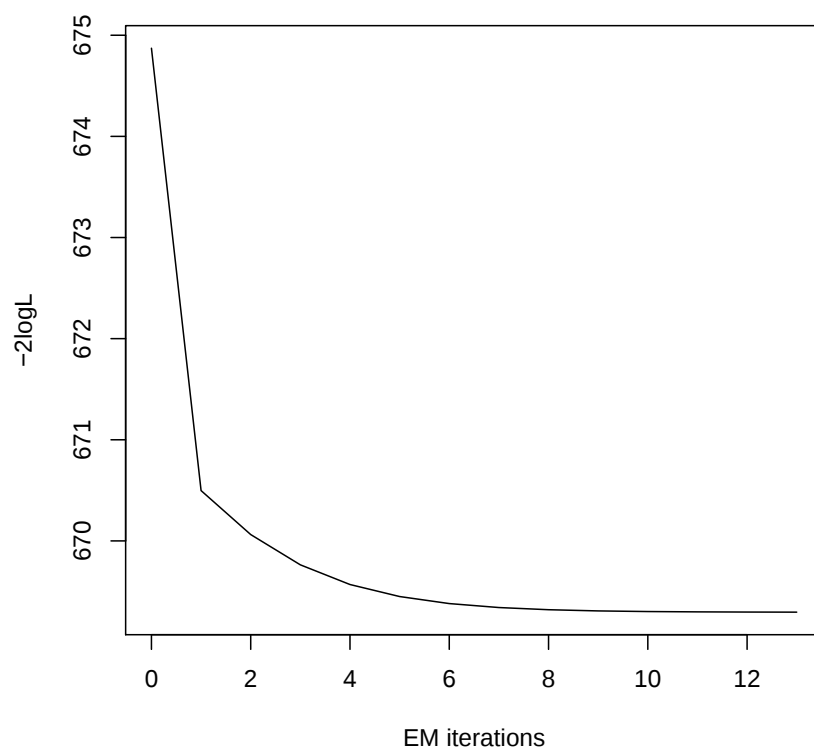
Residual Deg. of Freedom: 1191

at cycle: 1

Global Deviance: 669.2956

AIC: 687.2956

SBC: 733.1063



The fitted `m1` has been obtained by maximizing the marginal distribution of y i.e. $f_Y(y|\theta, \sigma_\gamma) = \int f_Y(y|\theta, \gamma) f_\gamma(\gamma|\sigma_\gamma) d\gamma$. A very similar model can be obtained by using an ordinary GAMLSS

model with a additive term `random()` for the factor `id`. In the later cases the (joint) likelihood maximized is $f_Y(y|\theta, \gamma)f_\gamma(\gamma|\hat{\sigma}_\gamma)$ is maximized for a fixed σ_γ or equivalently for a fixed $\lambda_\gamma = 1/\sigma_\gamma^2$. The Empirical Bayes Predictions (EBP) for μ of the marginal model `m1` should be almost identical to the the predictor for μ of the ordinary GAMLSS model. Also the beta coefficients for both models should be very similar but not identical. We now fit the ordinary GAMLSS models for comparison. The Empirical Bayes Predictions for `m2` are stored in `m1$ebp`. Compare this to the predictor for μ in model `m2` in which λ is fixed equal to $1/\hat{\sigma}_\gamma^2$ where $\hat{\sigma}_\gamma = 0.79894$ is obtained from the coefficient of `z` in the fitted model `m1` [given by `summary(m1)`].

```
> m2 <- gamlss(time ~ age + xero + female + cosine + sine + height +
+   stunted + random(id, lambda = 1/(0.79894)^2), data = xe,
+   family = BI)
```

GAMLSS-RS iteration 1: Global Deviance = 577.9343

GAMLSS-RS iteration 2: Global Deviance = 577.9343

```
> summary(m2)
```

```
*****
```

```
Family:  c("BI", "Binomial")
```

```
Call:  gamlss(formula = time ~ age + xero + female + cosine + sine +
  height + stunted + random(id, lambda = 1/(0.79894)^2), family = BI,
  data = xe)
```

Fitting method: RS()

```
-----
Mu link function:  logit
```

```
Mu Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.47521	0.172544	-14.3454	4.741e-43
age	-0.03176	0.006505	-4.8825	1.196e-06
xero	0.58781	0.439960	1.3361	1.818e-01
female	-0.40644	0.221719	-1.8331	6.704e-02
cosine	-0.56829	0.168932	-3.3640	7.935e-04
sine	-0.15796	0.170134	-0.9284	3.534e-01
height	-0.04516	0.023892	-1.8902	5.898e-02
stunted	0.18956	0.395971	0.4787	6.322e-01

```
-----
No. of observations in the fit:  1200
```

```
Degrees of Freedom for the fit:  52.30702
```

```
Residual Deg. of Freedom:  1147.693
```

```
at cycle:  2
```

```
Global Deviance:      577.9343
```

```
AIC:      682.5483
```

```
SBC:      948.795
```

```
*****
```

	studyname	year	d	n	fac	study
1	Blondal	1989.00	24.00	90.00	1	1.00
2	Campbell	1991.00	21.00	105.00	1	2.00
3	Fagerstrom	1982.00	23.00	50.00	1	3.00
4	Fee	1982.00	15.00	172.00	1	4.00
5	Garcia	1989.00	5.00	38.00	1	5.00
6	Garvey	2000.00	17.00	203.00	1	6.00
7	Gross	1995.00	6.00	46.00	1	7.00
8	Hall	1985.00	10.00	36.00	1	8.00
9	Hall	1987.00	14.00	68.00	1	9.00
10	Hall	1996.00	28.00	103.00	1	10.00
11	Hjalmarson	1984.00	16.00	100.00	1	11.00
12	Huber	1988.00	11.00	60.00	1	12.00
13	Jarvis	1982.00	9.00	58.00	1	13.00
14	Jensen	1991.00	28.00	82.00	1	14.00
15	Killen	1984.00	6.00	20.00	1	15.00
16	Killen	1990.00	112.00	617.00	1	16.00
17	Malcolm	1980.00	3.00	121.00	1	17.00
18	McGovern	1992.00	40.00	127.00	1	18.00
19	Nakamura	1990.00	5.00	30.00	1	19.00
20	Niaura	1994.00	4.00	89.00	1	20.00
21	Niaura	1999.00	2.00	31.00	1	21.00
22	Pirie	1992.00	50.00	211.00	1	22.00
23	Puska	1979.00	21.00	113.00	1	23.00
24	Schneider	1985.00	6.00	30.00	1	24.00
25	Tonnesen	1988.00	12.00	53.00	1	25.00
26	Villa	1999.00	10.00	26.00	1	26.00
27	Zelman	1992.00	18.00	58.00	1	27.00
28	Blondal	1989.00	37.00	92.00	2	1.00
29	Campbell	1991.00	21.00	107.00	2	2.00
30	Fagerstrom	1982.00	30.00	50.00	2	3.00
31	Fee	1982.00	23.00	180.00	2	4.00
32	Garcia	1989.00	21.00	68.00	2	5.00
33	Garvey	2000.00	75.00	405.00	2	6.00
34	Gross	1995.00	37.00	131.00	2	7.00
35	Hall	1985.00	18.00	41.00	2	8.00
36	Hall	1987.00	30.00	71.00	2	9.00
37	Hall	1996.00	24.00	98.00	2	10.00
38	Hjalmarson	1984.00	31.00	106.00	2	11.00
39	Huber	1988.00	31.00	54.00	2	12.00
40	Jarvis	1982.00	22.00	58.00	2	13.00
41	Jensen	1991.00	90.00	211.00	2	14.00
42	Killen	1984.00	16.00	44.00	2	15.00
43	Killen	1990.00	129.00	600.00	2	16.00
44	Malcolm	1980.00	6.00	73.00	2	17.00
45	McGovern	1992.00	51.00	146.00	2	18.00
46	Nakamura	1990.00	13.00	30.00	2	19.00
47	Niaura	1994.00	5.00	84.00	2	20.00
48	Niaura	1999.00	1.00	31.00	2	21.00
49	Pirie	1992.00	75.00	206.00	2	22.00
50	Puska	1979.00	29.00	116.00	2	23.00
51	Schneider	1985.00	9.00	30.00	2	24.00
52	Tonnesen	1988.00	23.00	60.00	2	25.00
53	Villa	1999.00	11.00	21.00	2	26.00
54	Zelman	1992.00	23.00	58.00	2	27.00

Table 13.2: The meta analysis data

Chapter 14

General random effect models

14.1 Random effects model in gamlss

Rigby and Stasinopoulos (2005) define the original formulation of a GAMLSS random effects model as follows. Let $\mathbf{y}^T = (y_1, y_2, \dots, y_n)$ be the n length vector of the response variable. Also for $k = 1, 2, 3, 4$, let $g_k(\cdot)$ be known monotonic link functions relating the distribution parameters to explanatory variables by

$$\begin{aligned} g_1(\boldsymbol{\mu}) &= \boldsymbol{\eta}_1 = \mathbf{X}_1\boldsymbol{\beta}_1 + \sum_{j=1}^{J_1} \mathbf{Z}_{j1}\boldsymbol{\gamma}_{j1} \\ g_2(\boldsymbol{\sigma}) &= \boldsymbol{\eta}_2 = \mathbf{X}_2\boldsymbol{\beta}_2 + \sum_{j=1}^{J_2} \mathbf{Z}_{j2}\boldsymbol{\gamma}_{j2} \\ g_3(\boldsymbol{\nu}) &= \boldsymbol{\eta}_3 = \mathbf{X}_3\boldsymbol{\beta}_3 + \sum_{j=1}^{J_3} \mathbf{Z}_{j3}\boldsymbol{\gamma}_{j3} \\ g_4(\boldsymbol{\tau}) &= \boldsymbol{\eta}_4 = \mathbf{X}_4\boldsymbol{\beta}_4 + \sum_{j=1}^{J_4} \mathbf{Z}_{j4}\boldsymbol{\gamma}_{j4}. \end{aligned} \tag{14.1}$$

where $\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\nu}, \boldsymbol{\tau}$ and $\boldsymbol{\eta}_k$ are vectors of length n , $\boldsymbol{\beta}_k^\top = (\beta_{1k}, \beta_{2k}, \dots, \beta_{J'_k k})$ is a parameter vector of length J'_k , $\mathbf{X}_k$ is a known design matrix of order $n \times J'_k$, $\mathbf{Z}_{jk}$ is a fixed known $n \times q_{jk}$ design matrix and $\boldsymbol{\gamma}_{jk}$ is a q_{jk} dimensional random variable which is assumed to distributed as $\boldsymbol{\gamma}_{jk} \sim N_{q_{jk}}(\mathbf{0}, \mathbf{G}_{jk}^{-1})$, where $\mathbf{G}_{jk}^{-1}$ is the (generalized) inverse of a $q_{jk} \times q_{jk}$ symmetric matrix $\mathbf{G}_{jk} = \mathbf{G}_{jk}(\boldsymbol{\lambda}_{jk})$ which may depend on a vector of hyperparameters $\boldsymbol{\lambda}_{jk}$, and where if $\mathbf{G}_{jk}$ is singular then $\boldsymbol{\gamma}_{jk}$ is understood to have an improper prior density function proportional to $\exp\left(-\frac{1}{2}\boldsymbol{\gamma}_{jk}^T \mathbf{G}_{jk} \boldsymbol{\gamma}_{jk}\right)$.

The model in (2.1) allows the user to model each distributional parameter as linear function of explanatory variables and/or linear functions of stochastic variables (random effects). We have to restrict the stochastic variables $\boldsymbol{\gamma}$'s to be normally distributed since that simplifies some of the equations in the fitting procedure of the model. This restriction can be lifted in general but we are not going to attempt this here.

14.2 Posterior mode estimation of the parameters β and random effects γ given λ

For the GAMLSS model (14.1) an empirical Bayesian argument can be used to obtain maximum *a posteriori* (MAP), or posterior mode, estimation (see Berger, 1985) of both the β_k 's and the γ_{jk} 's assuming normal, possibly improper, priors for the γ_{jk} 's.

We show below that this is equivalent to maximizing a penalized likelihood. To show this we shall use arguments developed in the statistical literature by Wahba (1978), Silverman (1985), Green (1985), Kohn and Ansley (1988), Speed (1991), Green and Silverman (1994), Verbyla *et al.* (1999), Hastie and Tibshirani (2000) and Fahrmeir and Lang (2001).

The components of a GAMLSS model (14.1) are

- $\mathbf{y}$, the response vector of length n
- $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p)$, design matrices for fixed explanatory variables
- $\beta^T = (\beta_1^T, \dots, \beta_p^T)$, the beta parameters in the model
- $\mathbf{Z} = (\mathbf{Z}_{11}, \mathbf{Z}_{21}, \dots, \mathbf{Z}_{J_1 1}, \dots, \mathbf{Z}_{1p}, \mathbf{Z}_{2p}, \dots, \mathbf{Z}_{J_p p})$, design matrices for the random effects
- $\gamma = (\gamma_{11}, \gamma_{21}, \dots, \gamma_{J_1 1}, \dots, \gamma_{1p}, \gamma_{2p}, \dots, \gamma_{J_p p})$, random effects
- $\lambda = (\lambda_{11}, \lambda_{21}, \dots, \lambda_{J_1 1}, \dots, \lambda_{1p}, \lambda_{2p}, \dots, \lambda_{J_p p})$, hyperparameters

Assume that the joint distribution of all the components in (2.1) is given by

$$f(\mathbf{y}, \beta, \gamma, \lambda) = f(\mathbf{y}|\beta, \gamma) f(\gamma|\lambda) f(\lambda) f(\beta) \quad (14.2)$$

where $f(\mathbf{y}|\beta, \gamma)$ and $f(\gamma|\lambda)$ are conditional distributions for $\mathbf{y}$ and γ and $f(\lambda)$ and $f(\beta)$ are appropriate priors for λ and β respectively and $\mathbf{X}$ and $\mathbf{Z}$ are assumed fixed and known throughout. Assuming now that the hyperparameters λ are fixed and assuming a constant improper prior for β , then the posterior distribution for β and γ given $\mathbf{y}$ and λ is given by

$$f(\beta, \gamma|\mathbf{y}, \lambda) \propto f(\mathbf{y}|\beta, \gamma) f(\gamma|\lambda). \quad (14.3)$$

Model (14.1) assumes conditionally independent y_i for $i = 1, 2, \dots, n$ given (β, γ) and assumes that the γ_{jk} 's have independent normal (possibly improper) prior distributions, for the random $\gamma_{jk} \sim N(0, \mathbf{G}_{jk}^{-1})$, where $\mathbf{G}_{jk}^{-1}$ is a (possibly generalized) inverse of the $q_{jk} \times q_{jk}$ symmetric matrix $\mathbf{G}_{jk} = \mathbf{G}_{jk}(\lambda_{jk})$ which may depend on hyperparameters λ_{jk} . Hence, the parameters β^T and the random effects γ , for given hyperparameters λ , can be estimated by posterior mode (MAP) estimation, which is equivalent to maximization of the penalized, ℓ_p , or hierarchical, ℓ_h , likelihood see Lee and Nelder (1996, 2001, 2001b) since

$$\begin{aligned} \log f(\beta, \gamma|\mathbf{y}, \lambda) &= \log f(\mathbf{y}|\beta, \gamma) + \log f(\gamma|\lambda) + c_1(\mathbf{y}, \lambda) \\ &= \ell_h + c_1(\mathbf{y}, \lambda) \\ &= \sum_{i=1}^n \log f(y_i|\beta, \gamma) - \frac{1}{2} \sum_{k=1}^p \sum_{j=1}^{J_k} \gamma_{jk}^T \mathbf{G}_{jk} \gamma_{jk} + c_2(\mathbf{y}, \lambda) \\ &= \ell_p + c_2(\mathbf{y}, \lambda) \end{aligned} \quad (14.4)$$

14.3 Estimation of the random effects hyper-parameters λ

The hyperparameters λ can be estimated within a Classical likelihood framework for random effects by maximizing the marginal likelihood for (β, λ) given $\mathbf{y}$, i.e. $L(\beta, \lambda|\mathbf{y}) = \int f(\mathbf{y}|\beta, \gamma)f(\gamma|\lambda)d\gamma$. The maximization of $L(\beta, \lambda|\mathbf{y})$ over β and λ involves high dimensional integration so any approach to maximizing it will be computer intensive. Note that the MLE for β from this approach will not in general be the same as the MAP estimator for β described in the previous section.

In REML estimation, effectively a non-informative (constant) prior is assumed for β and both γ and β are integrated out of the joint density $f(\mathbf{y}, \gamma, \beta|\lambda)$ to give the marginal likelihood $L(\lambda|\mathbf{y})$, which is maximized over λ .

In a fully Bayesian inference for the GAMLSS model, the posterior distribution of (β, γ, λ) is obtained from (14.2), e.g. using MCMC; see Fahrmeir and Tutz (2001) or Fahrmeir and Lang (2001).

The above methods of estimation of the hyperparameters λ are in general highly computationally intensive: the ML and REML methods require high dimensional integration, while the fully Bayes method requires MCMC.

The following two methods (a) and (b), not requiring such computational intensity, are considered for hyperparameter estimation in GAMLSS.

The methods are summarised in the following algorithm:

- PROCEDURE 1.: Estimate the hyperparameters λ by one of the following two methods:
 - (a) minimizing a profile generalized Akaike Information criterion (GAIC) over λ
 - (b) maximizing the approximate marginal density (or profile marginal likelihood) for λ using a Laplace approximation
 - (c) approximately maximize the marginal density for λ using an EM algorithm
- PROCEDURE 2. For fixed current hyperparameters λ , use the GAMLSS (RS or CG) algorithm to obtain posterior mode (MAP) estimates of (β, γ) .

Procedure 2 is nested within procedure 1 and a numerical algorithm is used to estimate λ .

The estimation of λ by method (b) using the marginal likelihood, while more rigorous, is not easy to program in general. Method (c) is best suited to random coefficient models. The more relaxed approach by method (a) using a generalized Akaike criterion (GAIC) is easiest to program.

We now consider the methods in more detail:

(a) minimizing a profile GAIC over λ

The GAIC criterion, Akaike (1983), was considered by Hastie and Tibshirani (1990, p160 and p261) for hyperparameter estimation in generalized additive models (GAMs). In GAMs a cubic smoothing spline function $h(x)$ is used to model the dependence of a predictor on explanatory variable x . For a single smoothing spline term, since λ is related to the smoothing degrees of freedom $df = tr(\mathbf{S})$, selection (or estimation) of λ may be achieved by minimizing $GAIC(\#)$, defined in Chapter 2, over λ .

When the model contains p cubic smoothing spline functions in different explanatory variables, then the corresponding p smoothing hyperparameters $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_p)$ can be jointly

estimated by minimizing $\text{GAIC}(\#)$ over $\boldsymbol{\lambda}$. Note however that with multiple smoothing splines $\sum_{j=1}^p \text{tr}(\mathbf{S}_j)$ is only an approximation to the full model complexity degrees of freedom.

The $\text{GAIC}(\#)$ criterion can be applied more generally to estimate hyperparameters $\boldsymbol{\lambda}$ in the distribution of random effects terms. The (model complexity) degrees of freedom df needs to be obtained for models with random effects terms. This has been considered by Hodges and Sargent (2001). The degrees of freedom of a model with a single random effects term can be defined as the trace of the random effect (shrinkage) smoother $\mathbf{S}$, i.e. $df = \text{tr}(\mathbf{S})$. As with smoothing terms, when there are other terms in the model $\sum_{j=1}^p \text{tr}(\mathbf{S}_j)$ is only an approximation to the full model complexity degrees of freedom. The full model complexity degrees of freedom for model (14.1) is given by $df = \text{tr}\{\mathbf{A}^{-1}\mathbf{B}\}$ where $\mathbf{A}$ is defined in Appendix C and $\mathbf{B}$ is obtained from $\mathbf{A}$ by omitting the matrices $\mathbf{G}_{jk}$ for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$.

(b) maximizing the approximate marginal density (or profile marginal likelihood) of $\boldsymbol{\lambda}$ using a Laplace approximation

For GLMMs, Breslow and Clayton (1993) use a first order Laplace integral approximation to integrate out the random effects $\boldsymbol{\gamma}$ and approximate the marginal likelihood, leading to estimating equations based on penalized quasiliquid (PQL) for the mean model parameters and pseudo normal (reml) likelihood for the dispersion components. Breslow and Lin (1995) extend this to a second order Laplace approximation.

Lee and Nelder (1996) take a similar approach estimating the dispersion components using a first order approximation to the Cox-Read (1987) profile likelihood which eliminates the nuisance parameters $\boldsymbol{\beta}$ from the marginal likelihood, which they call an adjusted profile h -likelihood. Lee and Nelder (2001a) extend this to a second order approximation.

Here we consider an approximate Bayesian approach. Assuming a uniform improper prior for both $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$ then from (14.2) the posterior marginal of $\boldsymbol{\lambda}$ is given by

$$f(\boldsymbol{\lambda}|\mathbf{y}) = \int \int \frac{e^{\ell_h}}{f(\mathbf{y})} d\boldsymbol{\gamma} d\boldsymbol{\beta} \quad (14.5)$$

where $\ell_h = \ell_h(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \log f(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\gamma}) + \log f(\boldsymbol{\gamma}|\boldsymbol{\lambda}) = \ell_p + \frac{1}{2} \sum_{k=1}^p \sum_{j=1}^{J_k} [\log |\mathbf{G}_{jk}| - q_{jk} \log(2\pi)]$ and where ℓ_p is given by (2.6). Using a first order Laplace approximation, Tierney and Kadane (1986), to the integral (14.5) gives

$$f(\boldsymbol{\lambda}|\mathbf{y}) \approx \frac{e^{\hat{\ell}_h}}{f(\mathbf{y})} |\hat{\mathbf{D}}/(2\pi)|^{-\frac{1}{2}} \quad (14.6)$$

where $\hat{\ell}_h = \ell_h(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}})$ and

$$\hat{\mathbf{D}} = \mathbf{D}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}) = - \left[\begin{array}{cc} \frac{\partial^2 \ell_h}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} & \frac{\partial^2 \ell_h}{\partial \boldsymbol{\beta} \partial \boldsymbol{\gamma}^T} \\ \frac{\partial^2 \ell_h}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\beta}^T} & \frac{\partial^2 \ell_h}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^T} \end{array} \right]_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}, \boldsymbol{\gamma}=\hat{\boldsymbol{\gamma}}} \quad (14.7)$$

is the observed information matrix, evaluated at $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}(\boldsymbol{\lambda})$ and $\hat{\boldsymbol{\gamma}} = \hat{\boldsymbol{\gamma}}(\boldsymbol{\lambda})$, the MAP estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ given each fixed $\boldsymbol{\lambda}$. [Note that matrix $\mathbf{D}$ is a rearrangement of matrix $\mathbf{A}$ from Appendix C.] Alternatively, as a form of generalization of reml estimation of $\boldsymbol{\lambda}$, consider an approximate profile log-likelihood for $\boldsymbol{\lambda}$, denoted here as $\ell(\boldsymbol{\lambda})$, given by replacing $D(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}})$ by the expected information $\hat{\mathbf{H}} = \mathbf{H}(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}})$ giving

$$\ell(\boldsymbol{\lambda}) = \hat{\ell}_h - \frac{1}{2} \log |\hat{\mathbf{H}}/(2\pi)|. \quad (14.8)$$

This is closely related to the adjusted profile h -likelihood of Lee and Nelder (1996, 2001a, 2001b).

Estimation of λ can be achieved by maximizing (14.6) or (14.8) over λ (e.g. using a numerical maximization algorithm). The maximum value of $\ell(\lambda)$ given by (14.8) is called the maximized reml marginal likelihood, $RML = \ell(\hat{\lambda})$. The corresponding reml marginal deviance (RMDEV) is given by $RMDEV = -2\ell(\hat{\lambda})$.

Different random effects models (for the same fixed effects models) can be compared using the maximized reml marginal likelihood [$RML = \ell(\hat{\lambda})$] or equivalently the minimized reml marginal deviance $-2\ell(\hat{\lambda})$.

Different fixed effects models (for the same random effect model) can be compared using the maximized (Laplace approximated) marginal likelihood ($ML = \ell(\hat{\beta})$) or equivalently the minimized marginal deviance $MDEV = -2\ell(\hat{\beta})$ where $\ell(\hat{\beta})$ is obtained by maximizing the (Laplace approximated) marginal likelihood of β (eliminating the random effects γ) i.e.

$$\ell(\beta) \approx \ell_h(\beta, \hat{\gamma}) - \frac{1}{2} \log |\hat{H}/(2\pi)|. \quad (14.9)$$

where $\hat{H} = E[\partial \ell_h^2 / \partial \gamma \gamma^T]$ evaluated at $(\beta, \hat{\gamma})$ where $\hat{\gamma}$ is the MAP estimator of γ given β and ℓ_h is given in Section 14.2(b), conditional (for all the different fixed effect models) on fixed chosen (random effects) hyperparameters $\hat{\lambda}$. See Pinhero and Bates (2000), Section 2.4.2, for a discussion of hypothesis testing for fixed effect terms.

(c) approximately maximizing the marginal likelihood for λ using an (approximate) EM algorithm

An approximate EM algorithm was used by Fahrmeir and Tutz (2001), p298-303 and by Diggle *et al.* (2002) p172-175 to estimate hyperparameters in GLMMs and is similarly applied here to approximately maximize over λ the marginal likelihood of λ , $L(\lambda)$ (or equivalently the posterior marginal distribution of λ for a non-informative uniform prior).

In the E step of the EM algorithm, $M(\lambda|\hat{\lambda}) = E[\log f(\mathbf{y}, \beta, \gamma|\lambda)]$, is approximated, where the expectation is over the posterior distribution of (β, γ) given $\mathbf{y}$ and $\lambda = \hat{\lambda}$, i.e. $f(\beta, \lambda|\mathbf{y}, \hat{\lambda})$, where $\hat{\lambda}$ is the current estimate of λ , giving, apart from a function of $\mathbf{y}$,

$$M(\lambda|\hat{\lambda}) = -\frac{1}{2} \sum_{k=1}^p \sum_{j=1}^{J_k} \left[\text{tr}\{\mathbf{G}_{jk}[\hat{\gamma}_{jk} \hat{\gamma}_{jk}^T + \hat{V}(\hat{\gamma}_{jk})]\} - \frac{1}{2} \log |\mathbf{G}_{jk}| \right] \quad (14.10)$$

where $\hat{\gamma}_{jk}$ and $\hat{V}(\hat{\gamma}_{jk})$ are the posterior mode and curvature (i.e. submatix of $\mathbf{A}^{-1}$) of γ_{jk} from the MAP estimation in Appendix C.

In the M step of the EM algorithm, $M(\lambda|\hat{\lambda})$ is maximised over λ by a numerical maximization algorithm (e.g. the function `optim` in the R package). If $\mathbf{G}_{jk} = \mathbf{G}_k$ for $j = 1, 2, \dots, J_k$ and $k = 1, 2, \dots, p$, and the $\mathbf{G}_k$ are unconstrained positive definite symmetric matrices (e.g. in a random coefficients model), then (14.10) can be maximized explicitly giving, for $k = 1, 2, \dots, p$,

$$\hat{\mathbf{G}}_k^{-1} = \frac{1}{J_k} \sum_{j=1}^{J_k} [\hat{\gamma}_{jk} \hat{\gamma}_{jk}^T + \hat{V}(\hat{\gamma}_{jk})] \quad (14.11)$$

14.4 Examples

14.4.1 Hodges HMO data example

Data summary:

R data file: hodes in package **gamlss** of dimensions 341×6

variables

state : a factor with 45 levels

plan : a two-character code that identifies plans within state declared here as factor with 325 levels

prind : the total premium for an individual

prfam : the total premium for a family

enind : the total enrollment of federal employees as individuals

enfam : the total enrollment of federal employees as families.

purpose: to demonstrate the fitting of simple random effect model in **gamlss**

conclusion: A BCT distribution and random effect for μ is adequate

Here we consider a one factor random effects model for response variable health insurance premium (**prind**) with **state** as the random factor. The data were analyzed in Hodges (1998).

Hodges (1998) modelled the data using a normal conditional model for y_{ij} given γ_j the random effect in the mean for state j , and a normal distribution for γ_j , i.e. his model is expressed by $y_{ij}|\mu_{ij}, \sigma \sim N(\mu_{ij}, \sigma^2)$, $\mu_{ij} = \beta_1 + \gamma_j$, $\sigma^2 = \beta_2$ and $\gamma_j \sim N(0, \sigma_1^2)$, independently for $i = 1, 2, \dots, n_j$ and $j = 1, 2, \dots, J$, where i indexes the observations within states.

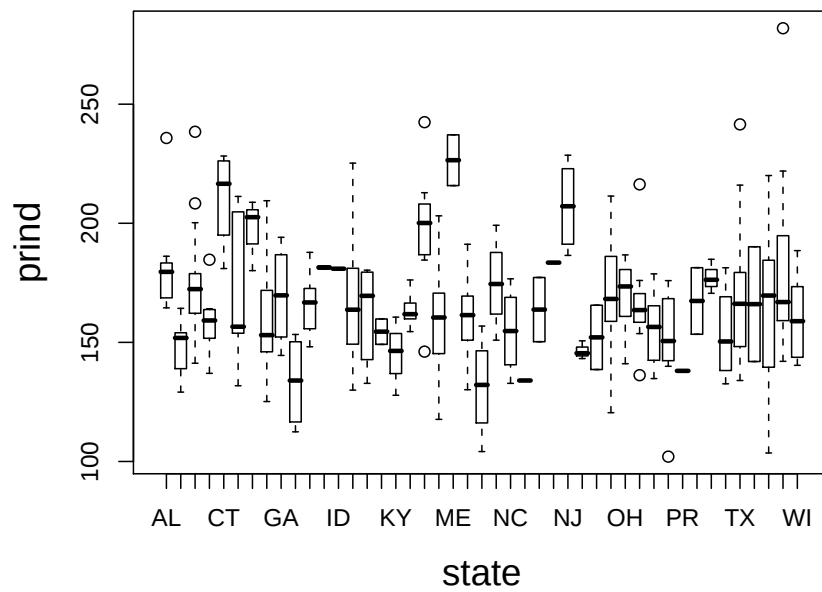
Figure 14.1 provides box plots of **prind** against **state**, showing the variation in the location and scale of **prind** between states and a positively skewed (and possible leptokurtic) distribution of **prind** within states. Although Hodges (1998) used an added variable diagnostic plot to identify the need for a Box-Cox transformation of y , he did not model the data using a transformation of y .

```
> library(gamlss)
> data(hodges)
> with(hodges, plot(prind ~ state, cex = 1, cex.lab = 1.5, cex.axis = 1,
+   cex.main = 1.2))
```

In the discussion of Hodges (1998), Wakefield commented as follows. 'If it were believed that there were different within-state variances then one possibility would be to assume a hierarchy for these also', while in his reply Hodges also suggested treating the 'within state precisions or variances as draws from some distribution'.

Hence we consider a Box-Cox t model which allow both for skewness and kurtosis in the conditional distribution of y given the parameters $\theta = (\mu, \sigma, \nu, \tau)$ and allows for possible differences between states in the location, scale and shape of the conditional distribution of y , by including a random effect term in each of the models for the parameters μ , σ , ν and τ , i.e. we assume a general model where, independently for $i = 1, 2, \dots, n_j$ and $j = 1, 2, \dots, J$,

$$y_{ij}|\mu_{ij}, \sigma_{ij}, \nu_{ij}, \tau_{ij} \sim BCT(\mu_{ij}, \sigma_{ij}, \nu_{ij}, \tau_{ij})$$

Figure 14.1: HMO data: box plots of `prind` against `stage`

where

$$\begin{aligned}\mu_{ij} &= \beta_1 + \gamma_{j1} \\ \log \sigma_{ij} &= \beta_2 + \gamma_{j2} \\ \nu_{ij} &= \beta_3 + \gamma_{j3} \\ \log \tau_{ij} &= \beta_4 + \gamma_{j4}\end{aligned}\tag{14.12}$$

and where $\gamma_{jk} \sim N(0, \sigma_k^2)$ independently for $j = 1, 2, \dots, J$ and $k = 1, 2, 3, 4$.

Using a Akaike information criterion, i.e. GAIC(2), for hyperparameter selection, as discussed in Section 14.3 lead to the following selected random effect parameters $\sigma_1 = 19.41$, $\sigma_2 = 0.0991$ and $\sigma_3 = \sigma_4 = 0$ i.e. constants for ν and τ (corresponding to $df_\mu = 36.29$, $df_\sigma = 4.1$, $df_\nu = 0$ and $df_\tau = 0$). The selection is achieve in R using the following commamnds:

```
> mod1 <- quote(gamlss(prind ~ random(state, df = p[1]), sigma.fo = ~random(state,
+   p[2]), nu.fo = ~random(state, p[3]), tau.fo = ~random(state,
+   p[4]), data = hodges, family = BCT, trace = FALSE))
> op <- find.hyper(mod1, par = c(40, 10, 3, 3), lower = c(0.001,
+   0.001, 0.001, 0.001), upper = c(45, 45, 10, 10), pen = 2)
```

and the results are:

```
> op

$par
[1] 36.293128  4.100129  0.001000  0.001000

$value
[1] 3085.359

$counts
function gradient
      14      14

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

The estimated degrees of freedom for the random effects for ν and τ are at the boundary of the search indicating that are effectively zero. That is, there is no need for random effects in those two parameters. Searching again for degrees of freedom for μ and σ after setting the ν and τ as constant results to similar degrees of freedom for μ and σ as the previous search. Now we refit the final model with constants for ν and τ .

```
> m1 <- gamlss(prind ~ random(state, df = 36.3), sigma.fo = ~random(state,
+   4.1), data = hodges, family = BCT)
```

```
GAMLSS-RS iteration 1: Global Deviance = 3005.369
GAMLSS-RS iteration 2: Global Deviance = 3004.668
GAMLSS-RS iteration 3: Global Deviance = 3004.602
```

```

GAMLSS-RS iteration 4: Global Deviance = 3004.572
GAMLSS-RS iteration 5: Global Deviance = 3004.56
GAMLSS-RS iteration 6: Global Deviance = 3004.554
GAMLSS-RS iteration 7: Global Deviance = 3004.550
GAMLSS-RS iteration 8: Global Deviance = 3004.549
GAMLSS-RS iteration 9: Global Deviance = 3004.548
GAMLSS-RS iteration 10: Global Deviance = 3004.547

> m1$mu.coefSmo

[[1]]
[1] 0.002653376

> m1$sigma.coefSmo

[[1]]
[1] 101.6494

> sqrt(1/(m1$mu.coefSmo[[1]]))

[1] 19.41335

> sqrt(1/(m1$sigma.coefSmo[[1]]))

[1] 0.09918538

> summary(m1)

*****
Family:  c("BCT", "Box-Cox t")

Call:
gamlss(formula = prind ~ random(state, df = 36.3), sigma.formula = ~random(state,
  4.1), family = BCT, data = hodes)

Fitting method: RS()

-----
Mu link function:  identity
Mu Coefficients:
      Estimate Std. Error t value    Pr(>|t|)
(Intercept)   165.1      1.036   159.3 1.887e-295

-----
Sigma link function:  log
Sigma Coefficients:
      Estimate Std. Error t value    Pr(>|t|)
(Intercept)   -2.252    0.04489  -50.15 2.475e-158

-----
Nu link function:  identity

```

Nu Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.06804	0.3865	-0.176	0.8604

Tau link function: log

Tau Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.079	0.2845	7.309	1.939e-12

No. of observations in the fit: 341
Degrees of Freedom for the fit: 42.40000
Residual Deg. of Freedom: 298.6
at cycle: 10

Global Deviance: 3004.547
AIC: 3089.347
SBC: 3251.819

Since $\hat{\nu} = \hat{\beta}_3 = -0.068$ is close to zero, the fitted model for the conditional distribution of y_{ij} given $(\mu_{ij}, \sigma_{jk}, \tau)$ is approximately defined by $\sigma_{ij}^{-1} \log(y_{ij}/\mu_{ij}) \sim t_{\hat{\tau}}$, a t distribution with $\hat{\tau} = \exp(\hat{\beta}_4) = 7.93$ degrees of freedom, for $i = 1, 2, \dots, n_j$ and $j = 1, 2, \dots, J$.

Figure 14.2 plots the sample and fitted medians of **prind** against **state** (ordered by the sample median). The fitted values of σ (not shown here) vary very little. The heterogeneity in the sample variances of **prind** between the states (in figure 14.1) seems to be primarily due to sampling variation caused by the high skewness and kurtosis in the conditional distribution of Y (rather than either the variance-mean relationship or the random effect in σ). Figure 14.3 plots the sample and approximate fitted coefficients of variation of **prind** against **state**.

Profile plots for each of ν and τ , give 95% intervals $(-1.01, 0.88)$ for ν and $(3.87, 33.54)$ for τ respectively, indicating considerable uncertainty about these parameters. The fitted model suggests a log transformation for Y , while the added variable plot used by Hodges (1998) suggested a Box-Cox transformation parameter $\nu = 0.67$ which, although rather different, still lies within the 95% interval for ν .

The maximized reml marginal likelihood, $\text{RML} = \ell(\hat{\lambda})$, given by maximizing (14.8) over λ , can be used to compare different random effects model having the same fixed effects models, i.e. constants for all four parameters of the BCT distribution in the example above. This can be used to investigate whether random effects are needed for both μ and σ .

Table 14.1 gives the resulting values for the reml marginal deviance, i.e. $\text{RMDEV} = -2 \times \text{RML}$, and the corresponding reml marginal AIC values, i.e. $\text{RMAIC} = \text{RMDEV} + 2 \times \text{df}$, where df is the hyperparameter degrees of freedom, i.e. the number of random effects hyperparameters.

The best model (with the lowest RMAIC) is the BCT model with random effects for μ only. Note that since the best model has random effects for the predictor of μ only we can now fit different models using the function `gamlssNP()` which uses Gaussian quadrature.

Table 14.2 gives the resulting values for the marginal deviance, i.e. $\text{MDEV} = -2 \times \text{ML}$, and the corresponding marginal AIC values, i.e. $\text{MAIC} = \text{MDEV} + 2 \times \text{df}$, where df is the fixed effects degrees of freedom, i.e. the number of fixed effects parameters. The best model (with the lowest MAIC) is the log t distribution model. Hence in conclusion the overall selected model

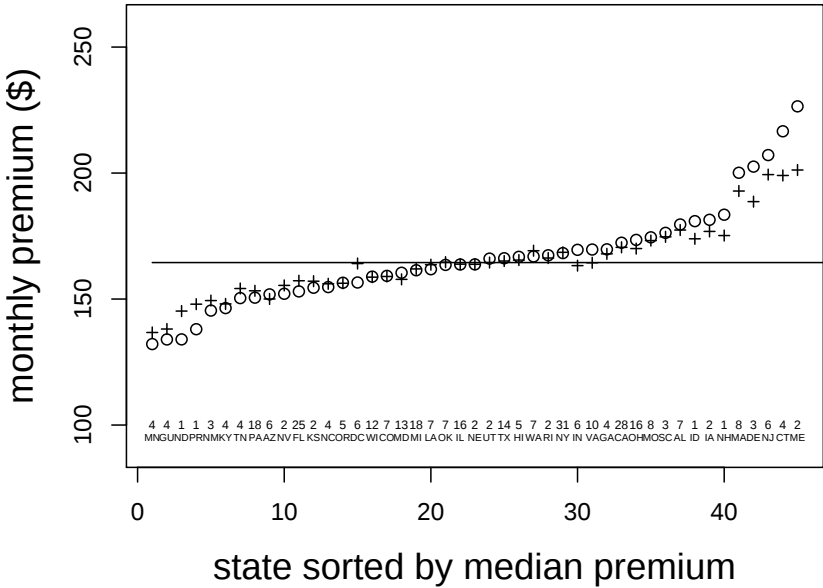


Figure 14.2: HMO data: the sample (o) and fitted medians (+) of `prind` against `state`

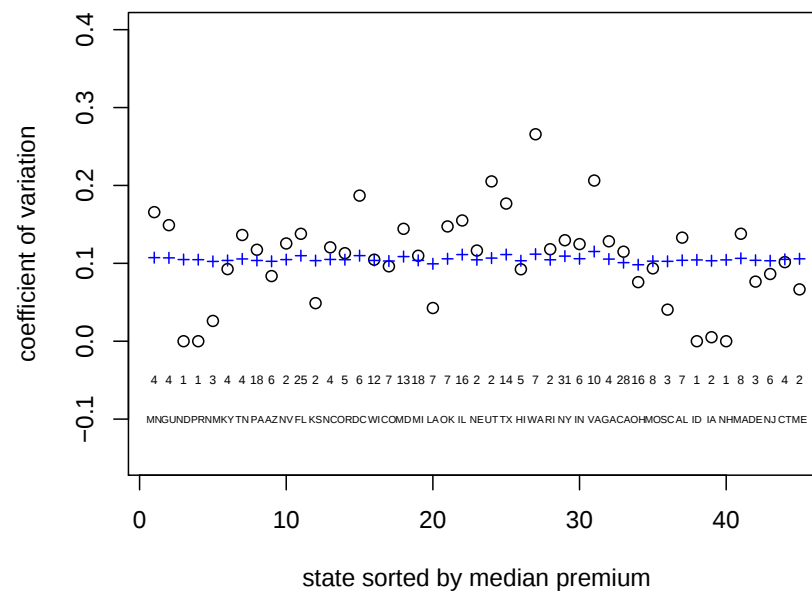


Figure 14.3: HMO data: The sample and approximate fitted coefficients of variation of `prind` against `state`

is a log t distribution model with random effects for μ only. Table 14.2 was obtained using the following commands.

```
> library(gamlss.mx)
> mNO <- gamlssNP(prind ~ 1, data = hedges, random = ~1 | state,
+   mixture = "gq", K = 20, family = NO)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..
EM algorithm met convergence criteria at iteration   11
Global deviance trend plotted.

> mBCT <- gamlssNP(prind ~ 1, data = hedges, random = ~1 | state,
+   mixture = "gq", K = 20, family = BCT)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..
EM algorithm met convergence criteria at iteration   22
Global deviance trend plotted.

> mlogt <- gamlssNP(prind ~ 1, data = hedges, random = ~1 | state,
+   mixture = "gq", K = 20, family = BCT, nu.fix = T, nu.start = 0.001)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..11 ..12 ..13 ..14 ..15 ..
EM algorithm met convergence criteria at iteration   22
Global deviance trend plotted.

> mlogNO <- gamlssNP(prind ~ 1, data = hedges, random = ~1 | state,
+   mixture = "gq", K = 20, family = LOGNO)

1 ..2 ..3 ..4 ..5 ..6 ..7 ..8 ..9 ..10 ..
EM algorithm met convergence criteria at iteration   10
Global deviance trend plotted.

> AIC(mBCT, mNO, mlogNO, mlogt, k = 0)

      df      AIC
mBCT    5 3115.723
mlogt    4 3115.729
mlogNO   3 3120.428
mNO      3 3137.513

> AIC(mBCT, mNO, mlogNO, mlogt, k = 2)

      df      AIC
mlogt    4 3123.729
mBCT     5 3125.723
mlogNO   3 3126.428
mNO      3 3143.513
```

Table 14.1: Comparison of random effects models in BCT model for Hodges data

Random effects model	RMDEV	df	RMAIC
for μ and σ	3118.6	2	3122.6
for μ only	3118.8	1	3120.8
no random effects	3142.6	0	3142.6

Table 14.2: Comparison of distribution models with random effect for μ only for Hodges data

Distribution model	MDEV	df	MAIC
BCT	3115.7	5	3125.7
Log t	3115.7	4	3123.7
Log Normal	3120.4	3	3126.4
Normal	3137.5	3	3143.5

Appendix A

Distributions in the `gamlss` packages

The distributions in Tables 2.1 and 2.2 can be divided into three categories depending on the type of response variable:

- continuous type
- binomial type (with binary as special case)
- counts or discrete type

Sections A.1, A.2 and A.3 cover the distributions available in `gamlss` packages of each of the three types above respectively. In each case the specific parameterization(s) used by `gamlss` for each of the distributions is given. For each parameterization of each distribution listed below, functions for the probability density function (pdf), cumulative distribution function (cdf), inverse cdf (i.e. quantile) function and random number generation are given by putting each of the letters `d`, `p`, `q` and `r` respectively before the `gamlss.family` name for the particular distribution parameterization. For example, for the parameterization of the normal distribution given by (A.1) below, denoted by $\text{NO}(\mu, \sigma)$, the corresponding `gamlss.family` functions `dNO`, `pNO`, `qNO` and `rNO` define its pdf, cdf, inverse cdf and random number generation respectively.

A.1 Continuous two parameter distributions on $\Re$

A.1.1 Normal (or Gaussian) distribution (`NO`, `NO2`, `NOF`)

First parameterization (`NO`)

The normal distribution is the default of the argument `family` of the function `gamlss()`. The parameterization used for the normal (or Gaussian) probability density function (pdf), denoted by $\text{NO}(\mu, \sigma)$, is

$$f_Y(y|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(y - \mu)^2}{2\sigma^2} \right] \quad (\text{A.1})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$ and $\sigma > 0$. The mean of Y is given by $E(Y) = \mu$ and the variance of Y by $\text{Var}(Y) = \sigma^2$, so μ is the mean and σ is the standard deviation of Y .

Second parameterization (NO2)

NO2(μ, σ) is a parameterization of the normal distribution where μ represents the mean and σ represents the variance of Y , i.e. $f_Y(y|\mu, \sigma) = (1/\sqrt{2\pi\sigma}) \exp[-(y - \mu)^2/(2\sigma)]$.

Normal family (of variance-mean relationships) (NOF)

The function **NOF**(μ, σ, ν) defines a normal distribution family with three parameters. The third parameter ν allows the variance of the distribution to be proportional to a power of the mean. The mean of **NOF**(μ, σ, ν) is equal to μ while the variance is equal to $Var(Y) = \sigma^2|\mu|^\nu$, so the standard deviation is $\sigma|\mu|^{\nu/2}$. The parametrization of the normal distribution given in the function **NOF**(μ, σ, ν) is

$$f(y|\mu, \sigma, \nu) = \frac{1}{\sqrt{2\pi}\sigma|\mu|^{\nu/2}} \exp \left[-\frac{(y - \mu)^2}{2\sigma^2|\mu|^\nu} \right] \quad (\text{A.2})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$ and $-\infty < \nu < \infty$.

The function **NOF**(μ, σ, ν) is appropriate for normally distributed regression type models where the variance of the response variable is proportional to a power of the mean. Models of this type are related to the "pseudo likelihood" models of Carroll and Rubert (1987) but here a proper likelihood is maximized. The ν parameter here is not designed to be modelled against explanatory variables but is a constant used as a device allowing us to model the variance mean relationship. Note that, due to the high correlation between the σ and ν parameters, the **mixed()** method argument is essential in the **gamlss()** fitting function. Alternatively ν can be estimated from its profile function, obtained using **gamlss** package function **prof.dev()**.

A.1.2 Logistic distribution (LO)

The logistic distribution is appropriate for moderately kurtotic data. The parameterization of the logistic distribution, denoted here as **LO**(μ, σ), is given by

$$f_Y(y|\mu, \sigma) = \frac{1}{\sigma} \left\{ \exp \left[-\left(\frac{y - \mu}{\sigma} \right) \right] \right\} \left\{ 1 + \exp \left[-\left(\frac{y - \mu}{\sigma} \right) \right] \right\}^{-2} \quad (\text{A.3})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$ and $\sigma > 0$, with $E(Y) = \mu$ and $Var(Y) = \pi^2\sigma^2/3$, Johnson *et al.* (1995) p 116.

A.1.3 Gumbel distribution (GU)

The Gumbel distribution is appropriate for moderately negative skew data. The pdf of the Gumbel distribution (or extreme value or Gompertz), denoted by **GU**(μ, σ), is defined by

$$f_Y(y|\mu, \sigma) = \frac{1}{\sigma} \exp \left[\left(\frac{y - \mu}{\sigma} \right) - \exp \left(\frac{y - \mu}{\sigma} \right) \right] \quad (\text{A.4})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$ and $\sigma > 0$, with $E(Y) = \mu - \gamma\sigma \simeq \mu - 0.57722\sigma$ and $Var(Y) = \pi^2\sigma^2/6 \simeq 1.64493\sigma^2$. See Crowder *et al.* (1991) p 17.

A.1.4 Reverse Gumbel distribution (RG)

The reverse Gumbel distribution, which is also called is the **type I extreme value distribution** is a special case of the generalized extreme value distribution, [see Johnson *et al.* (1995) p 2 and p 75]. The reverse Gumbel distribution is appropriate for moderately positive skew data. The pdf of the reverse Gumbel distribution, denoted by $\mathbf{RG}(\mu, \sigma)$ is defined by

$$f_Y(y|\mu, \sigma) = \frac{1}{\sigma} \exp \left\{ - \left(\frac{y - \mu}{\sigma} \right) - \exp \left[- \frac{(y - \mu)}{\sigma} \right] \right\} \quad (\text{A.5})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$ and $\sigma > 0$, with $E(Y) = \mu + \gamma\sigma \simeq \mu + 0.57722\sigma$ and $Var(Y) = \pi^2\sigma^2/6 \simeq 1.64493\sigma^2$. [Note that if $Y \sim RG(\mu, \sigma)$ and $W = -Y$, then $W \sim GU(-\mu, \sigma)$.]

A.2 Continuous three parameter distributions on $\Re$

A.2.1 Exponential Gaussian distribution (exGAUS)

The pdf of the ex-Gaussian distribution, denoted by $\mathbf{exGAUS}(\mu, \sigma)$, is defined as

$$f_Y(y|\mu, \sigma, \nu) = \frac{1}{\nu} \exp \left[\frac{\mu - y}{\nu} + \frac{\sigma^2}{2\nu^2} \right] \Phi \left(\frac{y - \mu}{\sigma} - \frac{\sigma}{\nu} \right) \quad (\text{A.6})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$ and $\nu > 0$, and where Φ is the cdf of the standard normal distribution. Since $Y = Y_1 + Y_2$ where $Y_1 \sim N(\mu, \sigma^2)$ and $Y_2 \sim EX(\nu)$ are independent, the mean of Y is given by $E(Y) = \mu + \nu$ and the variance is given by $Var(Y) = \sigma^2 + \nu^2$. This distribution has also been called the lagged normal distribution, Johnson *et al.*, (1994), p172.

A.2.2 Power Exponential distribution (PE, PE2)

First parameterization (PE)

The power exponential distribution is suitable for leptokurtic as well as platykurtic data. The pdf of the power exponential family distribution, denoted by $\mathbf{PE}(\mu, \sigma, \nu)$, is defined by

$$f_Y(y|\mu, \sigma, \nu) = \frac{\nu \exp[-|\frac{z}{c}|^\nu]}{2c\sigma\Gamma(\frac{1}{\nu})} \quad (\text{A.7})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$ and $\nu > 0$ and where $z = (y - \mu)/\sigma$ and $c^2 = \Gamma(1/\nu)[\Gamma(3/\nu)]^{-1}$.

In this parameterization, used by Nelson (1991), $E(Y) = \mu$ and $Var(Y) = \sigma^2$. Note that $\nu = 1$ and $\nu = 2$ correspond to the Laplace (i.e. two sided exponential) and normal distributions respectively, while the uniform distribution is the limiting distribution as $\nu \rightarrow \infty$.

The cdf of Y is given by $F_Y(y) = \frac{1}{2} [1 + F_S(s)\text{sign}(z)]$ where $S = |z/c|^\nu$ has a gamma distribution with pdf $f_S(s) = s^{1/\nu} \exp(-s)/\Gamma(\frac{1}{\nu})$.

Second parameterization (PE2)

An alternative parameterization, the power exponential type 2 distribution, denoted by $\mathbf{PE2}(\mu, \sigma, \nu)$, is defined by

$$f_Y(y|\mu, \sigma, \nu) = \frac{\nu \exp[-|z|^\nu]}{2\sigma\Gamma(\frac{1}{\nu})} \quad (\text{A.8})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$ and $\nu > 0$ and where $z = (y - \mu)/\sigma$. Here $E(Y) = \mu$ and $Var(Y) = \sigma^2/c^2$, where $c^2 = \Gamma(1/\nu)[\Gamma(3/\nu)]^{-1}$.

See also Johnson *et al.*, 1995, volume 2, p195, equation (24.83) for a re-parameterized version by Subbotin (1923).

A.2.3 t family distribution (TF)

The t family distribution is suitable for modelling leptokurtic data, that is, data with higher kurtosis than the normal distribution. The pdf of the t family distribution, denoted here as $\mathbf{TF}(\mu, \sigma, \nu)$, is defined by

$$f_Y(y|\mu, \sigma, \nu) = \frac{1}{\sigma B\left(\frac{1}{2}, \frac{\nu}{2}\right) \nu^{\frac{1}{2}}} \left[1 + \frac{(y - \mu)^2}{\sigma^2 \nu} \right]^{-\frac{\nu+1}{2}} \quad (\text{A.9})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$ and $\nu > 0$, where $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ is the beta function. The mean and variance of Y are given by $E(Y) = \mu$ and $Var(Y) = \sigma^2 \nu / (\nu - 2)$ when $\nu > 2$. Note that $T = (Y - \mu)/\sigma$ has a standard t distribution with ν degrees of freedom, given by Johnson *et al.* (1995), p 363, equation (28.2).

A.3 Continuous four parameter distributions on $\Re$

A.3.1 Exponential Generalized Beta type 2 distribution (EGB2)

The pdf of the exponential generalized beta type 2 distribution, denoted by $EGB2(\mu, \sigma, \nu, \tau)$, is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = e^{\nu z} \{ |\sigma| B(\nu, \tau) [1 + e^z]^{\nu+\tau} \}^{-1} \quad (\text{A.10})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $-\infty < \sigma < \infty$, $\nu > 0$ and $\tau > 0$, and where $z = (y - \mu)/\sigma$, McDonald and Xu (1995), equation (3.3). Here $E(Y) = \mu + \sigma [\Psi(\nu) - \Psi(\tau)]$ and $Var(Y) = \sigma^2 [\Psi^{(1)}(\nu) + \Psi^{(1)}(\tau)]$, from McDonald (1996), p437.

A.3.2 Generalized t distribution (GT)

This pdf of the generalized t distribution, denoted by $\mathbf{GT}(\mu, \sigma, \nu, \tau)$, is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \tau \left\{ 2\sigma \nu^{1/\tau} B(1/\tau, \nu) [1 + |z|^\tau / \nu]^{\nu+(1/\tau)} \right\}^{-1} \quad (\text{A.11})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$ and $\tau > 0$, and where $z = (y - \mu)/\sigma$, McDonald (1991) and McDonald and Newey (1988). Here $E(Y) = \mu$ and $Var(Y) = \sigma^2 \nu^{2/\tau} B\left(\frac{3}{\tau}, \nu - \frac{2}{\tau}\right) / B\left(\frac{1}{\tau}, \nu\right)$, from McDonald (1991) p274.

A.3.3 Johnson SU distribution (JSUo, JSU)

First parameterization (JSUo)

This is the original parameterization of the Johnson S_u distribution, Johnson (1949). The parameter ν determines the skewness of the distribution with $\nu > 0$ indicating negative skewness and $\nu < 0$ positive skewness. The parameter τ determines the kurtosis of the distribution. τ

should be positive and most likely in the region above 1. As $\tau \rightarrow \infty$ the distribution approaches the normal density function. The distribution is appropriate for leptokurtotic data.

The pdf of the original Johnson's S_u , denoted here as **JSUo**(μ, σ, ν, τ), is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{\tau}{\sigma} \frac{1}{(r^2 + 1)^{\frac{1}{2}}} \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2} z^2 \right] \quad (\text{A.12})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$ and $\tau > 0$, and where

$$z = \nu + \tau \sinh^{-1}(r) = \nu + \tau \log \left[r + (r^2 + 1)^{\frac{1}{2}} \right], \quad (\text{A.13})$$

where $r = (y - \mu)/\sigma$. Note that $Z \sim \mathbf{NO}(0, 1)$. Here $E(Y) = \mu - \sigma\omega^{1/2} \sinh(\nu/\tau)$ and $Var(Y) = \sigma^2 \frac{1}{2}(\omega - 1) [\omega \cosh(2\nu/\tau) + 1]$, where $\omega = \exp(1/\tau^2)$.

Second parameterization (JSU)

This is a reparameterization of the original Johnson S_u distribution, Johnson (1949), so that parameters μ and σ are the mean and the standard deviation of the distribution. The parameter ν determines the skewness of the distribution with $\nu > 0$ indicating positive skewness and $\nu < 0$ negative. The parameter τ determines the kurtosis of the distribution. τ should be positive and most likely in the region above 1. As $\tau \rightarrow \infty$ the distribution approaches the normal density function. The distribution is appropriate for leptokurtic data.

The pdf of the Johnson's S_u , denoted here as **JSU**(μ, σ, ν, τ), is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{\tau}{c\sigma} \frac{1}{(r^2 + 1)^{\frac{1}{2}}} \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2} z^2 \right] \quad (\text{A.14})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$, $\tau > 0$, and where

$$z = -\nu + \tau \sinh^{-1}(r) = -\nu + \tau \log \left[r + (r^2 + 1)^{\frac{1}{2}} \right], \quad (\text{A.15})$$

$$r = \frac{y - (\mu + c\sigma w^{\frac{1}{2}} \sinh \Omega)}{c\sigma},$$

$$c = \left\{ \frac{1}{2}(w - 1) [w \cosh(2\Omega) + 1] \right\}^{-\frac{1}{2}},$$

$w = \exp(1/\tau^2)$ and $\Omega = -\nu/\tau$. Note that $Z \sim \mathbf{NO}(0, 1)$. Here $E(Y) = \mu$ and $Var(Y) = \sigma^2$.

A.3.4 Normal-Exponential- t distribution (NET)

The NET distribution is a four parameter continuous distribution, although in **gamlss** it is used as a two parameter distribution with the other two of its parameters fixed. It was introduced by Rigby and Stasinopoulos (1994) as a robust method of fitting the mean and the scale parameters of a symmetric distribution as functions of explanatory variables. The NET distribution is the abbreviation of the Normal-Exponential-Student- t distribution and is denoted by **NET**(μ, σ, ν, τ), for given values for ν and τ . It is normal up to ν , exponential from ν to τ and

Student- t with $(\nu\tau - 1)$ degrees of freedom after τ . Fitted parameters are the first two parameters, μ and σ . Parameters ν and τ may be chosen and fixed by the user. Alternatively estimates of the third and forth parameters can be obtained, using the **gamlss** function **prof.dev()**.

The pdf of the normal exponential t distribution, denoted here as **NET** (μ, σ, ν, τ) , is given by Rigby and Stasinopoulos (1994) and defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{c}{\sigma} \begin{cases} \exp\left\{-\frac{z^2}{2}\right\}, & \text{when } |z| \leq \nu \\ \exp\left\{-\nu|z| + \frac{\nu^2}{2}\right\}, & \text{when } \nu < |z| \leq \tau \\ \exp\left\{-\nu\tau \log\left(\frac{|z|}{\tau}\right) - \nu\tau + \frac{\nu^2}{2}\right\}, & \text{when } |z| > \tau \end{cases} \quad (\text{A.16})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 1$, $\tau > \nu$ ¹, and where $z = (y - \mu)/\sigma$ and $c = (c_1 + c_2 + c_3)^{-1}$, where $c_1 = \sqrt{2\pi} [1 - 2\Phi(-\nu)]$, $c_2 = \frac{2}{\nu} \exp\left\{-\frac{\nu^2}{2}\right\}$ and $c_3 = \frac{2}{(\nu\tau-1)\nu} \exp\left\{-\nu\tau + \frac{\nu^2}{2}\right\}$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution. Here μ is the mean of Y .

A.3.5 Sinh-Arcsinh (SHASH)

The pdf of the Sinh-Arcsinh distribution, denoted by **SHASH** (μ, σ, ν, τ) , Jones(2005), is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{c}{\sqrt{2\pi}\sigma(1+r^2)^{1/2}} e^{-z^2/2} \quad (\text{A.17})$$

where

$$z = \frac{1}{2} \left\{ \exp[\tau \sinh^{-1}(r)] - \exp[-\nu \sinh^{-1}(r)] \right\}$$

and

$$c = \frac{1}{2} \left\{ \tau \exp[\tau \sinh^{-1}(r)] + \nu \exp[-\nu \sinh^{-1}(r)] \right\}$$

and $r = (y - \mu)/\sigma$ for $-\infty < y < \infty$, where $-\infty < \mu < +\infty$, $\sigma > 0$, $\nu > 0$ and $\tau > 0$. Note $\sinh^{-1}(r) = \log(u)$ where $u = r + (r^2 + 1)^{1/2}$. Hence $z = \frac{1}{2} (u^\tau - u^{-\nu})$. Note that $Z \sim \mathbf{NO}(0, 1)$. Hence μ is the median of Y .

A.3.6 Skew Exponential Power type 1 distribution (SEP1)

The pdf of the skew exponential power type 1 distribution, denoted by **SEP1** (μ, σ, ν, τ) , is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{2}{\sigma} f_{Z_1}(z) F_{Z_1}(\nu z) \quad (\text{A.18})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$ and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and f_{Z_1} and F_{Z_1} are the pdf and cdf of $Z_1 \sim PE2(0, \tau^{1/\tau}, \tau)$, a power exponential type

¹since NET involves the Student- t distribution with $(\nu\tau-1)$ degrees of freedom

2 distribution with $f_{Z_1}(z) = \alpha^{-1} \exp[-|z|^\tau/\tau]$, where $\alpha = 2\tau^{(1/\tau)-1}\Gamma(1/\tau)$. This distribution was introduced by Azzalini (1986) as his type I distribution.

Here $E(Y) = \mu + \sigma E(Z)$ and $Var(Y) = \sigma^2 V(Z) = \sigma^2 \{E(Z^2) - [E(Z)]^2\}$ where $Z = (Y - \mu)/\sigma$ and $E(Z) = \text{sign}(\nu)\tau^{1/\tau} [\Gamma(\frac{2}{\tau})/\Gamma(\frac{1}{\tau})] pBEo\left(\frac{\nu^\tau}{1+\nu^\tau}, \frac{1}{\tau}, \frac{2}{\tau}\right)$, and $E(Z^2) = \tau^{2/\tau}\Gamma(\frac{3}{\tau})/\Gamma(\frac{1}{\tau})$, where $pBEo(q, a, b)$ is the cdf of an original beta distribution $BEo(a, b)$ evaluated at q , Azzalini (1986), p202-203.

The skew normal type 1 distribution, denoted by **SN1**(μ, σ, ν), a special case of **SEP1**(μ, σ, ν, τ) given by $\tau = 2$, has mean and variance given by $E(Y) = \mu + \sigma \text{sign}(\nu) \{2\nu^2 / [\pi(1 + \nu^2)]\}^{1/2}$ and $Var(Y) = \sigma^2 \{1 - 2\nu^2 / [\pi(1 + \nu^2)]\}$, Azzalini (1985), p174. Note that **SN1** is not currently implemented as a specific distribution, but can be obtained by fixing $\tau = 2$ in **SEP1** using the arguments `tau.start=2, tau.fix=TRUE` in `gamlss()`.

A.3.7 Skew Exponential Power type 2 distribution (SEP2)

The pdf of the skew exponential power type 2 distribution, denoted by **SEP2**(μ, σ, ν, τ), is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{2}{\sigma} f_{Z_1}(z) \Phi(\omega) \quad (\text{A.19})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$, and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $\omega = \text{sign}(z)|z|^{\tau/2}\nu\sqrt{2/\tau}$ and f_{Z_1} is the pdf of $Z_1 \sim PE2(0, \tau^{1/\tau}, \tau)$ and $\Phi(\omega)$ is the cdf of a standard normal variable evaluated at ω .

This distribution was introduced by Azzalini (1986) as his type II distribution and was further developed by DiCiccio and Monti (2004). The parameter ν determines the skewness of the distribution with $\nu > 0$ indicating positive skewness and $\nu < 0$ negative. The parameter τ determines the kurtosis of the distribution, with $\tau > 2$ for platykurtic data and $\tau < 2$ for leptokurtic.

Here $E(Y) = \mu + \sigma E(Z)$ and $Var(Y) = \sigma^2 V(Z)$ where

$$E(Z) = \frac{2\tau^{1/\tau}\nu}{\sqrt{\pi}\Gamma(\frac{1}{\tau})(1+\nu^2)^{(2/\tau)+(1/2)}} \sum_{n=0}^{\infty} \frac{\Gamma(\frac{2}{\tau} + n + \frac{1}{2})}{(2n+1)!!} \left(\frac{2\nu^2}{1+\nu^2}\right)^n \quad (\text{A.20})$$

and $E(Z^2) = \tau^{1/\tau}\Gamma(\frac{3}{\tau})/\Gamma(\frac{1}{\tau})$, where $(2n+1)!! = 1.3.5...(2n-1)$, DiCiccio and Monti (2004), p439.

For $\tau = 2$ the **SEP2**(μ, σ, ν, τ) distribution is the skew normal type 1 distribution, Azzalini (1985), denoted by **SN1**(μ, σ, ν), while for $\nu = 1$ and $\tau = 2$ the **SEP2**(μ, σ, ν, τ) distribution is the normal density function, **NO**(μ, σ).

A.3.8 Skew Exponential Power type 3 distribution (SEP3)

This is a "spliced-scale" distribution with pdf, denoted by **SEP3**(μ, σ, ν, τ), defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{c}{\sigma} \left\{ \exp\left[-\frac{1}{2}|\nu z|^\tau\right] I(y < \mu) + \exp\left[-\frac{1}{2}\left|\frac{z}{\nu}\right|^\tau\right] I(y \geq \mu) \right\} \quad (\text{A.21})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$, and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $c = \nu\tau/[(1 + \nu^2)2^{1/\tau}\Gamma(\frac{1}{\tau})]$, Fernandez, Osiewalski and Steel (1995). Note that $I()$ is an indicator function, where $I(u) = 1$ if u is true and $I(u) = 0$ if u is false.

Note that μ is the mode of Y . Here $E(Y) = \mu + \sigma E(Z)$ and $Var(Y) = \sigma^2 V(Z)$ where $E(Z) = 2^{1/\tau} \Gamma(\frac{2}{\tau}) (\nu - \frac{1}{\nu}) / \Gamma(\frac{1}{\tau})$ and $E(Z^2) = 2^{2/\tau} \Gamma(\frac{3}{\tau}) (\nu^3 + \frac{1}{\nu^3}) / [\Gamma(\frac{1}{\tau}) (\nu + \frac{1}{\nu})]$, Fernandez, Osiewalski and Steel (1995), p1333, eqns. (12) and (13).

The skew normal type 2 distribution, Johnson *et al.* (1994) p173, denoted by **SN2**(μ, σ, ν), (or two-piece normal) is a special case of **SEP3**(μ, σ, ν, τ) given by $\tau = 2$.

A.3.9 Skew Exponential Power type 4 distribution (SEP4)

This is a "spliced-shape" distribution with pdf, denoted by **SEP4**(μ, σ, ν, τ), defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{c}{\sigma} \{ \exp[-|z|^\nu] I(y < \mu) + \exp[-|z|^\tau] I(y \geq \mu) \} \quad (\text{A.22})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$, and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $c = [\Gamma(1 + \frac{1}{\nu}) + \Gamma(1 + \frac{1}{\tau})]^{-1}$, Jones (2005). Note that μ is the mode of Y .

Here $E(Y) = \mu + \sigma E(Z)$ and $Var(Y) = \sigma^2 V(Z)$ where $E(Z) = c [\frac{1}{\tau} \Gamma(\frac{2}{\tau}) - \frac{1}{\nu} \Gamma(\frac{2}{\nu})]$ and $E(Z^2) = c [\frac{1}{\nu} \Gamma(\frac{3}{\nu}) + \frac{1}{\tau} \Gamma(\frac{3}{\tau})]$.

A.3.10 Skew t type 1 distribution (ST1)

The pdf of the skew t type 1 distribution, denoted by **ST1**(μ, σ, ν, τ), is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{2}{\sigma} f_{Z_1}(z) F_{Z_1}(\nu z) \quad (\text{A.23})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$ and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and f_{Z_1} and F_{Z_1} are the pdf and cdf of $Z \sim TF(0, 1, \tau)$, a t distribution with $\tau > 0$ degrees of freedom, with τ treated as a continuous parameter. This distribution is in the form of a type I distribution of Azzalini (1986).

A.3.11 Skew t type 2 distribution (ST2)

The pdf of the skew t type 2 distribution, denoted by **ST2**(μ, σ, ν, τ), is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{2}{\sigma} f_{Z_1}(z) F_{Z_2}(w) \quad (\text{A.24})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$, and $\tau > 0$, and where $z = (y - \mu)/\sigma$, $w = \nu \lambda^{1/2} z$ and $\lambda = (\tau + 1)/(\tau + z^2)$ and f_{Z_1} is the pdf of $Z_1 \sim TF(0, 1, \tau)$ and F_{Z_2} is the cdf of $Z_2 \sim TF(0, 1, \tau + 1)$. This distribution is the univariate case of the multivariate skew t distribution introduced by Azzalini and Capitanio (2003).

Here the mean and variance of Y are given by $E(Y) = \mu + \sigma E(Z)$ and $Var(Y) = \sigma^2 V(Z)$ where $E(Z) = \nu \tau^{1/2} \Gamma(\frac{\tau-1}{2}) / [\pi^{1/2} (1 + \nu^2)^{1/2} \Gamma(\frac{\tau}{2})]$ for $\tau > 1$ and $E(Z^2) = \tau/(\tau - 2)$ for $\tau > 2$, Azzalini and Capitanio (2003), p382.

A.3.12 Skew t type 3 distribution (ST3)

This is a "spliced-scale" distribution with pdf, denoted by **ST3**(μ, σ, ν, τ), defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{c}{\sigma} \left\{ 1 + \frac{z^2}{\tau} \left[\nu^2 I(y < \mu) + \frac{1}{\nu^2} I(y \geq \mu) \right] \right\} \quad (\text{A.25})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$, and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $c = 2\nu / [\sigma(1 + \nu^2) B(\frac{1}{2}, \frac{\tau}{2}) \tau^{1/2}]$, Fernandez and Steel (1998).

Note that μ is the mode of Y . The mean and variance of Y are given by $E(Y) = \mu + \sigma E(Z)$ and $Var(Y) = \sigma^2 V(Z)$ where $E(Z) = 2\tau^{1/2}(\nu^2 - 1) / [(\tau - 1)B(\frac{1}{2}, \frac{\tau}{2}) \nu]$ and $E(Z^2) = \tau(\nu^3 + \frac{1}{\nu^3}) / [(\tau - 2)(\nu + \frac{1}{\nu})]$, Fernandez and Steel (1998), p360, eqn. (5).

A.3.13 Skew t type 4 distribution (ST4)

This is a "spliced-shape" distribution with pdf, denoted by $ST4(\mu, \sigma, \nu, \tau)$, defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{c}{\sigma} \left\{ \left[1 + \frac{z^2}{\nu} \right]^{-(\nu+1)/2} I(y < \mu) + \left[1 + \frac{z^2}{\tau} \right]^{-(\tau+1)/2} I(y \geq \mu) \right\} \quad (\text{A.26})$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $\nu > 0$ and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $c = 2 [\nu^{1/2} B(\frac{1}{2}, \frac{\nu}{2}) + \tau^{1/2} B(\frac{1}{2}, \frac{\tau}{2})]^{-1}$.

Here $E(Y) = \mu + \sigma E(Z)$ and $Var(Y) = \sigma^2 V(Z)$ where $E(Z) = c \left[\frac{1}{\tau-1} - \frac{1}{\nu-1} \right]$, provided $\nu > 1$ and $\tau > 1$, and $E(Z^2) = \frac{c}{2} \{ [\tau^{3/2} B(\frac{1}{2}, \frac{\tau}{2}) / (\tau - 2)] + [\nu^{3/2} B(\frac{1}{2}, \frac{\nu}{2}) / (\nu - 2)] \}$, provided $\nu > 2$ and $\tau > 2$.

A.3.14 Skew t type 5 distribution (ST5)

The pdf of the skew t distribution type 5, denoted by $\mathbf{ST5}(\mu, \sigma, \nu, \tau)$, Jones and Faddy (2003), is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{c}{\sigma} \left[1 + \frac{z}{(a+b+z^2)^{1/2}} \right]^{a+1/2} \left[1 - \frac{z}{(a+b+z^2)^{1/2}} \right]^{b+1/2}$$

for $-\infty < y < \infty$, where $-\infty < \mu < \infty$, $\sigma > 0$, $-\infty < \nu < \infty$ and $\tau > 0$, and where $z = (y - \mu)/\sigma$ and $c = [2^{a+b-1}(a+b)^{1/2} B(a, b)]^{-1}$ and $\nu = (a-b) / [ab(a+b)]^{1/2}$ and $\tau = 2/(a+b)$.

Here $E(Y) = \mu + \sigma E(Z)$ where $E(Z) = (a-b)(a+b)^{1/2} \Gamma(a - \frac{1}{2}) \Gamma(a - \frac{1}{2}) / [2\Gamma(a)\Gamma(b)]$ and $Var(Y) = \sigma^2 V(Z)$ where $E(Z^2) = (a+b) [(a-b)^2 + a+b-2] / [4(a-1)(b-1)]$, Jones and Faddy (2003), p162.

A.4 Continuous one parameter distribution in $\Re^+$

A.4.1 Exponential distribution (EXP)

This is the only one parameter continuous distribution in **gamlss** packages. The exponential distribution is appropriate for moderately positive skew data. The parameterization of the exponential distribution, denoted here as $\mathbf{EXP}(\mu)$, is defined by

$$f_Y(y|\mu) = \frac{1}{\mu} \exp \left\{ -\frac{y}{\mu} \right\} \quad (\text{A.27})$$

for $y > 0$, where $\mu > 0$ and where $E(Y) = \mu$ and $Var(Y) = \mu^2$.

A.5 Continuous two parameter distribution in $\mathcal{R}^+$

A.5.1 Gamma distribution (GA)

The gamma distribution is appropriate for positively skew data. The pdf of the gamma distribution, denoted by **GA**(μ, σ), is defined by

$$f_Y(y|\mu, \sigma) = \frac{1}{(\sigma^2\mu)^{1/\sigma^2}} \frac{y^{\frac{1}{\sigma^2}-1} e^{-y/(\sigma^2\mu)}}{\Gamma(1/\sigma^2)} \quad (\text{A.28})$$

for $y > 0$, where $\mu > 0$ and $\sigma > 0$. Here $E(Y) = \mu$ and $Var(Y) = \sigma^2\mu^2$. This a reparameterization of Johnson *et al.* (1994) p 343 equation (17.23) obtained by setting $\sigma^2 = 1/\alpha$ and $\mu = \alpha\beta$.

A.5.2 Log Normal distribution (LOGNO, LNO)

Log Normal distribution (LOGNO)

The log-normal distribution is appropriate for positively skew data. The pdf of the log-normal distribution, denoted by **LOGNO**(μ, σ), is defined by

$$f_Y(y|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{1}{y} \exp \left\{ -\frac{[\log(y) - \mu]^2}{2\sigma^2} \right\} \quad (\text{A.29})$$

for $y > 0$, where $\mu > 0$ and $\sigma > 0$. Here $E(Y) = \omega^{1/2}e^\mu$ and $Var(Y) = \omega(\omega - 1)e^{2\mu}$, where $\omega = \exp(\sigma^2)$.

Log normal family (i.e. original Box-Cox) (LNO)

The **gamlss** function **LNO**(μ, σ, ν) allows the use of the Box-Cox power transformation approach, Box and Cox (1964), where the transformation $Y(\nu)$ is applied to Y in order to remove skewness, where $Z = (Y^\nu - 1)/\nu$ (if $\nu \neq 0$) + $\log(Y)$ (if $\nu = 0$). The transformed variable Z is then assumed to have a normal $NO(\mu, \sigma)$ distribution. The resulting distribution for Y is denoted by **LNO**(μ, σ, ν). When $\nu = 0$, this results in the distribution in equation (A.29). For values of ν different from zero we have the resulting three parameter distribution

$$f_Y(y|\mu, \sigma, \nu) = \frac{y^{\nu-1}}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{(z - \mu)^2}{2\sigma^2} \right] \quad (\text{A.30})$$

for $y > 0$, where $\mu > 0$, $\sigma > 0$ and $-\infty < \nu < \infty$, and where $z = (y^\nu - 1)/\nu$ (if $\nu \neq 0$) + $\log(y)$ (if $\nu = 0$). The distribution in (A.30) can be fitted for fixed ν only, e.g. $\nu = 0.5$, using the following arguments of **gamlss**(): **family=LNO**, **nu.fix=TRUE**, **nu.start=0.5**. If ν is unknown, it can be estimated from its profile likelihood. Alternatively instead of (A.30), the more orthogonal parameterization of (A.30) given by the BCCG distribution in Section A.6.1 can be used.

A.5.3 Inverse Gaussian distribution (IG)

The inverse Gaussian distribution is appropriate for highly positive skew data. The pdf of the inverse Gaussian distribution, denoted by **IG**(μ, σ) is defined by

$$f_Y(y|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2 y^3}} \exp \left[-\frac{1}{2\mu^2\sigma^2 y} (y - \mu)^2 \right] \quad (\text{A.31})$$

for $y > 0$, where $\mu > 0$ and $\sigma > 0$ with $E(Y) = \mu$ and $Var(Y) = \sigma^2 \mu^3$. This is a reparameterization of Johnson *et al.* (1994) p 261 equation (15.4a), obtained by setting $\sigma^2 = 1/\lambda$.

A.5.4 Weibull distribution (WEI, WEI2, WEI3)

First parameterization (WEI)

There are three version of the two parameter Weibull distribution implemented into the **gamlss** package. The first, denoted by **WEI**(μ, σ), has the following parameterization

$$f_Y(y|\mu, \sigma) = \frac{\sigma y^{\sigma-1}}{\mu^\sigma} \exp \left[- \left(\frac{y}{\mu} \right)^\sigma \right] \quad (\text{A.32})$$

for $y > 0$, where $\mu > 0$ and $\sigma > 0$, [see Johnson *et al.* (1994) p629]. The mean and the variance of Y in this parameterization (A.32) of the two parameter Weibull are given by $E(Y) = \mu \Gamma(\frac{1}{\sigma} + 1)$ and $Var(Y) = \mu^2 \left\{ \Gamma(\frac{2}{\sigma} + 1) - [\Gamma(\frac{1}{\sigma} + 1)]^2 \right\}$, from Johnson *et al.* (1994) p632. Although the parameter μ is a scale parameter, it also affects the mean of Y . The median of Y is $m_Y = \mu(\log 2)^{1/\sigma}$, see Johnson *et al.* (1994), p630.

Second parameterization (WEI2)

The second parameterization of the Weibull distribution, denoted by **WEI2**(μ, σ), is defined as

$$f_Y(y|\mu, \sigma) = \sigma \mu y^{\sigma-1} e^{-\mu y^\sigma} \quad (\text{A.33})$$

for $y > 0$, where $\mu > 0$ and $\sigma > 0$, Johnson *et al.* (1994), p686. The mean of Y in this parameterization (A.33) is $E(Y) = \mu^{-1/\sigma} \Gamma(\frac{1}{\sigma} + 1)$ and the variance of Y is $Var(Y) = \mu^{-2/\sigma} \left\{ \Gamma(\frac{2}{\sigma} + 1) - [\Gamma(\frac{1}{\sigma} + 1)]^2 \right\}$.

In the second parameterization of the Weibull distribution the two parameters μ and σ are highly correlated, so the RS method of fitting is very slow and therefore the **CG**() method of fitting should be used.

Third parameterization (WEI3)

This is a parameterization of the Weibull distribution where μ is the mean of the distribution. This parameterization of the Weibull distribution, denoted by **WEI3**(μ, σ), is defined as

$$f_Y(y|\mu, \sigma) = \frac{\sigma}{\beta} \left(\frac{y}{\beta} \right)^{\sigma-1} \exp \left\{ - \left(\frac{y}{\beta} \right)^\sigma \right\} \quad (\text{A.34})$$

for $y > 0$, where $\mu > 0$ and $\sigma > 0$ and where $\beta = \mu/\Gamma(\frac{1}{\sigma} + 1)$. The mean of Y is given by $E(Y) = \mu$ and the variance $Var(Y) = \mu^2 \left\{ \Gamma(\frac{2}{\sigma} + 1) [\Gamma(\frac{1}{\sigma} + 1)]^{-2} - 1 \right\}$.

A.6 Continuous three parameter distribution in $\mathbb{R}^+$

A.6.1 Box-Cox Cole and Green distribution (BCCG)

The Box-Cox Cole and Green distribution is suitable for positively or negatively skew data. Let $Y > 0$ be a positive random variable having a Box-Cox Cole and Green distribution, denoted

here as **BCCG**(μ, σ, ν), defined through the transformed random variable Z given by

$$Z = \begin{cases} \frac{1}{\sigma\nu} \left[\left(\frac{Y}{\mu} \right)^\nu - 1 \right], & \text{if } \nu \neq 0 \\ \frac{1}{\sigma} \log\left(\frac{Y}{\mu}\right), & \text{if } \nu = 0 \end{cases} \quad (\text{A.35})$$

for $0 < Y < \infty$, where $\mu > 0$, $\sigma > 0$ and $-\infty < \nu < \infty$, and where the random variable Z is assumed to follow a truncated standard normal distribution. The condition $0 < Y < \infty$ (required for Y^ν to be real for all ν) leads to the condition $-1/(\sigma\nu) < Z < \infty$ if $\nu > 0$ and $-\infty < Z < -1/(\sigma\nu)$ if $\nu < 0$, which necessitates the truncated standard normal distribution for Z .

Hence the pdf of Y is given by

$$f_Y(y) = \frac{y^{\nu-1} \exp(-\frac{1}{2}z^2)}{\mu^\nu \sigma \sqrt{2\pi} \Phi(\frac{1}{\sigma|\nu|})} \quad (\text{A.36})$$

where z is given by (A.35) and $\Phi()$ is the cumulative distribution function (cdf) of a standard normal distribution.

If the truncation probability $\Phi(-\frac{1}{\sigma|\nu|})$ is negligible, the variable Y has median μ . The parameterization (A.35) was used by Cole and Green (1992) who assumed a standard normal distribution for Z and assumed that the truncation probability was negligible.

A.6.2 Generalized gamma distribution (GG, GG2)

First parameterization (GG)

The specific parameterization of the generalized gamma distribution used here and denoted by **GG**(μ, σ, ν) was used by Lopatzidis and Green (2000), and is defined as

$$f_Y(y|\mu, \sigma, \nu) = \frac{|\nu| \theta^\theta z^\theta \exp\{-\theta z\}}{\Gamma(\theta)y} \quad (\text{A.37})$$

for $y > 0$, where $\mu > 0$, $\sigma > 0$ and $-\infty < \nu < \infty$ and where $z = (y/\mu)^\nu$ and $\theta = 1/(\sigma^2\nu^2)$.

The mean and variance of Y are given by $E(Y) = \mu \Gamma(\theta + \frac{1}{\nu}) / [\theta^{1/\nu} \Gamma(\theta)]$ and $Var(Y) = \mu^2 \left\{ \Gamma(\theta) \Gamma(\theta + \frac{2}{\nu}) - [\Gamma(\theta + \frac{1}{\nu})]^2 \right\} / \left\{ \theta^{2/\nu} [\Gamma(\theta)]^2 \right\}$. Note that **GG2** is not currently implemented in **gamlss**.

Second parameterization (GG2)

A second parameterization, given by Johnson *et al.*, (1995), p401, denoted by **GG2**(μ, σ, ν), is defined as

$$f_Y(y|\mu, \sigma, \nu) = \frac{|\mu| y^{\mu\nu-1}}{\Gamma(\nu) \sigma^{\mu\nu}} \exp\left\{-\left(\frac{y}{\sigma}\right)^\mu\right\} \quad (\text{A.38})$$

for $y > 0$, where $-\infty < \mu < \infty$, $\sigma > 0$ and $\nu > 0$.

The mean and variance of $Y \sim \mathbf{GG2}(\mu, \sigma, \nu)$ can be obtained from those of **GG**(μ, σ, ν) since $\mathbf{GG}(\mu, \sigma, \nu) \equiv \mathbf{GG2}(\nu, \mu\theta^{-1/\nu}, \theta)$ and $\mathbf{GG2}(\mu, \sigma, \nu) \equiv \mathbf{GG}(\sigma\nu^{1/\mu}, [\mu^2\nu]^{-1/2}, \mu)$.

A.6.3 Generalized inverse Gaussian distribution (GIG)

The parameterization of the generalized inverse Gaussian distribution, denoted by $\mathbf{GIG}(\mu, \sigma, \nu)$, is defined as

$$f_Y(y|\mu, \sigma, \nu) = \left(\frac{c}{\mu}\right)^\nu \left[\frac{y^{\nu-1}}{2K_\nu\left(\frac{1}{\sigma^2}\right)} \right] \exp \left[-\frac{1}{2\sigma^2} \left(\frac{cy}{\mu} + \frac{\mu}{cy} \right) \right] \quad (\text{A.39})$$

for $y > 0$, where $\mu > 0$, $\sigma > 0$ and $-\infty < \nu < \infty$, where $c = [K_{\nu+1}(1/\sigma^2)] [K_\nu(1/\sigma^2)]^{-1}$ and $K_\lambda(t) = \frac{1}{2} \int_0^\infty x^{\lambda-1} \exp\{-\frac{1}{2}t(x+x^{-1})\} dx$.

Here $E(Y) = \mu$ and $Var(Y) = \mu^2 [2\sigma^2(\nu+1)/c + 1/c^2 - 1]$. $\mathbf{GIG}(\mu, \sigma, \nu)$ is a reparameterization of the generalized inverse Gaussian distribution of Jorgensen (1982). Note also that $\mathbf{GIG}(\mu, \sigma, -0.5) \equiv \mathbf{IG}(\mu, \sigma\mu^{-1/2})$ a reparameterization of the inverse Gaussian distribution.

A.6.4 Zero adjusted Inverse Gaussian distribution (ZAIG)

The zero adjusted inverse Gaussian distribution is appropriate when the response variable Y takes values from zero to infinity including zero, i.e. $[0, \infty)$. Hence $Y = 0$ has non zero probability ν . The pdf of the zero adjusted inverse Gaussian distribution, denoted by $\mathbf{ZAIG}(\mu, \sigma, \nu)$, is defined by

$$f_Y(y|\mu, \sigma, \nu) = \begin{cases} \nu & \text{if } y = 0 \\ (1-\nu) \frac{1}{\sqrt{2\pi\sigma^2 y^3}} \exp \left[-\frac{1}{2\mu^2\sigma^2 y} (y-\mu)^2 \right] & \text{if } y > 0 \end{cases} \quad (\text{A.40})$$

for $0 \leq y < \infty$, where $0 < \nu < 1$, $\mu > 0$ and $\sigma > 0$ with $E(Y) = (1-\nu)\mu$ and $Var(Y) = (1-\nu)\mu^2(\nu + \mu\sigma^2)$.

A.7 Continuous four parameter distribution in $\mathbb{R}^+$

A.7.1 Box-Cox t distribution (BCT)

Let Y be a positive random variable having a Box-Cox t distribution, Rigby and Stasinopoulos (2006), denoted by $\mathbf{BCT}(\mu, \sigma, \nu, \tau)$, defined through the transformed random variable Z given by (A.35), where the random variable Z is assumed to follow a truncated t distribution with degrees of freedom, $\tau > 0$, treated as a continuous parameter.

The pdf of Y , a $\mathbf{BCT}(\mu, \sigma, \nu, \tau)$ random variable, is given by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{y^{\nu-1} f_T(z)}{\mu^\nu \sigma F_T\left(\frac{1}{\sigma|\nu|}\right)} \quad (\text{A.41})$$

for $y > 0$, where $\mu > 0$, $\sigma > 0$ and $-\infty < \nu < \infty$, and where z is given by (A.35) and $f_T(t)$ and $F_T(t)$ are respectively the pdf and cumulative distribution function of a random variable T having a standard t distribution with degrees of freedom parameter $\tau > 0$, ie $T \sim t_\tau \equiv \mathbf{TF}(0, 1, \tau)$. If the truncation probability $F_T(-\frac{1}{\sigma|\nu|})$ is negligible, the variable Y has median μ .

A.7.2 Box-Cox power exponential distribution (BCPE)

Let Y be a positive random variable having a Box-Cox power exponential distribution, Rigby and Stasinopoulos (2004), denoted by $\mathbf{BCPE}(\mu, \sigma, \nu, \tau)$, defined through the transformed random variable Z given by (A.35), where the random variable Z is assumed to follow a truncated

standard power exponential distribution with power parameter, $\tau > 0$, treated as a continuous parameter.

The pdf of Y , a $\mathbf{BCPE}(\mu, \sigma, \nu, \tau)$ random variable, is given by (A.41), where $f_T(t)$ and $F_T(t)$ are respectively the pdf and cumulative distribution function of a variable T having a standard power exponential distribution, $T \sim \mathbf{PE}(0, 1, \tau)$. If the truncation probability $F_T(-\frac{1}{\sigma|\nu|})$ is negligible, the variable Y has median μ .

A.7.3 Generalized Beta type 2 distribution (GB2)

This pdf of the generalized beta type 2 distribution, denoted by $GB2(\mu, \sigma, \nu, \tau)$, is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = |\sigma| y^{\sigma\nu-1} \left\{ \mu^{\sigma\nu} B(\nu, \tau) [1 + (y/\mu)^\sigma]^{\nu+\tau} \right\}^{-1} \quad (\text{A.42})$$

for $y > 0$, where $\mu > 0$, $-\infty < \sigma < \infty$, $\nu > 0$ and $\tau > 0$, McDonald and Xu (1995), equation (2.7). The mean and variance of Y are given by $E(Y) = \mu B(\nu + \frac{1}{\sigma}, \tau - \frac{1}{\sigma}) / B(\nu, \tau)$ for $-\nu < \frac{1}{\sigma} < \tau$ and $E(Y^2) = \mu^2 B(\nu + \frac{2}{\sigma}, \tau - \frac{2}{\sigma}) / B(\nu, \tau)$ for $-\nu < \frac{2}{\sigma} < \tau$, McDonald (1996), p434.

A.8 Continuous two parameter distribution in $\mathfrak{R}[0, 1]$

A.8.1 Beta distribution (BE, BEo)

The beta distribution is appropriate when the response variable takes values in a known restricted range, excluding the endpoints of the range. Appropriate standardization can be applied to make the range of the response variable (0,1), i.e. from zero to one excluding the endpoints. Note that $0 < Y < 1$ so values $Y = 0$ and $Y = 1$ have zero density under the model.

First parameterization (BEo)

The original parameterization of the beta distribution, denoted by $BEo(\mu, \sigma)$, has pdf given by $f_Y(y|\mu, \sigma) = \frac{1}{B(\mu, \sigma)} y^{\mu-1} (1-y)^{\sigma-1}$ for $0 < y < 1$, with parameters $\mu > 0$ and $\sigma > 0$. Here $E(Y) = \mu/(\mu + \sigma)$ and $Var(Y) = \mu\sigma(\mu + \sigma)^{-2}(\mu + \sigma + 1)^{-1}$.

Second parameterization (BE)

In the second parameterization of the beta distribution below the parameters μ and σ are location and scale parameters that relate to the mean and standard deviation of Y . The pdf of the beta distribution, denoted by $\mathbf{BE}(\mu, \sigma)$, is defined by

$$f_Y(y|\mu, \sigma) = \frac{1}{B(\alpha, \beta)} y^{\alpha-1} (1-y)^{\beta-1} \quad (\text{A.43})$$

for $0 < y < 1$, where $\alpha = \mu(1 - \sigma^2)/\sigma^2$ and $\beta = (1 - \mu)(1 - \sigma^2)/\sigma^2$, $\alpha > 0$, and $\beta > 0$ and hence $0 < \mu < 1$ and $0 < \sigma < 1$. [Note the relationship between parameters (μ, σ) and (α, β) is given by $\mu = \alpha/(\alpha + \beta)$ and $\sigma = (\alpha + \beta + 1)^{-1/2}$.] In this parameterization, the mean of Y is $E(Y) = \mu$ and the variance is $Var(Y) = \sigma^2\mu(1 - \mu)$.

A.8.2 Beta inflated distribution (BEINF)

The beta inflated distribution is appropriate when the response variable takes values in a known restricted range including the endpoints of the range. Appropriate standardization can be applied to make the range of the response variable $[0,1]$, i.e. from zero to one including the endpoints. Values zero and one for Y have non zero probabilities p_0 and p_1 respectively. The probability (density) function of the inflated beta distribution, denoted by **BEINF** (μ, σ, ν, τ) is defined by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \begin{cases} p_0 & \text{if } y = 0 \\ (1 - p_0 - p_1)^{\frac{1}{B(\alpha, \beta)}} y^{\alpha-1} (1-y)^{\beta-1} & \text{if } 0 < y < 1 \\ p_1 & \text{if } y = 1 \end{cases} \quad (\text{A.44})$$

for $0 \leq y \leq 1$, where $\alpha = \mu(1 - \sigma^2)/\sigma^2$, $\beta = (1 - \mu)(1 - \sigma^2)/\sigma^2$, $p_0 = \nu(1 + \nu + \tau)^{-1}$, $p_1 = \tau(1 + \nu + \tau)^{-1}$ so $\alpha > 0$, $\beta > 0$, $0 < p_0 < 1$, $0 < p_1 < 1 - p_0$. Hence **BEINF** (μ, σ, ν, τ) has parameters $\mu = \alpha/(\alpha + \beta)$ and $\sigma = (\alpha + \beta + 1)^{-1/2}$, $\nu = p_0/p_2$, $\tau = p_1/p_2$ where $p_2 = 1 - p_0 - p_1$. Hence $0 < \mu < 1$, $0 < \sigma < 1$, $\nu > 0$ and $\tau > 0$.

A.8.3 Generalized Beta type 1 distribution (GB1)

The generalized beta type 1 distribution is defined by assuming $Z = Y^\tau / [\nu + (1 - \nu)Y^\tau] \sim BE(\mu, \sigma)$. Hence, the pdf of generalized beta type 1 distribution, denoted by **GB1** (μ, σ, ν, τ) , is given by

$$f_Y(y|\mu, \sigma, \nu, \tau) = \frac{\tau \nu^\beta y^{\tau\alpha-1} (1 - y^\tau)^{\beta-1}}{B(\alpha, \beta) [\nu + (1 - \nu)y^\tau]^{\alpha+\beta}} \quad (\text{A.45})$$

for $0 < y < 1$, where $\alpha = \mu(1 - \sigma^2)/\sigma^2$ and $\beta = (1 - \mu)(1 - \sigma^2)/\sigma^2$, $\alpha > 0$ and $\beta > 0$. Hence, **GB1** (μ, σ, ν, τ) has adopted parameters $\mu = \alpha/(\alpha + \beta)$, $\sigma = (\alpha + \beta + 1)^{-1/2}$, ν and τ , where $0 < \mu < 1$, $0 < \sigma < 1$, $\nu > 0$ and $\tau > 0$. The beta **BE** (μ, σ) distribution is a special case of **GB1** (μ, σ, ν, τ) where $\nu = 1$ and $\tau = 1$.

A.9 Binomial type data

A.9.1 The Binomial distribution (BI)

The probability function of the binomial distribution, denoted here as **BI** (n, μ) , is given by

$$p_Y(y|n, \mu) = P(Y = y|n, \mu) = \frac{n!}{y!(n-y)!} \mu^y (1 - \mu)^{n-y}$$

for $y = 0, 1, 2, \dots, n$, where $0 < \mu < 1$, (and n is a known positive integer), with $E(Y) = n\mu$ and $Var(Y) = n\mu(1 - \mu)$. See Johnson *et al.* (1993), p 105 where $\mu = p$.

A.9.2 Beta Binomial distribution (BB)

The probability function of the beta binomial distribution denoted here as **BB** (n, μ, σ) is given by

$$p_Y(y|\mu, \sigma) = \frac{\Gamma(n+1)}{\Gamma(y+1)\Gamma(n-y+1)} \frac{\Gamma(\frac{1}{\sigma})\Gamma(y + \frac{\mu}{\sigma})\Gamma[n + \frac{(1-\mu)}{\sigma} - y]}{\Gamma(n + \frac{1}{\sigma})\Gamma(\frac{\mu}{\sigma})\Gamma(\frac{1-\mu}{\sigma})} \quad (\text{A.46})$$

for $y = 0, 1, 2, \dots, n$, where $0 < \mu < 1$ and $\sigma > 0$ (and n is a known positive integer). Note that $E(Y) = n\mu$ and $Var(Y) = n\mu(1 - \mu) \left[1 + \frac{\sigma}{1+\sigma}(n-1)\right]$.

The binomial $\mathbf{BI}(n, \mu)$ distribution is the limiting distribution of $\mathbf{BB}(n, \mu, \sigma)$ as $\sigma \rightarrow 0$. For $\mu = 0.5$ and $\sigma = 0.5$, $\mathbf{BB}(n, \mu, \sigma)$ is a uniform distribution.

A.10 Count data

A.10.1 Poisson distribution (PO)

Poisson distribution

The probability function of the Poisson distribution, denoted here as $\mathbf{PO}(\mu)$, is given by

$$p_Y(y|\mu) = P(Y = y|\mu) = \frac{e^{-\mu}\mu^y}{y!} \quad (\text{A.47})$$

where $y = 0, 1, 2, \dots$, where $\mu > 0$, with $E(Y) = \mu$ and $Var(Y) = \mu$. [See Johnson *et al.* (1993), p 151.] The moment ratios of the distribution are given by $\sqrt{\beta_1} = \mu^{-0.5}$ and $\beta_2 = 3 + \mu^{-1}$ respectively. Note that the Poisson distribution has the property that $E[Y] = Var[Y]$ and that $\beta_2 - \beta_1 - 3 = 0$. The coefficient of variation of the distribution is given by $\mu^{-0.5}$. The index of dispersion, that is, the ratio $Var[Y]/E[Y]$ is equal to one for the Poisson distribution. For $Var[Y] > E[Y]$ we have overdispersion and for $Var[Y] < E[Y]$ we have underdispersion or repulsion. The distribution is skew for small values of μ , but almost symmetric for large μ values.

A.10.2 Negative Binomial distribution (NBI, NBII)

First parameterization: Negative Binomial type I (NBI)

The probability function of the negative binomial distribution type I, denoted here as $\mathbf{NBI}(\mu, \sigma)$, is given by

$$p_Y(y|\mu, \sigma) = \frac{\Gamma(y + \frac{1}{\sigma})}{\Gamma(\frac{1}{\sigma})\Gamma(y+1)} \left(\frac{\sigma\mu}{1+\sigma\mu}\right)^y \left(\frac{1}{1+\sigma\mu}\right)^{1/\sigma}$$

for $y = 0, 1, 2, \dots$, where $\mu > 0$, $\sigma > 0$ with $E(Y) = \mu$ and $Var(Y) = \mu + \sigma\mu^2$. [This parameterization is equivalent to that used by Anscombe (1950) except he used $\alpha = 1/\sigma$, as pointed out by Johnson *et al.* (1993), p 200, line 5.]

Second parameterization: Negative Binomial type II (NBII)

The probability function of the negative binomial distribution type II, denoted here as $\mathbf{NBII}(\mu, \sigma)$, is given by

$$p_Y(y|\mu, \sigma) = \frac{\Gamma(y + \mu/\sigma)\sigma^y}{\Gamma(\mu/\sigma)\Gamma(y+1)(1+\sigma)^{y+\mu/\sigma}}$$

for $y = 0, 1, 2, \dots$, where $\mu > 0$ and $\sigma > 0$. Note $E(Y) = \mu$ and $Var(Y) = (1 + \sigma)\mu$, so σ is a dispersion parameter [This parameterization was used by Evans (1953) as pointed out by Johnson *et al.* (1993) p 200 line 7.]

A.10.3 Poisson-inverse Gaussian distribution (PIG)

The probability function of the Poisson-inverse Gaussian distribution, denoted by **PIG**(μ, σ), is given by

$$p_Y(y|\mu, \sigma) = \left(\frac{2\alpha}{\pi}\right)^{\frac{1}{2}} \frac{\mu^y e^{1/\sigma} K_{y-\frac{1}{2}}(\alpha)}{(\alpha\sigma)^y y!}$$

where $\alpha^2 = \frac{1}{\sigma^2} + \frac{2\mu}{\sigma}$, for $y = 0, 1, 2, \dots, \infty$ where $\mu > 0$ and $\sigma > 0$ and $K_\lambda(t) = \frac{1}{2} \int_0^\infty x^{\lambda-1} \exp\{-\frac{1}{2}t(x+x^{-1})\}dx$ is the modified Bessel function of the third kind. [Note that the above parameterization was used by Dean, Lawless and Willmot (1989). It is also a special case of the **gamlss.family** distribution **SI**(μ, σ, ν) when $\nu = -\frac{1}{2}$.]

A.10.4 Delaporte distribution (DEL)

The probability function of the Delaporte distribution, denoted by **DEL**(μ, σ, ν), is given by

$$p_Y(y|\mu, \sigma, \nu) = \frac{e^{-\mu\nu}}{\Gamma(1/\sigma)} [1 + \mu\sigma(1 - \nu)]^{-1/\sigma} S \quad (\text{A.48})$$

where

$$S = \sum_{j=0}^y \binom{y}{j} \frac{\mu^y \nu^{y-j}}{y!} \left[\mu + \frac{1}{\sigma(1-\nu)} \right]^{-j} \Gamma\left(\frac{1}{\sigma} + j\right)$$

for $y = 0, 1, 2, \dots, \infty$ where $\mu > 0$, $\sigma > 0$ and $0 < \nu < 1$. This distribution is a reparameterization of the distribution given by Wimmer and Altmann (1999) p 515-516 where $\alpha = \mu\nu$, $k = 1/\sigma$ and $\rho = [1 + \mu\sigma(1 - \nu)]^{-1}$. The mean of Y is given by $E(Y) = \mu$ and the variance by $Var(Y) = \mu + \mu^2\sigma(1 - \nu)^2$.

A.10.5 Sichel distribution (SI, SICHEL)

First parameterization (SI)

The probability function of the first parameterization of the Sichel distribution, denoted by **SI**(μ, σ, ν), is given by

$$p_Y(y|\mu, \sigma, \nu) = \frac{\mu^y K_{y+\nu}(\alpha)}{(\alpha\sigma)^{y+\nu} y! K_\nu(\frac{1}{\sigma})} \quad (\text{A.49})$$

where $\alpha^2 = \frac{1}{\sigma^2} + \frac{2\mu}{\sigma}$, for $y = 0, 1, 2, \dots, \infty$ where $\mu > 0$, $\sigma > 0$ and $-\infty < \nu < \infty$ and $K_\lambda(t) = \frac{1}{2} \int_0^\infty x^{\lambda-1} \exp\{-\frac{1}{2}t(x+x^{-1})\}dx$ is the modified Bessel function of the third kind. Note that the above parameterization is different from Stein, Zucchini and Juritz (1988) who use the above probability function but treat μ , α and ν as the parameters. Note that $\sigma = [(\mu^2 + \alpha^2)^{\frac{1}{2}} - \mu]^{-1}$.

Second parameterization (SICHEL)

The second parameterization of the Sichel distribution, Rigby, Stasinopoulos and Akantziliotou (2008), denoted by **SICHEL**(μ, σ, ν), is given by

$$p_Y(y|\mu, \sigma, \nu) = \frac{(\mu/c)^y K_{y+\nu}(\alpha)}{y! (\alpha\sigma)^{y+\nu} K_\nu(\frac{1}{\sigma})} \quad (\text{A.50})$$

for $y = 0, 1, 2, \dots, \infty$, where $\alpha^2 = \sigma^{-2} + 2\mu(c\sigma)^{-1}$. The mean of Y is given by $E(Y) = \mu$ and the variance by $Var(Y) = \mu + \mu^2 [2\sigma(\nu + 1)/c + 1/c^2 - 1]$.

A.10.6 Zero inflated poisson (ZIP, ZIP2)

First parameterization (ZIP)

Let $Y = 0$ with probability σ and $Y \sim Po(\mu)$ with probability $(1 - \sigma)$, then Y has a zero inflated Poisson distribution, denoted by **ZIP** (μ, σ, ν) , given by

$$p_Y(y|\mu, \sigma) = \begin{cases} \sigma + (1 - \sigma)e^{-\mu}, & \text{if } y = 0 \\ (1 - \sigma)\frac{\mu^y}{y!}e^{-\mu}, & \text{if } y = 1, 2, 3, \dots \end{cases} \quad (\text{A.51})$$

See Johnson *et al* (1993), p 186, equation (4.100) for this parametrization. This parametrization was also used by Lambert (1992). The mean of Y in this parametrization is given by $E(Y) = (1 - \sigma)\mu$ and its variance by $Var(Y) = \mu(1 - \sigma)[1 + \mu\sigma]$.

Second parameterization (ZIP2)

A different parameterization of the zero inflated poisson distribution, denoted by **ZIP2** (μ, σ, ν) , is given by

$$p_Y(y|\mu, \sigma) = \begin{cases} \sigma + (1 - \sigma)e^{-\left(\frac{\mu}{1-\sigma}\right)}, & \text{if } y = 0 \\ (1 - \sigma)\frac{\mu^y}{y!(1-\sigma)^y}e^{-\left(\frac{\mu}{1-\sigma}\right)}, & \text{if } y = 1, 2, 3, \dots \end{cases} \quad (\text{A.52})$$

The mean of Y in (A.52) is given by $E(Y) = \mu$ and the variance by $Var(Y) = \mu + \mu^2 \frac{\sigma}{(1-\sigma)}$.

Bibliography

- [1] **Aitkin, M.** (1996). A general maximum likelihood analysis of overdispersion in generalised linear models. *Statist. Comput.*, **6**: 251–262.
- [2] **Aitkin, M.** (1999). A general maximum likelihood analysis of variance components in generalized linear models. *Biometrics*, **55**: 117–128.
- [3] **Aitkin, M.** (1999a). Meta -analysis by random effede modelling in generalised linear models. *Statistics in Medicine*, **18**: 2343–2351.
- [4] **Akaike, H.** (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**: 716–723.
- [5] **Akaike, H.** (1983). Information measures and model selection. *Bulletin of the International Statistical Institute*, **50**: 277–290.
- [6] **Akantziliotou, K. Rigby, R. A. and Stasinopoulos, D. M.** (2002). The R implementation of Generalized Additive Models for Location, Scale and Shape. In: Stasinopoulos, M. and Touloumi, G. (eds.), *Statistical modelling in Society: Proceedings of the 17th International Workshop on statistical modelling*, pp. 75–83. Chania, Greece.
- [7] **Ambler, G.** (1999). *fracpoly(): Fractional Polynomial Model*. S-PLUS.
- [8] **Anscombe, F. J.** (1950). Sampling theory of the negative binomial and logarithmic series approximations. *Biometrika*, **37**: 358–382.
- [9] **Azzalini, A.** (1986). Further results on a class of distributions which includes the normal ones. *Statistica*, **46**: 199:208.
- [10] **Azzalini, A. and Bowman, A. W.** (1990). A look at some data on the Old Faithful geyser. *Applied Statistics*, **39**: 357–365.
- [11] **Azzalini, A. and Capitanio, A.** (1985). A class of distributions which includes the normal ones. *Scand. J. Statist.*, **12**: 171–178.
- [12] **Azzalini, A. and Capitanio, A.** (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew *t*-distribution. *J. R. Statist. Soc. B*, **65**: 367–389.
- [13] **Besag, J. and Higdon, P.** (1999). Bayesian analysis of agriculture field experiments (with discussion). *Journal of the Royal Statistical Society, B*, **61**: 691–746.

- [14] **Besag, J., York, J. and Mollie, A.** (1991). Bayesian image restoration, with to applications in spacial statistics (with discussion). *Annals of the Institute of Statistical Mathematics*, **43**: 1–59.
- [15] **Box, G. E. P. and Cox, D. R.** (1964). An analysis of transformations (with discussion). *J. R. Statist. Soc. B.*, **26**: 211–252.
- [16] **Box, G. E. P. and Tiao, G. C.** (1973). *Bayesian Inference in Statistical Analysis*. Wiley, New York.
- [17] **Breslow, N. E. and Clayton, D. G.** (1993). Approximate inference in generalized linear mixed models. *J. Am. Statist. Ass.*, **88**: 9–25.
- [18] **Carroll, R. J. and Ruppert, D.** (1982). Robust estimation in heteroscedastic linear models. *Ann. Statist.*, **10**: 429–441.
- [19] **Chambers, J. M. and Hastie, T. J.** (1992). *Statistical Models in S*. Chapman & Hall, London.
- [20] **Chappas, C. and Corina-Borja, M.** (2006). A stylometric analysis of newspapers, periodicals and news scripts. *Journal of Quantitative Linguistics*, **13**: 285–312.
- [21] **Chitty, L. S., Altman, D. G., Henderson, A., and Campbell, S.** (1994). Charts of fetal size: 3, abdominal measurements. *Br. J. Obstetr.*, **101**: 125–131.
- [22] **Claeskens, G. and Hjort, N. L.** (2003). The focused information criterion. *J. Am. Statist. Ass.*, **98**: 900–916.
- [23] **Cleveland, W. S., Grosse, E. and Shyu, M.** (1993). Local Regression Models. In: Chambers, J. and Hastie, T. (eds.), *Statistical Modelling in S*, pp. 309–376. Chapman and Hall: New York.
- [24] **Cole, T. J. and Green, P. J.** (1992). Smoothing reference centile curves: the LMS method and penalized likelihood. *Statist. Med.*, **11**: 1305–1319.
- [25] **Consul, P. C.** (1989). *Generalized Poisson Distributions*. Marcel Dekker, New York.
- [26] **Crisp, A. and Burridge, J.** (1994). A note on nonregular likelihood functions in heteroscedastic regression models. *Biometrika*, **81**: 585–587.
- [27] **Crowder, M. J., Kimber, A. C., Smith R. L. and Sweeting, T. J.** (1991). *Statistical Analysis of Reliability Data*. Chapman and Hall, London.
- [28] **CYTEL Software Corporation** (2001). *EGRET for Windows*. CYTEL Software Corporation, Cambridge, Massachusetts.
- [29] **D’Agostino, R. B., Balanger, A. and D’Agostino Jr., R. B.** (1990). A suggestion for using powerful and informative tests of normality. *American Statistician*, **44**: 316–321.
- [30] **Davidian, M. and Carroll, R. J.** (1988). A note on extended quasi-likelihood. *J. R. Statist. Soc.*, **50**: 74–82.
- [31] **Davis, G. C. J. and Kutner, M. H.** (1976). The lagged normal family of probability density functions applied to indicator-dilution curves. *Biometrics*, **32**: 669–675.

- [32] **de Boor, C.** (1978). *A Practical Guide to Splines*. Springer, New York.
- [33] **de Jong, P. and Heller, G. Z.** (2007). *Generalized Linear Models for Insurance Data*. Cambridge University Press.
- [34] **Dean, C., Lawless, J. F. and Willmot, G. E.** (1989). A mixed Poisson-inverse-Gaussian regression model. *Canadian Journal of Statistics*, **17**: 171–181.
- [35] **Dempster, A., Laird, N. and Rubin, D.** (1977). Maximum likelihood from incomplete data via EM algorithm (with discussion). *J. R. Statist. Soc.*, **39**: 1–38.
- [36] **DiCiccio, T. J. and Monti, A. C.** (2004). Inferential Aspects of the Skew Exponential Power Distribution. *J. Am. Statist. Ass.*, **99**: 439–450.
- [37] **Diggle, P. J., Heagerty, P., Liang, K. -Y. and Zeger, S. L.** (2002). *Analysis of Longitudinal Data, 2nd ed.* Oxford University Press, Oxford.
- [38] **Draper, D.** (1995). Assessment and propagation of model uncertainty (with discussion). *J. R. Statist. Soc. B.*, **57**: 45–97.
- [39] **Dunn, P. K. and Smyth, G. K.** (1996). Randomised quantile residuals. *J. Comput. Graph. Statist.*, **5**: 236–244.
- [40] **Efron, B.** (1986). Double exponential families and their use in generalized linear regression. *J. Am. Statist. Ass.*, **81**: 709–721.
- [41] **Eilers, P. H. C. and Marx, B. D.** (1996). Flexible smoothing with B-splines and penalties (with comments and rejoinder). *Statist. Sci.*, **11**: 89–121.
- [42] **Evans, D. A.** (1953). Experimental evidence concerning contagious distributions in ecology. *Biometrika*, **40**: 186–211.
- [43] **Fahrmeir, L. and Tutz, G.** (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models, 2nd ed.* Springer, New York.
- [44] **Fernandez, C. and Steel, M. F. J.** (1998). On Bayesian Modelling of Fat Tails and Skewness. *J. Am. Statist. Ass.*, **93**: 359–371.
- [45] **Fernandez, C., Osiewalski, J. and Steel, M. J. F.** (1995). Modeling and inference with v-spherical distributions. *J. Am. Statist. Ass.*, **90**: 1331–1340.
- [46] **Fredriks, A.M., van Buuren, S., Burgmeijer, R.J.F., Meulmeester, J.F., Beuker, R.J., Brugman, E., Roede, M.J., Verloove-Vanhorick, S.P. and Wit, J. M.** (2000a). Continuing positive secular change in The Netherlands, 1955-1997. *Pediatric Research*, **47**: 316–323.
- [47] **Fredriks, A.M., van Buuren, S., Wit, J.M. and Verloove-Vanhorick, S. P.** (2000b). Body index measurements in 1996-7 compared with 1980. *Archives of Childhood Diseases*, **82**: 107–112.
- [48] **Gange, S. J., Munoz, A., Saez, M. and Alonso, J.** (1996). Use of the beta-binomial distribution to model the effect of policy changes on appropriateness of hospital stays. *Appl. Statist.*, **45**: 371–382.

- [49] **Gelman, A. Carlin, J. B. Stern, H. S. and Rubin, D. B.** (2004). *Bayesian Data Analysis, 2nd ed.* Chapman and Hall/CRC, London.
- [50] **Gibbons, J. F. and Mylroie, S.** (1973). Estimation of impurity profiles in ion-implanted amorphous targets using joined half-Gaussian distributions. *Appl. Phys. Lett.*, **22**: 568–569.
- [51] **Gilchrist, R.** (2000). Regression models for data with a non-zero probability of a zero response. *Commun. Statist. Theory Meth.*, **29**: 1987–2003.
- [52] **Green, P. J.** (1985). Linear models for field trials, smoothing and cross-validation. *Biometrika*, **72**: 527–537.
- [53] **Green, P. J. and Silverman, B. W.** (1994). *Nonparametric Regression and Generalized Linear Models.* Chapman and Hall, London.
- [54] **Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J. and Ostrowski, E.** (1994). *A handbook of small data sets.* Chapman and Hall, London.
- [55] **Hansen, B.** (1994). Autoregressive conditional density estimation. *International Economic Review*, **35**: 705–730.
- [56] **Hastie, T.** (2006). **gam**: *Generalized Additive Models*. R package version 0.98.
- [57] **Hastie, T. J. and Tibshirani, R. J.** (1990). *Generalized Additive Models.* Chapman and Hall, London.
- [58] **Hastie, T. J. and Tibshirani, R. J.** (1993). Varying coefficient models (with discussion). *J. R. Statist. Soc. B.*, **55**: 757–796.
- [59] **Hastie, T. J. and Tibshirani, R. J.** (2000). Bayesian backfitting. *Stat. Sci.*, **15**.
- [60] **Hinde, J.** (1982). Compound Poisson regression models. In: Gilchrist, R. (ed.), *GLIM 82, Proceedings of the International Conference on Generalised Linear Models*, pp. 109–121. Springer, New York.
- [61] **Hjort, N. L. and Claeskens, G.** (2003). Frequentist model average estimation. *J. Am. Statist. Ass.*, **98**: 879–899.
- [62] **Hodges, J. S.** (1998). Some algebra and geometry for hierarchical models, applied to diadnostics. *J. R. Statist. Soc. B.*, **60**: 497–536.
- [63] **Hodges, J. S. and Sargent, D. J.** (2001). Counting degrees of freedom in hierarchical and other richly-parameterised models. *Biometrika*, **88**: 367–379.
- [64] **Ihaka, R. and Gentleman, R.** (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, **5**: 299–314.
- [65] **Johnson, N. L.** (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, **36**: 149–176.
- [66] **Johnson, N. L., Kotz, S. and Balakrishnan, N.** (1994). *Continuous Univariate Distributions, Volume I, 2nd edn.* Wiley, New York.

- [67] **Johnson, N. L., Kotz, S. and Balakrishnan, N.** (1995). *Continuous Univariate Distributions, Volume II, 2nd edn.* Wiley, New York.
- [68] **Johnson, N. L., Kotz, S. and Kemp, A. W.** (2005). *Univariate Discrete Distributions, 3rd edn.* Wiley, New York.
- [69] **Jones, M. C.** (2005). In discussion of Rigby, R. A. and Stasinopoulos, D. M. (2005) Generalized additive models for location, scale and shape,. *Applied Statistics*, **54**: 507–554.
- [70] **Jones, M. C. and Faddy, M. J.** (2003). A skew extension of the t distribution, with applications. *J. Roy. Statist. Soc B*, **65**: 159–174.
- [71] **Jørgensen, B.** (1982). *Statistical Properties of the Generalized Inverse Gaussian Distribution, Lecture Notes in Statistics No.9.* Springer-Verlag, New York.
- [72] **Karlis, D. and Xekalaki, E.** (2008). The Polygonal Distribution. In: Arnold, B. C., Balakrishnan, N., Minué, M and Sarabia, J. M. (eds.), *Mathematical and Statistical Modeling in Honor of Enrique Castillo*. Birkhauser: Boston.
- [73] **Kohn, R. and Ansley, C. F.** (1998). Equivalence between Bayesian smoothness prior and optimal smoothing for function estimation. In: Spall, J. C. (ed.), *Bayesian Analysis of Time Series and Dynamic models*, pp. 393–430. Dekker: New York.
- [74] **Kohn, R. and Ansley, C. F. and Tharm, D.** (1991). The performance of cross-validation and maximum likelihood estimators of spline smoothing parameters. *J. Amer. Statist. Ass.*, **86**: 1042–1050.
- [75] **Lambert, D.** (1992). Zero-inflated Poisson Regression with an application to defects in Manufacturing. *Technometrics*, **34**: 1–14.
- [76] **Lambert, P. and Lindsey, J.** (1999). Analysing financial returns using regression models based on non-symmetric stable distributions. *Applied Statistics*, **48**: 409–424.
- [77] **Lange, K.** (1999). *Numerical Analysis for Statisticians*. Springer, New York.
- [78] **Lange, K. L., Little, R. J. A. and Taylor, J. M. G.** (1989). Robust statistical modelling using the t distribution. *J. Am. Statist. Ass.*, **84**: 881–896.
- [79] **Lee, Y. and Nelder, J. A.** (1996). Hierarchical generalized linear models (with discussion). *J. R. Statist. Soc. B.*, **58**: 619–678.
- [80] **Lee, Y. and Nelder, J. A.** (2001a). Hierarchical generalised linear models: A synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**: 987–1006.
- [81] **Lee, Y. and Nelder, J. A.** (2001b). Modelling and analysing correlated non-normal data. *Statistical Modelling*, **1**: 3–16.
- [82] **Lee, Y. and Nelder, J. A.** (2006). Doudle Hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**: 139–185.
- [83] **Lindsey, J.** (1999). On the use of correction for overdispersion. *Applied Statistics*, **48**: 553–561.

- [84] **Lindsey, J. K.** (1999). *Models for Repeated Measurements (Second Edition)*. Oxford University Press, Oxford.
- [85] **Lopatzidis, A. and Green, P. J.** (2000). Nonparametric quantile regression using the gamma distribution. *submitted for publication*.
- [86] **Madigan, D. and Raftery, A. E.** (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *J. Am. Statist. Ass.*, **89**: 1535–1546.
- [87] **Marx, B.** (2003). *ps(): P-Spline Code for GAM's and Univariate GLM Smoothing*. S-PLUS.
- [88] **McCullagh, P. and Nelder, J. A.** (1989). *Generalized Linear Models, 2nd edn*. Chapman and Hall, London.
- [89] **McDonald, J. B.** (1991). Parametric models for partially adaptive estimation with skewed and leptokurtic residuals. *Economic Letters*, **37**: 273–278.
- [90] **McDonald, J. B.** (1996). Probability Distributions for Financial Models. In: Maddala, G. S. and Rao, C. R. (eds.), *Handbook of Statistics, Vol. 14*, pp. 427–460. Elsevier Science.
- [91] **McDonald, J. B. and Newey, W. K.** (1988). Partially adaptive estimation of regression models via the generalized t distribution. *Econometric Theory*, **4**: 428–457.
- [92] **McDonald, J. B. and Xu, Y. J.** (1995). A generalisation of the beta distribution with applications. *Journal of Econometrics*, **66**: 133–152.
- [93] **Nandi, A. K. and Mämpel, D.** (1995). An extension of the generalized Gaussian distribution to include asymmetry. *J. Franklin Inst.*, **332**: 67–75.
- [94] **Nelder, J. A.** (2001). Correspondance. *Statistician*, **50**: 209–211.
- [95] **Nelder, J. A. and Lee, Y.** (1992). Likelihood, quasi-likelihood and pseudolikelihood: some comparisons. *J. R. Statist. Soc. B.*, **54**: 273–284.
- [96] **Nelder, J. A. and Pregibon, D.** (1987). An extended quasi-likelihood function. *Biometrika*, **74**: 221–232.
- [97] **Nelder, J. A. and Wedderburn, R. W. M.** (1972). Generalized linear models. *J. R. Statist. Soc. A.*, **135**: 370–384.
- [98] **Nelson, D. B.** (1991). Conditional heteroskedasticity in asset returns: a new approach. *Econometrica*, **59**: 347–370.
- [99] **Ortega, J. M. and Rheinboldt, W. C.** (1970). *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York.
- [100] **P., W. M. and Jones, M. C.** (1999). *Kernel Smoothing*. Chapman & Hall, Essen, Germany.
- [101] **Parzen, E.** (1984). Nonparametric statistical data modelling. *Journal of the American Statistical Association*, **74**: 105–131.

- [102] **Perks, W. F.** (1932). On some experiments in the graduation of mortality statistics. *Journal of the Institute of Actuaries*, **58**: 12–57.
- [103] **Pham-Gia, T. and Duong, Q. P.** (1989). The generalized beta and F distributions in statistical modelling. *Mathematical and Computer Modelling*, **13**: 1613–1625.
- [104] **Pinheiro, J. C. and Bates, D. M.** (2000). *Mixed-Effects Models in S and S-Plus*. Springer. ISBN 0-387-98957-0.
- [105] **Quesenberry, C. and Hales, C.** (1980). Concentration bands for uniformly plots. *Journal of Statistical Computation and Simulation*, **11**: 41–53.
- [106] **Raftery, A. E.** (1996). Approximate Bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika*, **83**: 251–266.
- [107] **Raftery, A. E.** (1999). Bayes Factors and BIC, comment on 'A critique of the Bayesian Information Criterion for Model Selection'. *Sociological Methods & Research*, **27**: 411–427.
- [108] **Reinsch, C.** (1967). Smoothing by spline functions. *Numerische Mathematik*, **10**: 177–183.
- [109] **Rigby, R. A. and Stasinopoulos, D. M.** (1994). Robust fitting of an additive model for variance heterogeneity. In: Dutter, R. and Grossmann, W. (eds.), *COMPSTAT : Proceedings in Computational Statistics*, pp. 263–268. Physica, Heidelberg.
- [110] **Rigby, R. A. and Stasinopoulos, D. M.** (1996a). A semi-parametric additive model for variance heterogeneity. *Statist. Comput.*, **6**: 57–65.
- [111] **Rigby, R. A. and Stasinopoulos, D. M.** (1996b). Mean and dispersion additive models. In: Hardle, W. and Schimek, M. G. (eds.), *Statistical Theory and Computational Aspects of Smoothing*, pp. 215–230. Physica, Heidelberg.
- [112] **Rigby, R. A. and Stasinopoulos, D. M.** (2000). Construction of reference centiles using mean and dispersion additive models. *Statistician*, **49**: 41–50.
- [113] **Rigby, R. A. and Stasinopoulos, D. M.** (2001). The GAMLSS project: a flexible approach to statistical modelling. In: Klein, B. and Korsholm, L. (eds.), *New Trends in Statistical Modelling: Proceedings of the 16th International Workshop on Statistical Modelling*, pp. 249–256. Odense, Denmark.
- [114] **Rigby, R. A. and Stasinopoulos, D. M.** (2004). Smooth centile curves for skew and kurtotic data modelled using the Box-Cox Power Exponential distribution. *Statistics in Medicine*, **23**: 3053–3076.
- [115] **Rigby, R. A. and Stasinopoulos, D. M.** (2005). Generalized additive models for location, scale and shape, (with discussion). *Appl. Statist.*, **54**: 507–554.
- [116] **Rigby, R. A. and Stasinopoulos, D. M.** (2006). Using the Box-Cox t distribution in GAMLSS to model skewness and kurtosis. *Statistical Modelling*, **6**: 209–229.
- [117] **Rigby, R. A. Stasinopoulos, D. M. and K., A.** (2008). A framework for modelling overdispersed count data, including the Poisson-shifted generalized inverse Gaussian distribution. *Computational Statistics and Data analysis*, **n print**.

- [118] **Royston, P. and Altman, D. G.** (1994). Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling (with discussion). *Appl. Statist.*, **43**: 429–467.
- [119] **Royston, P. and Wright, E. M.** (2000). Goodness-of-fit statistics for age-specific reference intervals. *Statistics in Medicine*, **19**: 2943–2962.
- [120] **SAS Institute Inc.** (2000). *Enterprise Miner Software, Version 4*. SAS Institute Inc, Cary, North Carolina.
- [121] **Schumaker, L. L.** (1993). *Spline Functions : Basic Theory*. Krieger, Melbourne Florida.
- [122] **Schwarz, G.** (1978). Estimating the dimension of a model. *Ann. Statist.*, **6**: 461–464.
- [123] **Sheather, S. J. and Jones, M. C.** (1991). A reliable data-based bandwidth selection method for kernel density estimation. *J. R. Statist. Soc. B.*, **53**: 683–690.
- [124] **Sichel, H. S.** (1992). Anatomy of a generalized inverse Gaussian-Poisson distribution with special applications to bibliometric Studies. *Information Processing and Management*, **28**: 5–17.
- [125] **Silverman, B. W.** (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). *J. R. Statist. Soc. B.*, **47**: 1–52.
- [126] **Silverman, B. W.** (1988). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall.
- [127] **Skrondal, A. and S., R.-H.** (2004). *Generalized Latent Variable modelling*. Chapman & Hall.
- [128] **Smith, R. L. and Naylor, J. C.** (1987). A comparison of maximum likelihood and Bayesian estimators for the three-parameter Weibull distribution. *Appl. Statist.*, **36**: 358–369.
- [129] **Speed, T. P.** (1991). Comment on 'That BLUP is a good thing: the estimation of random effects' (by G. K. Robinson). *Statist. Sci.*, **6**: 42–44.
- [130] **Stasinopoulos, D. M.** (2006). Contribution to the discussion of the paper by Lee and Nelder, Double hierarchical generalized linear models. *Appl. Statist.*, **55**: 171–172.
- [131] **Stasinopoulos, D. M., Rigby, R. A. and Akantziliotou, C.** (2007). Instructions on how to use the GAMLSS package in R, Second Edition. Technical Report 01/08, STORM Research Centre, London Metropolitan University, London.
- [132] **Stasinopoulos, D. M., Rigby, R. A. and Fahrmeir, L.** (2000). Modelling rental guide data using mean and dispersion additive models. *Statistician*, **49**: 479–493.
- [133] **Stein, G. Z., Zucchini, W. and Juritz, J. M.** (1987). Parameter Estimation of the Sichel Distribution and its Multivariate Extension. *Journal of American Statistical Association*, **82**: 938–944.
- [134] **Subbotin, M. T.** (1923). On the law of frequency of errors. *Mathematicheskii Sbornik*, **31**: 296–301.

- [135] **Tamura, R. N. and Young, S. S.** (1987). A stabilized moment estimator for the beta binomial distribution. *Biometrics*, **43**: 813–824.
- [136] **Theodossiou, P.** (1998). Financial data and the skewed generalized t distribution. *Management Science*, **44**: 1650–1661.
- [137] **Tierney, L. and Kadane, J. B.** (1986). Accurate approximations for posterior moments and marginal densities. *J. Am. Statist. Ass.*, **81**: 82–86.
- [138] **van Buuren, S. and Fredriks, M.** (2001). Worm plot: a simple diagnostic device for modelling growth reference curves. *Statistics in Medicine*, **20**: 1259–1277.
- [139] **Venables, W. N. and Ripley, B. D.** (2000). *S Programming*. Springer. ISBN 0-387-98966-8.
- [140] **Venables, W. N. and Ripley, B. D.** (2002). *Modern Applied Statistics with S. Fourth Edition*. Springer. ISBN 0-387-98825-4.
- [141] **Verbyla, A. P., Cullis, B. R., Kenward, M. G. and Welham, S. J.** (1999). The analysis of designed experiments and longitudinal data by using smoothing splines (with discussion). *Appl. Statist.*, **48**: 269–311.
- [142] **Wade, A. M. and Ades, A. E.** (1994). Age-related reference ranges : Significance tests for models and confidence intervals for centiles. *Statistics in Medicine*, **13**: 2359–2367.
- [143] **Wahba, G.** (1978). Improper priors, spline smoothing and the problem of guarding against model errors in regression. *J. R. Statist. Soc. B.*, **40**: 364–372.
- [144] **Wahba, G.** (1985). A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Ann. Statist.*, **4**: 1378–1402.
- [145] **Wedderburn, R. W. M.** (1974). Quasi-likelihood functions, generalised linear models and the Gauss-Newton method. *Biometrika*, **61**: 439–447.
- [146] **Wilkinson, G. N. and Rogers, C. E.** (1973). Symbolic description of factorial models for analysis of variance. *Appl. Statist.*, **22**: 392–399.
- [147] **Wimmer, G. and Altmann, G.** (1999). *Thesaurus of univariate discrete probability distributions*. Stamm Verlag, Essen, Germany.
- [148] **Winkelmann, R.** (1997). *Ecometric Analysis of Count Data*. Springer Verlag, Berlin.
- [149] **Wood, S. N.** (2001). mgcv: GAMs and Generalised Ridge Regression for R. *R News*, **1**: 20–25.
- [150] **Wright, E. M. and Royston, P.** (1997). A comparison of statistical methods for age-related reference intervals. *J. R. Statist. Soc. A.*, **160**: 47–69.

Index

- additive terms
 - cs(), 231
 - fp(), 27
 - lo, 242
 - ps(), 239
 - vc(), 235
- addterm, 32
- AIC, 31, 41, 192
- algorithm
 - EM, 158
 - RS, 88
 - RS-simple, 88
- algorithms, 29
 - CG, 29
 - RS, 29
- backfitting, 28
- backfitting
 - modified , 28
- centiles, 269
 - functions, 33
- checklink, 33
- coef, 31
- cubic smoothing splines
 - cs(), 37, 231
- data, 121
 - abdom, 34
 - abdom, 34, 100, 219
 - AEP, 147
 - aids, 193, 214, 215
 - alveolar, 143, 147
 - brains, 182
 - CD4, 89
 - computer, 121
 - crash helmet, 228
 - education, 293
 - enzyme, 163
 - geyser, 168, 172
 - glass, 65
 - hodges, 320
 - la, 252
 - LGAcclaims, 206
 - lice, 124
 - mcycle, 228, 240
 - meta, 303
 - old faithful geyser, 167
 - parzen, 59
 - pCD4, 89
 - rent, 228, 238
 - schools, 293
 - species, 129
 - students, 146
 - stylo, 127
 - tensile, 72
 - The AIDS data, 193
 - US pollution data, 193
 - usair, 193
- degree
 - lo(), 245
- degrees of freedom, 28
- deviance, 31, 41, 191
 - global deviance, 58
- df
 - lo(), 245
- distribution, 21, 50
 - zero inflated Poisson, 346
 - BCCG, 339
 - BCPE, 341
 - BCT, 341
 - beta, 342
 - beta inflated, 343
 - betabinomial, 343
 - binomial, 343
 - continuous, 50
 - Delaporte, 345
 - distHist(), 58
 - exponential, 337

- exponential Gaussian, 331
- Exponential Generalized Beta type 2, 332
- fit
 - discrete, 113
- fitting, 56
- gamma, 338
- Generalized Beta type 1, 343
- Generalized Beta type 2, 342
- Generalized Gamma, 340
- Generalized Inverse Gaussian, 341
- Generalized t, 332
- Gumbel, 330
- inverse Gaussian, 338
- Johnson
 - original, 332
 - reparameterized, 333
- log normal, 338
- logistic, 330
- negative binomial
 - type I, 344
 - type II, 344
- normal, 34, 329
- Poisson, 344
- Poisson-inverse Gaussian, 345
- powerexponential, 331
- reverse Gumbel, 331
- Sichel, 345
- Sinh-Arcsinh, 334
- Skew Exponential Power type 1, 334
- Skew Exponential Power type 3, 335
- Skew Exponential Power type 4, 336
- skew power exponential, 335
- skew t distribution type 5, 337
- Skew t type 1, 336
- Skewt type 2, 336
- Skewt type 3, 336
- Skewt type 4, 337
- t, 41, 332
- types, 49
- Weibull, 339
- zero adjusted Inverse Gaussian, 341
- distributions, 329
 - fit
 - Continuous, 59
- distributions
 - fit
 - count, 113
- dropterm, 32
- effective degrees of freedom , 38
- examples
 - abdominal circumference, 100
- extractAIC, 31
- find.hyper, 32
- finite mixtures, 157
 - no common parameters, 158
 - common parameters, 179
- fitted, 31, 38
- fitted.plot, 33
- formula, 31
- fractional polynomials
 - fp(), 27
- fractional polynomials
 - fp, 28
- function
 - addterm, 193
 - addterm(), 193, 194
 - arguments, 194
 - distHist(), 58
 - dropterm(), 193, 194
 - arguments, 194
 - find.hyper(), 213
 - gamlssMX(), 162
 - gamlssNP(), 181
 - nlgamlss(), 251
 - stepGAIC(), 193, 194, 196
 - arguments, 197
 - stepGAIC.CH(), 193, 201
 - arguments, 196
 - stepGAIC.VR(), 193, 194, 197, 204
 - arguments, 196
- functon
 - stepGAIC(), 194
- fv, 31
- GAIC, 41, 192
- GAMLSS
 - The model, 19
- gamlss
 - family, 21, 50, 329
 - function, 34
 - model, 19
 - nonlinear parametric, 21
 - nonlinear semi parametric, 21
 - packages, 30

- parametriclinear, 21
- gamlss family
 - BB, 21, 343
 - BCCG, 21, 50, 75, 339
 - BCPE, 21, 50, 75, 341
 - BCT, 21, 50, 75, 76, 341
 - BE, 21, 342
 - BEINF, 21, 343
 - BEOI, 21
 - BEZI, 21
 - BI, 21, 343
 - DEL, 21, 113, 345
 - EGB2, 21, 50, 76, 332
 - exGAUS, 21, 50, 331
 - EXP, 21, 50, 337
 - GA, 21, 50, 338
 - GB1, 21, 343
 - GB2, 21, 342
 - GG, 21, 50, 340
 - GG2, 340
 - GIG, 21, 50, 341
 - GT, 21, 50, 332
 - GU, 21, 50, 330
 - IG, 21, 50, 338
 - JSU, 21, 50, 333
 - JSUo, 21, 50, 332
 - LNO, 21, 50, 338
 - LO, 21, 50, 330
 - LOGNO, 21, 50, 338
 - NBI, 21, 113, 344
 - NBII, 21, 113, 344
 - NET, 21, 50, 333
 - NO, 21, 50, 329
 - NO2, 330
 - NOF, 21, 50, 330
 - PE, 21, 50, 331
 - PE2, 331
 - PIG, 21, 113, 345
 - PO, 21, 113, 344
 - RG, 21, 50, 331
 - RGE, 50
 - SEP, 50
 - SEP1, 21, 50, 334
 - SEP2, 21, 50, 335
 - SEP3, 21, 50, 335
 - SEP4, 21, 50, 336
 - SHASH, 21, 50, 334
 - SI, 21, 345
 - SICHEL, 21, 113, 345
 - ST1, 21, 50, 336
 - ST2, 21, 50, 336
 - ST3, 21, 50, 336
 - ST4, 21, 50, 337
 - ST5, 21, 50, 337
 - TF, 21, 50, 332
 - WEI, 21, 50, 339
 - WEI2, 21, 50, 339
 - WEI3, 21, 50, 339
 - ZAIG, 21, 341
 - ZIP, 21, 113, 346
 - ZIP2, 21, 346
- gamlss.family, 21
- gamlss.scope, 32
- global deviance, 41, 191
- histDist, 31
- likelihood, 56
 - finite mixtures, 158
 - regression, 87
- link function
 - show.link, 33
- loess
 - lo, 242
- make.link, 33
- model.frame, 31
- model.matrix, 31
- nl, 261
- nonlinear models, 251
- order
 - ps(), 242
- par.plot, 33
- parameters
 - distributional, 20
- pdf.plot, 33
- penalized likelihood, 21
- penalized splines
 - ps(), 239
- plot, 32
- predict, 31, 38
- print, 31
- prof.dev, 33
- prof.term, 33

- Q statistics, [33](#)
- Q.stats, [33](#)

- refit, [31](#)
- resid, [32](#), [38](#)
- residuals, [32](#)
 - quantile, [38](#)
- rqres, [33](#)

- Selecting hyperparameters, [213](#)
- show.link, [33](#)
- smoothing, [225](#)
 - components, [28](#)
- span
 - lo(), [245](#)
- stepGAIC, [32](#)
- stepGAIC.VR, [32](#)
- summary, [32](#)

- term.plot, [231](#), [233](#), [235](#)
- terms, [32](#)
- terms.plot, [33](#)
- TGD, [32](#)

- update, [31](#)

- varying coefficient
 - vc(), [235](#)
- VGD, [32](#)
- VGD1, [32](#)
- VGD2, [32](#)

- wp, [33](#), [39](#)